DRACULA 2.4

User Manual

August 2007

Dr. Ronghui Liu

Institute for Transport Studies
University of Leeds
Leeds LS2 9JT

Tel: +44 (0)113 343 5338
Fax: +44 (0)113 343 5334
E-mail: trarl@its.leeds.ac.uk

Web site: http://www.its.leeds.ac.uk/software/dracula/

Highlights of DRACULA Versions 2.4

This documentation describes the changes made in DRACULA Version 2.4 to be released in autumn 2007.

There have been many changes both in the way **DRACULA** models traffic flow dynamics (see the list below on "Modelling") and in improving the user interface ("Technical"). The main changes are listed below.

MODELLING

- Allow a choice of inputting either passenger flow or dwell time at the bus stops. (Default QPASSFLOW=T, input passenger flow).

- Model of time-dependent passenger demand profile. A same, global demand profile is applied to all O-D pairs using parameters P_NSTEP, P_TSTEP(n) and P_GONZO(n) to indicate the number of time steps, the period of step n and the matrix factor for step n. See Section 5.4 of the Manual.

- Correction in loading multiple time period assignments from multiple input .trp files.

- Under pre-vehicle-generation option (QNEWDEMAND=T), buses are generated separately from the other traffic because they are on fixed departure time.

TECHNICAL

- Include buses (entered both from .bus file and SATURN 66666 card) in the pre-specified vehicle pool, when QNEWDEMAND=T is set. (Use no_rt_P instead of no_grt_P in generate_veh_pool).

- Allow SATURN-style comment lines (lines beginning with a *) in the GIS inputs. (routines changed are read_gis_* in t5load.c) 25.08.2006

- Allow automatic search for detector inputs from *filename.DET*.

- Separate bus-based outputs in .out file (cf SATURN output format).

- Error corrected on results of passenger delay outputs in .PSN and .PAS files.

- Apply user-specific stop-line position to all node types, as specified via .ADD file.

- Allow off-line setting of default colours via parameters such as COL_FAC, COL_LANE, etc for the colour of the clock face, lane marking, etc. This provides facility to set preferred colours for all runs, as oppose to set them on-line during each simulation run. The new parameter names are listed in Appendix B.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 The DRACULA Concept

The dynamic network microsimulation framework **DRACULA** (Dynamic Route Assignment Combining User Learning and microsimulation) has been developed at the University of Leeds since 1993 (Liu et al 1995). It adapted a new approach to modelling road traffic networks whereby the emphasis is on the "microsimulation" of individual trip makers' choices and individual vehicles' movements. The model attempts to represent directly the behaviour of individual drivers and vehicles in real-time as these evolve from day to day. This is coupled with a detailed within-day traffic simulation model of the second-by-second movements of individual vehicles according to car-following, lane-changing rules and traffic controls. In combination they model the evolution of the traffic system over a representative number of days so that both within-day and between-day variabilities are included and interaction between the demand and supply modelled.

The full DRACULA framework combines a number of sub-models. The ***demand model*** represents the day-to-day variability in total demand within a fixed departure time period. It simulates for each potential traveller - based on individual drivers' knowledge of the network, their past experience and perceived network condition of the day - whether to travel, if so, the route to be taken, and the preferred departure time. This information is then passed to the ***traffic simulation model*** which represents the within-day variability of network conditions and simulates individual vehicles movements through the network following the routes chosen and records their travel performance. At the end of the day (the study period), a ***learning model*** updates the experiences of each individual and stores the information in their travel history files which, to a greater or lesser extent, influence their next day's choices.

The current release version, code named **DRACULA-MARS** (Microscopic Analysis of Road Systems), includes only the ***traffic simulation model*** and a simplified ***departure-time choice model*** of DRACULA. This version is designed primarily for existing SATURN users who can combine the SATURN route assignment with DRACULA traffic micro-simulation for detailed network design and/or short-term forecasting. Hence route assignment is an input exogenous to the model.

Within **DRACULA-MARS**, however, there are functions for choice of departure-time to be modelled (see §4.3), and a wide range of network variability and traffic dynamics. These include modelling of:

(a) variability in network demand (such as daily fluctuation in demand and temporal distribution of demand) and supply conditions (such as those due to weather and incidents);

(b) many dynamic phenomena, such as congestion build-up, blocking back, gap-acceptance and merging, more realistically;

(c) complex traffic controls such as flared approaches, shared lanes and dynamic responsive traffic signal controls;

(d) public transport operation and bus priority measures;

(e) dynamic traffic demand management measures (such as congestion pricing).

Through its animation, **DRACULA-MARS** gives a graphical representation of the network performance.

1.2 Overview of DRACULA Traffic Simulation Model

The traffic model in DRACULA is a micro-simulation of the movement of (pre-specified) vehicles through the network. Drivers follow their pre-determined routes and en-route they encounter signals, queues and interact with other vehicles on the road. A large number of such microscopic vehicle models have been developed in the past at varying levels of complexity and network size (e.g. in some the network is effectively a single intersection). An essential property of all such models is that the vehicles move in real-time and their space-time trajectories are determined by, e.g., car following and lane-changing models and network controls such as signals.

The simulation is based on fixed time increments; the speeds and positions of individual vehicles are updated at an increment of one second. Spatially, the simulation is continuous in that a vehicle can be positioned at any point along a link. The model includes an animated graphical display of vehicle movements in the network.

The simulation starts by loading the simulation parameters, network data including global and local variations and trip information (demand and routes determined by the demand model). It then runs through an iterative procedure at the pre-defined time increments, within which the following tasks are performed:

Simulation Loop

a. [Initialisation] Set within-day simulation clock t=0.

b. [Vehicle Generation] Generate new entry vehicles at their preferred departure-time and place them on their entrance links.

c. [Vehicle Movement] Loop through all vehicles in the network, and for each of them:
   i. check if the vehicle wants to change lane, if so whether the gaps are acceptable;
   ii. update the vehicle's speed and acceleration and advance it to its new position. At the end of the link, either remove the vehicle from the network (if it arrives at its destination) or pass it to its next link en-route.
   iii. Calculate vehicle emissions and fuel consumption, and record traffic performance measures;

d. [Animation Update] Update the graphical display if required.

e. [Traffic Control Update] For each signalised junction, update the stage change-over clock according to desired signal plans (fixed plans or responsive). Check if the any incident is to start or to finish.

f. [Time Update] If all drivers have finished their journey, terminate the day; otherwise increment the simulation clock and return to step b.

The simulation loop is depicted in Figure 1.1.

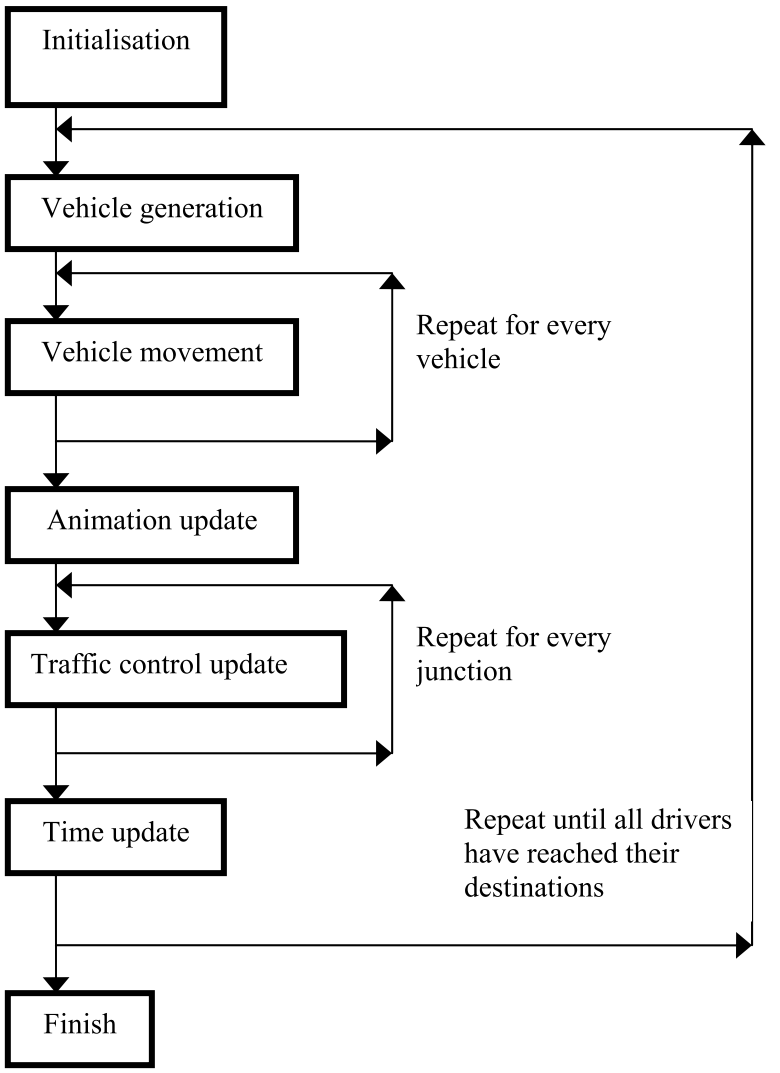


Fig. 1.1. The simulation loop.

Network Representation

The network is represented by zones, nodes, links and lanes. A zone is the source or sink of traffic where they enter or leave the network. A node is either external connecting to a zone or an intersection connecting to other nodes inside the network. In DRACULA, there is no restriction on the number of roads connecting an intersection.

A link is a directional roadway between two nodes and consists of one or more lanes. A link is specified by its upstream and downstream nodes, cruise speed, number of lanes, turns permitted to other outbound links from the downstream node, and restricted access (such as reserved lanes). For each permitted turn, the lane(s) in the link that can use this turn are specified and a marker describing its priority over opposing flows is given.

In the model traffic moves in lanes. A lane can be reserved for a particular type(s) of vehicles only (for example a reserved bus lane). The reservation is specified by its start and end position on the lane and, optionally, by a start and end time of its operation. See §4.2.

Vehicle Characteristics

Vehicles are individually represented; each has a set of individual characteristics including vehicle type (car, bus, guided-bus, taxi, heavy goods vehicle); vehicle length; desired minimum distance headway; normal and maximum acceleration; normal and maximum deceleration; desired speed (relative to the mean speed on any individual link) and a gap-acceptance parameter. These characteristics are randomly sampled from normal distributions representative of that type of vehicle. See §6.1.3.

Public transport vehicles are represented with additional information such as service number, service frequency, bus stops and average passenger flows at each bus stop, etc. See §5.

Vehicle Simulation

Vehicle movements in a network are determined by its desired movement (such as desired speed, lane choice), response to traffic regulations and interactions with neighboring vehicles. The simulation maintains a linked list of vehicles in each lane and moves individual vehicles according to a car-following model and a lane-changing model, and their response to traffic controls at intersections.

The car-following model calculates a vehicle's acceleration in response to its desired speed and the relative speed and distance of the preceding vehicle. Depending on the magnitude of the relative distance, a vehicle is classified into one of three regimes: free-moving, following or close-following.

The lane-changing model contains three steps: (1) obtain the lane-changing desires and define the type of changing, (2) select the target lane, and (3) change lane if all gaps are acceptable.

The model divides drivers' lane-changing desires into one of six types when drivers have to or want to change lane in order to:

(a) reach a bus stop on the link;
(b) avoid a restricted-use lane or incident;
(c) make their turn from the next junction;
(d) move into a lane reserved for their type;
(e) gain speed by overtaking a slower moving vehicle by changing to a lane in the same direction; or
(f) overtake a slower moving vehicle by changing to a lane in the opposite direction of traffic.

Simulation Outputs

As a measure of network performance, the simulation outputs (by default) network-, link- and route-specific measures such as total vehicle-hour, total vehicle-km, average travel time, speed, queue length, fuel consumption and pollutant emission over regular time periods. The length of the report time period can be defined by the user. At the user's request, the program may also output individual vehicles' second-by-second locations and speeds to provide space-time trajectories of the vehicles. A graphical animation of the vehicles' movements can also be shown in parallel with the simulation, giving the user a direct view of the traffic conditions on the network. See §9.

A new program, called **SPATULA** (§9.2), has been written to convert the link-based simulation results into a SATURN-style data file which can then be input to a **SATURN** program **P1X** for display and for comparison with SATURN simulated results.

2 SYSTEM REQUIREMENTS AND ARCHITECTURE

2.1 Computer System Requirements

DRACULA is written in C/C++ and compiled with Salford C/C++ and Salford/ClearWin+ for running on PCs under 32-bit Windows systems (Windows XP, 2000, NT, and earlier Windows systems). A typical specification for a system to run **DRACULA** would include:

Table 2.1 System requirements

	Requirements
Hardware	64MB RAM, 0.5GB hard disk, Windows XP or earlier systems
Software library	Salflibc.dll
Screen setting	True colour (32 bit)

2.2 Model Architecture and Data Requirements

There are two main programs in DRACULA-MARS:

(1) a pre-processor called **DRACPREP** which processes the input data and creates the required environment for the microsimulation; and

(2) a traffic microsimulation program called **DRACSIM**.

The data required for **DRACULA-MARS** includes: the travel demand data (in terms of routes used, flow on each route, and, optionally, drivers' preferred departure times), the network topology, traffic controls and simulation control parameters. Figure 2.1 shows the architecture of DRACULA-MARS and depicts the relationship between the two programs with the input data required and the output data files produced. Details of the input and output files are explained in §2.2.1 and §2.2.2. §2.2.3 explains file name conventions.

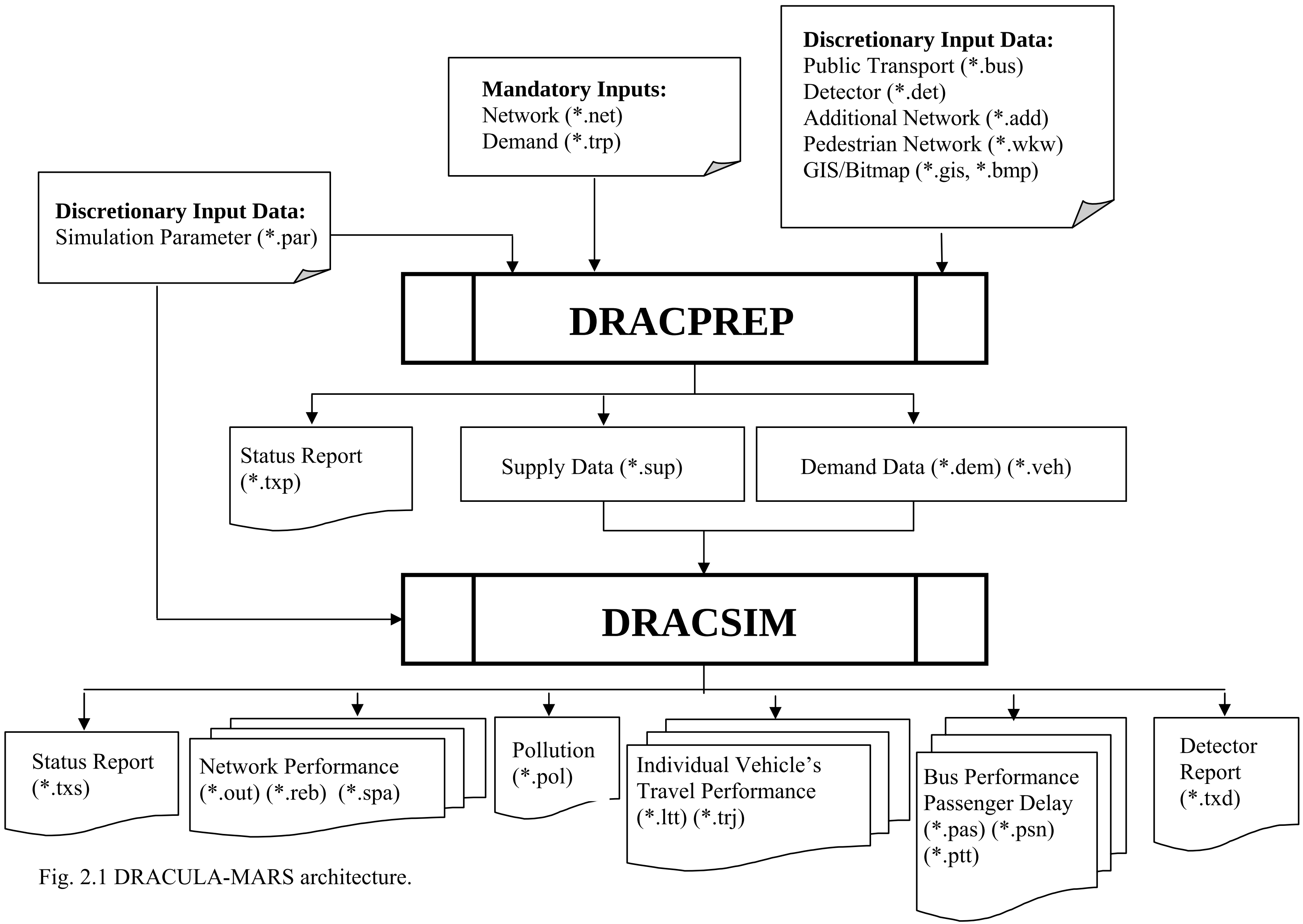


Fig. 2.1 DRACULA-MARS architecture.

2.2.1 DRACPREP

Function: **DRACPREP** creates the database, including the network topology, traffic regulation and traffic demand, which is to be used by **DRACSIM**. Tables 2.2 and 2.3 list the input and output data files for the program.

Table 2.2 Input data files for ***DRACPREP****.*

Data Type	**Details of Input Data**	**File Extension**
Basic inputs (mandatory)	Network description. See Section 4.2	.NET
	Travel demand in terms of routes and route flows (Section 4.3)	.TRP
Selective Inputs (discretionary)	Control parameters (Section 4.1).	.PAR
	Public transport services (Section 5.2)	.BUS
	Selective vehicle detector information (Section 8.2).	.DET
	Additional network data inputs (Section 6.2).	.ADD
	Pedestrian network and flows (Section 8.6).	.WKW

Table 2.3 Outputs from ***DRACPREP****.*

Data Type	**Details of Output Files**	**File extension**
Default	Text reports of the status of the process	.TXP
	Network supply description.	.SUP
	List of route and route flows.	.DEM
Optional	List of individual trip departure-time, route and vehicle characteristics. See 4.3.2.	.VEH

The text messages in file *.TXP* contain useful information on the status of the **DRACPREP** process. If **DRACPREP** fails to complete successfully, you will be prompted with a screen message asking you to check in the file for error messages. The file also contains warning messages; one particular warning is about the inconsistency between coded link length and crow-fly distance between node coordinates (Section 4.2.2.2). It is advisable to check the file for any warning and error messages before proceeding on to **DRACSIM**.

The data in *.SUP* and *.DEM* are information on the network description and aggregated demand re-formatted for **DRACSIM**. They are not meant for users to look at, nor to change. The outputs in *.VEH* list individual trip's departure-time, identification of the route it is to take, and characteristics of the vehicle. A detailed description of its format is given in Section 4.3.2.

2.2.2 DRACSIM

Function: **DRACSIM** performs the actual simulation of individual vehicles' movement in the network following fixed routes. Tables 2.4 and 2.5 list the input and output data files for the program.

Table 2.4 Input data files for ***DRACSIM****.*

Data Type	**Details of Input Data**	**File name and extension**
Basic inputs (mandatory)	Network supply description	.SUP
	List of routes and aggregated route flows	.DEM
Selective inputs (discretionary)	Simulation control parameters	.PAR
	Individual trip departure-time, route and vehicle characteristics.	.VEH
	GIS features. See Section 10.1	.GIS
	Bitmap background file, and associated coordinates. Section 10.2	.BMP .XYB
Global input (discretionary)	Specification of vehicle characteristics (Section 6.1.3).	VEH.TAB
	Emission and fuel consumption factors (Section 7).	POL.TAB

The "basic" and "selective" inputs are network dependent, whilst the "global inputs" are not.

Depending on the choices provided by the user in the parameter control file, different levels of details are output ranging from the most detailed second-by-second vehicle trajectories, to aggregated link, route and network measures. Format of the output files are described in Section 9.

*Table 2.5 Outputs from **DRACSIM**.*

Data type	Details of Output Files	File extension
Default	Text message reporting the status of simulation	.TXS
	Summary statistics on travel performance measures	.OUT
	Summary statistics on exit measures	.REB
	Network and link performance measures for use with SPATULA	.SPA
	Summary statistics on pollution measures	.POL
	Individual vehicles' link travel time	.LTT
Optional	Individual buses link travel time	.PTT
	Summary of passenger delays and bus dwell times	.PSN
	Detector records of passing vehicle' speed and type	.TXD
	Trajectories for selected vehicles	.TRJ

2.2.3 File Names

Unless specified otherwise via the method described below, all input data files for each network should be stored in the same directory. The outputs files from **DRACPREP** and **DRACSIM** will be in the same directory. Both the file name and the directory name should be a string of alpha-numeric characters without a white space in between. For example, a name should not be named as *OTLEY BASE***,** nor should it be stored in directory **c:\Program Files**.

The extensions of all the input and output files used in **DRACULA** (As listed in Tables 2.3-2.5) are fixed. For example, a file used to specify an input network description has to have an extension of *.NET*, and the output pollution measures can only be found in a file with extension *.POL*.

By default, all the input data files for the same network should have the same name and it is the name of the *.NET* file. For example, a network called ***OTLEY*** will have its parameter control file named *OTLEY.PAR*, trip file *OTLEY.TRP*, etc.

However, different names can be used for different input files for the same network. This can be done via parameters in the *.PAR* file (see Section 4.1 for the format of *.PAR* file). Table 2.6 lists the parameters used ("PARAMETER") and the extension of their corresponding files ("EXTENSION"). Only the names of the files need to be entered; **DRACPREP** and **DRACSIM** will automatically pick up the relevant extensions. For example, to specify *otley.BUS*, set *BUSFIL=otley*. The names can also be put inside inverted commas as in SATURN coding, e.g. for the above example *BUSFIL='otley'*.

Table 2.6 Parameters used to specify the file names.

PARAMETER	EXTENSION	NOTE
FILTRP	.TRP	* Input data, or
FLTRP(t)	.TRP	* Input trip data for time period t. Section 4.3.1.2
FILBUS	.BUS	* Input data
FILDET	.DET	* Input data
FILADD	.ADD	* Input data
FILWKW	.WKW	* Input data
FILGIS	.GIS	* Input data for dracsim
FILBMP	.BMP	Input data for dracsim
FILSUP	.SUP	* Transient data
FILDEM	.DEM	* Transient data
FILVEH	.VEH	* Input/output data
FILTXP	.TXP	Output data
FILLTT	.LTT	Output data
FILOUT	.OUT	Output data
FILPOL	.POL	Output data
FILREB	.REB	Output data
FILSPA	.SPA	Output data
FILTXS	.TXS	Output data
FILPTT	.PTT	Output data
FILPAS	.PAS	Output data
FILPSN	.PSN	Output data
FILTXD	.TXD	Output data
FILTRJ	.TRJ	Output data
FILPIG	.LPG	Input data. Section 4.3.3

This facility is useful to allow the same input file be used in a number of different scenario testing. For example, the same demanded trips *OTLEY.TRP* can be used as basis for two different network descriptions *OTLEY1.NET* and *OTLEY2.NET***.**

N.B.

Before **DRACPREP** can load in the correct network-coding format (see the two different ways of coding networks in Sections 4.2.1 and 4.2.2), the relevant parameter needs to be specified in *.PAR* file. Hence, the *.PAR* file has to bear the same name as that of the *.NET* file.

As an experiment, those filenames marked with * marked can also be specified in the network description file *.NET*.

Example

```
PARAMETERS
 PERIOD(MIN)=60
 NSEED=900
 LEFTDR=T
 QBUS=T
 FILTRP='otley.trp'
 FILBUS='otley.bus'
 FILADD='otley.add'
 FILDET='otley.det'
END
```

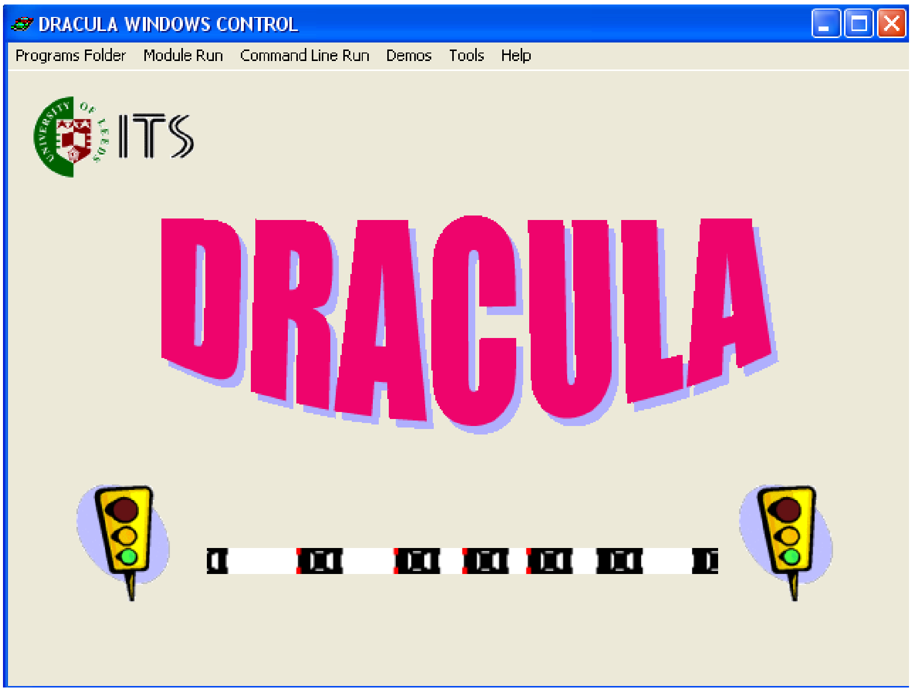
3 EXECUTING THE PROGRAMS

Programs within the **DRACULA-MARS** suite can be accessed using either the DRACWIN (Section 3.1) or the SATWIN (Section 3.6) windows-based front ends, or via batch mode commands under Command Prompt (Section 3.7).

3.1 DRACWIN Front-End

The **DRACPREP** and **DRACSIM** can be accessed via a Windows front-end for DRACULA from a program called ***DRACWIN.EXE***

Double click the icon '**DRACWIN**' on the desktop or **DRACWIN.EXE**, the interface 'DRACULA WINDOWS CONTROL' pops up to load the front-end.



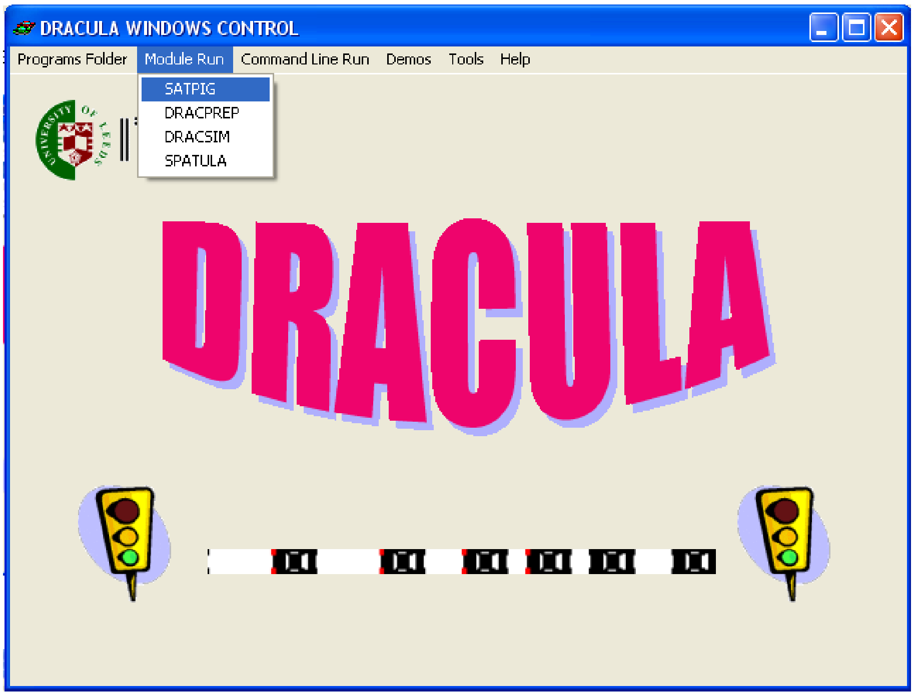
There are six menus on the top main menus:

Programs Folder:	To change the path of executive files, e.g. DRACSIM.exe, DRACPREP.exe, and SPATULA.BAT
Module Run:	To execute the programs: DRACPREP, DRACSIM, SPATULA
Command Line Run:	To run the executive programs with DOS Command
Demos:	To display DRACULA simulation runs on demonstration networks
Tools:	To access DOS/Prompt
Help:	Version Information

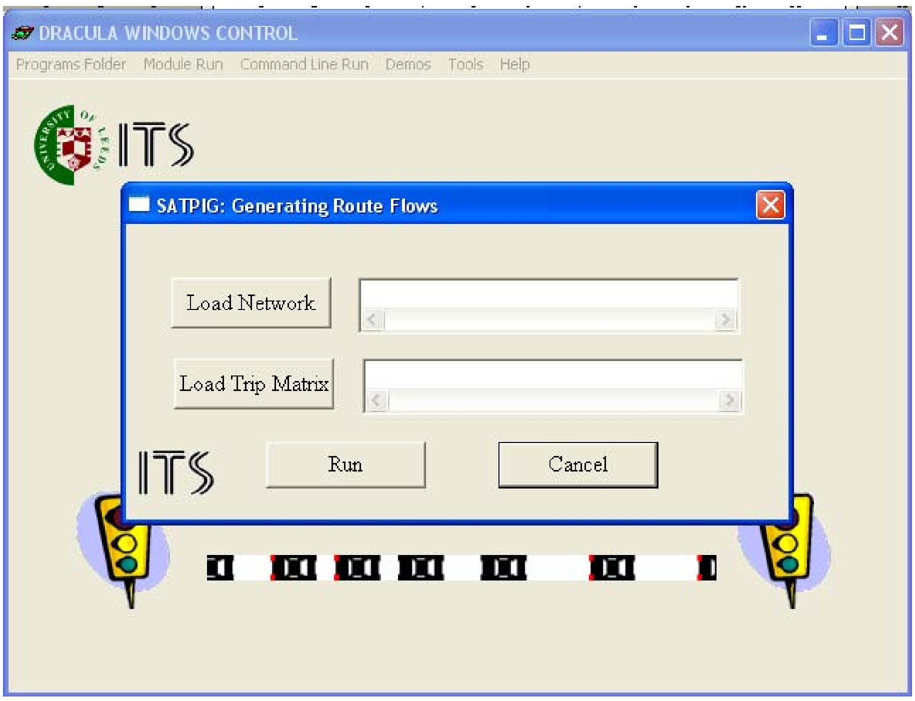
3.2 Running the SATPIG Program

SATPIG is a SATURN program to produce a text file of origin-destination route flows from a SATURN assignment. It was written in particular to facilitate interface with DRACULA; its output route flows are written to the .trp format required by DRACULA.

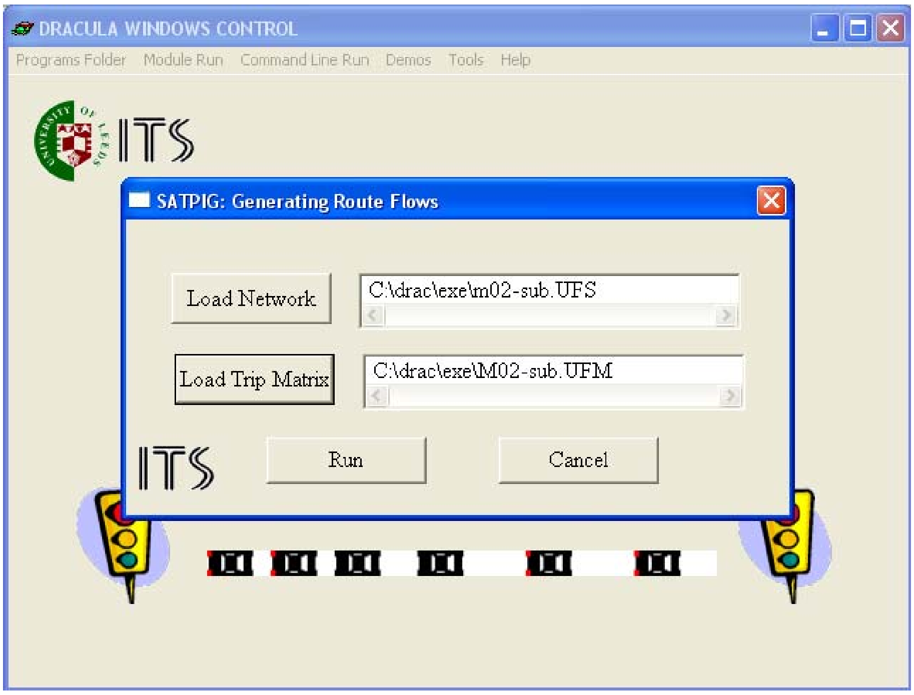
To start **SATPIG,** select from main menu 'Module Run'-'SATPIG'.



The dialogue 'SATPIG: Generating Route Flows' then pops up.



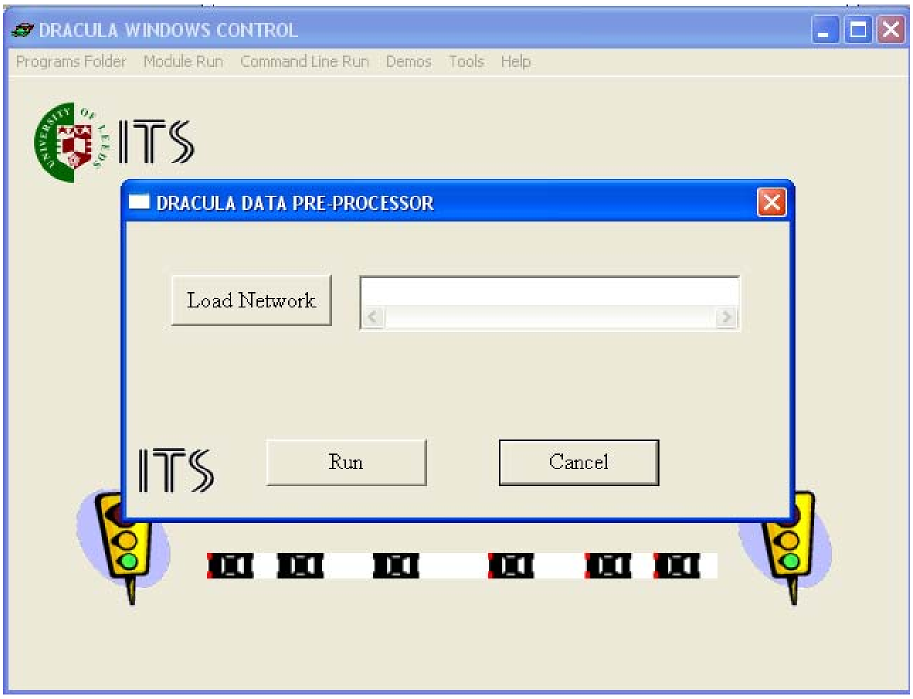
Step3. Click ‘Load Network’ button to load network (*.UFS), and click ‘Load Trip Matrix’ button to load trip matrix (*.UFM), or, type the network full path name (e.g. d:\Dracula\net1.ufs and d:\Dracula\net1.ufm) into the text boxes.



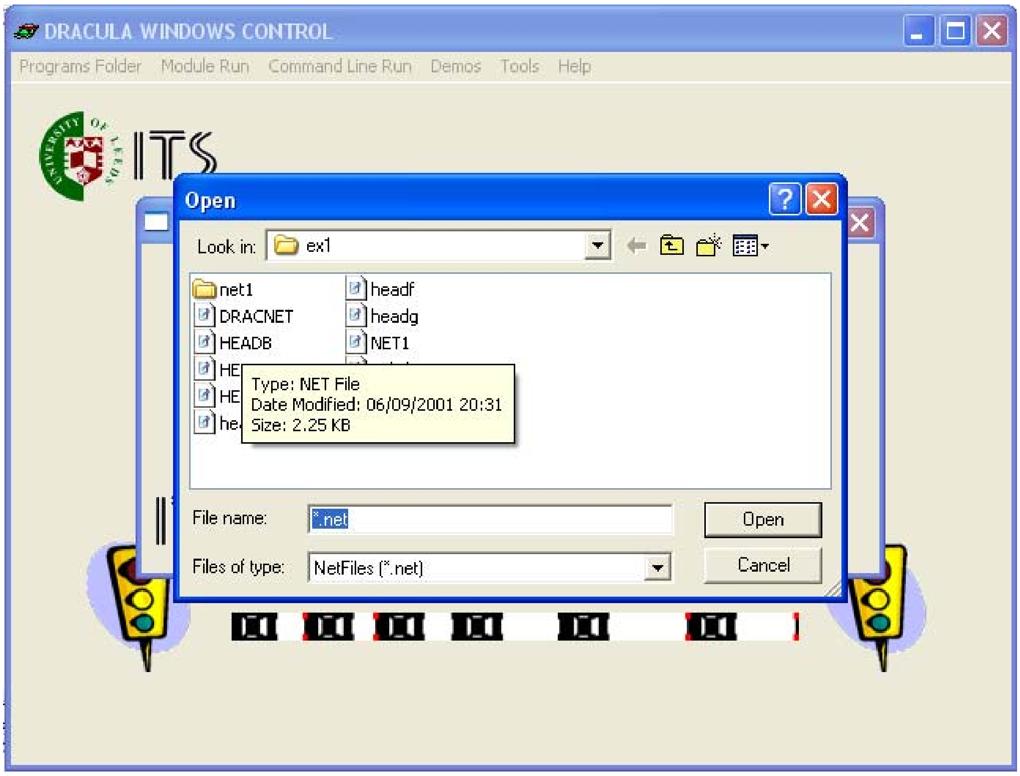
Finally, start the SATPIG by clicking ‘Run’ button.

3.3 Running the Pre-Processor

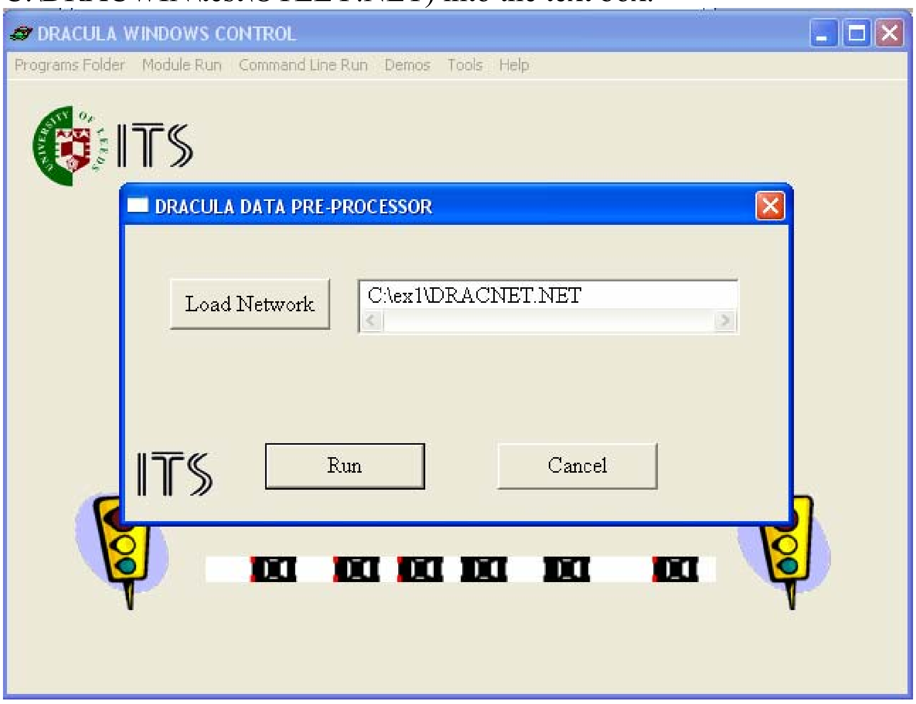
Start DRACPREP from main menu 'Module Run'-'DRACPREP'. The dialogue 'DRACULA DATA PRE-PROCESSOR' then pops up.



Click 'Load Network' button to load network, or, type the network full path name (e.g. d:\Dracula\net1.net) into the text box.



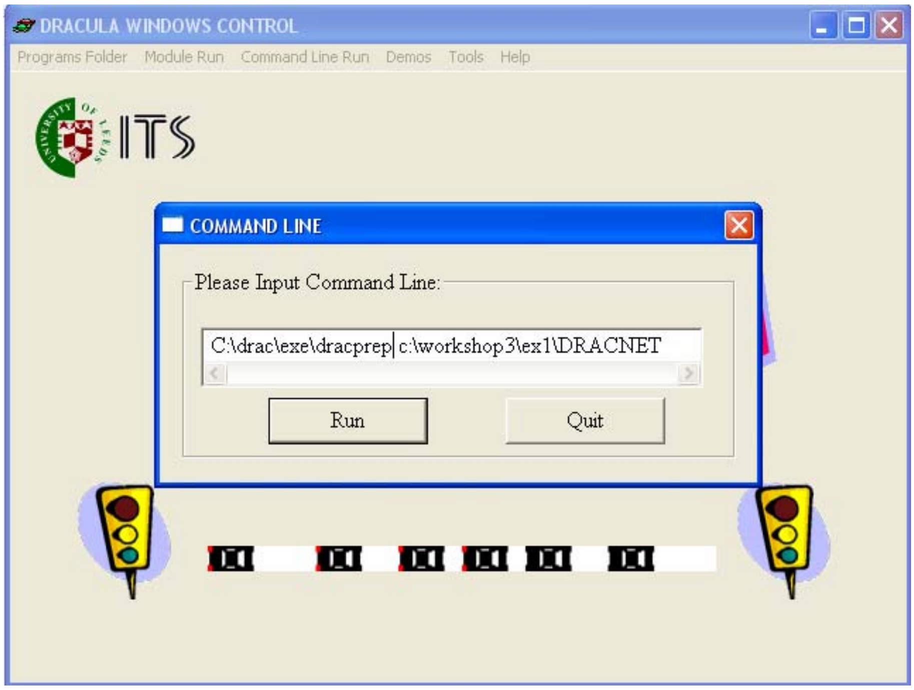
OR, input the network full path name (e.g. C:\DRACWIN\test\OTLEY.NET) into the text box.



Step 3. Start the pre-process by clicking 'Run' button.

Alternatively, the above steps can be carried out with one command from main menu-'**Command Line Run**' Prompt or DOS/Prompt, for example:

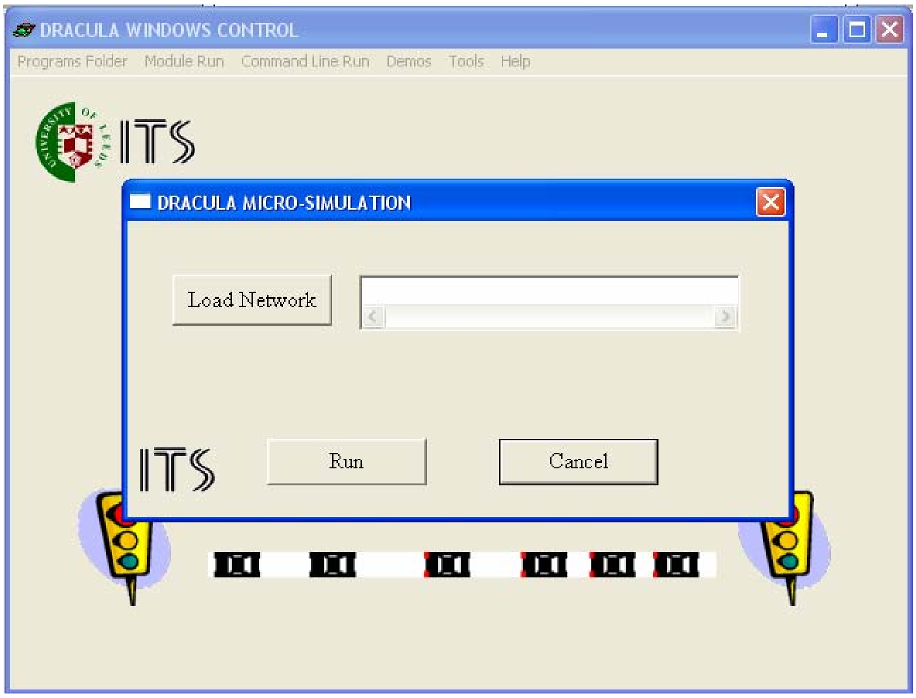
D:\DRACULA\DRACPREP *netname*



3.4 Running the Simulation

Step 1. Load the network from main menu ‘Module Run’-‘DRACSIM’.

Step 2. The dialogue ‘DRACULA MICRO-SIMULATION’ pops up.



Step 3. Click ‘Load Network’ button to load network.

OR, input the network full path name (e.g. C:\DRACWIN\test\OTLEY.SUP) into the text box.

Step 4. Start the simulation by clicking ‘Run’ button.

Alternatively, the above steps can be carried out with one command from main menu-‘**Command Line Run**’ Prompt or DOS/Prompt, for example

D:\DRACULA\DRACSIM *netname*

3.5 Running the SPATULA Data Processing

SATURN users also benefit from a new program **SPATULA** which converts DRACSIM time-dependent link performance measures to SATURN format. See Section 9.2 for details on how to load SPATULA outputs into **P1X** for data display.

Step 1. Run **SATNET** on the network data file to produce **.UFN** file

Step 2. Load the program from main menu 'Module Run'-'SPATULA'.

Step 3. Click 'Load Network' button to load network.
or,
Type the network full path name (e.g. C:\Dracula\net1.spa) into the text box.

Step 4. Start the process by clicking 'Run' button.

3.6 Accessing the Programs from SATWIN Front-End

SATURN users can access the **DRACULA-MARS** programs from within the SATURN Launch Pad, **SATWIN**. Select menu:

SATWIN/Module Run/DRACULA

which will lead you to the **DRACWIN** front-end. Follow the instructions in Section 3.2-3.5 to access the DRACULA-MARS programs.

3.7 Running the Programs in Batch Mode

Alternatively, the programs can be run with two commands from the main menu "**Command Line Run**" Prompt or DOS/Prompt as:

DRACPREP *netname*

DRACSIM *netname*

SPATULA *netname*

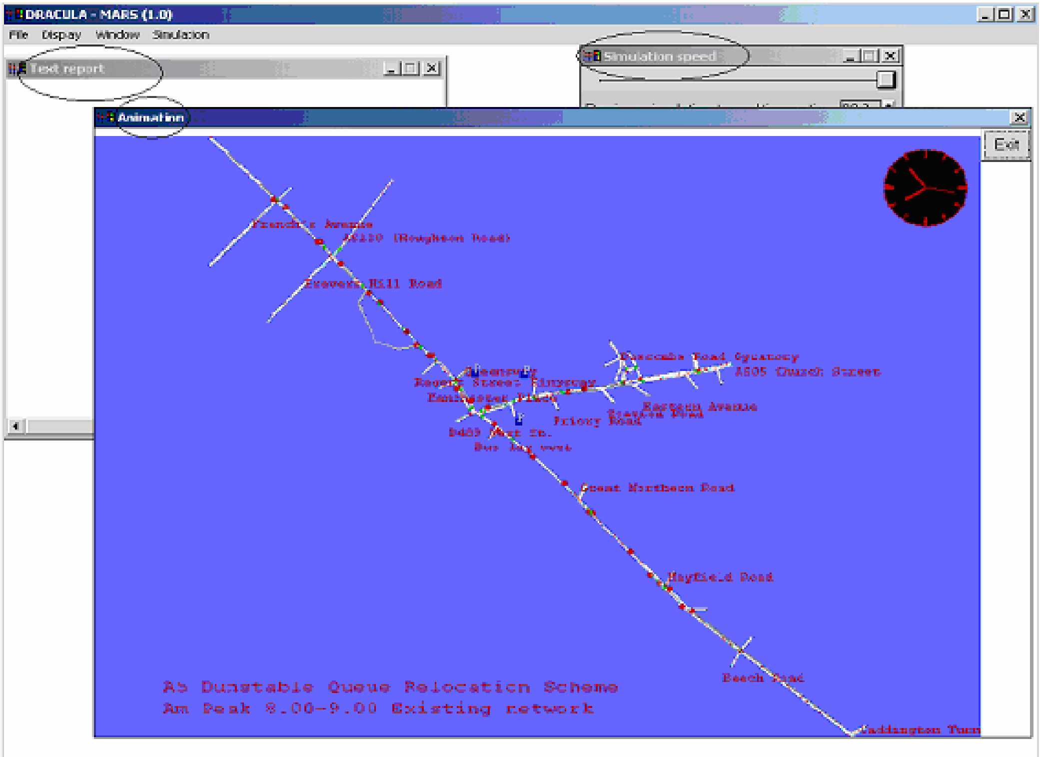
To load a bitmap image together with the network, type in command:
DRACSIM *netname bitmap*

3.8 The DRACSIM Window

3.8.1 The Windows

Upon starting the simulation, you will get a main (parent) window with a title bar across the top, reading "DRACULA – MARS (#.#)", where (#.#) gives the version number of the software.

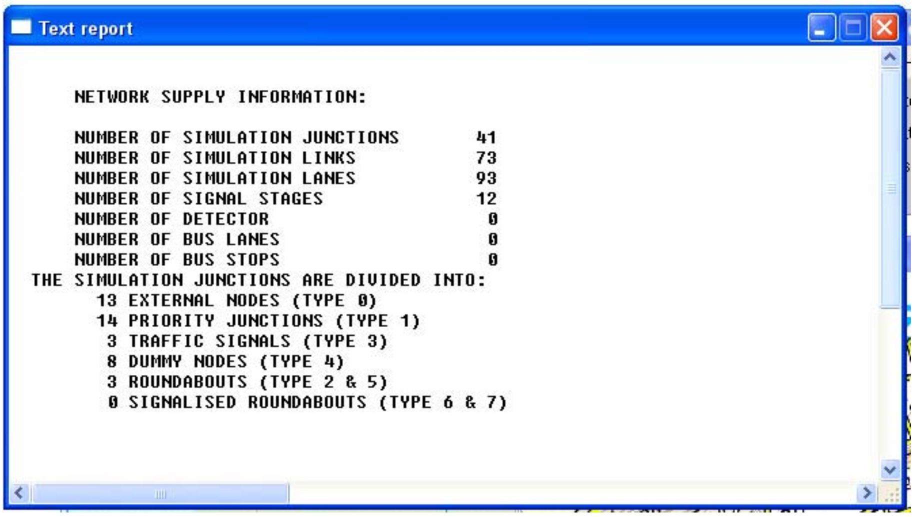
Inside this main DRACULA screen are three child windows: "Animation", "Text report" and "Simulation Speed". At the start of a simulation, the animation window lies on top of the text window. By clicking anywhere on any of the other two windows, you can bring it to the front. The child windows are restricted within the parent window.



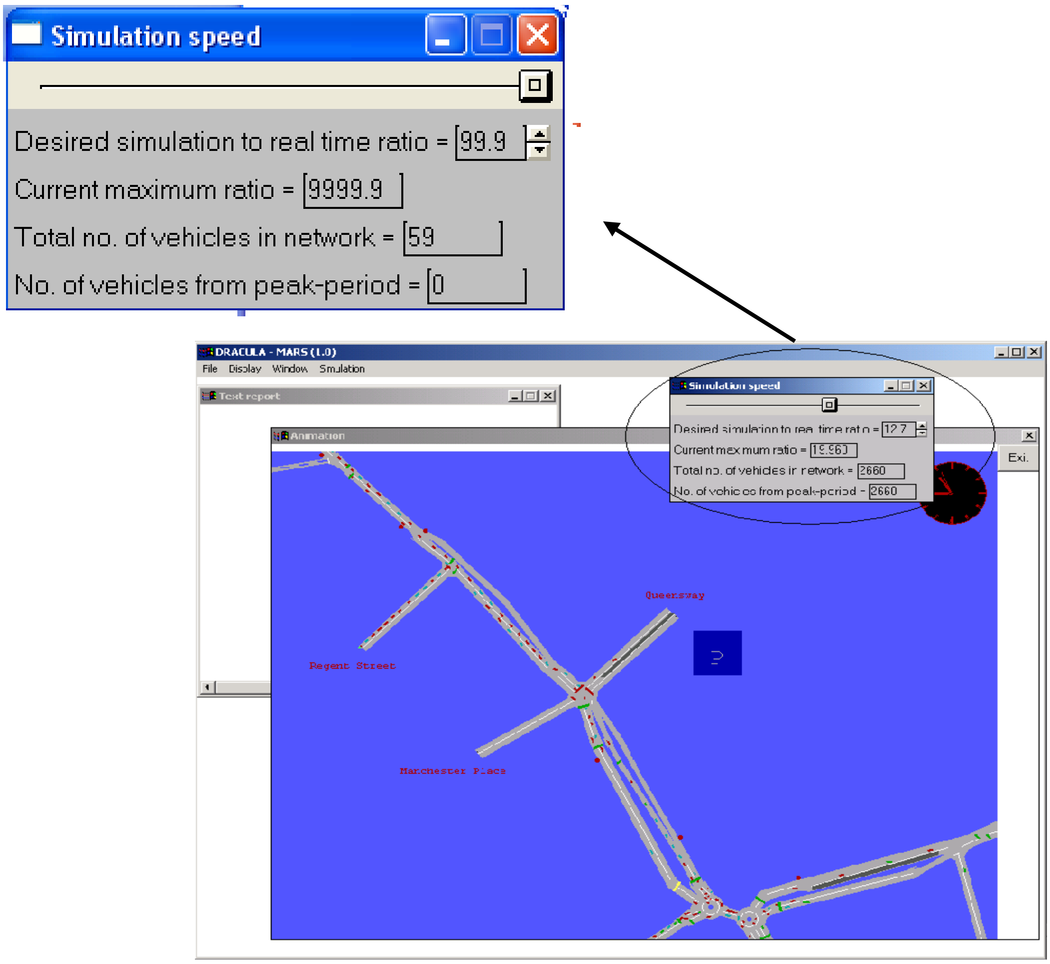
On top of the parent window are four menu bars which provide user interfaces to the simulation program. The details of the menu options are described in Section 3.7.2 and Appendix C.

The animation window displays the simulated network. To get a closer view of any part of the network, use the left-hand button on the mouse to select the area of interest, then click on menu option ***View/Box*** to zoom in.

The "text report" window is used to display information about the simulated network and messages regarding the simulation process at the end of the simulation. At the start of the simulation, the screen is blank.



The “simulation speed” window shown below allows users to control the simulation speed, normally turning the simulation slower so to be able to see the vehicle movements more clearly. The window also displays the number of vehicles in process and the number of vehicles from the demand period to be processed. The simulation ends when all vehicles from the demand period have completed their journey.



3.8.2 The Menus

Appendix C provides a full of the menu options in **DRACSIM**.

3.8.2.1 "File" Menu

Load network - Enable the user to load the network to be simulated
Load BMP map - To load bitmap background map
Screens - Enable the user to choose two colour settings: 256 colours or 16/32 bits

3.8.2.2 "Display" Menu

Network Info - Related information given in the Text Report Window
Demand Info - Related information given in the Text Report Window

Node Number - Show the nodes numbers on the network

Bus-stop Number - Show the bus-stop numbers

Background map - toggle between display or not the bitmap background map

GIS-ALL - Ditto for all of the network GIS information coded in the .GIS input data file, except for curved link which will always be displayed if given in .GIS file.

GIS-polygon - Ditto for GIS polygons

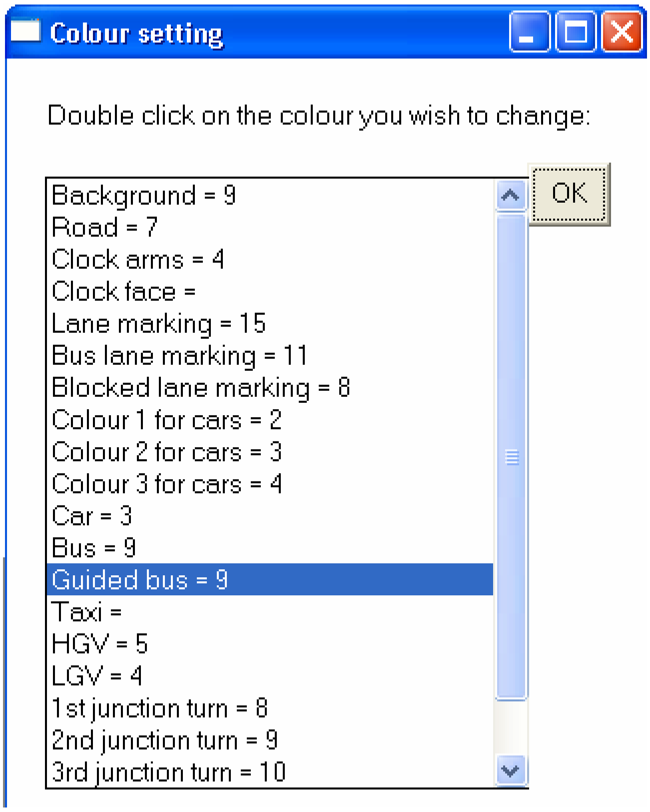
GIS-polyline - Ditto for GIS poly-lines

GIS-icons - Ditto for GIS icons

GIS-text - Ditto for GIS texts

GIS-CurvPt - Ditto for the points along a curved link

GIS-CurvPt2 - Ditto for the points along, and the A- and B_node number of, a curved link
Colours - A window will pop up which allows changes to the default colour settings for the animation.



Click on the colour to change and enter the corresponding integer number as defined in Table 3.1.

Table 3.1 Colour coding

Code	Colour
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	GREY
8	DARKGREY
9	LIGHTBLUE
10	LIGHTGREEN
11	LIGHTCYAN
12	LIGHTRED
13	LIGHTMAGENTA
14	YELLOW
15	WHITE

3.8.2.3 "View" Menu

Full Network - enables the whole simulated network to be displayed in the animation window.

Full background - ditto for background map

Window - Option to control viewing changes to network only, background only, or both.

Box - enable the user to get a closer view of any part of the network. Use the left-hand button on the mouse to select the area of interest, then click on menu option ***Box*** to zoom in.

Options ***Zoom/Pan/Left/Right/Up/Down*** enable the user to move to any interested parts of the simulated network.

The last five options, ***Scale2Pt BMP, Drag BMP, Set BMP X/Y, Save BMP X/Y,*** and ***Fit whole BMP to screen*** enable scaling and positioning of bitmap background map.

3.8.2.4 "Simulation" Menu

Display vehicle by - provides options to display the colour of vehicles by type, junction turning, route or randomly.

Animation by - options to display the animation by individual vehicles, or aggregated link delay.

4 BASIC HIGHWAY TRAFFIC NETWORK AND DEMAND

The input data required to run DRACULA-MARS are divided into fundamental data and advanced data. The former is essential to run the model, while the advanced data allows optional model features to be activated. This section provides an introductory discussion on the structure and the fundamental input data files that are used to build a basic highway traffic network model.

Most of the data are input via text files which can be created or edited with a text editor (e.g. **EDIT**, **BRIEF**, **WORDPAD** or **NOTEPAD**). It is advisable not to use any word-processor for the editing. SATURN users can use **PMAKE** for network editing.

4.1 Control Parameters File

The control parameters define the options for input data, output data and traffic simulation. They are stored in an ASCII text file with the same name as that of the network description and extension *.PAR*. All parameters have default values; so if a parameter is not specified, its default value is used. If only the default values are used for this run, the file does not need to be created - hence it being one of the selective inputs (see Section 2.2).

The parameter should contain at least a beginning and an end identifier: PARAMETERS and END. Any parameter to be specified must always be set between the two identifiers. Any text or space outside identifiers PARAMETERS and END is ignored by the program, and so can be used to insert comments.

Comments can be put on the same line as the parameters, but they must be (a) entered after the parameter definition and (b) separated from the parameter definition by at least one space. Parameter names are case insensitive, so for example, parameter “gonzo” is read in as the same as “GONZO”.

The parameter values can be logical (T or F stands for TRUE or FALSE respectively), integer or real parameters. In the case of a real parameter value, the number can be input as either real value or integer numbers.

Some of the parameters are used in the pre-processing stage, others affect the behaviour of the simulation. The list of user-definable parameters is long, many of the variables and their use are complicated. The full list of the parameters used in **DRACPREP** and **DRACSIM** are given in Appendix A and B respectively, with pointers to appropriate sections. A subset of parameters, given in Tables 4.1 and 4.2 for **DRACPREP** and **DRACSIM** respectively, is suggested as reasonable starting points where explicit choices need to be made.

Table 4.1 Parameters defining the options of basic input to ***DRACPREP***

Parameter	Defaults	Definition
NSEED	800	Random number seed. An integer number (preferably of three or more digits) to trigger a pseudo random number generator. Used both in DRACPREP and DRACSIM.
QSATNET	T	T if to take SATURN-format simulation network; F to use the free-format network coding as described in Section 4.2.2.
QSATPIG	F	F if to take routes and route flows from input file *.TRP* of format defined in Section 4.3.1; Else if T, take them from *.LPG* files (4.3.3)
QBUS	F	T if to load bus service, bus stops and bus lanes from *.BUS* file (Section 5.2.1)
LEFTDR	T	F if driving on the right (e.g. European driving)
XYUNIT	1.0	Factor converting one unit of node/zone coordinates to metres. Default: 1coordinate unit =1metre. If XYUNIT=10, then 1 unit of coordinate=10m. Same as used in SATURN, so can also be specified in the network description file *.NET* (Section 4.2.1)
GONZO	1.0	Demand factor: number of vehicles (of all types) are factored by GONZO.
VCPCU	1.0	Per car unit used to convert flow given in pcu or pcu/hr to number of vehicles or vehicles/hr. Default: 1 pcu = 1 vehicle of type CAR. See Sections 4.3.3.1.
CIRC_SPEED	27	Roundabout circulating speed (km/hr). See Section 4.2.2.3
CIRC_RADIUS	4	Minimum radius of roundabout central island (metres). See Section 4.2.2.3.
QRBLANE	T	If T the number of lanes on a roundabout is limited to at least equal to the maximum lanes on entry arms. F if the circulating lanes are determined from SATURN roundabout capacity. See Section 4.2.2.3.
NSTEP	1	Number of time periods. See Section 4.3.1
NMUC	1	Number of user classes. See Section 4.3.2

*Table 4.2 Parameters defining the basic options for **DRACSIM***

Parameter	Defaults	Definition
NSEED	800	Random number seed
NSEED2	4321	Random number seed used to generate random arrival headways.
TMAIN	60	Simulation time period (minutes). Over-written by LTP in SATURN network coding (Section 4.2.2.1)
TWARM	10	Warm-up period (minutes)
TCOOL	10	Cooling-off period (minutes)
TOUTPUT	10	Frequency of output statistics (minutes)
PPM	1.0	Pence per minute, used for calculating generalised costs. See Section 9.3
PPK	0.0	Pence per km. Section 9.3

An example of a **.PAR** file is shown below.

Example

```
&PARAM
 CIRC_SPEED=20
 NMUC=3
 MUC(1)=1
 MUC(2)=8
 MUC(3)=16
&END
```

Tip

Note that the parameters file needs to be created only if non-default values are to be used. For example, using the above file, the roundabout circulating speed would be reduced to 20 km/hr from the default 27 km/hr (see Section 6.2). There are three user classes in the traffic flow; they are of vehicle types CAR, TAXI and LGV (see Section 6.1.2 for definitions of vehicle type).

4.2 Basic Highway Network

The network data is stored in file *.NET*.

A DRACULA network is represented by zones, nodes, links and lanes. Zones are an abstract concept representing the sources or sinks of traffic where they enter or leave the network. They could be used to represent a housing estate, a car park, or simple connections with traffic from outside the network.

A node is either external connecting to a zone or an internal intersection connecting to other nodes. There is no restriction on the number of roads connected to an intersection.

A link is a directional roadway between two nodes and consists of one or more lanes. A link is specified by its upstream and downstream nodes, cruise speed, number of lanes, and turns permitted to other outbound links from the downstream node. For each permitted turn, the lane(s) in the link that can use this turn are specified and a marker describing its priority over opposing flows is given. A lane can be reserved for a particular type(s) of vehicles only (for example a reserved bus lane). This section defines coding of a basic highway network which has no reserved lanes; the latter is described in Section 5.4.

Zones and nodes are named with a number. There should be a unique number for each zone and a unique number for each node. The numbers do not need to be sequential. In addition nodes and zones may have the same numbers. DRACULA network coding adopted a convention used in SATURN to distinguish a zone number from a node number by including the character "C" with the zone numbers.

A *.NET* file may use one of two different coding conventions: (a) that used by **SATURN** networks (Section 4.2.2) and (2) a DRACULA-specific set of records (4.2.1). Selection of (a) is made according to parameter QSATNET=T to be specified in .PAR file. Note that the SATURN-formatted files contain a lot of data that is not required by **DRACULA** and equally do not contain all the information which **DRACULA** requires. Nevertheless they offer a convenient starting point for users with existing SATURN networks.

Tip

4.2.1 DRACULA Network Coding

To describe a basic highway network, there are five sections of records; records 1 and 5 are discretionary whilst records 2-4 are mandatory. Each section is preceded by a line containing a string beginning with character "&" and followed by a section name, and terminated by a line containing a single record 99999. The five sections are:

1. Section 1 – File name description. Preceded by "&FNAME".

2. Section 2 – Node and link description. Preceded by "&LINKS". Within this Section, there are three types of records:

    1) Record type 1 – Node description (mandatory). One record for each node.

    2) Record type 2 – Link description (mandatory). One record for each link in strict clockwise[1] order but starting with any arbitrary link.

    3) Record type 3 – Traffic signal stage description. Only required for traffic signalised nodes. One record per stage is input.

3. Section 3 – Description of zones. Preceded by "&ZONES". One record per zone.

4. Section 4 – Node/zone coordinates. Preceded by "&COORD". One record per zone/node coordinates.

5. Section 5 – Partially signalised roundabout. Preceded by "&SRBS". See Section 4.2.3 for details.

Coding instructions and a worked example for a basic network are given in the following sub-sections. The "NAME" provides an convenient shorthand for data fields for later discussion, while "DATA" column records the number of fields for the record.

Data fields are free-format; there is no column restriction. The only requirements are: (a) different field being separated by a space unless specified otherwise, and (b) each record type being entered on a separate line with a line-break immediately following the last data entry of the record. There should be no white space(s) at the end of a record.

4.2.1.1 File Name Specification

File names in this section must be preceded by a single line containing the string "&FNAME" and terminated by a single line containing "99999". The names should be entered one per line. The following files can be specified this way are listed in Table 2.6, marked with an *.

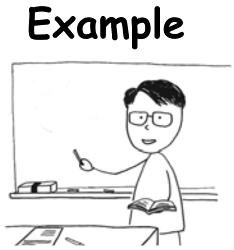
[1] Counter-clockwise under drive on the right

An example of filename specification is shown below:

```
&FNAME
 FILTRP=ottrip1
 FILBUS=otbus
99999
```

or

```
&FNAME
 FILTRP='ottrip1.trp'
 FILBUS='otbus.bus'
99999
```



4.2.1.2 Node and Link Data Formats

Data in this section must be preceded by a single line containing the string "&LINKS" and terminated by a single line containing "99999".

For internal nodes, all three record types of Section 2 data are required to give a complete description of the nodes, its links, and signal timings (for traffic signal control intersection). For external nodes, only those starred (*) variable "NAME"s of record types 1) and 2) need to be included.

DATA	NAME	DESCRIPTION
***********RECORD TYPE 1 – NODE DATA ***************		
1	NODE*	Node number
2	NIN*	Number of arms at the node
3	JTYPE*	Node type: 0 - for external nodes 1 - for priority junctions 2 - for roundabouts with or without U-turn 3 - for traffic signals 4 - for a dummy node (5 not in use currently) 6 - for a new roundabout model, see 4.2.3. (7 not in use currently) 8 - for giveway-to on-coming traffic, see 4.2.5 9 - for giveway-to-the-right, see 4.2.4.
4	QYELL	a character "y" or "Y" entered immediately after JTYPE – no space in between JTYPE and QYELL to indicate the junction is in yellow-box control. See Section 6.2.
5	NSTAGE	Number of stages – traffic signals only (JTYPE=3)
	or	or
	HICD	half inscribe circular diameter (metres) – roundabouts only (JTYPE=2 or 6), see 4.2.3
	or	or
	ZERO	0 for other junction type
6	OFFSET	Relative offset – traffic signals only
	or	or
	NRBL	Number of roundabout lanes – roundabouts only
	or	or

	ZERO	0 for other junction type
7	LCY	Cycle time for this node (seconds), for traffic signals
	or	or
	CSPEED	Roundabout circulating speed (kph), see 4.2.3 and 6.2.
	or	or
	ZERO	0 for other junction type
8	GAP	Junction-specific gap-acceptance value (seconds). See 6.2.1.
	or	or
	BLANK	left blank if not to specify

******** RECORD TYPE 2 – LINK DATA ********

One record for each of the NIN links:

1	ANODE*	Node at the upstream end of the link
2	LANES*	Number of entry lanes for this link 0 if the link is one-way from 'NODE' to 'ANODE'
3	SPEED *	Link free-flow speed (kph)
4	IDIST*	Distance (meters) measured from the centre of 'NODE' to the centre of 'ANODE'. See 4.2.2.

Data for the first turn from this link, clockwise from left for left-hand driving and anti-clockwise from right for right-hand driving condition:

5	LANE1	First lane used by this turn (counted from the nearside/kerbside, kerbside lane=1) 0 if the turn is banned.
6	TPM	Turn priority marker for this turn. This is to be entered immediately following LANE1 – there should be no space between LANE1 and TPM. See 4.2.1.5.
7	LANE2	Last lane used by this turn. 0 if the turn is banned.

Data for the second turn from this link:

…. Same entries as for the first turn

Data for the last turn from this link. The maximum number of turns (being possible or banned) equals to the number of arms to the junction.

…. Same entries as for the first turn

******** RECORD TYPE 3 – SIGNAL DATA ********

One record for each of the NSTAGE stages:

1	STAGL	Duration of stage (seconds)
2	INTG	Duration of following inter-green (seconds)
3	GNA(1)	The entry-node for the first turn
4	GNC(1)	The exit-node for the first turn
5	GNA(2)	The entry-node for the second turn
6	GNC(2)	The exit-node for the second turn

etc etc… with all permitted turns coded on the same line

NOTES:

(i) If any exit-node GNC() above is zero it is assumed that ALL turning movements from that entry node GNA() (except of course any prohibited movements indicated by zero turning lanes LANE1 and LANE2) are allowed.

(ii) Code all movements in one stage on the same line.

(iii) In versions priori to January 2004 (version 2.1), at most two links are allowed in any one stage. There could be as many turns from the two links as required in each stage. This restriction has now been removed.

4.2.1.3 Zone Data Formats

Data in this section must be preceded by a single line containing string “&ZONES” and terminated by a single line containing “99999”.

DATA	**NAME**	**DESCRIPTION**
1	ZONE	Zone number
2	NODEA	Upstream node of the link connecting the zone
3	NODEB	Downstream node of the link connecting the zone

N.B. Strictly speaking, node NODEA should be of type “external” where traffic enters to and leaves from the network. In SATURN-formatted networks (Section 4.2.2), it is possible that a zone is connected to an internal node. In that case, DRACULA treats the node as though it is “external” such that no traffic control applies to traffic entering/leaving the network at that node.

4.2.1.4 Node and Zone Coordinates

Data in this section must be preceded by a single line containing string “&COORDS” and terminated by a single line containing “99999”.

DATA	**NAME**	**DESCRIPTION**
1	NODEN	Node or zone number, a character C in front of a number denotes a zone. See note (i)
2	XCORD	Its X co-ordinate; see note (ii)
3	YCORD	Its Y co-ordinate

NOTES:

Tip

(i) All nodes and zones should be given co-ordinates. An error is registered if co-ordinates are not given for any node or zone and **DRACPREP** will terminate un-successfully.

(ii) In principle, the units used for XCORD and YCORD are arbitrary. They are converted to real "size" in meters by parameter XYUNIT specified in parameter file *.PAR* (Table 4.1). However, it is strongly recommended that co-ordinates are defined in metres and that XYUNIT is set to 1.0.

(iii) Nodes or zones within this section which have not already appeared in the Node or Link records are ignored by the program.

4.2.1.5 Turn Priority Marker (TPM)

(N.B. for right-hand drive read left for right in this note and vice-versa.)

The markers describe which traffic flows oppose a turning movement. The naming of TPM follows that of SATURN. In addition, there are DRACULA-specific TPMs. The full range of turn priority markers modelled in DRACULA are:

G - Used at a priority junction to indicate the turn which must give way (from a minor road). As defined in SATURN, "a G priority marker indicates either a 'give-way' or a 'sharing' movements whereby a turn marked G gives way to all 'major' turning movements (i.e. those without any marker on an X) but 'shares' with other give-way movements" (van Vliet & Hall, 2002).

X - For opposed right turn which must cross a major flow from the opposite direction. This could be a major flow turning right at a priority junction or at a signalised junction.

F - For a permanent filter at traffic signals which has 100% green all the time.

M - Merge with one other (and only one) stream of traffic at a priority junction. See Note (i).

S - A turn with a clear *nearside* exit. Used mainly for motorway entry slip roads to indicate the slip road traffic has a clear exit onto the acceleration lane. In case of a two-lane slip road, the first lane traffic has a clear exit onto the acceleration lane and the second lane traffic will "merge" with the motorway traffic. See Note (ii).

C - Used to indicate a turn that has a clear (or protected) exit to the *far side*. Unlike an S turn, a C turn is to the far side and usually takes the far-side lane on exit link. Can be used for both priority and signalised junction. See Note (iii).

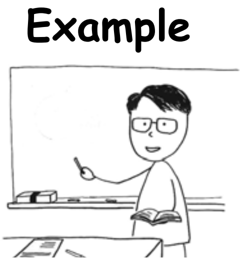
Z - Termed as "zipped" turns to replace a G turn at a priority junction where under congested conditions, the priority rule is changed to "one goes, another goes": the two streams of traffic take in turns to merge into the exit link. See Note 4.

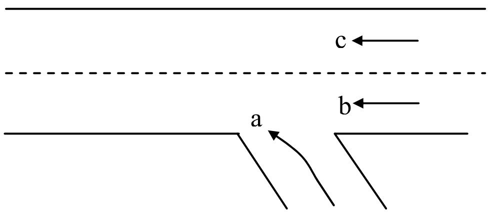
BLANK – no opposing flow.

NOTES:

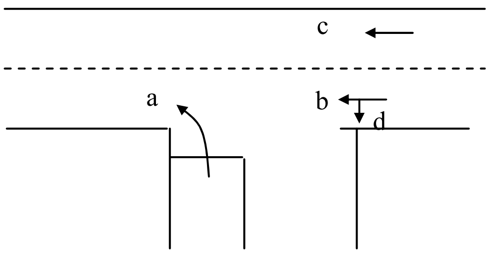
(i) (M) A turn coded M only needs to give-way to one stream (one lane) of traffic from right (for driving on the left); the stream from the right is called the opposing turn for the M turn. Generally an M turn shares the same exit with its opposing turn and they both exit into the nearside lane in exit link.

In the examples below, turn *a* coded as M gives way to *b* but not to *c*. By default, stream *b* turns into the same exit link as *a*. In a normally give-way situation (e.g. G turn), turn *a* would have to give way to both streams *b* and *c*.





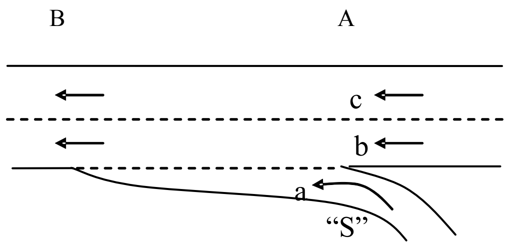
Give-way to b, not c.
Exit to lane 1



Give-way to b, not c.
No conflict with d.
Exit to first lane

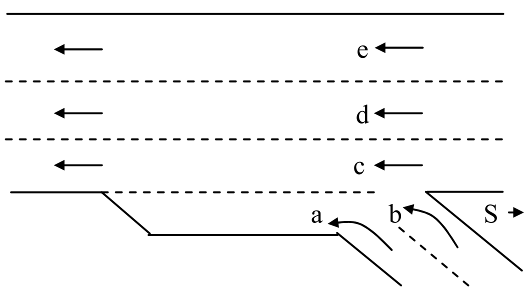
The merge facility can also be used to model “Y” junction where two streams of traffic with equal priority, or 2-lane link goes into 1-lane. There the rule is such that whichever vehicle arrives first to the stop line gets priority.

(ii) (S) Used on motorway slip road which leads to an acceleration lane, a turn marked S has a clear exit lane. In the example below, turn *a* enters the acceleration lane without any conflict with the motorway traffic *b* and *c* (e.g. does not need to give way to *b* or *c*). From the acceleration lane, the traffic then tries to merge with motorway traffic by changing lane.

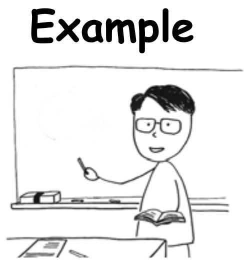


Modelled as two "junctions" at node A and B. The slip road traffic *a* coded with an S turn has a clear exit to lane 1 on link A-B.

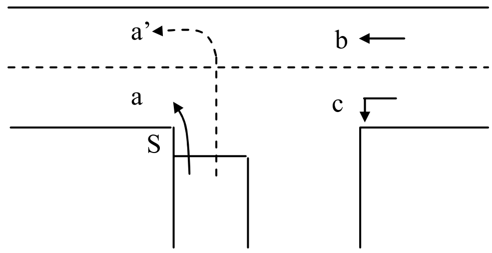
In the example below, a two-lane slip road leads to one acceleration lane. The on-ramp turning movement can also be coded as S. The nearside stream of on-ramp traffic *a* will enter the acceleration lane, whilst stream *b* will "merge" with motorway traffic *c* as though it has an M turn-priority marker.



A 2-lane slip road
b merge with *c*
a has a clear to lane 1



The S turn is not restricted to motorway slip roads. The left turning movement in the following example can also be modelled as an S turn. Whilst stream *a* has a clear nearside exit, stream *a'* will merge with stream *b*.

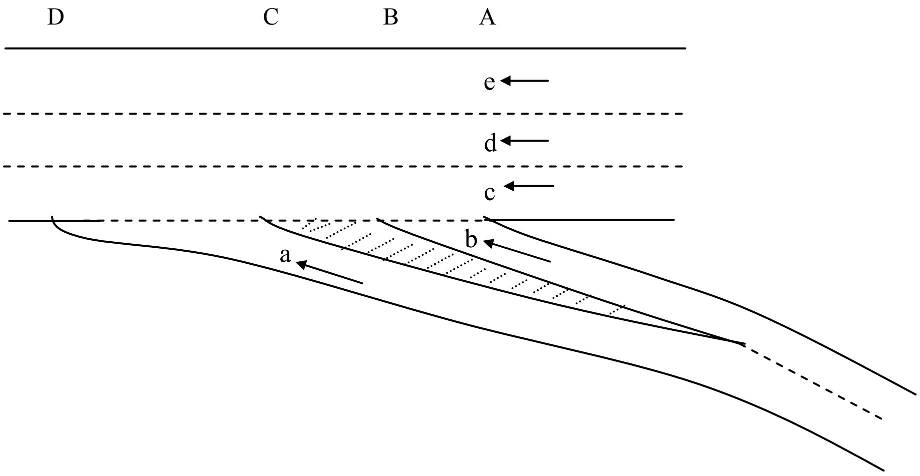


1-lane exit
Clear first lane exit.
Exit to first lane only.

In general, an S turn can be used in any nearside turn which has a clear exit lane. However, an S turn is currently restricted to 2 lanes for the turn, e.g. LANE=1 and LANE2≤2 (c.f. Section 4.2.1.2). A far-side turn which has a clear exit should be coded as a C turn, see Note (iii) below.

For a "staged" motorway entry (see example below), each of the two entry streams *a* and *b* are best modelled as a separate S turn entering two separate motorway junctions. Hence stream *b* of turn S joins the motorway at node A, whilst stream *a* of an S turn enters the motorway acceleration lane at node C.

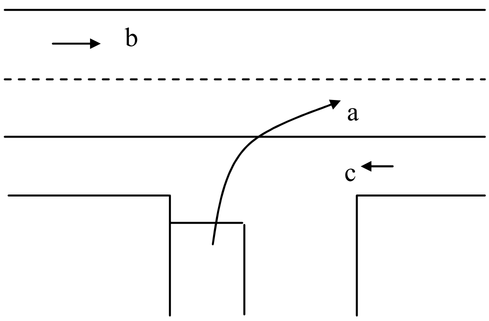
Example



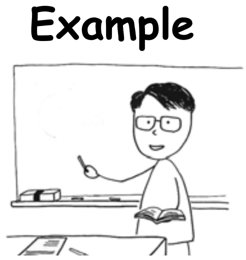
(iii) (C) A C turn is used to describe a turning movement to the right, or far side (for driving on the left) which also has a clear exit. This is the same as the C TPMod (turning modifier) used in **SATURN** (Section 4.2.2.3).

The general rule for a C turn is that (1) it has its own exclusive exit lane; (2) it does not need to give way to traffic streams sharing the same exit; however (3) it does need to give way to priority crossing traffic.

In the following example, turning movement *a* has a clear exit. It gives way to conflicting traffic *c* on the far-side, but not to traffic *b* from near-side which turns to the same exit link.



Give-way to *c* on far-side, not to *b* on near-side
Exit to far-side lane in exit link



(iv) (Z) A Z turn works as a G turn at a priority junction in free-flow situation. During extreme congestion at the priority junction, the Z turn and its higher-priority counter part will take in turns to enter the junction (and merge into the exit link). The congestion threshold where such a transition occurs is modelled by a speed threshold: if the average speed on the major road is below the threshold, the rule changes from G to Z. The speed threshold is specified via a global parameter VZTPM (Appendix B). (still under development!!!)

4.2.1.6 Link Length and Junction Layout

The coded link-length is defined in both **DRACULA** and **SATURN** as the distance between the centre of the upstream junction and the centre of the downstream junction, along the path of the road. The path may be a straight line between the upstream and downstream nodes, or it could be curved making the length of the path longer than crow-fly distance between the upstream and downstream nodes.

However, unlike SATURN which doe not model the physical size of a junction (except for Node Graphics display). DRACULA explicitly models the physical layout of intersections and estimates the size of an intersection from the number of lanes of the links and a fixed lane-width therefore the "effective" or "inter stop line" length of a link in DRACULA is the link-length taking away the distance between the centre of the upstream junction to the entry of the link and the distance between the centre of the downstream junction to the stopline of the link.

Tip

It can happen that a negative length is calculated in the above method. This could be due to an un-realistically short link-length coded in SATURN which was used to connect two very close intersections, or due to an incorrect interpretation of junction size (see Section 4.2.2.3). When a negative link-length is detected, a warning message is given in the pre-processing report file ***.TXP***. If a negative link-length is not corrected, the simulation will still run with vehicles "jumping" over the link without stopping. However, it is advisable to check the report file for any such warnings.

In many networks (particularly SATURN-based), XYUNIT is left to its default value, 1.0, even though the node co-ordinates input are not in metres. This does not affect the SATURN results as it only uses the link-length as defined in node/link records. However, DRACULA simulates individual vehicles' second-by-second movements over the length of the link, and then projects the movements onto the physical length of the link in animation. It, therefore, requires the perfect matching of the physical length of a link with the link-length as interpreted from the data coded in the simulation node/link records. Try to make a link's crow-fly distance (if it is on a straight-line) or its physical length (if on a curved road) match with the link length. Check file *.TXP* for those grossly incomparable.

Experience shows that the true value of XYUNIT in SATURN networks is often 10 (i.e. co-ordinates are defined to the nearest 10 metres).

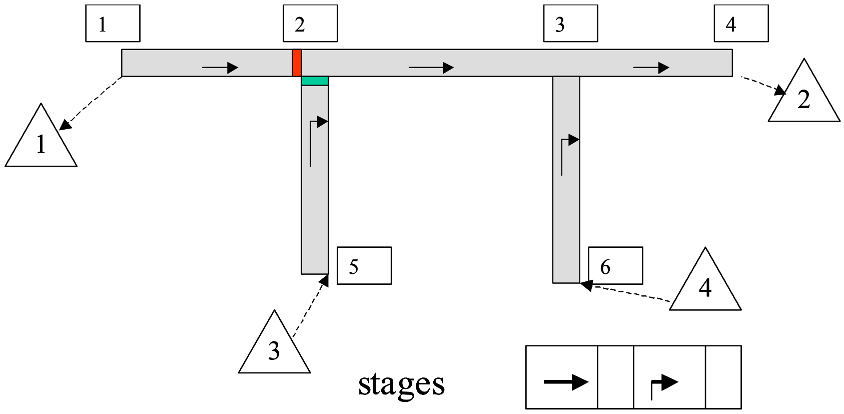
If one wants to use crow-fly distance as the link-length, set the following parameter in .PAR file:

QCROWFLY=T

In addition, **DRACPREP** sets various defaults in case of implausible coding. These include setting the link length to the crow-fly distance if it was coded as zero or left blank. Similarly, a 40 km/hr speed is assigned as link free-flow speed should it was coded as zero of blank.

4.2.1.7 An example

An example coding is given below to describe the network shown in Fig. 4.2 with four external nodes, two intersections – one signalised (node 2) and the other priority control (node 3). Traffic from link 6-3 gives way to traffic on link 2-3. The route flow for this example is given in Section 4.3.1.



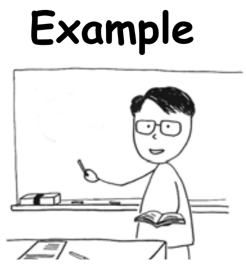
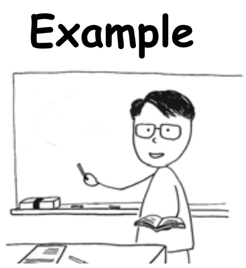


Fig. 4.2 An example network. Numbers in rectangles display the node numbers, while those in triangles are the zone numbers. Junction number 2 is signalised; a stage diagram with permitted turning movements and stage length is shown.

```
NETWORK:  dtest.net
&LINKS
      1      1       0
              2       0
       2      3       3   2   60   0
              1       1   30.0  200  1  1   0  0
                 3       0
              5       1   30.0  200  0  0   1  1
                    37   3   1 0
                    17   3   5 0
       3      3       1
              2       1   30.0  200  1  1   0  0
                 4      0
              6       1   30.0  200  0  0   1G  1
        4      1       0
              3       1   30.0  200
        5      1       0
                 2       0
        6      1       0
                 3       0
99999
&ZONES
       1   1 2
       2   4 3
       3   5 2
       4   6 3
99999
&COORD
       1       -300    0
       2       -100       0
       3       100     0
       4       300    0
       5       -100 -200
       6       100  -200
       C1       -500   0
       C2        500   0
       C3       -150    -250
       C4      150 -250
99999
```



4.2.2 SATURN Network Coding

SATURN simulation networks can be used as the base for a DRACULA highway network. To start with an existing SATURN network data file, rename the SATURN network data file *.DAT* to *.NET*.

DRACULA uses the simulation network of SATURN ***only***. See Chapter 6 of the SATURN Manual for network coding, in particular coding for simulation network (record 11111) and node coordinates (55555). If there are buffer links in the network, you need first to convert them into simulation links within SATURN. ***[Hint: use SATURN P1X/Network editor]***

Not all sections/records of a SATURN network data file are used in DRACULA. The ones adopted by **DRACULA** are:

1 – OPTION SPECIFICATION RECORD
2 – PARAMETER SPECIFICATION RECORD
3 – SIMULATION NODES AND LINKS (RECORD 1)
4 – SIMULATION ZONES (RECORD 2)
5 – NODE AND/OR ZONE COORDINATES (RECORD 5)
6 – FIXED (BUS) ROUTES (RECORD 6)

Most of the SATURN records on simulation nodes, links, zones (including buffer zones) are used straight-forwardly by DRACULA in a similar way as they are used in SATURN. For a detailed description of SATURN network coding, readers are referred to Chapter 5 and 6 of the SATURN Manual (van Vliet & Hall, 2002).

Within the sections of data adopted, the SATURN variables used and their interpretation into DRACULA model variables are given in the following sub-sections, except for the bus routes (data record 5 above) which is described in Section 5. Particular attention is given to the adaptation and interpretation of data records which are different from their SATURN use and/or are not used by DRACULA.

DRACULA does not accept SATURN sub-files using the "&INCLUDE" facility. Users will have to embed the data into the main *.NET* file.

4.2.2.1 Option and parameter specifications

SATURN parameter specification records set user-defined model parameters for the network. They are defined at the top of a SATURN network file under FPARAM. The parameters used by **DRACPREP** relate primarily to those used in the network building and interpretation; these parameters are given in Table 4.3.

DRAPREP can only read in one parameter per line. So if any two of the parameters of Table 4.3 are entered on the same line in a network file, one needs to edit the *.NET* file to make them appear on separate lines.

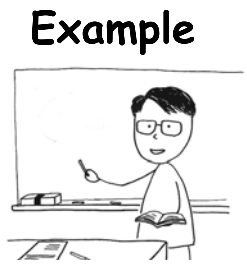
Table 4.3 ***SATURN*** *option and parameters used by* ***DRACULA****.*

Parameter	Default	Description
PASSQ	.FALSE.	A SATURN option. TRUE if to have multiple time period assignment
AUTOX	.FALSE.	If TRUE any uncoded external simulation nodes are automatically coded using the best available data. ***Further testing required. Contact the author if you use this function.***
DUTCH	.FALSE.	If TRUE node numbers take up to 10 digits.
EZBUS	.FALSE.	If TRUE the bus route data on the 66666 data records are in “free format”. By default they are in fixed column format. ***Further testing required***
FREEXY	.FALSE.	If TRUE, then the X, Y coordinates in the 55555 records are input as “free format”. ***Further testing required***
LEFTDR	.TRUE.	If FALSE right-hand drive assumed.
SPEEDS	.FALSE.	If TRUE travel speed (in kph) rather than travel times (in seconds) are input in the simulation link records.
LTP	30	Duration of the simulated time period (in minutes). Over-writes the TMAIN specified in .PAR file.
NOMADS	1	Number of user classes to be assigned separately (see Section 5.3.2)
PPK	0.0	Pence per kilometre 0 used to convert distances into generalised costs.
PPM	1.0	Pence per minute – converts times into generalised costs.
VCPCU(v)	1.0	The pcu value per vehicle in user class v, or for all vehicles in the trip matrix if unscripted.
XYUNIT	1.0	Number of metres corresponding to an integer value of 1 as used to define node/zone coordinates.
XYFORM	2I5	The format used to define node coordinates.
IFGIS(n)	.TRUE.	Control the display of GIS features. n=1,2,…7 is for GIS polygon, polyline, icon, text, name, curved link and node coordinate respectively. Currently only options n=1,2,3, and 4 are used by DRACULA.

In addition, certain DRACULA input/output filenames can be specified within the SATURN parameter specification record (e.g. record &PARAM). The DRACULA file names can be specified this way are listed in Table 2.6, marked with an *.

An example of filename specification is shown below:

```
&OPTION
&END
Otley Basic Network 1985
&PARAM
SPEEDS = T
LTP = 60
XYUNIT = 1
FILTRP = ottrip1
FILBUS = otbus
&END
```



4.2.2.2 Simulation Node Data

Most of the SATURN simulation node data are used directly by DRACULA in the same way as they are used in SATURN. The following node data are interpreted into DRACULA data:

NAME	DESCRIPTION
JTYPE	Type 9 junction for giveway-to-the-right traffic control. See Section 4.2.4.
JCIR	Time to circle roundabout. See Section 4.2.2.6
RSAT	Roundabout capacity. See Section 4.2.2.6

The node data not used by DRACULA is:

NUC	Number of time units per cycle

4.2.2.3 Simulation Link and Turn Data

The SATURN simulation link and turn data not used by DRACULA are:

STACK	Link stack capacity
QSTAR	Marker for speed-flow curves
LSAT	Turn saturation flow. See N.B. (i)

and the whole section on "Link speed flow data".

N.B.

(i) Junction or turn saturation flows are determined by car-following rules. They are outcome of micro-simulation models.

The types of SATURN turn priority parker (TPM) used by DRACULA are:

G	-	a give way turn at priority junction
X	-	Opposed right turn from a major road
M	-	Merge
F	-	A permanent green filter
BLANK	-	No opposing flow

In addition, the following TPMs which are specific to DRACULA can also be specified in the SATURN-format networks (see Section 4.2.1.5 for details):

S	-	For motorway slip roads
Z	-	A “zipped” turn, a normally give way turn (G turn) which makes alternative turns with the priority stream into the junction (Section 4.2.1.5)

SATURN distinguishes hooked or not hooked opposite pairs of right-turning G or X movements. In DRACULA such pair of movements is always modelled as not hooked. Hence, DRACULA discards SATURN turn modifier D and E. The only SATURN turn modifier (TPMod) used by DRACULA is C for a clear or reserved exit lane for G or X movements. This is modelled in the same way as the C turn in Section 4.2.1.5.

As with the DRACULA-format coding of traffic signals, there could be at most two links in any one stage. There are, however, no limit on the number of turns that can be coded in each stage as long as the turns are from the two links.

4.2.2.4 Simulation Zone Data

Currently the SATURN simulation zone data are used in by DRACULA in two ways:

(i) to aggregate simulation statistics by origin-destination pairs; and
(ii) to be used in conjunction with the day-to-day demand model (Section 10).

The zone data are not affecting the simulation directly. Hence it is not crucial to have zones coded, or embedded in the main *.NET* file.

4.2.2.5 Node Coordinates and Link Length

If the coordinates of a node are not assigned, SATURN tries to “extrapolate/interpolate” its coordinates from those of its neighbours. Since the node coordinates are used purly for display purposes in SATURN (not affecting it simulation), SATURN treats such coding problem simply as a WARNING.

As described later, node coordinates determines some of the basic parameters in DRACULA which affect the results of the simulation. Hence DRACULA requires coordinates of all nodes be assigned.

See Section 4.2.1.6 for DRACULA interpretation of junction layout and entry-to-stopline link distance.

4.2.2.6 Roundabouts

All SATURN roundabouts are assumed to operate whereby entry to the roundabout is controlled by “Give Way” markings and priority must be given to traffic approaching from the right (or left for right-hand driving). Roundabouts, which are fully or partially operated by traffic signal controls at the entries, must be represented in SATURN as a series of one-way links. Section 4.2.3 describes a DRACULA model of roundabouts which can take into account both Give-way and signal controlled roundabouts.

A roundabout in a SATURN network is specified in terms of the circulation time and circulating capacity, whilst DRACULA requires the physical diameter of and the number of lanes on the roundabout. The SATURN network data (in node records) used by DRACULA to represent a roundabout are:

RSAT	Maximum roundabout capacity in pcu/hr.
JCIR	Roundabout circulating time (seconds)

By default, the number of lanes on the roundabout (NRBL) is estimated from the maximum roundabout capacity, RSAT, assuming 1800 veh/hr/lane and the maximum number of lanes on any entry lane (NL_{entry}):

$$NRBL = \max(RSAT/1800, NL_{entry}) \qquad (4.1a)$$

Alternatively, one can make NRBL depend only on RSAT as:

$$NRBL = RSAT/1800 \qquad (4.1b)$$

To choose equation (1b), set a parameter QRBLANE=F in **.PAR* file.

DRACULA makes an assumption about the free-flow circulating speed and then estimates the inscribed diameter of the roundabout from JCIR:

$$HCID = \max(R_{island}, JCIR \times V_{circ} / 2\pi + NRBL \times W_{lane}) \qquad (4.2)$$

where:

HCID - half of the inscribed diameter (i.e. radius of outer boundary of a roundabout),

R_{island} - a user-defined minimum radius of the central island (metre),

W_{lane} - the width of a lane which is fixed to 4 metres, and

V_{circ} - the free-flow circulating speed. Default: 27 km/hr.

Variable V_{circ} can be specified via a control parameter CIRC_SPEED in the parameter file *net.PAR*. The default circulating speed of 27 km/hr is taken from a study by McDonald et al (1984). This is a global variable which applies to all roundabouts in a network. To make a roundabout-specific circulating speed, see description in Section 6.2.

If the diameter and the number of lanes on any of the roundabouts in the network are known, one can over-write the default estimations with the actual data by manually altering the values of circulating time (JCIR) and the maximum roundabout capacity (RSAT) to match those observed. Alternatively, one can define HCID, V_{circ}, R_{island} and NRBL for the roundabout concerned via roundabout-specific input in *.ADD* file (see Section 6.2).

4.2.3 Signalised Roundabout Model

The roundabout models described in the basic DRACULA networks (Section 4.2.1) and in SATURN networks (Section 4.2.2.3) represent those operating with give-way entries. The model described in this section can represent roundabouts which are fully or partially operated by traffic signal controls at the entries. In fact, the logic developed is generic which allows both give-way and signal controlled roundabouts to be represented.

More detailed representation, this roundabout model allows information about the number of lanes, lane markings on the roundabout, angles of entry/exit approaches and traffic control on approaches to be specified. In addition, this model can represent different characteristics for different sections of a roundabout. For example, you can model a roundabout partially operates as two lanes and partially operates as three lanes. You can also model a partially signalised roundabout: some arms operate as priority approaches and others signal controlled. For these reasons, the new roundabout model is sometimes referred to as a partially signalised roundabout model.

This model can be used in-conjunction with both the basic DRACULA and SATURN-format networks. This section of data should be entered in *.NET* file after the main network descriptions (described in Sections 4.2.1 and 4.2.2) have been coded. The section should be preceded by “&SRBS”, and ending with a string “99999”.

This section of data contains three (mandatory) record types:

1 - Node description, as in the basic network coding
2 - Description of approach arm to the roundabout
3 - Sections on roundabout

4.2.3.1 Coding instruction

For DRACULA networks, the “DATA” column specifies the number of fields for the record. As with the DRACULA network coding convention, data fields are entered in free-format and only a white space is required to separate two fields. For SATURN networks, one need to look at “(COL)” which indicate the precise column(s) where each data field needs to be entered.

DATA(COL)	**NAME**	**DESCRIPTION**
***********RECORD TYPE 1 - NODE DATA ***************		
1 (1-5)	NODE	Node number
2 (6-10)	NIN	Number of links at the node
3 (11-15)	JTYPE	Node type: 6 for the new roundabout model
4(16-20)	HICD	half inscribe circular diameter (metres), JTYPE=6 only
5(20-25)	NRBL	Number of roundabout lanes. Note (i)
6(26-30)	CSPEED	Roundabout circulating speed (kph).
***** RECORD TYPE 2.1 – ROUNDABOUT APPROACH DATA *****		
1(6-10)	RBNODE	Node number at the approach. Note (iii)
2(11-15)	RJTYPE	Node type: 1 for priority approach 3 for signalised approach
3(16-20)	RNSTAGE or 0	Number of stages if RJTYPE=3 or if RJTYPE=1
4(21-25)	ROFFSET or 0	signal offset if RJTYPE=3 or if RJTYPE=1
5(26-30)	RLCY or 0	cycle time if RJTYPE=3 or if RJTYPE=1
6(35-40)	RSHIFT	entry and exit separation (meters) See Note (ii).
***** RECORD TYPE 2.2 – ROUNDABOUT LINK DATA *****		
1(10-15)	RANODE	Upstream node of the roundabout section.
2(15-20)	RTIM	Free flow speed (km/hr) on this section
3(20-25)	RDIST	Length of the section (m). Not in use.
4(26-30)	RLANES	Number of lanes of the section. Note (i)
5(31-35)	RLMARK1	Lane marking for the first lane
6(36-40)	RLMARK2	Lane marking for the second lane
7(40-45)	..etc	..etc for all RLANES

***** RECORD TYPE 2.3 – SIGNAL DATA (AS IN SATURN)*****

1(10-15)	STAGL	Duration of stage (seconds)
2(15-20)	INTG	Duration of following inter-green (seconds)
3(20-25)	NGM	The number of node entries which follow as GNA(1), GNC(1), GNA(2), GNC(2), …, with two entries per green movements.
4(26-30)	GNA(1)	The A_node for the first green movement.
5(31-35)	GNC(1)	The C_node for this turn.
6(36-40)	GNA(2)	The second A_node
7(40-45)	GNC(2)	The second C_node
8(45-50)	etc	etc.

Note:

(i) Different sections of a roundabout could operate with different number of lanes. NRBL should be the maximum number of lanes on the roundabout, while RLANES the number of lanes along specific section. NRBL is used in this signalised roundabout model purely for display purpose: the roundabout will be shown with equal number of lanes. In the simulation, traffic follows RLANES in each section. The example given in 4.2.3.2 shows a roundabout operating as 4 lanes on half of the circulating carriageway and 5 lanes on the other half. Careful observation of the animation would reveal that the inner-most lane on the 4-lane half of the carriageway is never being used by traffic.

(ii) For large roundabouts, there are often separation between an entry and an exit of an arm. RSHIFT is to represent partly such physical separation and partly the entry angle of a roundabout approach.

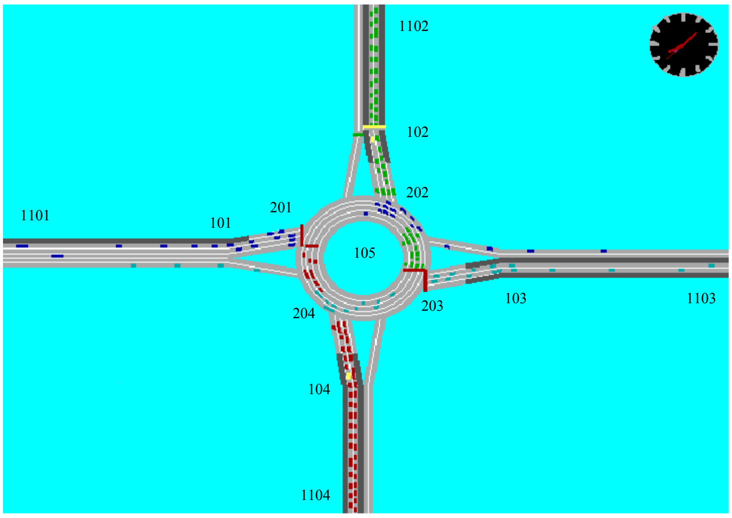
(iii) The NIN number of RBNODE needs to be entered in strictly clockwise order (or anti-clockwise for driving on right) and in the same order as the ANODE being entered in the Node Record (Section 4.2.1.2).

(iv) Lane marking is given such: turning left is 2, right 4 and straight 8. A shared lane is simply to add the two turning numbers together, i.e. a RLMARK=10 indicates a shared left and straight ahead turnings.

4.2.3.2 An Example

Example

The network description of the Stanwell Moor Roundabout which is partially signalised and with spiral marking is given below. See the additional coding for B-node 105.




```
M25 Stanwell Moor Roundabout June 1998 with Indirect Signals
on A3044 North
11111
 1101     1    0
        101    2    70  200
  101     2    4
       1101    4    70  200 1800  1 4
        105    2    70  200 1800  1 2
 1102     1    0
        102    2    70  200
  102     2    3     1   28   50
       1102    4    70  200 1800  1 4
        105    2    70  200 1800F 1 2
              34    16     2 1102  105
 1103     1    0
        103    2    70  200
  103     2    4
       1103    4    70  200 1800  1 4
        105    2    70  200 1800  1 2
 1104     1    0
        104    2    70  200
  104     2    4
       1104    4    70  200 1800  1 4
        105    2    70  200 1800  1 2
  105     4    6    85     5   30
        101    4    70  200   900S 1 1 1500S 2 3  800S 3 4
        102    4    70  200   900G 1 1 1500G 1 2  800G 3 4
        103    4    70  200   900S 1 1 1500S 2 3  800S 3 4
        104    4    70  200   900G 1 1 1500G 2 3  800G 3 4
99999
22222
…..
99999
33333
…..
99999
55555
  105 2000 2000
….
99999
&SRBS
 105      4     6    85     5    30
        201     6     2     0    50    40
             204     4     2    10    12     4
              23     5     2   101   105
```

```
          14    8    2  204  201
   202     2    2    0    0   40
         201    4    2   10   12    4
   203     6    2    0   50   40
         202    5    2   10    8    4    4
          23    5    2  103  105
          14    8    2  202  203
   204     2    0    0    0   40
         203    5    2    2    8   12    4
99999
```

4.2.3.3 Lane Choice Modelling at Signalised Roundabouts

Rather than representing a roundabout as a series of one-way links, a distinct advantage of this model is its ability to represent the lane choice of traffic approaching a roundabout. Even though a great deal more information is provided to describe such a roundabout and information can be provided for different sections of the roundabout, the model treats the roundabout simply as an intersection, still. Vehicles approaching such a roundabout need only to know which exit (from the roundabout) to turn, as they would know for approaching other types of intersections. They would choose an appropriate lane according to their turning exit and to the lane-markings on the approaching arms. In this way, they would get into the appropriate lane on roundabout and change lane on the roundabout only to make their exits.

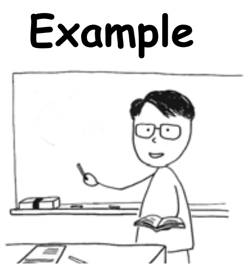
The workings of the roundabout, in terms of vehicle joining, circulating and leaving the roundabout and lane-changing on the roundabout, are all worked out internally by the model. Therefore, when specifying the path/route of a trip, you only need to give the node number of the roundabout and not the node numbers of each approach. For the example in Section 4.2.3.2, a path from zone 5001 to zone 5003 will go through nodes 1101, 101, 105, 103 and 1103.

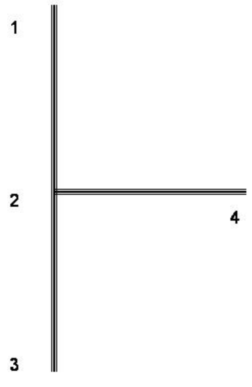
In the traditional way of representing a roundabout as a series of one-way links, such lane-choice behaviour of traffic approaching a roundabout gets lost as all the lanes on the approaching arm permit the same turn. There would also be too much unrealistic lane-changing on those one-way roundabout links.

4.2.4 Model for Give-way-to-the-Right Junction Control

In the Netherlands and part of Belgium, a type of junction control is such that one should always give-way to all traffic from the right (for driving on the right). This type of junction is modelled as a type 9 junction in DRACULA (see node coding in Section 4.2.1.2).

The following example shows the general rules for this type of traffic control. The example is for driving on the left, though he model can also be applied for driving on the right with the turns appropriately “mirrored”.





An example T-junction operating with GWTTR rule.

Erik Versteegt of TNO provided description to the rules which are involved in this type of junction. They are:

- Give way to all (e.g. crossing or merging) traffic from the right (GWTTR rule)
- Traffic going straight ahead on a road has priority over turning traffic on the same road (the 'X' turns)
- Turning traffic with a 'short corner' goes before the 'long corner' (corner rule)

Hence for the above example, the priorities are:

Turn 1-2-4 gives way to turn 3-2-1 (based on the X rule)
Turn 1-2-4 gives way to turn 3-2-4 (corner rule)

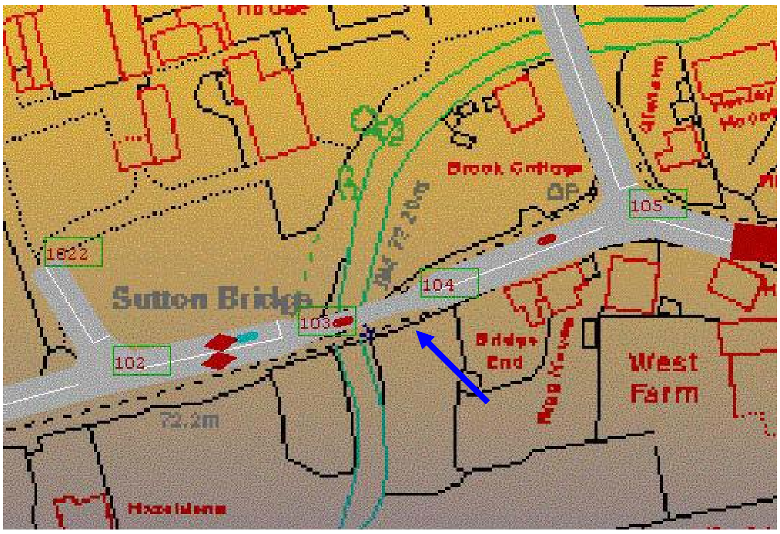
Turn 3-2-1 gives way to turn 4-2-3 (GWTTR rule)
Turn 3-2-1 gives way to turn 4-2-1 (GWTTR rule)

Turn 4-2-3 gives way to turn 1-2-3 (GWTTR rule)
Turn 4-2-3 gives way to turn 1-2-4 (GWTTR rule)

The X turn can already be modelled in **DRACULA**. The corner rule is implicitly modelled in **DRACULA** with the traffic from the short corner getting onto the junction quicker than those from the longer corner.

4.2.5 Model for Give way to On-Coming Traffic

Often seen on rural country roads are sections of the road which are too narrow to accommodate simultaneous crossing of vehicles from both directions. For example, the narrow bridge ("Sutton Bridge") pointed by the blue arrow in the picture below. Some of them are just wide enough for allow two cars to cross, but not a large vehicle with another vehicle.



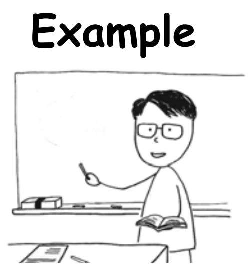
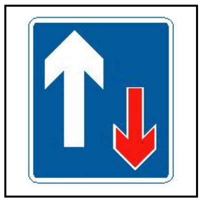


Fig. 4.1. A network model of a narrow bridge over the area modelled.

There are two types of traffic operations on such narrow roads. On some of these narrow roads or sections of a road, especially the busier ones, traffic signs are put up to indicate which direction of traffic should give priority to vehicles from opposite direction (sign in the picture (a) below), and signs to indicate to the vehicles who have priority over the oncoming traffic (picture (b)).

(a) Give priority to vehicles from opposite direction



(b) Have priority over vehicles from opposite direction

Fig. 4.2. Traffic signs used in the UK.

The second type of operation applies to those quieter, less traffic, roads. Here the roads may be wide enough for two cars but not enough for a large vehicle and another vehicle. Traffic operation is this case, in terms of who has priority, is left to the drivers themselves.

Two modelling features are introduced to model these narrow roads and access to the road. First, the narrow road or section of a road should be modelled as a separate link, a narrow link. The nodes at each end of the narrow link should be coded as of type JTYPE=8 (see Section 4.2.1.2).

If accessing to the narrow road or bridge is controlled by give-way sign (sign (a)), then the entry arm to the bridge should be coded with a 'G' turn-priority-marker (TPM, Section 4.2.1.5). The opposite, with high priority, would have TPM=BLANK.

If the narrow road is not controlled with a give-way sign, then the entry arms from both direction should have TPM=BLANK.

In the example of Sutton Bridge, the section of road over the bridge is modelling as link from 103-104 for the east bound traffic, and link 104 to 103 for west bound traffic. A ‘give way to oncoming vehicles’ sign is put up for the east bound traffic, indicated in the network plot by a solid stop-line downstream of link 102 to 203. The node and link coding (in card 11111 of SATURN format) for the bridge is as follows, and node the turn 102-103-140 has a TPM of ‘G’:

```
  103    2    8
       104    1   48   88 2520  1 1
       102    1   49  128 2520G 1 1
  104    2    8
       105    1   45  150 2520  1 1
       103    1   45   88 2520  1 1
```

Example

Fig. 4.3. An example coding of a narrow bridge.

Note:

Tip

(i) Such narrow links can only be connected to one other link, e.g. junction of type 8 could only have two arms, e.g. NIN=2 (Section 4.2.1.2).
(ii) There could be only one lane on such links.

The second feature introduced to the model is one used to describe the characteristics of the narrow bridge or link itself and to model the traffic operation over the bridge. A narrow link is characterised by its width, measured as a fraction of the normal lane width. The operation of the traffic over the narrow bridge is characterised by the types of vehicles to give-way and a sight distance over the other end of the bridge. The data should be coded in the additional network input data file **.ADD** and within the headings “&BRIDGE” and “99999”, as follows:

DATA	NAME	DESCRIPTION
1	ANODE	Upstream node number of the link. Note (i)
2	BNODE	Downstream node number
3	VTYPE	Vehicle types to give way. See Note (ii).
4	FWIDTH	Fraction of a normal lane width. Note (iii)
5	SIGHT	Sight distance (metre). Note (iv)

Note:

(i) A narrow bridge is still a two-way road, coded in the main network description as two uni-directional links. Both these links should be coded here in **.ADD**.

Tip

(ii) A “give way to oncoming vehicles” sign normally indicates give way to all types of vehicles. Here VTYPE should be coded as 63 (see Section 6.1.2 for definition of vehicle types). Where a road is only narrow for a large vehicle plus another vehicle, VTYPE can be used to indicate the actual types of large vehicles which are of problems. In the “Example 2” below, the road is wide enough for two cars, but are too narrow if there is a LGV or HGV vehicle present. VTYPE=48 is the sum of type 16 vehicles (LGV) and type 32 vehicles (HGV).

(iii) The normal lane width is modelled as a fixed value at 4 metre’s wide. Given that this is a narrow link, hence FWIDTH should be less than 1.0. However, sum of the FWIDTHs of the two-way links should be greater or equal to 1.0.

(iv) A vehicle approaching a narrow bridge needs not only look at the vehicles already on the bridge, but also those vehicles approaching the bridge on the other side. The SIGHT sets out the distance downstream of the link approaching the bridge from the other end by which the subject vehicle would look out.

Example

```
Example 1:
&BRIDGE
 103 104 63  0.5 20
 104 103 63  0.5 20
99999
```

```
Example 2:
&BRIDGE
 103 104 48  0.8 20
 104 103 48  0.8 20
99999
```

The above “Example 1” is used together with the link coding presented in Fig. 4.3 to describe that the bridge has a combined width of a normal lane-width. A give-way sign is put up on approach from link 102-103 (Fig. 4.3). Hence the VTYPE=63 coding here, where traffic from link 102-103 will have to give-way to all types of traffic. They would give-way not only to all types of traffic already on link 103-104, but also to all types of traffic 20 metres downstream on link approaching node 104 from the other direction, e.g. link 105-104 (Figs. 4.1 and 4.3).

Even though the give-way sign is only put up on node 103 side of the bridge for traffic from link 102-103, vehicles approaching the bridge from link 105-104 should also give-way to any vehicle travelling on link 103-104. This is indicated by the VTYPE=63 for link 104-103.

4.3 Highway Travel Demand

For the current release of DRACULA-MARS, travel demand is an exogenous input. At the most detailed level, the demand is modelled at the individual level: individual trip-maker, the route they take on the day, preferred departure time and their individual driving behaviour (Section 4.3.2). Such information can be generated by the DRACULA day-to-day route choice model (Section 11) or any other microscopic route choice models.

At a more aggregated level, travel demand is represented in terms of average flows (vehicles/hr) over assigned routes and time periods. The latter representation of travel demand, described in Section 4.3.1, is a more practical one given the current level of data available (i.e. OD trip matrix rather than individual trip-makers) and can be readily linked with other assignment packages such as SATURN (see Section 4.3.3).

Different from SATURN and other assignment models, traffic flow in DRACULA is represented in terms of number of trips per hour (or number of vehicles per hour) as opposed to pcus per hour. So when converting SATURN flows into DRACULA flows, conversion from pcu to trip is required (see Section 4.3.3).

4.3.1 Average Travel Demand

Though greater effort has been made to link DRACULA directly with SATURN assignments, it is also possible to link DRACULA with other assignment packages with relatively minimal effort required from the user. The users need simply to convert the results from their assignment model to the required DRACULA free-standing format and store them in text file *.TRP.

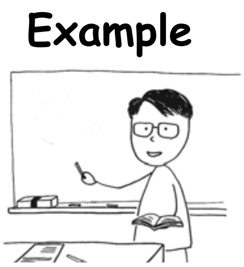
The format for a single time-period assignment is shown below. Results from a multiple user-class assignment can be represented in this way (see also Section 4.3.1.1). For multiple time-periods assignment, route choice for each time period should be written to one *.TRP* file (Section 4.3.1.2).

The data records should be preceded by a line containing a string “&ROUTES” and terminated with line containing “99999”. Users can enter extra “comment” lines before the preceding and terminating lines. As for network coding (Section 4.1.2), the entries of data records are in free format – only a space is required to separate two data entries.

DATA	NAME	DESCRIPTION
1	OZONE	Origin zone number
2	DZONE	Destination zone number
3	MUC	The user class following this route. See 4.3.1.1.
4	FLOW	Route flow rate (trips/hr) for the subject time period See 4.3.1.2.
6	QSTART	A character symbol "%" to indicate the start of the node list en-route
7	NODE(1)	First node en-route.
8	NODE(2)	Second note en-route
etc	etc	
9	QEOL	a symbol "+" at the end of a line to indicate continuation of node list on next line. No space between the last node on line and the symbol.
etc	etc	
10	QEND	a symbol "%" to indicate the end of node list. No space between the last node en-route and the symbol

N.B. If all or part of a route follows a straight line (more or less), it is not necessary to define every node along the line, but only the nodes at the bends or at the ends.

The route flow for example network ***dtest*** (shown in Section 4.2.1.5) are:




```
NETWORK: dtest.trp
&ROUTES
 1 2   1  200  % 1  2  3 4%
 3 2   1  100  %5  2+
         3 4%
 4 2   1 100  %6  3 4%
99999
```

4.3.1.1 User class vs. vehicle type

Multiple user classes are used in assignment models such as SATURN to refer to trips which differ with respect to either, as defined in Van Vliet and Hall (2002):

(a) vehicle type;
(a) their criteria for route choice;
(b) network restrictions;
(c) demand characteristics (e.g. elasticities).

For simulation of supply-side effects of travel, the more relevant distinctions are vehicle types and network restrictions. For example, lorries and cars have different physical size and acceleration profiles which have an impact on their journey performance. Equally cars and taxis would be different if there were taxi-only lanes.

In DRACULA, the terms "user class" and "vehicle types" are often used inter-changeably to refer to the physical/mechanical characteristics of vehicles and restrictions on them in network access. Each user class must be associated with a vehicle type. For the definition of vehicle types used in DRACULA, see Section 6.1.2.

More than one user class or vehicle type may use the same route. It can be noted that user classes or vehicles types are embedded within the coding of a route. If a route is used by more than one user classes, each user class should be entered with the respective MUC specified with MUC refers to the first, second,…, and the n-th of user class using this route.

By default, all user classes are assumed to be of vehicle type "CAR". To specify different vehicle types, use the following parameters in the *.PAR* file:

Table 4.4 Specification of user classes

Parameter	**Defaults**	**Description**
NMUC	1	Number of user classes in the demand input. Maximum 10 user classes.
MUC(v)	1	Vehicle type for user class v. See Section 6.2. for numerical representation of vehicle type. Type 1 vehicles are CARS.

4.3.1.2 Time-dependent demand with variable route choices

Temporal distribution of demand can be represented in DRACULA by step functions indicating the level of demand for each time period (step) and the route assignment for the time period.

N.B. Time periods discussed here are sections of time within the main (demand) period. To conduct a simulation of the main period, a warm-up time period before and a cooling-off period after the main period are also simulated. The warm-up and cooling-off periods are described in Section 6.1.1.

It should be noted that the coding of "routes" in DRACULA permits input of a single time period assignment only. Results for multiple time-period assignment can be load into DRACULA via separate **.TRP** files: each file contains the route assignment for each time period. Table 4.5 lists the essential parameters used to specify a temporal demand distribution and file names of route assignment for each time period.

Table 4.5 Specification for a temporal demand distribution with time-dependent route choice.

Parameter	**Defaults**	**Description**
NSTEP	1	Number of time steps. Maximum 10 steps.
TSTEP(1)	60	Duration (minutes) for the first time period.
TSTEP(t)	0	Duration (minutes) for time period t (t=2,3,..10)
QPASSQ	T	Set it to be T to load time-dependent route choice from files TRPNAM(*)
FILTRP(t)		File name (with extension .TRP) of route choice for time period t (t=1,2,…10)

4.3.1.3 Time-dependent demand with fixed route choices

It is possible to model simply a temporal demand profile without route re-assignment (i.e. with a fixed route assignment for all time periods). In this case, only one *.TRP* file is required (which contains the fixed route assignment) and the file name can be specified via parameter FILTRP or FILTRP(1) in the *.PAR* file. Table 4.6 lists the parameters to be specified in *.PAR* in order to specify the temporal demand distribution with fixed route choice.

Table 4.6 Specification for temporal demand with fixed route choice.

Parameter	**Defaults**	**Description**
NSTEP	1	Number of time steps. Maximum 10 steps.
TSTEP(1)	60	Duration (minutes) for time period t
TSTEP(t)	0	Duration (minutes) for time period t (t=2, 3, ..10)
QPASSQ	F	Leave it as F is to assume fixed assignment over NSTEP time period.
GONZO	1	Matrix factor for the whole time period
GONZO(t)	1	Matrix factor for time period t

4.3.2 Individual Trips

By default, the DRACULA-MARS traffic simulation requires individual drivers' route and departure-time choice for the day. When exogenous input on such detailed demand description is not available, the program **DRACPREP** can still generate a list of individual drivers whose routes are generated based on the aggregated demand provided, hence *.VEH* is referred to as an output from **DRACPREP** in Section 2.2.

One advantage to feeding a pre-defined list of vehicles to DRACSIM, as opposed to generating vehicles within DRACSIM, is to enable consistency in the driver/vehicle characteristics simulated between/among different scenario tests. To do multiple simulation runs on the same scenario (see Section 9.5), however, it is advisable not to run DRACSIM with the same list of vehicles.

To enable **DRACPREP** to generate *net.VEH* file, set in *net.PAR* file the parameter:

QVEHPOOL = T

The format of the *.VEH* file contains two records:

RECORD TYPE 1	-	Total number of vehicles to travel (NVEH), entered on the first line of the file.
RECORD TYPE 2	-	Route choice, departure time choice, and characteristics for each vehicles. One record per line and for each of the NVEH vehicles, in ascending order of departure times.

DATA	NAME	DESCRIPTION
***********RECORD TYPE 1 – TOTAL DEMAND ***************		
1	NVEH	Total number of vehicles to travel
***********RECORD TYPE 2 – INDIVIDUAL VEHICLE***********		
1	VEHID	Vehicle identification, a unique number
2	DEPT	Departure time (seconds), measured from the beginning of the "demand" period.
3	RTID	Route identification (i), a number pointing to the i-th route specified in file *.TRP*
4	VTYPE	Vehicle type (see 6.1.2 for definition)
5	LEN	Length of the vehicle (metres)
6	SMIN	Minimum safety distance headway (metres)
7	TREAC	Reaction time (seconds), fixed at 1.0 second
8	ANORM	Normal acceleration (m/s^2)
9	AMAX	Maximum acceleration (m/s^2)
10	DNORM	Normal deceleration ($-m/s^2$)
11	DMAX	Maximum deceleration ($-m/s^2$)
12	SPEED	Speed factor
13	GAP	Gap-acceptance factor

The vehicle characteristics ("DATA" entries 4 - 13) are drawn from distributions which try to match as closely as possible the behaviour of drivers in the study area. Details on the distributions used can be found in Section 6.1.3.

4.3.3 Route Assignment from SATURN

A SATURN program **SATPIG** can be used to produce a file of origin-destination route flows from a SATURN assignment. To run **SATPIG,** use a command line such as:

SATPIG net trips

where:	net.ufs	-	Input post-assignment network file;
	trips.ufm	-	Input trip matrix;
	net.**TRP**	-	Output route file

The outputs file *net.TRP* are in the format described in Section 4.3.1, and hence can be loaded in by DRACULA straight away.

In versions earlier than **SATURN 10.3,** the route flows were stored in file *net.LPG* which combines SATURN line printer and route file. The format of *.LPG* files are very different from that of *.TRP*. To enable **DRACPREP** to load route flows from *.LPG* format, set the following parameter in *.PAR* file:

QSATPIG = T

4.3.3.1 PCU vs. vehicles

Flows in SATURN are measured in pcu/hr; they need to be converted to vehicles/hr. A conversion factor PCU can be specified in *.PAR* file for **DRACPREP** to carry out the conversion for a single user class assignment (or single type of vehicles simulated). For multiple user class assignment, each user class should have a specific value of PCU factor input in *.PAR* file, as indicated by a subscript. Thus PCU(2) = 1.5 would assign a pcu factor to user class 2. Table 4.7 lists the parameters used to specify user-class-dependent PCU values.

Table 4.7 Specification of PCU values for single and multiple user class.

Parameter	Defaults	Description
VCPCU	1	PCU value per vehicle for all vehicles in the trip matrix
VCPCU(v)	1	PCU value per vehicle for user class v

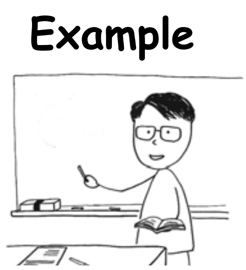
When loading demand from **SATPIG** outputs, **DRACPREP** will convert SATURN flow in pcu/hr (X_{pcu}) to DRACULA flow in veh/hr (X_{veh}) as follows:

$$X_{veh} = X_{pcu} / PCU \qquad (4.3)$$

Hence, a SATURN flow of 200 pcu/hr with PCU=2 would result a flow of 100 vehicles/hr for DRACULA.

Assuming that there are three user classes, car, taxi and HGV using the <net> road network, passenger car unit equivalent for car is indicated as VCPCU(1) = 1, VCPUC(2) = 1.3 for large taxi and VCPCU(3) for HGV. It is easy to see that the reference number for each user class should match with the 'level' for that user class as defined in the stacked matrix of travel demand. An example of *.PAR* is given below:

```
PARAMETERS
 VCPCU(1) = 1
 VCPCU(2) = 1.3
 VCPCU(3) = 3
END
```



Bus flows in SATURN

Buses in **SATURN** are coded as fixed flows along the specified routes via 66666 records in the *.DAT* file. They are coded in unit of service frequency (i.e. number of buses per hour per service). Bus flows are not included in the SATURN demand matrix. When SATURN loads the bus flows, it first converts them from buses/hr into PCUs/hr after multiplying by the BUSPCU factor (defined in the SATURN network parameters list), then assigns them on to the road network.

As DRACULA simulates individual vehicles as opposed to traffic flow measured in PCUs, it uses the SATURN bus frequency data specified in the 66666 records directly. Hence, there is no need to define any PCU value for buses to convert SATURN bus data to DRACULA bus data.

4.3.3.2 Multiple user class assignment

To import results from a SATURN multiple-user-class assignment to DRACULA, set the corresponding PCU values (Section 4.3.3.1) and vehicle types for each user class (4.3.1.1) in *.PAR* file.

4.3.3.3 Multiple time period assignment from SATURN

The “quasi-dynamic” assignment modelling of SATURN models time-dependent matrices and the effect of over-capacity queues on time period assignment. See Chapter 17 of the SATURN Manual on how to set temporal demand in *.DAT* file and how to run **SATTPX** and **SATSUMA** to produce multiple time period assignments.

The result is a series of SATURN *.UFM* and *.UFS* files, one pair for each time period. Running **SATPIG** on each pair of the *.UFM* and *.UFS* files, the SATURN route choice for each time period is extracted to DRACULA-format *.TRP* files.

Then following the instructions given in Section 4.3.1.2 and Table 4.5 to feed these data into **DRACPREP.**

4.3.3.4 Removal of sparsely-utilised routes

Some routes generated by SATURN have very small flows. To remove such routes from DRACULA simulation, the following global parameter can be used to set a limit on the absolute minimum flow a route a selected by DRACULA.

Table 4.8 Parameter to limit small-flow routes

Parameter	**Defaults**	**Description**
FIJPMIN	0.0	Minimum flow (veh/hr) below which a route is not selected for DRACULA simulation

5 MODELLING PUBLIC TRANSPORT

5.1 The Model

5.1.1 Introduction

This section describes the developments made to adapt the DRACULA microscopic traffic simulation model in order to meet the requirements for the modelling of traffic management measures for public transport. The work was funded by the UK Engineering and Physical Science Research Council (EPSRC) as part of a project to look at traffic management issues for kerb guided bus systems.

Specifically, developments have been made in DRACULA in order to model:

- public transport (buses and guided buses) services;
- reserved bus lanes, bus stops and bus laybys;
- passenger volumes and their effect on bus dwell time;
- selective vehicle detection;
- journey time prediction for forecasting the arrival of the bus at the stopline from its upstream detection;
- a variety of responsive signal control policies, such as extension of green, early termination of red, and payback to maintain offsets in subsequent stages.

One of the objectives of the EPSRC-funded project was to compare guided bus operation with traditional bus priority measures, such as reserved bus lanes and bus signal priority. The next few sections describe the developments made in DRACULA to represent bus operations and reserved bus lanes. Modelling of selective-vehicle responsive signal priority measures is described in Section 8.

5.1.2 Modelling Public Transport Service

The public transport services in the model are described by:

- service number;
- vehicle type (bus or guided bus);
- service frequency (veh/hr);
- departure time of first service
- fixed route in terms of nodes through the network;
- a list of bus stops en-route.

Only the departure time of the service (via a fixed hourly service frequency) is modelled. The bus schedule (in terms of route timing points) will be included in the near future.

There are two distinct types of buses modelled: ordinary buses and guided buses. The distinction is made in terms of both vehicle characteristics and with the traffic regulations governing their movement on the streets.

The characteristics for each vehicle are randomly selected from a normal distribution at the start of each trip, but are fixed during the trip. Section 6.1.3 describes the full range of vehicle types and characteristics modelled in DRACULA and how they can be defined. The default average values used for buses and guided buses are listed in the following table, compared with those for cars. The default coefficient of variation is 0.1 for all parameters.

Table 5.1Vehicle characteristics

	Car	Bus	Guided Bus
Length (m)	4.5	7.5	7.5
Min. safety distance (m)	1.0	1.0	1.0
Normal acceleration (m/s^2)	1.5	1.5	1.5
Max. acceleration (m/s^2)	2.0	1.6	1.6
Normal deceleration (-m/s^2)	2.0	1.5	1.5
Max. deceleration (-m/s^2)	5.0	2.5	2.5
Speed factor	1.0	1.0	1.0
Gap-acceptance factor	1.0	0.5	0.5

The risk factor here represents a gap-acceptance behaviour – see Section 6.1.3 for full description. A risk factor of 0.5 is used for the two types of buses to represent the bus drivers' lane-changing behaviour: they tend to take smaller gaps than car drivers to push into the main stream of traffic or to an adjacent lane.

5.1.3 Modelling Bus Stops and Bus Dwell Time

An ordinary bus-stop is a single sign on the road side and a bus stops alongside the sign on the road to pick up or put down passengers, thereby blocking the following traffic in that lane. A bus-layby, however, provides a space for the bus to pull into at the bus-stop and thus allows following traffic to pass. A bus-layby in the model is represented as a special type of bus-stop.

A bus-stop is described by the following data:

- the bus-stop identification;
- the link it is on;
- location of the bus-stop on the link, measured from the entry of the link;
- type of bus-stop (stop or layby);
- length of the stop (in case of layby);
- lateral location on the link (kerb side or median);

- average passenger arrival rate (passengers/hour) at the bus-stop.

There can be several bus-stops on one link. One bus-stop can be used by more than one bus service.

Bus movements in presence of a bus stop

Before entering a link with a bus-stop, a bus will try to get into a lane which both permits its next junction turning movement and leads it onto the lane with the bus-stop. Failing to do so, the bus would first get into the next link, then look for gaps to move to the lane with the bus-stop.

Entering in a link with bus-stop(s), a bus would always aim to travel in the lane where the bus-stop lies (or adjacent lane in case of layby). This may involve changing to the lane with the bus-stop. The bus would only look to get into a lane that permits its next junction turning *after* it has passed the last bus-stop on the link. A bus only stops at its own bus-stops as listed in its service data. A bus in a layby does not affect the traffic on the road. In the model, a bus which stopped at a layby is "removed" from the stream of traffic on the road. Once it has picked up all the waiting passengers, the bus will try to re-merge into the traffic stream according to gap-acceptance rules.

The passenger volume at each bus stop is represented by an average hourly flow (λ), which is converted to a uniform rate of arrival.

.

Passenger flow and bus dwell time

The bus dwell time (T_d) at a bus-stop is related to the number of passengers (N) waiting at the bus-stop in the following way:

$$T_d = a_1(1 - p_s)N + a_2 p_s N + b \tag{5.1}$$

where a_1 is the time it takes for one passenger to get on a bus including paying to get a ticket, a_2 the time it takes for a seasonal ticket holder to get on the bus, p_s the proportion of passengers using seasonal bus tickets, and b the time for the door of bus to be opened and closed. A value of a_1=4 second/passenger, and b=5 second are suggested (Clark et al 1996). Different values can be specified by the users via parameter setting (see coding in Section 5.2).

In the model, a bus has unlimited passenger capacity. Passenger routes are not modelled, nor is delay due to passengers egress. Once a bus stops at a bus-stop, it will pick up all the waiting passengers regardless of the number of passengers and regardless of where they want to go. Consequently, when a bus stops at a bus-stop, no other buses need to stop at that bus-stop. A bus only stops if there are passengers waiting.

While a bus is picking up passengers at a bus-stop, any more passengers arriving at the bus-stop will be added to the passenger queue and the bus dwell time will be prolonged accordingly. However, once a bus is in the mode of closing its door or has closed its door, it would not re-open its door for the new arrivals. In the case of a bus-layby, even if the bus is still waiting to leave the layby, passengers newly arrived at the layby still have to wait for the next bus. The next bus "sees" the waiting passenger and will aim to stop at the layby, which helps to create gaps for the waiting bus to leave the layby.

5.1.4 Modelling Reserved Lanes

Reserved lanes are specified in the model by:

- the link identification;
- location on the link (kerb or median lane);
- type(s) of vehicles reserved for the lane;
- set backs at the begining and end of the link;
- start and end time of the reservation, measured from the start of the simulation.

Lane Choice in presence of a reserved lane

It is assumed that bus drivers all have perfect knowledge of the network. Thus they know if there is a reserved bus lane in the next link well in advance, and will try to move into a lane (in the current link) which leads it naturally (geometrically) into the reserved lane in the next link. If it is not possible, a bus would go into the next link first, then keep looking for chances to move into the reserved lane, as long as the reserved lane permits its next junction turning, or else it is not too close to the end of the reservation.

In contrast, a private vehicle driver only obtains information about their next link when they approach the junction. If the start of the reserved lane (in the next link) is further than 50 metres from the upstream end of the link, the driver would ignore the reservation and choose to get into a lane in the next link according to its position in the current link and the traffic situation in the next link.

If the set-back of the reservation at the beginning of the next link is less than 50 metres long, the private vehicle driver would choose a lane from the remaining lanes in the next link, merging with other traffic from the same link if necessary.

Moving in a link containing a reserved bus lane in action, a private vehicle would start to try to get off the bus lane 50 metres before the start of the reservation. If there is no setback at the downstream end of the link and that is the only lane that permits the vehicle's next junction turning, the private vehicle **will** take the reserved lane on approaching the junction. If there is a setback at the downstream end, vehicles (of all types) may use that lane as long as it permits their next junction turning and by moving into that lane will increase their speeds.

5.2 Data Files

Public transport data are stored in file **.BUS**. The file contains three sections of records, each is preceded with a line containing a string beginning with a character "&" followed by a record name, and ended with a line containing a string "99999". The three sections are:

1. Section 1 – Bus service routes. Preceded by "&BUS_SERVICE". See Section 5.2.
   Within this Section, there are three types of records:

   1) Record type 1 – Service description

   2) Record type 2 – Service route.

   3) Record type 3 – Bus stops en-route.

2. Section 2 – Bus stops. Preceded by "&BUS_STOP" (5.2)

3. Section 3 – Reserved bus lane. Preceded by "&BUS_LANE" (5.2).
   Within this Section, there are two types of records:

   1) Record type 1 – Link description

   2) Record type 2 – reserved lane description

To code in bus service routes, bus stops and reserved bus lanes from scratch, follow the format described in the next three sub-sections and entered via text file *net.BUS*.

In addition, the following parameters can be set in *.PAR* file to specify bus related variables:

Table 5.2 Parameters of bus operation

Parameter	Default	Definition
QBUS	F	Set T to allow loading from **.BUS** file.
PFASTBRD	0	Percentage (%) passengers holding seasonal bus tickets.
TBOARD1	4.0	Ticket-purchasing and boarding time per passenger (sec/person) i.e. constant $\boldsymbol{a_1}$ in Eq 5.1.
TBOARD2	1.0	Boarding time (second/person) for reason-ticket holders. i.e. constant $\boldsymbol{a_2}$ in Eq. 5.1
TDOOR	5.0	Door opening and closing time for buses (sec), i.e. the constant $\boldsymbol{b}$ in Eq 5.1.

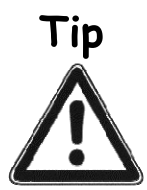
5.2.1 Bus service data

Data in this section must be preceded by a single line containing string "&BUS_SERVICE" and terminated by a single line containing "99999".

DATA	NAME	DESCRIPTION
***** RECORD 1 – SERVICE DESCRIPTION*****		
1	BUSNO	Name of bus route, alpha-numeric string of up to five characters
2	TRROUTE	One of the following characters entered in the space immediately after BUSNO: 'T' if the route is two-way 'R' if the route is defined in reverse order otherwise leave blank. See Note (i).
3	VTYPE	Type of buses, see Section 6.1.2.
4	TSTART	Time the first service starts (minutes), from the beginning of the simulation.
5	FREQ	Frequency of the service (minutes)
***** RECORD 2 – BUS ROUTES *****		
1	NNODES	Number of nodes en route. See Note (ii)
2	NODE(1)	First node en-route
3	NODE(2)	Second node en-route
etc	etc	
11	NODE(10)	The 10th node en-route
Start with a new line		
12	NODE(11)	The 11th node en-route
etc	etc	
***** RECORD 3 – LIST OF BUS STOPS *****		
1	NSTOP	Number of bus stops en-route. See Note (iii)
2	STOP(1)	First bus stop for the service
3	STOP(1)	Second bus stop for the service
etc	etc	
4	STOP(10)	The 10th bus stop
Start with a new line		
11	STOP(11)	The 11th bus stop
	etc	etc

NOTE:

a. If TRROUTE='T' or 't' and the node list is A…Z, then a return bus route with the nodes in exactly reversed order (e.g. Z,…A) is automatically created by **DRACPREP** and added to the bus route list. If however TRROUTE='R' or 'r', then only one route is assumed whose nodes are Z,…A.



The R (or r) option allows one to code a 2-way route which differs marginally in the two directions. Thus if the "out" direction is A…IJL…Z and the "in" direction is Z…LKI…Z then one may code it using A…IKL…Z, i.e. the same as the out direction apart from one node.

This is a similar function as that exists in SATURN. Hence the SATURN users may wish to create the .BUS file with the same data as coded in SATURN network data record 66666.

Bus route along a straight line

b. If all or parts of a bus route follows a straight line (more or less), it is not necessary to define every node along the line, but only the nodes at the corners or at the ends. The NNODES should be the number of corners coded instead of the actual number of nodes en-route.

c. All bus stops en-route should be coded.

Note in SATURN networks, public services are coded as routes under card 6. Use this information to help code up the public transport services in the following format. If any of the links on the bus routes are really part of a link in the real network, as with the reserved lanes, de-code the bus-links to lanes.

Use of SATURN bus routes

Bus routes coded in SATURN networks can be converted to DRACULA data. Following the procedure below:

If there are bus routes to be extracted from existing SATURN network, go to step 1, else go to step 4:

Step 1: Set parameter QSATNET=T, QBUS=F in *net.PAR* file

Step 2: Run **DRACPREP**. On completion, you should have a *net.BUS* file created which has three records beginning with:

&BUS_SERVICE
&BUS_STOP
&BUS_LANE

Each record ends with a string 99999. There should only be data entries under the first record.

Step 3: go to Step 4 for further data-editing

Step 4: Set parameter QBUS=T in *net.PAR* file

Step 5: Edit text file *net.BUS* following the instructions below to add more, or build from scratch, bus routes/bus stops/bus lanes.

***** RECORD 2 – BUS ROUTES *****

1	OZONE	Origin zone number
2	DZONE	Destination zone number
3	NNODES	Number of nodes en route
4	NODE(1)	First node en-route
5	NODE(2)	Second node en-route
etc	etc	
13	NODE(10)	The 10th node en-route

Start with a new line

14	NODE(11)	The 11th node en-route
etc	etc	

If you have coded the network with this old format, you can still use them by setting a parameter QOLDBUS=T in your *.PAR* file.

5.2.2 Bus stop data

Data in this section must be preceded by a single line containing the string "&BUS_STOP" and terminated by a single line containing "99999".

DATA	NAME	DESCRIPTION
1	STOPNO	Bus stop number
2	ANODE	Upstream node number of the link on which the stop lies
3	BNODE	Downstream node number of the link
4	POSIT	Location of bus stop on link measured from link entry (m)
5	SIDE	Side of the road where the bus stop lies: 0=curb side, 1=medium side
6	TYPE	Type of bus stop: 0=ordinary, 1=layby
7	LENGTH	Layby length (m)
8	PFLOW	Average pedestrian flow (pedestrian/hr) or average dwell time (sec) to the bus stop
9	COVPFLOW	Coefficient of variation (COV) of pedestrian flow or COV of dwell time.

In general, public transport means buses (including park & ride buses) and guided buses which follow fixed routes in a network. If their routes can be specified, taxi services can also be coded as a public service in this file. But, generally, taxi trips are included as part of the private trip matrix.

5.2.3 Reserved bus lane data

Data in this section must be preceded by a single line containing the string “&BUS_LANE” and terminated by a single line containing “99999”.

DATA	NAME	DESCRIPTION
***** RECORD 1 – LINK DESCRIPTION *****		
1	ANODE	Upstream node of the link
2	BNODE	Downstream node of the link
3	NRESV	Number of reserved lanes on link
***** RECORD 2 – RESERVED LANE *****		
1	SIDE	Relative position of the lane on link, measured from kerb side: kerbside lane=1
2	RTYPE	Type(s) of vehicles the lane is reserved for
3	XSTART	Start position of the lane (metre), measured from link entry
4	XEND	End position of the lane (metre), measured from the stop-line of the link
5	TSTART	Start time (minutes) of the reservation, relative to the start of the simulation (at the beginning of warm-up period)
6	TEND	End of reservation time period (minutes)

Tip

N.B. the XSTART and XEND are measured relative to different point of the link.

In DRACULA a lane can be reserved for any type or a combination of several types of vehicle. Each type of vehicle has a unique (binary) number defined by DRACULA. See Section 6.1.2 for definition of vehicle types.

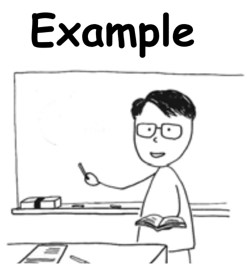
To specify a lane reserved for a single type of vehicle, simply give the number of that vehicle type (see Section 6.1.2). To specify a combination of more than one type of vehicles, use the sum of their vehicle types. For example, a lane which has a reserved type 2 is for bus only, 6 for bus and guided-bus, 10 for bus, guided-bus and taxi. A lane reserved for type 63 is for all types of vehicle (e.g. not reserved), and 0 means it is a blocked lane. One can therefore model a local incident that causes (partial) lane closure for a specified period of time.

5.3 An Example

The following example public transport data inputs show that:

- there are two public transport routes in the network, both served by buses;
- the first service (number 30) begins its operation 10 minutes after the start of the simulation period and runs every 10 minutes along a route that passes 17 nodes and stops at 4 bus-stops.
- the service number 33 does not stop anywhere en-route (it is probably an express bus);
- there are two reserved lanes in the network: they are along links 1511-721 and 721-1511. Both reserved for buses with setback of 150 metres and operate all day.

```
&BUS_SERVICE
  30    2  10  10
              17   7211  721   722   723  727  731  783  782   12   15
                          13    21  9903  9908  132  182  9910
              4      101  102   104   107
  33    2    8  20
             12     773   777  782    12  15   13    21  9903  9908  132
                        182   9910
              0
99999
&BUS_STOP
 101  722   723 100  0 1 20 25
 102   15    13    5  0 0   0 35
 104   21 9903 130  0 0   0 20
 107  132   182   90  0 1 20 25
99999
&BUS_LANE
 1511  721   1
          1  2  0 150  0 36000
  721 1511   1
         1  2  0 150  0 36000
99999
```



5.4 Guideway for Guided Buses

A guideway is generally thought to be best represented as a separate link in a network. Code the guideway links in the network data file, under card 1.

5.5 Time-Dependant Passenger Demand

Similar to the model of time-dependant traffic demand with fixed route assignment (ie Section 4.3.1.3), the passenger flows may also be modelled as time-dependant step functions. The average passenger flows arriving at each bus stop (and its random variation) are specified as above (see Section 5.2.2).

By using the following parameters, to be specified in .PAR file, a time-dependant passenger demand profile can be described. Note that this is a global profile which applies to passenger flows to all bus stops.

Table 5.3 Specification for time-dependant passenger demand

Parameter	**Defaults**	**Description**
P_NSTEP	1	Number of time steps. Maximum 10 steps.
P_TSTEP(1)	60	Duration (minutes) for the first time period.
P_TSTEP(t)	0	Duration (minutes) for time period t (t=2,3,..10)
P_GONZO(1)	1.0	Passenger demand factor for the first time period
P_GONZO(t)	0.0	Passenger demand factors for time period t=2,3,…, 10

6 SIMULATION SETTINGS

The traffic model in DRACULA is based on a discrete-time micro-simulation of individual vehicles' space-time movements, with an time increment of 1 second. The simulation is based on individual vehicles' characteristics, their car-following, lane-changing and gap-acceptance behaviour and traffic controls in the network. Simulation results are affected by the input vehicles' characters, on parameters describing their driving behaviours and on conditions and controls in the network. This section describes the settings that could affect the simulation. These settings are divided into global ones which affect the whole network and throughout the whole simulation time periods, and local settings which affect only parts of the network and during parts of the time.

6.1 Global Settings

6.1.1 Simulation Periods

Three time periods were simulated in DRACULA; these are schematically depicted in Fig. 6.1.

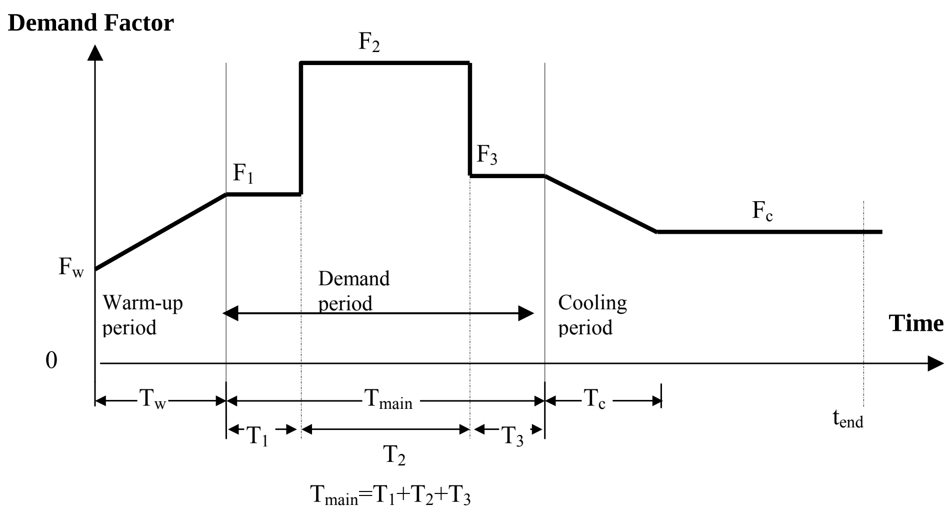


Fig. 6.1 Simulation time periods and representation of temporal demand profile.

The "**demand period**" is the main simulation period, specified via parameter "PERIOD" in DRACULA network coding (Section 4.2.1), and LTP from a SATURN network (Section 4.2.2). The demand period is typically one hour representing the peak period. There could be a number of sub-periods within the main study period to represent the temporal distribution of demand (Section 4.3.1.2). The demand levels for each time period (F_1, F_2, etc in Fig. 6.1) are represented as a proportion of some nominal demand, for example an average demand level.

A **warm-up period** (T_w) is simulated with demand linearly increasing from a fraction F_w of the peak level to the peak as a way of ensuring that the traffic from the demand period does not start with an empty network.

At the end of the main period, the simulated demand linearly decreases from the peak level to a level F_c equivalent to that of an off-peak over a "**cooling-off period**" (T_c) and thereafter stays at that level until the end of the simulation (t_{end}).

Various time periods and associated demand levels shown in Fig. 6.1 can be specified via input parameters in .**PAR** file.

Symbols	**Parameters**	**Defaults**
T_w	TWARM	15 min
F_w	FWARM	0.5 (see Note)
T_c	TCOOL	15 min
F_c	FCOOL	0.5 (see Note)
T_{main}	TMAIN	60 min
	NSTEP	
F_1	FSTEP(1)	
F_2	FSTEP(2)	
etc		
T_1	TSTEP(1)	
T_2	TSTEP(2)	
etc		

Note: For historic reasons, the warm-up and cooling of flows are modelled as a proportion to that of the peak demand. Hence a F_w =0.5 means that the demand during the warm-up period will increasing linearly from half of the peak demand (of that of the first time period) to the peak demand. Similarly, over the cooling off period, the demand will be reduced linearly from the peak demand to, by default, half the peak level (F_c=0.5).

By default, a simulation run in DRACULA ends when all vehicles departing during the demand period have finished their journey. The end of simulation is also indicated with the "No. of vehicles from peak-period" on the Simulation Speed screen becoming zero. The actual simulation time will then depends on the level of congestion in the network and the lengths of the journeys.

For example, if the longest journey in a simulation takes 10min to complete in free-flow situation, one would expect the simulation will terminate in 15-20 minute's with moderate congestion in the network. It is possible, however, grid-lock may occur in the simulation. In this case, the simulation could run forever. To eliminate such situation, the simulation will only run for a total of three times of the demand period (counting after the warm-up period); after that period, the simulation will terminate and an error message will be reported. Such situation may also occur if the average journey time is longer than three times of the demand period specified.

It is possible to terminate the simulation at the end of the main demand period (i.e. after TWARM+TMAIN). This can be done by setting the following parameter value in the .par file:

QMAINSIM=T

The choice between running a simulation over a pre-defined time period and tracking all vehicles from the demand period to their destinations (and therefore an unspecified time period) depends on the simulation outputs one wants – please see Section 9.4 for a distinction between the network performance measures and the supply costs of trips.

6.1.2 Vehicle Types

Currently there are seven types of vehicles defined in DRACULA. Each type of vehicle is represented by a name and a unique number. The names and the numbers of the seven types of vehicles are listed in the table below.

Table 6.2 Definition of vehicle types

Name	Number	Description
DUMMY	0	Dummy vehicles
CAR	1	Small passenger cars
BUS	2	Bus type 1
GBUS	4	Bus type 2, for example guided bus
TAXI	8	Taxi
LGV	16	Light goods vehicles
HGV	32	Heavy goods vehicles

The names are given according to their default types, for example, type "TAXI" of number 8 represents, by default, taxi. However, there is no reason why users cannot re-define the types according to their needs, as long as you specify the right characteristics for the right type (see next sub-section). For example, you could user "TAXI" to represent high-occupancy vehicles, or use it for trams, as long as you give the appropriate specification of the characters for your chosen types of vehicles. Take another example, if you need to have three different categories of buses, for example, ordinary buses, express buses and P&R buses, you could "borrow" of the other types that you don't need and re-define the characters accordingly. The only limitation is that you can not have more than six vehicle-types (type "DUMMY" is reserved for some special use and cannot be replaced). Development is underway to incorporate more vehicle types or user classes into the model.

6.1.3 Vehicle Characteristics

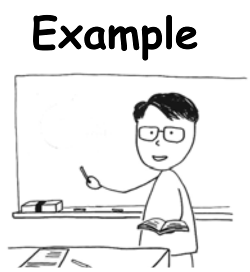
There are nine parameters used in DRACULA to describe the physical and behavioural characteristics of each vehicle-driver unit. They are:

1) length (metre)
2) minimum safety distance (metre)
3) reaction time (second)
4) normal acceleration (m/s2)
5) maximum acceleration (m/s2)
6) normal deceleration (-m/s2)
7) maximum deceleration (-m/s2)
8) desired speed factor
9) gap acceptance factor

The users can specify/change the distributions of the vehicle parameters via a text file, **VEH.TAB.** The format of the file, together with the default characteristics for the default types of vehicles are listed below.

The file starts with an identifier &VEH_PARAM and ends with &END. For each type of vehicle, there are four rows of nine records representing the nine vehicle characteristics. The first row for each vehicle type contains the mean values of its nine parameters. The second row contains the coefficients of variation, the third contains the minimum allowed values of each parameter, and the fourth contains the maximum. As an example, the following file lists the default values.

veh.tab file




```
&VEH_PARAM
CAR
 4.5  1.0  1.0  1.5  2.0  2.5  5.0  1.00  1.0
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 3.5  0.8  1.0  1.0  1.0  1.0  1.0  0.50  0.5
 5.5  1.2  1.0  5.0  5.0  5.0  6.5  2.00  2.0
BUS
 7.5  1.0  1.0  1.5  1.6  1.5  2.5  1.00  0.5
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 5.0  0.8  1.0  0.8  0.8  1.0  1.0  0.50  0.2
10.0  1.2  1.0  2.0  2.0  4.0  4.0  1.50  1.0
GBUS
 7.5  1.0  1.0  1.5  1.6  1.5  2.5  1.00  0.5
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 5.0  0.8  1.0  0.8  0.8  1.0  1.0  0.50  0.2
10.0  1.2  1.0  2.0  2.0  4.0  4.0  1.50  1.0
TAXI
 4.5  1.0  1.0  1.5  2.0  2.5  5.0  1.00  1.0
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 3.5  0.8  1.0  1.0  1.0  1.0  1.0  0.50  0.5
 5.5  1.2  1.0  5.0  5.0  5.0  6.5  2.00  2.0
LGV
 7.5  1.0  1.0  1.2  1.6  1.5  2.5  0.80  1.0
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 5.0  0.8  1.0  0.8  0.8  1.0  1.0  0.50  0.5
10.0  1.2  1.0  2.0  2.0  4.0  4.0  1.50  2.0
HGV
 7.5  1.0  1.0  1.2  1.6  1.5  2.5  0.80  1.0
 0.1  0.1  0.0  0.1  0.1  0.1  0.1  0.10  0.1
 5.0  0.8  1.0  0.8  0.8  1.0  1.0  0.50  0.5
10.0  1.2  1.0  2.0  2.0  4.0  4.0  1.50  2.0
&END
```

6.1.4 Traffic Composition and Demand Factor

When the origin-destination (O-D) trip matrix and route assignment for each user class or vehicle type are known, they can be loaded onto the simulation according to the description given in Section 4.3. This section describes how mixed traffic can be represented with knowledge of only an average trip matrix and assignment.

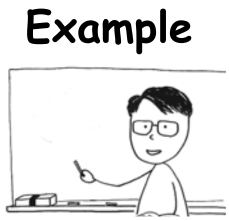
In practice, a single trip matrix is often collected combining several types of traffic. If one knows the average distribution of user classes/vehicle types among the matrix, e.g. how many different types of vehicles there are in the network and, on average, what proportions of the trip matrix are made up by these vehicle types, the following method can be used to approximate the traffic composition. Based on the proportions of each type of vehicles within the overall demand as specified by parameters in **.PAR** file (Table 6.3), the model randomly assigns a type to each individual vehicles according to a probability distribution.

A demand factor via parameter GONZO is used in DRACULA to change the overall level of demand. This is useful when simulating the effect of different traffic demand levels (with fixed route choice) on networks.

The name of this parameter is borrowed from SATURN, where the GONZO value affects not only the level of demand but also, in congested networks, the route assignment results.

Table 6.3 Parameters used to specify traffic composition and demand factor

Parameters	Defaults	Definition
GONZO	1.0	Demand factor.
VCPCU	1.0	PCU value for the overall demand.
PBUS	0	Proportion (%) of total demand is buses. For 10%, enter 10.
PTAXI	0	Proportion (%) of total demand is made up by taxis.
PHGV	0	Proportion of total demand is HGVs
PLGV	0	Proportion (%) of total demand is LGVs



For example, if GONZO=1.5 and VCPCU=1.2, the program **DRACPREP** will first multiply the whole demand by these two factor, and then assigns the proportions, PBUS, PTAXI, PHGV, PLGV of the resulted demand to be of type BUS, TAXI, HGV and LGV. The rest is of CAR type.

The proportions should be specified such that, if PBUS=10, it means 10% of the total demand is of type BUS. Clearly, the total should be less than 100, e.g. PBUS+PTAXI+PHGV+PLGV<100. Indeed they should be much less than 100 in most of cases as the traffic is composed mostly by CAR.

6.1.5 Car-Following Models

Car-following models represent the longitudinal interaction between vehicles. The speed of the following vehicle is modified in the light of the relative speed and position of the preceding vehicle. The parameters required to determine such longitudinal progress of vehicles are: the **desired speed,** the **desired minimum headway,** the **reaction time**, the **rate of acceleration** and the **rate of deceleration.** The implementation of such parameters in DRACULA is via text input file **VEH.TAB**; details are described in Section 6.1.3.

6.1.6 Gap-Acceptance Models

Gap-acceptance models are used to describe how drivers make the decision to merge or cross at intersections. They are so called because they represent the critical gap (seconds) in the opposing stream(s) of traffic that the driver feel safe to accept: if the gap is greater or equal to the critical gap, then merge or cross; otherwise wait.

The key parameter for gap-acceptance models is the **normal acceptable gap** (expressed in seconds) for the manoeuvre being contemplated. Some gap-acceptance models use a fixed value for each manoeuvre, DRACULA allows the minimum gap to be reduced if the traffic is very heavy and moves slowly – a representation of frustration on the part of the waiting traffic. DRACULA uses the time a driver has been waiting to find an acceptable gap as a **stimulus** to induce use of a **reduced gap**. The reduced gap is modelled as a linear function, reducing from the normal gap (GAP) to a minimum gap (MIN_GAP) linearly between two waiting times, GAP_TSTART, and GAP_TEND (Fig. 6.2).

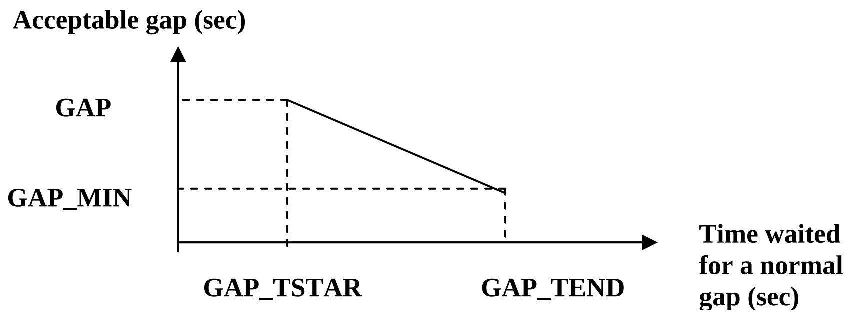


Fig. 6.2 Modelling gap-acceptance behaviour.

Table 6.4 Global gap-acceptance parameters

Parameters	Defaults	Definition
GAP	3.0	Normal acceptable gap (seconds)
GAP_MIN	1.0	Minimum acceptable gap (seconds)
GAP_TSTART	60	Stimulus to induce use of reduced gap: time waited for a normal gap (seconds)
GAP_TEND	180	Stimulus to induce use of MIN_GAP: time waited for a normal gap (seconds)
RB_GAP	3.0	Acceptable gap (seconds) used upon entering roundabouts
RB_VISIBILITY	50	Distance on roundabout which the entry traffic would look back for potential conflicts (metre)
FGW_CAR	0.5	Courtesy factor, fraction of traffic who are willing to giveway for minor-flow cars or cars wanting to change lane
FGW_BUS	0.8	Fraction of traffic willing to giveway to buses

Roundabout visibility

For gap-acceptance at roundabouts, RB_GAP is used instead of GAP. A limit on how far back on the roundabout circulating track an approaching traffic can view, RB_VISIBILITY, is also introduced.

Willingness to create gaps

DRACULA also tries to model the fact that vehicles in the priority flow may take pity on vehicles waiting for a gap and may deliberately slow down in order to create one – in which case at least one parameter will be required to indicate their **willingness to create gaps.** This willingness is modelled as a random proportion of priority flow, with two proportions FGW _BUS and FGW_CAR representing the different willingness drivers may show to buses and to other traffic.

Individual drivers' gap-acceptance values

To represent the variable gap-acceptance behaviour among drivers, DRACULA assigns to each driver a gap-acceptance factor. You can see from Section 6.1.3 that, among the seven vehicle characteristics is a gap-acceptance factor and a speed factor. These factors represent a kind of risk or aggressiveness of the driver: a higher speed factor represents a tendency for driving faster; a smaller gap-acceptance factor the tendency to take smaller gaps, hence more risky.

The safety gap (seconds) the driver actually uses is the multiple of this gap-acceptance factor with an average gap value (seconds). The reason we specify a “gap-acceptance factor” rather than the actual “gap (seconds)” to each driver-vehicle is to cater for the fact that the actual gaps required by the same driver may vary from site to site and from time to time.

6.1.7 Lane-Changing Models

Lane-changing models consider the individual driver’s intention and ability to change lanes. An *intention* to change lanes will reflect the advantage to be gained (eg an increase in speed or an avoidance of delay) or the need to do so (eg in order to comply with a traffic regulation, to avoid an incident in the current lane, or to prepare for a turning movement). The driver’s intention to change lanes may be modelled differently depending on whether the lane-change is “discretionary” or “mandatory”. The intention to make a discretionary lane-change may be triggered when the time advantage to be gained by changing lanes exceeds some **critical value** whereas the intention to make a mandatory lane-change will generally be triggered by **rules**.

DRACULA divides drivers’ lane-changing desires into one of five types when drivers have to or want to change lane in order to:

(a) reach a bus stop on the link;
(b) avoid a restricted-use lane or incident;
(c) make their turn from the next junction;
(d) move into a lane reserved for their type; or
(e) gain speed by overtaking a slower moving vehicle.

The first three types are mandatory, i.e. the lane-changing has to be carried out by a certain position on the current link; the other two types are discretionary. Whether a discretionary lane-change can be carried out depends on the actual traffic conditions. For example, a vehicle would only change lane to gain speed if the speed offered by the adjacent lane is higher by a pre-defined factor. Section 8.6 describes another, special type of lane-changing: overtaking on a single carriage way using the road space in opposite direction.

When a vehicle wishes to change lane, it looks for a target lane. The target lane is generally determined by the lane-changing requirement, except in the case of overtaking which is only permitted from the nearside to the offside. Once it has chosen a target lane, it examines the “lead” and “lag” gaps in its target lane, in a way which is analogous to a gap-acceptance model.

DRACULA allow drivers to anticipate the need for a change of lane. It uses variables XAPPRO_JNCT and TAPPRO_JNCT (see Section 6.1 and 6.2) to determine the distance and time to lane-changing **target point,** such as a bus stop, beginning of an incident or the stop-line of a junction in case of types (1), (b) or (c) lane changing.

DRACULA also allows for the **willingness to create gaps** by kind hearted drivers in the target lane (Section 6.1.6) and **variation in the gap depending on the urgency of the desire to change lanes** – the closer to the target point the smaller the gaps they would be willing to accept.

For overtaking, DRACULA uses the following two control parameters: if the current speed is below F2_OVERTAKE times its desired speed and the speed-gain in target lane would be over F1_OVERTAKE times its current speed, an overtaking desire is triggered. The parameters can be set in **.par** file.

Table 6.5 Parameters specifying lane-changing desires and "cooling" period after one lane-changing movement

Parameters	**Defaults**	**Definition**
F1_OVERTAKE	2.0	Threshold speed factor, change lane only if the speed can be improved that this factor.
F2_OVERTAKE	0.8	Lower boundary factor in desired speed that will trigger a over-taking desire
TINLANE_CAR	15	Time (seconds) a passenger car keeps in the lane it has just changed into before making another lane-changing attempt.
TINLANE_BUS	30	Ditto for buses
TINLANE_HGV	30	Ditto for heavy goods vehicles

Generally drivers make one lane-changing movement at a time. It takes some time for them to adjust to and to assess the new "environment" before attempting another lane-changing move. This behaviour is modelled through a "cooling-off" period assigned to each lane-changing movement, during that period the vehicle is not permitted to another lane-changing move.

The cooling period is represented with global parameters TINLANE_CAR, TINLANE_BUS and TINLANE_HGV which can be specified in .par file. We model different cooling periods for different type of vehicles to represent the ability of different types of vehicles to move laterally.

By specifying a cooling period greater than one simulation time step (e.g. 1 second), we eliminate chances of any "swooping", e.g. a vehicle crosses several lanes at a time such as a sudden change from lane 3 to lane 1, happening in the model.

6.1.8 Reaction to Traffic Control at Intersections

Drivers start to react to traffic controls at intersections when they are within a certain distance of the intersection, called “junction reaction zone” This distance varies with the speed of the traffic: if the traffic is moving at a higher speed, it will “feel” the control from the downstream intersection earlier. Similarly, the lane markings for turning movements may only be displayed within certain distance to the junction where the drivers find they may have to change lane in order to get into the correct lane for their junction turning movements.

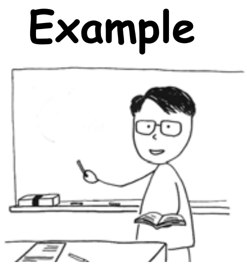
A global parameter is used to represent this effect, TAPPRO_JNCT, a time headway to the downstream intersection when the vehicle will react to junction control and their desired (according to their fixed route) junction turning movement. In the simulation, this time headway is then converted into a distanced-based *junction reaction zone* for each link according to the free-flow speed of that link:

$$RX_l = TAPPRO\_JNCT \times V_l^o$$

where RX_l is the distance on link l to the downstream intersection whereby the effect of junction is felt by the traffic, V_l^0 is the free-flow speed of link l. In the simulation, it is this localised, link-specific RX_l which plays an effect (see also how to set RX_l directly for link l in Section 6.2).

Up the Version 2.2 (i.e. programmes released before 2005), a semi-time based variable, XAPPRO_JNCT, is used to describe the junction reaction zone. XAPPRO_JNCT is defined as the distance (metres), per every 30 km/hr of a speed. For example, if the link free-flow speed is 30 km/hr, then its reaction zone starts 50 metres (by default) upstream of the junction. If the link free-flow speed is 60 km/hr, then the zone starts 100 metres upstream of the junction. The conversion from the variable XAPPRO_JNCT to the new variable TAPPRO_JNCT is then:

$$TAPPRO\_JNCT = \frac{XAPPRO\_JNCT}{30 \times \frac{1000}{3600}} = 0.12 \times XAPPRO\_JNCT$$



and

$$RX_l = 0.12 \times XAPPRO\_JNCT \times V_l^o$$

As long as TAPPRO_JNCT is specified for a network (in its .par file), its value will be used in the simulation, regardless whether or not XAPPRO_JNCT is also specified.

The response to junction control or lane-markings for junction turning restrictions could be in the form of stopping at red signals, or, in the case of mandatory lane-changing, drivers become more “desperate” to change lane if they are within such a distance from their target point.

On approaching a give-way stop-line and within the distance XAPPRO_JNCT, drivers would first aim to stop by the stop-line. When they get closer to the junction and can view the other approaching traffic, they could speed up and move off if gaps are available. A parameter in DRACULA, XVIEW_JNCT, is used to describe the distance to the stop-line where drivers could become aware of traffic flow on other approaches.

Table 6.6 Parameters specifying response to traffic control at intersection

Parameters	Defaults	Definition
TAPPRO_JNCT	18.0	Estimated time (sec) before approaching a junction from when a vehicle begins to react to the junction control. Estimated based on the current cruising speed. Sections 6.1, and XAPPRO_JNCT
XAPPRO_JNCT	150.0	Distance (metres per 30kph) upstream of a stop-line where a vehicle starts to react to junction control
XVIEW_JNCT	6.0	Distance (metre) upstream of a stop-line where a vehicle starts to look for gaps in the conflicting traffic

6.2 Local Settings

6.2.1 Junction- or Link-Specific Gap Acceptance Parameters

SATURN specifies node-based gap values in its network description files (see SATURN Manual, Chapter 6, for details). For example, in SATURN networks, one can specify parameters GAP and GAPR for priority junction and GAPM for merge. These node-based gap acceptance values are read in by DRACULA and used as the mean gaps for the junction concerned.

The records in this section define link-specific gap acceptance parameters. They give the ability to define gap acceptance not only on a junction by junction basis but also approach by approach. Wherever link-dependent gaps are given, they over-write the global parameters as defined in **.PAR** file and the node-based gaps as specified in SATURN node records.

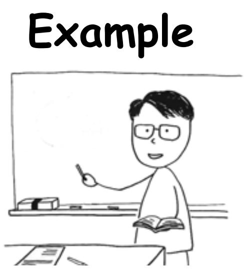
To specify gap acceptance parameters for a given link, use the additional network inputs in text file **.ADD**. Insert a section in the file beginning with “&GAPS” and ending with “99999”. Enter the links and the gap values in between the start and end delimiters, following the format below.

DATA	NAME	DESCRIPTION
*****	LINK GAP DATA *****	
1	ANODE	Upstream node
2	BNODE	Downstream node
3	GAP	average gap value (seconds) of the link
4	MIN_GAP	minimum gaps (seconds) of the link
5	GAP_TSTART	time (sec) since waiting before reducing gaps
6	GAP_TEND	time (sec) since waiting before taking MIN_GAP

The example below suggests that vehicles approaching node 21 along link 31-21 take an average gap of 5 seconds during the first 60 seconds of waiting time, then gradually reducing their acceptable gap to the minimum gap of 2 seconds at/after having waited 120 seconds. Vehicles along link 55-21 have an average gap of 4.8 sec, minimun gap 1.2 sec, and two waiting times are 80 and 160 sec respectively.

net.ADD

```
&GAP
        31  21  5  2  60  120
        55 21  4.8  1.2  80 160
99999
&STOP
          5  20  5.5
99999
&RZONE
          5  20  85.0
99999
&CIRC
     1115    20
```



```
    1116    30  -1  20 .2   4
99999
```

The rules for using various level of gap values are:

Tip

Different gap-acceptance values

(1) by default, all approaches/links use the default global gap values, e.g. GAP=3.5 seconds, MIN_GAP=1 sec.
(2) If gap values are specified in the DRACULA parameter file, .par file, they will be used for all approaches/links;
(3) If node-based gaps are given, as in 11111 records, they will be used as the GAP and MIN_GAP values for links approaching the node; and finally
(4) if link-based gap values are specified in network data files, as described in this section, these values will be used for the link.

6.2.2 Stop-line Location

By default, the stop lines are positioned at a distance of 3 meters upstream from the corner of the road. This distance can be specified by parameter X_STOPLINE in *.PAR* file, which applies to all stop lines in the network.

It is possible to specify stop line locations for individual links, as additional network data in file *.ADD*. Enter the records in between markers &STOP and 99999 as follows:

DATA	NAME	DESCRIPTION
***** LINK GAP DATA *****		
1	ANODE	Upstream node
2	BNODE	Downstream node
3	XSTOP	stop-line location (metre) upstream from the corner of the road.

In the example net**.ADD** shown in Section 6.2, the stopline on link from node 5 to node 20 lies 5.5 meters upstream from the corner.

6.2.3 Reaction zone to junction control

A reaction to traffic control at the downstream junction can be specified as a global variable (see Section 6.2), or be specified for individual links via input in file *.ADD*. Enter the records in between markers &RZONE and 99999 as follows:

DATA	NAME	DESCRIPTION
***** JUNCTION REACTION ZONE *****		
1	ANODE	Upstream node
2	BNODE	Downstream node
3	XRZONE	distance (metre) upstream from the junction

In the example net**.ADD** shown in Section 6.2, the reaction zone on link 5 - 20 starts 85 meters upstream to the junction.

6.2.4 Roundabout Specification

A roundabout is specified through a number of variables:

HICD	-	Radius of outer boundary of the roundabout
NRBL	-	Number of circulating lanes
R_{island}	-	Radius of the central island
V_{circ}	-	free-flow circulating speed

Section 4.2.2.6 described how some of these variables (e.g. HICD and NRBL) are interpreted from information coded for a SATURN-format roundabout and others specified as global variables (e.g. V_{circ} and R_{island}) in *.PAR* file.

Now all the above variables can be specified for individual roundabout via additional network data file ***.ADD***. Enter the data in between record markers &CIRC and 99999, as:

DATA	NAME	DESCRIPTION
1	NODE	Node number of the roundabout
2	CSPEED	Circulating speed (km/hr)
3	RISLAND	Radius of central island (m)
4	HICD	Radius of the outer boundary (m)
5	NRBL	Number of circulating lanes

NOTE

Tip

1. Data for each roundabout should be specified on a separate line
2. Not all four fields for a roundabout need to be specified. If a field is left blank or a value –1 is entered, the default (global) value for the variable will be used. If a field is left blank, all other files afterwards have to be left blank, too. Otherwise, the program will read the field after as that meant to take the default.

The example shown in Section 6.2 defines, for roundabout of node number 1115, a circulating speed of 20kph and leaves all other variables their default values. The example also specify for roundabout 1116, a circulating speed of 30kph, radius of the central island that of the default value, outer radius of 20.2 meters, and with 4 circulating lanes.

6.2.5 Road Works and Incidents

DRACULA can model planned road works and regular incidents (such as illegally parked vehicles), where location and duration can be specified. Such incidents are modelled as lane closures for the specified period and locations of the lane involved. To specify such lane closure or blockage, specify the lane as a reserved lane and reserve it to DUMMY vehicles; see Section 5.4 for the specification of reserved lanes.

DRACULA does not model irregular incidents, such as accidents, whose location, duration and frequency of occurrence can not be pre-defined.

6.2.6 Yellow Box Junction

To specify a junction as a yellow box, simply add a letter 'Y' or 'y' immediately after the node-type specification in network coding, e.g. at column 16 of Node Data record of a SATURN network.

6.2.7 Motorway Merge

The acceleration lane on a motorway merging section can be coded as a "dead-end" lane (no turning movements can be made from this lane), as part of a motorway link. The length of this link being that of the acceleration section. The one-ramp traffic entering the motorway can be coded with a turn-priority-marker S (see Section 4.2.1.5).

6.2.8 Flared Approaches

There are two ways to model a flared approach. One method is to insert a dummy node before the flared section and model the flared section as a separate link.

The DRACULA recommended method is to model the whole link with the number of lanes on the flared section, and model the first part of the lane (before the flared section) as blocked (see Section 5.2.3).

Example

Example: a one-lane link of 100m length, from node 1111-1112, flares into two lanes on approaching its downstream node 1112. The flared section is 20m and is flared to the kerbside.

Model solution:

- code the link with two lanes in the network description file (*net.NET*).
- code the first 80m of the kerb lane as a reserved lane which is reserved for "DUMMY" type vehicles. (Hint: code reserved lane in *net.BUS* file record &BUS_LANE).
- Run **DRACPREP** and **DRACSIM**. You can change the reserved lane colour using menu Display/colour.

7 EMISSION AND FUEL CONSUMPTION MODELS

It is well known that vehicles produce more harmful emissions when operating in acceleration and deceleration modes than in cruising mode. The DRACULA emission and fuel consumption models takes into account explicitly vehicles' four different driving modes (acceleration, deceleration, cruising and idling) and calculates emission by pollutant and fuel consumption for each individual vehicle based on their instantaneous speed and acceleration.

The emission factors used are taken from the QUARTET Deliverables No. 2 (QUARTET, 1992). Emission rates for three pollutants are available: CO, NOx and un-burnt HydroCarbon (HC) emission. For vehicles cruising at a constant speed, the emission factors (grams/second) are assumed to be a function of speed, as shown in the table below.

Table 7.1 Emission factors for driving at constant speed

Pollutant\ Speed(kph)	10	20	30	40	50	60	70
CO	0.06	0.091	0.13	0.129	0.09	0.11	0.177
NO_x	0.0006	0.0006	0.0017	0.0022	0.0042	0.0050	0.0058
HC	0.0063	0.0078	0.0083	0.0128	0.0097	0.0117	0.0136

The emission factors for the cruising mode were given as discrete speed points (at every 10kph). A linear interpretation is used to get emission factors at speeds in between any of the two listed in the table. In other driving modes (idling, accelerating, or decelerating), constant emission factors are assumed (Table 7.2).

Table 7.2 Emission factors for idling, accelerating and decelerating

	Idling	Accelerating	Decelerating
CO	0.06	0.377	0.072
NOX	0.0008	0.01	0.005
HC	0.0067	0.02	0.0067

The fuel-consumption factors are taken from Ferreira (1982) and DOT (1991). It has assumed that fuel consumption factors are constant for vehicles that are in idling and decelerating modes, and vary as a function of speed for cruising vehicles. For vehicles that are in acceleration mode, the fuel consumption factor (f) is a function of both speed (v) and acceleration (a) of the vehicle:

$$f = c_0 + c_a \times a \times v \tag{7.1}$$

where c_0 and c_1 are two constants given in Ferreira (1982).

The models are developed to be flexible so that new values (in terms of new emission rates for the existing pollutants and for new pollutants) can be incorporated as they become available. The new factors can be fed into **DRACSIM** via a look-up table *POL.TAB*, so long as the relationship between traffic condition (driving mode) and emission remains the same. The following example file contains the default factors used in **DRACSIM**:

POL.TAB

```
Pollution Table:
Emission rates (g/sec) and Fuel comsumption rates (ml/sec)
for 7 different types of vehicles: DUMMY, CAR, BUS, GBUS, LGV,
HGV
and pollutants: CO, NOX, HC, FUEL
&POL_RATE
CAR
CO
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.06  0.091 0.13  0.129 0.09 0.11 0.177
 0.06
 0.377
 0.072
NOx
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.0006 0.0006 0.0017 0.0022 0.0042 0.0050 0.0058
 0.0008
 0.01
 0.0005
HC
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.0063 0.0078 0.0083 0.0128 0.0097 0.0117 0.0136
 0.0067
 0.02
 0.0067
FUEL
 12
  10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 100.0 110.0
120.0
  0.26 0.29 0.34 0.40 0.48 0.59 0.74 0.92 1.15 1.43 1.77 2.17
  0.333
  0.42 0.26
  0.537
&ENDPOL
BUS
CO
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.06  0.091 0.13  0.129 0.09 0.11 0.177
 0.06
 0.377
 0.072
NOx
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.0006 0.0006 0.0017 0.0022 0.0042 0.0050 0.0058
 0.0008
 0.01
 0.0005
HC
 7
 10.0  20.0  30.0  40.0  50.0  60.0  70.0
 0.0063 0.0078 0.0083 0.0128 0.0097 0.0117 0.0136
 0.0067
 0.02
 0.0067
FUEL
 12
  10.0 20.0 30.0 40.0 50.0 60.0 70.0 80.0 90.0 100.0 110.0
120.0
  0.26 0.29 0.34 0.40 0.48 0.59 0.74 0.92 1.15 1.43 1.77 2.17
  0.333
  0.42 0.26
  0.537
&ENDPOL
&END
```

It is possible to output pollution by individual vehicles. In the current version, only accumulated emission and fuel consumption measures for each link and for the whole network are stored and output at regular time intervals.

DRACULA does not model pollution dispersion. The modelled emission (by link and by time period) can be easily linked to dispersion model packages available such as ADMS.

Tip

Due to the many uncertainties in the model inputs, the pollution model is better suited for comparing strategies than for producing absolute emission levels.

8 ADVANCED SIMULATION APPLICATIONS

8.1 Selective Vehicle Detectors

Detectors are placed on the road as line sources on a lane. If at time (t_0-1) seconds the front bumper of a vehicle is before a detector and at the following second (t_0) its front bumper is at or has just passed the detector (regardless whether the rear-end of the vehicle has or has not passed the detector), then a vehicle detection is triggered. The actual detection time (in 0.1 second) is interpreted from the speed and location of the vehicle at time t. With such a definition, a vehicle stopped on top of a detector will not trigger more than one detection.

The input data consists of:

- detector identification;
- the lane where the detector is placed;
- location of the detector from the stopline.

DRACPREP scans for detector inputs in the *.DET* file automatically. It tries to locate a *.DET* or file name specified via parameter DETFIL in the *.PAR* file.

The detector file contains two parts: (1) detector data and (2) inputs of an alternative signal plan. The second part of the data is linked to a "plan-selection" responsive signal control which is to be explained in Section 8.3.5. There are two sections of records:

1. Section 1 – Detection description. Preceded by a line containing a string "&DETECTOR" and terminated by a line containing "99999". One record per detector.

2. Section 2 – Description of alternative signals. Preceded by "&ALT_SIGNAL" and terminated by "99999". Within this section are two records:

    1) Record type 1 - Signalised intersection description. One record per node.

    2) Record type 2 - Signal plan description. One record per signal plan.

The coding format is:

DATA	NAME	DESCRIPTION
***** SECTION 1 – DETECTOR DATA *****		
1	DETID	Detector number
2	QDEM	A character, to be entered immediately after DETID, indicating the type of responsive signal control to trigger: if QDEM='d' or 'D', to trigger demand-dependent signal controls described in Section 8.2.2; if QDEM='t' or 'T' coupled with a detector with QDEM='r' or 'R', to trigger a plan-selection signal as described in Section 8.2.4. Default: QDEM=BLANK, the detector is not used to trigger any responsive signal control, but maybe used to collect data at the detectors.
3	VTYPE	Types of vehicles to detect
4	QSTAR	A character to be entered immediately after VTYPE to indicate how the detector location is measured. If QSTAR='*', POSIT is measured relative to the stop-line of the link. By default, QSTAR is left blank where the detector location is measure relative to link entry.
5	NODE	Node number of the signalised junction to be controlled
6	ANODE	Upstream node number of the link where the detector is located
7	BNODE	Downstream node number of the link where the detector is located
8	SIDE	Relative position of the lane where the detector is located: 1=kerbside lane
9	POSIT	Location of the detector (meter)
10	RECALL or RDETID	If QDEM='d', maximum recall length (seconds) or if QDEM='t' or 'r', the DETID of the detector to be paired in plan-selection signal control
11	TCOOL	Cooling off period (seconds), if QDEM='d'.
***** SECTION 2 – ALTERNATIVE SIGNAL DATA *****		
***** RECORD 2.1 – NODE DATA *****		
1	NODE	Node number
2	NIN	Number of links at the node
3	JTYPE	Node type (should be 3 for traffic signals)
4	QYELL	a character "y" or "Y" entered immediately after JTYPE – no space in between JTYPE and QYELL, to indicate the junction is in yellow-box control
5	NSTAGE	Number of stages
6	OFFSET	Relative offset
7	LCY	Cycle time for this node (seconds)
***** RECORD 2.2 – SIGNAL DATA *****		
1	STAGL	Duration of stage (seconds)
2	INTG	Duration of following inter-green (seconds)
3	GNA(1)	The A-node for the first turn
4	GNC(1)	The C-node for the first turn (0 if all turns from A-node)
5	GNA(2)	The A-node for the second turn

6 GNC(2) The C-node for the second turn

Example

```
&DETECTOR
  5      63*   1012   1011   1012   1   50
  6d     2     1345   1123   1345   1   150  200
201T     2     727    723    728    1   400  202  10
202R     2     727    731    783    1   200  201  10
99999
&ALT_SIGNAL
 727  3   3   2   0   20
          35  5  731  723  728  0
          15  5  731  723  724  0
99999
```

This input detector file indicates that:

- there are four detectors in the network, identified by number 5,6,201 and 202;
- detector 5 lies 50 meters upstream from the stopline of the kerb-lane in link 1011-1012, and it is for detection of all types of vehicles;
- detector 6 lies 150 metres downstream from the entry of link 1123-1345 and is placed on the kerb-lane of the link. It detects buses only and triggers a demand-dependant signal control at junction 1345 with maximum recall length of 200 seconds.
- Detector 201 and 202 make a pair of co-ordinated detectors for a plan-selection signal control at junction 727; the first triggers and the latter cancels the plan-selection.
- The alternative signal plan for junction 727 is included in the second section of the data file.

8.2 Responsive Signal Controls

There are four types of traffic responsive signal controls modelled in DRACULA: they are all responsive controls to selective vehicles, e.g. they can be responsive only to selected types of vehicles such as buses. Section 9.7 describes a fifth control type: combined pedestrian and traffic responsive signals.

The four traffic control algorithms are:

(a) extension of green signals, in order, for example, to allow a bus pass the stop-line without delay;
(b) early termination of a red signal, in order to reduce bus delays;
(c) demand-dependent signals, which calls an additional signal phase when detection of a bus; and
(d) plan-selection signals, which switch to an alternative (fixed) signal plan when detecting a bus.

As these signal controls are selective vehicle ones, they work hand-in-hand with the data coding of the detectors described in Section 8.1. The relevant control parameters (Table 8.1) in the *.PAR* file.

Table 8.1 Parameters describing responsive signal controls

Parameters	**Defaults**	**Definition**
QBUS	F	T if there are PT services in the network
QPT_PRIORITY	F	T if to carry out extension or early-recall type of priority signal controls
EXTENSION	5	Maximum signal extension permitted (seconds)
RECALL	5	The amount of recall time of a red signal (seconds)
NCOOL_CYCLE	3	Number of cooling-off periods after a signal extension or recall

The pane-selection control is automatically activated when the appropriate detectors are set up.

8.2.1 Journey Time Prediction

Two methods are used to predict the time it takes for a vehicle to reach the stopline from its point of detection.

The first one simply assumes that the detected vehicle will be able to travel down to the stopline at its speed when it is detected. The second method considers possible queueing of the vehicle before it reaching the stopline. The simple queue model described in Clark et al (1996) is used to predict the back of the queue at the time of detection, using detected lane occupancy information.

8.2.2 Signal Extension and Recall

When a bus is detected at time t_0 and predicted to arrive at the stopline at time t_a, one of two actions may be performed:

Extension, which extends the bus green period in order to allow the bus to exit;

Recall, which terminates the bus red stage earlier in order to reduce the bus waiting time.

Figure 8.1 shows schematically the signal priority in a space-time diagram. Three bus trajectories from the detector to the stopline are drawn in dashed lines.

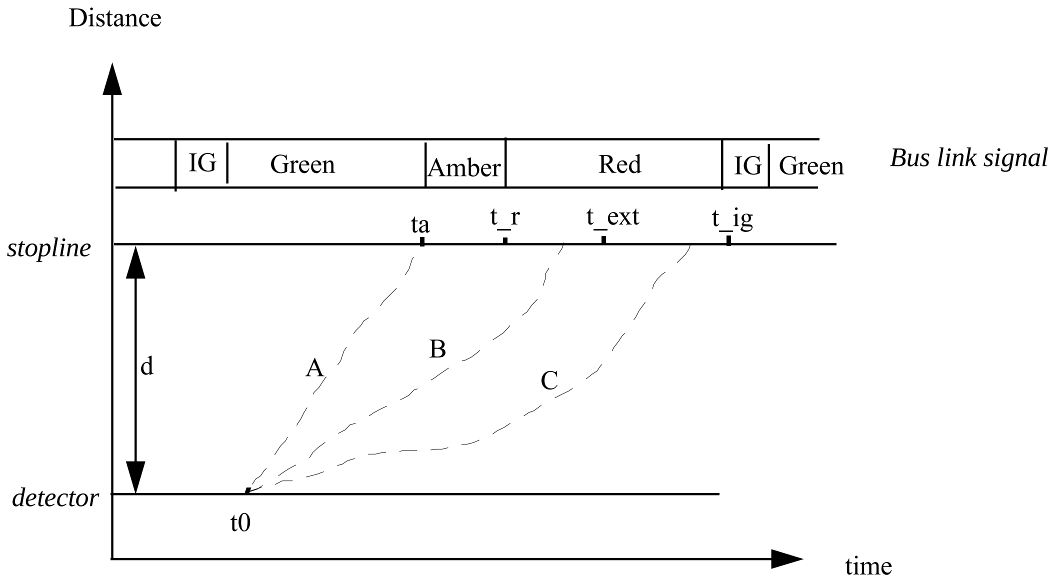


Figure 8.1 Space-time representation of bus signal priority

If a bus is predicted to arrive at the stopline just after the start of the red signal, the bus green stage and the following amber will be extended by just enough time to allow the bus to exit. The amount extended depends on the predicted bus arrival time, subject to a user-defined maximum (parameter EXTENSION) and to a minimum safety green for the subsequent stages affected.

If a bus is predicted to arrive during the red, but an extension is not appropriate, then the duration of the bus red stage may be reduced by a constant amount, defined by parameter RECALL. The stage length of other stages remain un-changed; the length of the current cycle is therefore changed.

Otherwise the signals will not be changed (case A in Figure 8.1).

Only one signal change (extension or recall) is permitted per cyclel and stage extension or recall respond to the first bus detected. In all cases, the constant intergreens must be maintained for safety reasons.

8.2.3 Payback

If an extension to the bus link green stage is triggered in one cycle, the extension time is removed from the bus link green stage in the following cycle, subject to a minimum safety green. This removed time is subsequently given back to the other link stage. In the case of an early recall, the amount of recall time is given back to the bus red stage in the following cycle. This procedure is named *payback*. It is designed to maintain signal offsets and the long-term degree of saturation on all links.

8.2.4 Cooling-Off Period

After an extension or recall in one cycle and a payback in the following cycle, there will be a recovery period during which no stage change is permitted. This period is named the *cooling-off period* and is intended to let the system settle down. The period is modelled as a multiple of the signal cycle, via parameter NCOOLING_CYCLE.

8.2.5 Plan-Selection Signal Control

This signal control strategy was supplied by Stuart Dagleish of York City Council and implemented in DRACULA to help design a quality bus corridor scheme in York.

The control strategy

The traffic signals are varied between two alternative fixed plans in response to detection of a selected type of vehicles, in practice buses. The alternative signal plan is called in at the detection of a bus upstream and cancelled at the detection of the bus downstream. While the signal is still on the alternative plan, further detection will reinforce the use of the alternative signal plan until they have all passed the downstream detector. A cooling-off period is enforced after a round of signal changes to ensure the system to settle down well.

Data inputs in DRACULA

The initial fixed-plan signal is coded in the network description file. The alternative fixed-plan signal is coded in the detector data file.

There need a pair of detectors: one upstream and one downstream of the traffic signal. They are of detector type T and R for the upstream detector and the downstream detector respectively. The detectors and information on the alternative signal plans is required in the data file, in a format similar to that in DRACULA free-format network coding. See Section 8.1 for coding of detectors and alternative signal plans.

8.3 Speed Advice and Intelligent Speed Adaptation

A mandatory speed control system or an advisory speed control system with known compliance rate can be modelled in DRACULA. The model was developed initially to represent the Intelligent Speed Adaptation (ISA) or External Vehicle Speed Control (EVSC) systems (Liu & Tate, 2004).

ISA is a technical device fitted to the vehicles, enabling the speed of the vehicle to be externally regulated. The external activation is achieved by a communication infrastructure in the form of roadside beacons, or an autonomous system using on-board digital maps combined with a Global Positioning System (GPS).

The main advantage of ISA relative to other forms of urban speed control measures (such as 20mph zones or traffic calming measures) is its flexibility. The system allows for different control speeds at different time of day and different location (e.g. outside schools and during school starting and finishing times), and under different traffic, roadway and weather conditions. The systems were designed initially as a speed management measure for the urban environment. However, there is no technical restriction of the systems being applied on motorways, where they can work in similar manner as controls by variable speed signs (such as that implemented on M25). In this latter application, the system aims to improve stability and homogeneity of traffic streams in order to reduce interactions/conflicts among vehicles, which in turn reduce accidents.

There are a number of variants of ISA systems:

1. Mandatory systems which automatically limit a vehicle’s maximum speed to either a prevailing fixed speed limit, or to a speed limit varying with road geometry; and
2. Voluntary systems which only provide speed limit warnings to the driver, such as the system proposed for Borläägn, Sweden (TEC 1998) which alerts the driver if they infringe a speed limit, and registers a violation if the warning is ignored. Driver compliance can be achieved either through the normal use of the vehicle control (the advisory systems) or by driver enacting the change, say by pressing a button to allow the vehicle’s speed be regulated to the speed limit (the driver selection systems).

For modelling purposes, a more useful distinction is the actual number (or percentage) of vehicles which are under or are complied to ISA control, and the speed limits on each section of the road. Later in the section, when reference is made to ISA penetration rate (i.e. the percentage of vehicle fleet equipped with the control devices) of the mandatory systems, it could equally referred to the level of compliance in voluntary systems. The speed limits are inputs to the simulation modelling; there can be more than one speed limit sections on one link which allows modelling a smaller speed limits at road bends, for example. Once specified at the start of a simulation, the speed limits do not change over the simulation period.

To enable speed control to be modelled, enter the relevant link speed limits in the additional network data file, *.ADD*, between “&ISA” and “99999” delimiters. The speed control may not apply to all parts of the network; so one needs only to enter the links that are under the speed control. If a link is not specified, it will be a link that is not under the speed control and is flat (has zero gradient).

There could be more than one speed limit specified for one link. The format of the inputs are:

DATA	NAME	DESCRIPTION
1	ANODE	Upstream node number of the link
2	BNODE	Downstream node number of the link
3	NSPEED	Number of speed-limit sections on link
4	START(1)	Start position of the first speed-limit section (metre) relative to link entry
5	END(1)	End position of the first speed-limit section (metre) relative to link entry
6	SPEED(1)	Speed limit (kph) for the section
	etc	etc for each speed-limit section
7	START(n)	Start position of the last speed-limit section (metre) relative to link entry, n=NSPEED
8	END(n)	End position of the last speed-limit section (metre) relative to link entry, n=NSPEED
9	SPEED(n)	Speed limit (kph) for the section

Use control parameter, PISA, in **.PAR** to specify the ISA system penetration rate.

Parameters	Defaults	Definition
PISA	0	Percentage vehicles fleet under ISA control. If PISA=10, then 10% penetration.

8.4 Modelling Dynamic Road Pricing Systems

Two dynamic road pricing systems have been implemented in DRACULA; they are: time-based pricing and the *Congestion Metering* system – the latter was first proposed for Cambridge by Oldridge (1990) and tested in an EU project ADEPT (Clark et al., 1993).

The underlying concept of the Congestion Metering system is that charges would only be levied when delays occur. This is possible by the setting of a *congestion threshold*, in the form of a critical time value for □ntersecti a certain unit of distance, above which a charge would be levied. The suggested threshold in Cambridge was three minutes to travel half a kilometre, equivalent to an average speed of 10kph (Oldridge, 1990). When the threshold is exceeded, the charge levied may be a pre-specified fixed unit (a figure of 20 Pence Sterling was mentioned in Cambridge) or a variable unit that depends on the degree of threshold violation (e.g. a charge in Pence Sterling per minute for all time taken above the threshold). Although calculation of the threshold is continuous (i.e. carried out at much more frequent intervals than the half □ntersect distance unit), charges cannot overlap, so that a fee will only be levied once for using any particular half-kilometre stretch of congested road.

Previous modelling work to investigate the route and demand choice aspects of a range of charging technologies in a static modelling context has been forced to rely on a much coarser specification of the Congestion Metering system (May and Milne, 2000). A faithful representation can only be achieved using a fully microscopic simulation model, where costs may be applied separately to individual vehicles rather than to aggregate flows. In particular, a feature of the threshold system is that the points in space at which charges are levied may not coincide with those at which major delays occur. This raises the question of how to model drivers' reactions to charges in terms of their perception of network travel conditions. For this reason, only a route-based, individual vehicle simulation model may fully represent the impacts of the most technologically advanced road user charging systems.

At the time of writing, an updated version is being made. Potential users may wish to contact the author for a test version.

8.5 Modelling Overtaking on Two-Lane Rural Roads

The links are classified into two types:

- links with double-white lines where vehicles can not move over to the other side of the road; or
- links without double-white lines where overtaking from the other side of the road is permitted.

By default, all links are the first type unless specified otherwise. A network can have a mixture of both types of links.

This section describes the logics behind DRACULA models of overtaking on rural two-lane roads without double-white lines. The model is largely based on the earlier work conducted in Australia and implemented in model TRARR.

The model first examines drivers' overtaking desires. If a vehicle falls within a "catching up region" – defined by a distance headway threshold (DRUB) and relative speed threshold (VRUB), its speed is severely constrained by the (slow moving) vehicle in front and anticipates to catch up within time (TRUB), a desire for overtaking will be triggered. The driver will then look for overtaking opportunities. If there is enough space (NLENGTH) in front of the vehicle for merging back after the overtaking, and the gap in the opposing traffic is sufficiently large (OGAP) and it is within a visibility range (SRUB), the driver will begin the overtaking manoeuvre at a speed faster and acceleration higher than his/her normal speed and acceleration by a factor of FVMAX and FAMAX respectively in order to complete the overtaking quickly.

The parameters specifying the overtaking behaviour are listed in Table 8.2; they can be set in *.PAR*. Some of these default values are from TRARR model (those with *), others are simply guess work which need sensitivity tests.

To specify the links without double-white line markings, enter the relevant links in the additional network data file,**.ADD**, between "&RURAL" and "99999" delimiters. If a link is not specified, by default, it prohibits overtaking using the other side of the road.

DATA	NAME	DESCRIPTION
1	ANODE:	Upstream node number of the link
2	BNODE:	Downstream node number of the link
3	VTYPE:	Type(s) of vehicles permitted for overtaking

The following example suggests that overtaking by all types of vehicles is permitted from link 101-109, whilst only CAR type is permitted to overtake from link 109-101.

Example

```
&RURAL
  101   109  63
  109   101   1
99999
```

Note that the permit for overtaking (road with or without double-white lines) is specified for links. Hence, to make overtaking from both sides of a road possible, one needs to code both links in.

*Table 8.2 Parameters specifying rural overtaking behaviour. The default values marked * are from the TRARR model.*

Parameter	Default	Note
DRUB	150*	Threshold distance (metre)
VRUB	5	Speed difference (kph)
TRUB	4*	Time threshold (seconds)
OGAP	10	Critical gap in opposing traffic (seconds)
F2RUB	0.8	Following to desired speed factor
FVRUB	1.15* for Cars 1.2* for HGVs	Max. speed to desired speed factor
FAMAX	1.5	Max. acceleration to normal acceleration factor
NLENGTH	5	No of vehicle length
SRUB	450*	Sight distance (metre)

Outputs of the overtaking movements and the total number of overtakings are summarised in file .TXS.

8.6 Pedestrian Simulation and Responsive Signal Controls

This section describes a methodology developed to simulate individual pedestrians' movements around a signalised intersection and their interactions with vehicular traffic. The pedestrian walkway networks are represented by pedestrian nodes and links. Pedestrians flows and paths are pre-determined and inputs to the model; pedestrians move along pedestrian links following fixed routes. Different pedestrian characteristics are modelled: law-obeying pedestrians who only cross the road when the "green man" is on, and two types of opportunistic pedestrians who may cross the road during red pedestrian signal if appropriate gaps available (the difference between the two depends on their patience).

A number of pedestrian actuated signal policies are implemented. Pedestrian stage length may vary with number of waiting pedestrians. A pedestrian stage may be extended for late crossing, or vice verse, be terminated early if all pedestrians have crossed the road. Similarly a pedestrian stage may be brought forward in excess demand, or vice verse, be cancelled in absence of pedestrian demand.

The model is implemented within **DRACSIM**. During the implementation of pedestrian simulation, a vehicle actuated signal policy is also represented where a vehicle green stage may be extended at the detection of approaching vehicle(s) by downstream detectors. Interactions between this vehicle-actuated and the pedestrian-actuated signal policies are dealt with in the model.

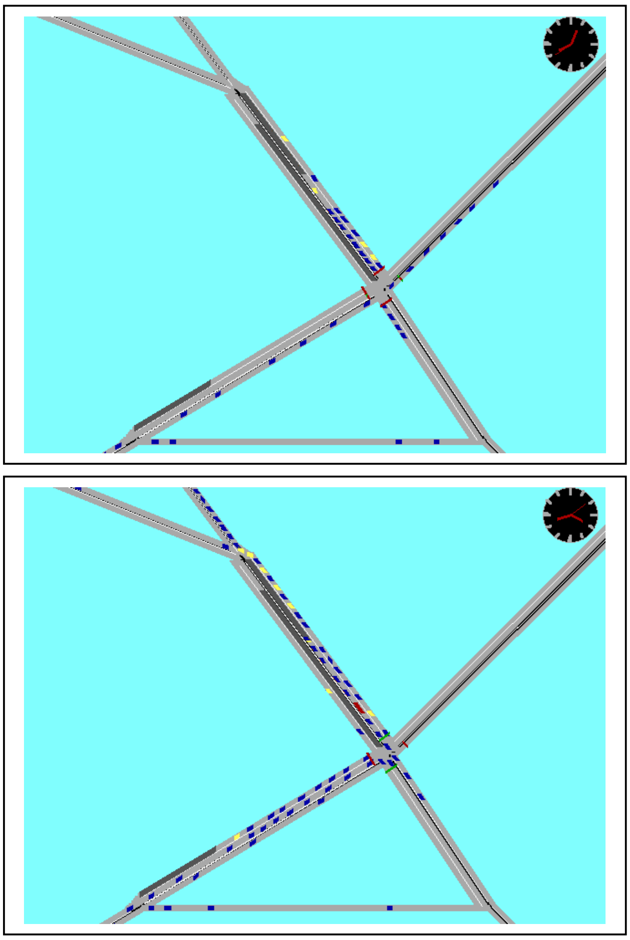
Further tests are to be conducted before its release. Prospect users may want to contact the author for a test version.

9 MODEL OUTPUTS AND EVALUATION

9.1 Summary of Output Files

The default outputs from a DRACULA simulation are the link, route and network based traffic condition measures and pollution measures at a regular time period and at the end of simulation. The output time period is user-specified, and can be as small as one second. The DRACULA simulation ends when all vehicles which entered the network within the "peak period" (Section 6.1.1) have completed their origin-destination movements.

The most detailed records, at the user's request, are the second-by-second individual vehicles' locations and speeds. The model also provide point- or loop-based detector measures on headway distribution, flow, occupancy and speed. For each bus service, the model summarizes the mean and standard deviation of total journey time and journey times in between stops, a measure which can help distinguish service delay due to traffic congestion from that due to poor management.



(a)

(b)

Figure 9.1 Simulated traffic conditions at Clifton Green intersection in the City of York. Two snapshots were taken at time 08:04 (a) and 08:20 (b) as shown by the clock on each snapshot. Vehicles are shown as coloured rectangles. Part of the network was blocked due to roadworks, which are shown in dark grey.

A graphical animation of the vehicles' movements can also be shown in parallel with the simulation, giving the user a direct view of the traffic condition on the network. Two snapshots from DRACULA simulation of an intersection are shown in Figure 9.1, which clearly shows the dynamics in traffic conditions.

Depending on application, DRACULA simulation produces a number of output files. These are all ASCII text files, the data format of the outputs are explained at the top of each file, and/or of each section.

- <net>**.OUT** – contains statistical measures on traffic congestion, defined by a "performance" and a "supply" measure of congestion, for link, route and network at regular time periods

- <net>**.SPA** – contains link-based performance measures (same as in <net>.**OUT**) at regular time periods, in a format to be used by the program **SPATULA** (Section 9.2) for SATURN.

- <net>**.REB** - contains statistics collected at link and network exits on queue length, flow, travel time and travel speed and variability of these measures; Also contained are speed distribution profiles.

- <net>**.POL** - Statistics on pollutant emission and fuel consumption measures for each link and for the whole network at regular time periods;

- <net>**.LTT** - Reports on individual vehicles route, departure-time, arrival time, and link-by-link travel time.

- <net>**.PTT** - Same as <netname>.ltt, but for buses only. They are also reported in <netname>.ltt.

- <net>**.PSN** - Records of individual buses' dwell time and passenger delay at bus stops;

- <net>**.TXD** - Individual vehicles' speed and time when passing detectors.

- <net>**.TXS** - General text reports. Also reports on each "rural" overtaking events – overtaking made from the opposite road space.

- <net>**.TRJ** – when specified, the program can also output second-by-second trajectories for vehicles from certain OD pairs, following certain routes, or randomly selected from all OD pairs.

9.2 The SPATULA program

SPATULA is a program that converts link-based simulation results from DRACULA to a format readable by the SATURN graphical display function **P1X**.

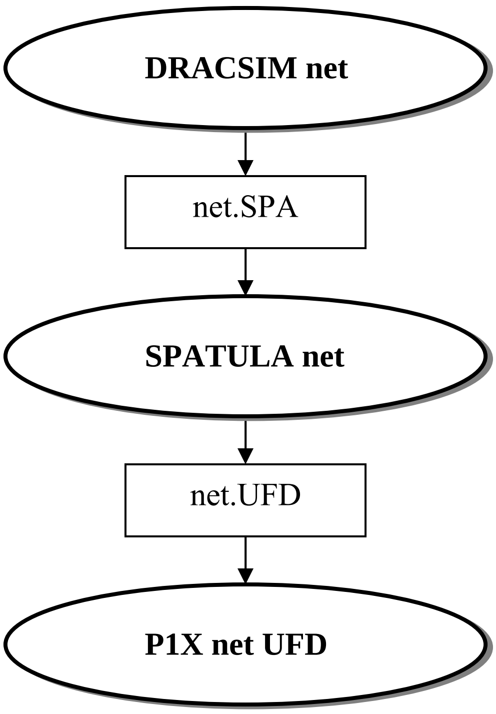
At the end of a simulation, **DRACSIM** will produce link-based performance measures by time period into a text file <net>**.SPA**. This file is only produced if a simulation ends "naturally", e.g. it is ended when all vehicles generated in the main demand period (Section 6.1.1) have completed their journey. This is because the process involved in producing **.SPA** output is quite time-consuming. Experience suggests that a number of iteration between parameter setting, network coding and simulation animation is often required before final numerical outputs are required for analysis.

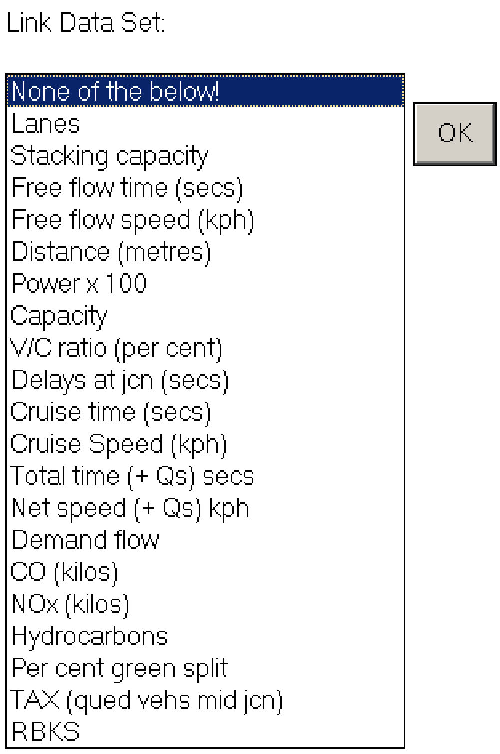
To run **SPATULA**, one first needs to run the network through SATNET to produce the SATURN-format .UFN file.

The input and output files for program **SPATULA** are:

network.**UFN** - Output from SATNET
network.**TRP** - Input DRACULA-format routes and route flows
network.**SPA** - Input DRACULA simulation results
network.**UFD** - Outputs in SATURN-format

The diagram below on the left depicts the processes and input/output files used.





In **P1X**, use Display Option/Choice of link annotation/By List, you will get to a window "Link Data Set" shown below on the right. Among those listed, the measures between "Delay at jcn(secs)" and "Hydrocarbons" are results from the DRACULA simulation. Others are SATURN inputs/outputs.

9.3 Generalised Cost

DRACULA generalised cost (C) is defined as:

$$C = PPM \times T + PPK \times D \quad (9.1)$$

where PPM is pence per minute
PPK pence per km
T total vehicle-minutes
D total vehicle-km

9.4 Performance vs Supply Measures

DRACULA makes clear distinction between the performance of a network and costs associated with a given demand (the supply costs), and produce summary statistics for each type of measures. The performance of a network or a single link can be measured in terms of flow performed and time performed in a defined period. They are engineering description of the performance of the link or network at a given point in time or over a given time period, and can be used to estimate the link or network equivalent of speed-flow relationships.

The supply costs reflect the costs experienced by a driver using the network at a given level of demand; and they can be used to describe the way in which costs of using net network rise as demand-levels increases. Since any journey through a network will pass through a number of different traffic states and the costs incurred will be affected by both the journey length and the route taken, as well as by the impacts of other demands on the network both at that time and in earlier time periods. In order to measure these costs, individual vehicles need to be "tracked" through the network. Thus the space-time domains used to measure performance curves and supply curves are different, as shown in Figure 9.2, and supply curves cannot be readily observed in the way that performance curves can.

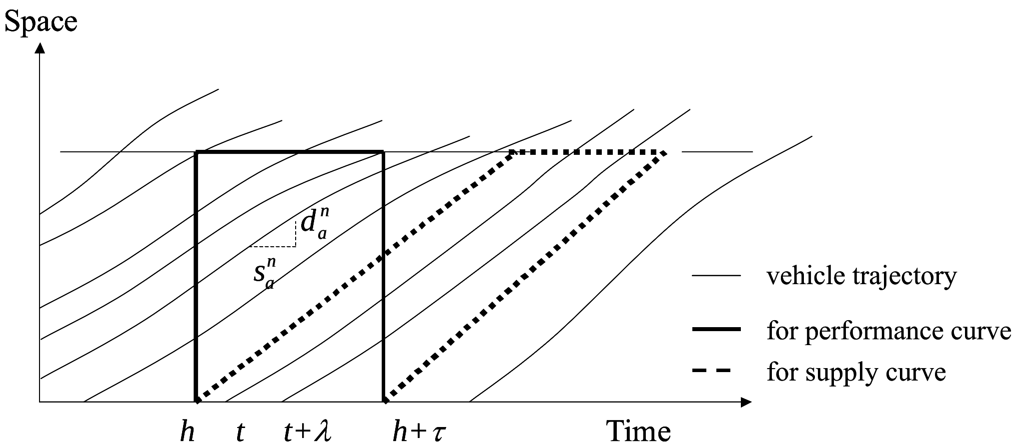


Figure 9.2 Simulation processing speed versus traffic density.

It can be seen that if all vehicle trajectories were parallel (representing a un-congested, homogeneous traffic flow), performance measures are equivalent to supply measures.

Performance curves relate the parameters of the traffic in a network at a given time. They are based on measurements of vehicle-km and vehicle–hr in a network in a given time period, and can be used to estimate network equivalents of speed-flow curves. These parameters can be measured, and have been used by earlier authors to describe the way in which costs of using the network rise as usage increases.

Supply curves reflect the costs experienced by a driver at a given level of demand, which will be affected by the journey length and the route taken, as well as by the impacts of other demands on the network both at the same time and in earlier time periods. Supply curves cannot be readily observed in the way that performance curves can.

The space-time domains used to measure performance curves and supply curves are different, as shown in Figure 9.2**.** The supply curves were generated through a vehicle-tracking approach (May et al, 2000), whereby individual vehicles' journey time and distance from origin to destination were used to give a generalised costs. The resulting generalised costs were then summarised by vehicles' departure-times to give supply costs for the time-period of interest. The supply curves are then represented as generalised costs per trip vs. the trips demanded over a given (departure-) time period.

The performance measures were generated through a time-slice approach, whereby vehicle trajectories (time and distance) within a specific time-period were used to get the generalised cost for the period. There was no distinction as to the departure-time of individual vehicles.

The program outputs, in file *.OUT*, for each link and for the whole network, at the end of current time period:

Performance Measures:

- Queuing delay (veh.hrs)
- Cruising time (veh.hrs)
- Total travel time (veh.hrs)
- Total travel distance (veh.kms)
- Average speed (km/hr)
- Performed flow (veh.km/hr)
- Average flow (veh/hr)
- Fuel consumption (litre)
- Total CO emission (Kg)
- Total NOx emission (Kg)
- Total HC emission (Kg)

Supply Measures:

- Total trips demanded (veh)
- Total travel time (veh.hrs)
- Total travel distance (veh.kms)
- Average flow (veh/hr)
- Average speed (km/hr)
- Average cost (pence)

9.5 Exit Measures

These are data collected when vehicles exit a link or the network. They do not include vehicles still travelling on the link or in the network. When a vehicle exits a link or the network, its travel time and distance over the link and throughout the journey are recorded. Travel time is averaged over the number of vehicles recorded. An average speed is derived from the average travel time and distance. The measures are disaggregated by vehicle types. The program outputs for each link and for each vehicle type, at the end of the current time period:

- Link free flow travel time (second)
- Number of vehicles exiting the link during the current time interval
- Average travel time in the current time interval (second)
- Standard deviation of the travel time (second)
- Maximum travel time in the current time interval (second)
- Average link travel speed (kph) in the current time period
- Number of vehicle exiting the link accumulated from the start of simulation
- Accumulated average travel time in the current time interval (min)
- Standard deviation of the accumulated travel time (min)
- Accumulated maximum travel time (min)
- Average link travel speed (kph)

If we divide "the number of vehicles exiting the link during the current time period" by the time period, we get a "**throughput**" for the link.

9.6 Pollution Measures

Pollution measures include measures of emission for pollutant CO, NOx, and HC, and measures of fuel consumption. Unlike travel condition measures which are measured only when vehicles exit a link or exit the network, pollution measures are recorded at every second for each vehicle in the network. The program outputs for each link and for the whole network time averages of the pollutant emission and fuel consumption for the current measuring time period (say 5 min interval) and for the whole simulation time period, in the following format and for each link and for the whole network:

At the end of current time period:

- CO emission (in kg) for the current output time interval
- NOx emission (in kg) for the current output time interval
- HC emission (in kg) for the current output time interval
- Fuel consumption (in litres) for the current output time interval
- CO emission (in kg) accumulated from the start of simulation
- Accumulated NOx emission (in kg)
- Accumulated HC emission (in kg)
- Accumulated fuel consumption (in litres)

9.7 Definition of Queue Length

The queue length of a lane is the distance from the stopline upstream to the end-of-queue (in meters). The queue length of a link is the longest queue length of all the lanes in the link. The end-of-queue of a lane is defined as the back of the first stationary vehicle from upstream (start) of the lane. Stationary is defined as speed less than 0.5m/s.

The queue-length for each link is recorded at every second of the simulation. At the end of each measuring time period, the program outputs the time averages of link queue-length, variance and the maximum queue length during the measuring time period in the following format:

- Sample time period (in sec) incremented by sec
- Time average queue length (in metre) in the current time interval
- Standard deviation of the queue length
- Maximum queue length in the current time interval
- Sample time period (in sec) from the start of simulation
- Accumulated average queue length in the current time interval
- Standard deviation of the accumulated queue length
- Accumulated maximum queue length (sec)

9.8 Multiple Simulation Runs

DRACULA is a stochastic simulation model; there are many random processes used/occurring during the simulation (e.g. vehicles' random parameters and drivers' random behaviour). Given the same network and demand inputs and the same simulation control options, the results from two or more runs with different random number seeds may differ. To get some confidence in the simulation predictions, it is a good practice to run the simulation several times with different random seeds NSEED or NSEED2 and to look at the averages and variances of any result from the number of simulations. At the moment, there is no automatic procedure within the supply model to help the user to do so. The users need to copy the output files from each simulation run either to a different set of file names or to a separate directory. All **DRACSIM** output files are in ASCII text format which can be loaded into standard database, such as **EXCEL,** from which statistical measurements such as averages and standard deviations can be calculated.

The number of simulation runs depends on the level of accuracy required. It should consider the level of variance introduced to the model (such as variance used in describing the distribution of vehicle characteristics) when considering the level of accuracy required. Typically, 10 simulation runs have been found to be acceptable.

10 SPECIAL OPTIONS AND FEATURES

10.1 GIS Data Files

To enhance background presentation of the study area, it is possible to include in the network plots geographical features such as rivers, railway lines, churches, and parks. The data required to model such features are termed GIS data, in line with the terminology used in SATURN.

The GIS features currently available are: enclosed polygons, polygon lines, icons (church, BR station, etc), text, and curved links. The data, to be stored in text file *.GIS*, follows exactly the same format to that in SATURN (see SATURN Manual, Appendix Z). In fact, the SATURN users can use **P1X** to generate the GIS features.

In versions prior to Version 2.1 released in October 2003, the SATURN-format GIS files have to be modified slightly before DRACSIM can load them. For example, the corners of a polygon, poly line or a curved link have to be given specifically by the user for DRACSIM. The newer versions do not require such information. The newer versions can take any GIS file created for earlier versions of DRACSIM programs.

The following set out the format for specifying GIS features. The format is the same as those used for SATURN. Those in *italic* are not used in DRACSIM.

10.1.1 Enclosed Polygons

Record 1.1 – Enclosed Polygons
Cols 1 – 5 11111

Record 1.2.1 – Start of a new polygon

Cols 1-10	X co-ordinate of the first corner
Cols 11-20	Y co-ordinate of the first corner
Cols 21-25	Pen colour (in range 1 to 16)
Cols 26-30	A non zero if the area is to be filled
Col 31-35	*Name of zone*
Col 39-50	An alphanumeric title for the area

Record 1.2.2 – Co-ordinates of the polygon's next (up to) 4 "corners":

Cols 1-10	X co-ordinate of the next+1 corner
Cols 11-20	Y co-ordinate of the next+1 corner
Cols 21-30	X co-ordinate of the next+2 corner
Cols 31-40	Y co-ordinate of the next+2 corner
etc.	

Record 1.3 – End of polygons
Cols 1 – 5 99999

10.1.2 Polylines

Record 2.1 – Polylines
Cols 1 – 5 22222

Record 2.2.1 – Start of a new polyline
Cols 1-10 X co-ordinate of the first corner
Cols 11-20 Y co-ordinate of the first corner
Cols 21-25 Pen colour (in range 1 to 16)
Cols 26-30 Line width in millimetres “on the screen”

Record 2.2.2 – Co-ordinates of the polygon’s next (up to) 4 “corners”:
Cols 1-10 X co-ordinate of the next+1 corner
Cols 11-20 Y co-ordinate of the next+1 corner
Cols 21-30 X co-ordinate of the next+2 corner
Cols 31-40 Y co-ordinate of the next+2 corner
etc.

Record 2.3 – End of polylines
Cols 1 – 5 99999

10.1.3 Icons

Record 3.1 – Icons
Cols 1 – 5 33333

Record 3.2.1 – Start of a new icon
Cols 1-10 X co-ordinate
Cols 11-20 Y co-ordinate
Cols 21-25 Pen colour (in range 1 to 16)
Cols 26-30 Icon height in millimetres “on the screen”
Cols 31-35 Icon number representing:
1 - for a monopoly-style house
2 - for a BR symbol
3 - for a car park
4 - for a church
5 - for a hospital

Record 3.2.2 – as 3.2.1 for the next icon
etc.

Record 3.3 – End of icons
Cols 1 – 5 99999

10.1.4 Text

Record 4.1 – Text
Cols 1 – 5 44444

Record 4.2.1 – Start of a new text
Cols 1-10 X co-ordinate of the centre of text
Cols 11-20 Y co-ordinate of the centre of text

Cols 21-25 Pen colour (in range 1 to 16)
Cols 26-30 Character height in millimetres "on the screen"
Cols 33-60 Text

Record 4.2.2 – as 3.2.1 for the next text
etc.

Record 4.3 – End of text
Cols 1 – 5 99999

10.1.5 Co-ordinates of Curved Links

Record 7.1 – Curved links
Cols 1 – 5 77777

Record 7.2.1 – Link Identification
Cols 1-10 Upstream node of link
Cols 11-20 Downstream node of link

Record 7.2.2 – Co-ordinates of the intermediate points
Cols 1-10 X co-ordinate of the next+1 corner
Cols 11-20 Y co-ordinate of the next+1 corner
Cols 21-30 X co-ordinate of the next+2 corner
Cols 31-40 Y co-ordinate of the next+2 corner
etc.

Record 7.3 – End of curved links
Cols 1 – 5 99999

10.2 Bitmap Background

Another way to enhance presentation of the study area is to display a background map of the area, such as an OS map or an aerial photograph, underneath the network plot. Currently within **DRACULA** such background map must be of ".bmp" format, as opposed to, e.g., .pcx, .gif, etc. formats. The other graphical formats may be converted into a .bmp format by making use of standard software such as Paint.

In order for **DRACSIM** to draw a bitmap background within the windowed area covered by a network plot it is necessary to know (a) the area covered by the full network and (b) the full area covered by the .bmp file. The full network area can be worked out by the programs from the coordinates of the nodes coded in the network data file. The area covered by the bitmap can be defined by the coordinates of its 4 corners.

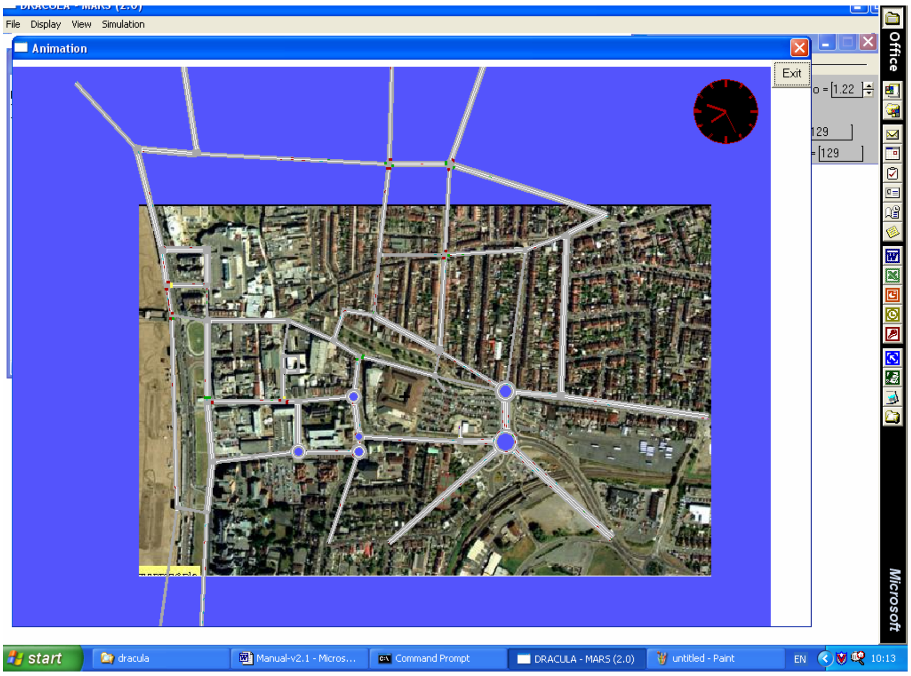
Thus for every .bmp file used by **DRACSIM**, say picture.bmp, it is necessary to set up a further (very small) file, named picture.xyb, which specifies the 4 corners of "picture" using the same coordinate system as that used by the network. The .xyb files consist of a single record (on the same line) containing 4 (real) values in the following order:

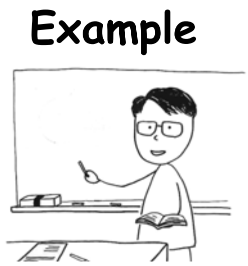
(a) XMIN - the east-west co-ordinate of the lower left-hand corner;
(b) XMAX - ditto for the upper right corner;
(c) YMIN - the north-south co-ordinate of the lower left-hand corner;
(d) YMAX - ditto for the upper right corner.

The .xyb file may be most conveniently set up the user assuming that the information is known in advance through knowing the source of the image.

Alternatively, if a bitmap is input into **DRACSIM** without a corresponding .xyb file being located, the full bitmap is drawn with its (XMIN, YMIN) corner to the bottom-left-hand corner of the screen.

The example below shows an aerial photograph drawn underneath a network plot. In this example, the network window displayed is larger than the area covered by the bitmap. Had it be the other way round, then the appropriate region of the bitmap would have been selected and suitably expanded. Thus a very useful property of the bitmap displays is that they “move” with the network window.





REFERENCES

Bonsall, P., Liu, R. and Young, W. (2005), Modelling safety-related driving behaviour – impact of parameter values. Transportation Research, 39A, 425-444.

Clark, D.J., Blythe, P.T., Thorpe, N. and Bourke, A. (1993) The ADEPT Project: 3. Congestion metering: the Cambridge trail. *Traffic Engineering & Control,* April 1993.

Clark, S.D, May, A.D & Montgomery, F.O (1996). Priotity management for urban artetials, transferability of techniques, implementation in NEMIS, *ITS Technical Note 387*, Institute for Transport Studies, University of Leeds.

Department of Transport, 1991, New car fuel consumption: the official figures December 1991, UK Dtp.

Ferreira, L.J.A. 1982, Car fuel consumption in urban traffic, the results of a survey in Leeds using instrumented vehicles. ITS Working Paper 162, Institute for Transport Studies, University of Leeds.

Liu R (2000) Simulation of plan-selection traffic signal control. Report prepared for York City Council.

Liu, R. (2005), The DRACULA dynamic network microsimulation model. In: Kitamura, R. Kuwahara, M. (Eds), Simulation Approaches in Transportation Analysis: Recent Advances and Challenges, Springer, pp23-56. ISBN0-387-24108-6.

Liu, R., Van Vliet, D. and Watling, D.P.(1995) DRACULA: Dynamic route assignment combining user learning and microsimulation, PTRC, Vol E, 1995.

Liu, R., van Vliet, D. and Watling, D. (2005) Microsimulation models incorporating both demand and supply dynamics. Transportation Research, 40A, 125-150

Liu, R., Clark S.D., Montgomery F.O. and Watling D.P. (1999a) Microscopic modelling of traffic management measures for guided bus operation. In *Selected Proceedings of 8th World Conference on Transport Research,* Vol 2 (eds. Meersman, H, Van de Voore E. & Winkelmans W), Elsevier,1999.

Liu, R. and Tate, J.E. (2004) Network effects of intelligent speed adaptation systems. *Transportation*. **31(3)**, 297-325.

May, A.D. and Milne, D.S. (2000) Effects of alternative road pricing systems on network performance. *Transportation Research A,* 34, 407-436.

Mauro, V. (1991) Evaluation of dynamic network control: simulation results using NEMIS urban microsimulator, Transportation Research Board 70th annual meeting, Washington DC.

McDonald, M., Hounsell, N.B & Kimber, R.M., Geometric delay at non-signalised □ntersections, *TRRL, SR810*, 1984.

Milne, D.S. (1997) Modelling the effects of urban road user charging. PhD thesis, Institute for Transport Studies, University of Leeds.

Oldridge, B. (1990) Electronic road pricing: an answer to traffic congestion? *Proc. Information Technology and Traffic Management.* HMSO (London).

QUARTET Deliverable No. 2, 1992, Assessment of current tools for environment assessment in QUARTET, DRIVE II Project V2018: QUARTET, September 1992.

Watling, D.P., DRACULA 1.0: user guide to the day-to-day model, ITS Technical Note 369, Institute for Transport Studies, University of Leeds, 1995.

Van Vliet, D. (1982) SATURN – a modern assignment model, *Traffic Engineering and Control*, 23(12), 578-581.

Van Vliet, D. And Hall, M. (2002) *SATURN 10.2 – User Manual.* Institute for Transport Studies, University of Leeds and W.S. Atkins. January 2002.

APPENDIX A : Control Parameters for Data Pre-Processing (DRACPREP)

Parameter	Defaults	Definition
CHRGBAND	500	Charge band (in meters) in the congestion pricing scheme. Coded in as integer value. Section 8.5
CHRGSEG	10	Segment (in metres) in the pricing scheme. Coded in as integer value. Section 8.5
CIRC_RADIUS	4	Minimum radius of roundabout central island (metres). Section 4.2.2.3.
CIRC_SPEED	27	Roundabout circulating speed (km/hr). See Section 4.2.2.3
FIJPMIN	0.0	Minimum flow (veh/hr) below which a route is not selected for DRACULA simulation
GAP	3.5	See definition in Appendix B
GAP_MIN	1.0	..
GAP_TSTART	60.0	..
GAP_TEND	180.0	..
GONZO	1.0	Demand factor for the whole trip matrix and all time periods Section 4.3.1.2
GONZO(t)	1.0	Matrix factor for time period t Section 4.3.1.3
LEFTDR	T	F if driving on the right (e.g. European driving) Section 4.1
MUC(v)	1	Vehicle type of user class v Section 4.3.3.2
NMUC	1	Number of user classes. Section 4.3.2
NSEED	800	Random number seed Section 4.
NSTEP	1	Number of time periods Sections 4.3.1.2, 4.3.1.3, 4.3.3.3
VCPCU	1.0	Per car unit used to convert flow given in pcu or pcu/hr to number of vehicles or vehicles/hr. Default: 1 pcu = 1 vehicle of type CAR. See Sections 4.3.3.1.
VCPCU(v)	1.0	PCU value per vehicle for user class v Section 4.3.3.1
PED_COV1	0	Coefficient of variance of the start time of a pedestrian route Section 8.7

PED_COV2	0	Coefficient of variance of the duration of a pedestrian route Section 8.7
PED_COV3	0	Coefficient of variance of the pedestrian flow Section 8.7
QBUS	F	T if to load bus service, bus stops and bus lanes from **.BUS** file Section 5.2.1
QCROWFLY	F	T if to use crow-fly distance as link length. Section 4.2.2.2.
QDETECTOR	F	T if to load detector data from .det file. Section 8.2
QEXPERT	F	If T the level of print out in .TXP file generally is such that only an "expert" would fully appreciate it.
QPASSQ	F	T if to load SATURN multiple time period assignment Section 4.3.3.3
QRBLANE	T	If T the number of lanes on a roundabout is limited to at least equal to the maximum lanes on entry arms. F if the circulating lanes are determined from SATURN roundabout capacity. Section 4.2.2.3.
QSATNET	T	T if to take SATURN-format simulation network; F to use the free-format network coding as described in Section 4.2.2.
QSATPIG	F	F if to take routes and route flows from input file **.TRP** of format; else if T, take them from **.LPG** file Sections 4.3.1, 4.3.3
QVEHPOOL	F	Set T to enable **DRACPRPEP** to generate individual vehicles list in **.VEH** file. Section 4.3.2
TRPNAM(t)		Name of input .TRP file for time period t Section 2.2.3
TSTEP(t)	1	Length (min) of time period t Sections 4.3.1.2, 4.3.3.3
XYUNIT	1.0	Factor converting one unit of node/zone coordinate to metres. If XYUNIT=10, then 1 unit of coordinate=10m. Same as used in SATURN, so can also be specified in the network description file **.NET** Section 4.2.1

APPENDIX B: List of Simulation Control Parameters (DRACSIM)

Parameters	Defaults	Definition
AMBER_PERIOD	3	Amber time period (second).
ARMCOLOUR or COL_ARM	4	Colour of displayed clock arms. Default colour: RED Section 3.7.2.2
COL_FACE	0	Ditto for the clock face. Default colour: BLACK
BKCOLOUR or COL_BKGROUND	9	Ditto for background display Default colour: LIGHTBLUE
COL_ROAD	7	Ditto for links Default colour: GREY
COL_LANE	15	Ditto for lane markings Default colour: WHITE
COL_FLARE	8	Ditto for flared approach Default colour: DARKGREY
COL_BUSLANE	11	Ditto for bus lane Default colour: LIGHTCYAN
CHRGRATE	1.0	Charge rate (pence/min) Section 8.5
CHRGTC	3.0	Charge threshold (min) Section 8.5
CHRGSEG	10	Charge segment (metre) Section 8.5
COL_DUMMY	0	The display colour of dummy vehicles – used to mark a blocked lane. Default black colour. The 16-bit colour palette is used; the numerical coding of the colour is defined as follows: 0 = BLACK, 1 = BLUE, 2=GREEN, 3=CYAN, 4=RED, 5=MAGENTA, 6=BORWN, 7=GREY, 8=DARKGREY, 9=LIGHTBLUE, 10=LIGHTGREEN, 11=LIGHTCRYAN, 12=LIGHTRED, 13=LIGHTMAGENTA, 14=YELLOW, 15=WHITE
COL_CAR	3	Ditto for cars
COL_BUS	9	Ditto for buses
COL_GBUS	9	Ditto for guided buses
COL_TAXI	0	Ditto for taxi
COL_LGV	5	Ditto for light goods vehicles

COL_HGV	4	Ditto for heavy goods vehicles
EXTENSION	5	Maximum signal extension permitted (sec) Section 8.4
F1_OVERTAKE	2.0	Speed improvement factor Section 6.1.7
F2_OVERTAKE	0.8	Speed constraint factor Section 6.1.7
FCOOL	0.5	Start level of traffic flow for the cooling-off period, as a proportion to the peak demand Section 6.1.1
FGW_BUS	0.8	fraction of traffic willing to give way to buses Section 6.1.6
FGW_CAR	0.5	Courtesy factor, fraction of traffic who are willing to give way to minor-flow cars or cars wanting to change lane Section 6.1.6
FIJPMIN	0.0	Absolute minimum flow (veh/hr) en-route. Section 4.3.3.4.
FWARM	0.5	Start level of traffic flow during warm-up period, as a proportion of the peak demand Section 6.1.1
GAP	3.0	Average acceptable gap (seconds) Section 6.1.6
GAP_MIN	1.0	Minimum acceptable gap (seconds) Section 6.1.6
GAP_TSTART	60	Waiting time (sec) before reducing acceptable gap Section 6.1.6
GAP_TEND	180	Waiting time (sec) before taking on MIN_GAP Section 6.1.6
GONZO	1.0	Flow factor Section 6.1.4
IFGIS(n)	F	Parameters as used in SATURN to control the display of GIS features. n=1, 2, …, 7 is for GIS polygon, polyline, icons, text , name, curved link and node coordinate respectively. Currently SATURN GIS names and node coordinates are not represented in DRACULA. Option IFGIS(5) is always set to True, e.g. the display of curved links are always ON if they are given in .GIS file.
LEFTDR	T	F if driving on the right (e.g. European driving)

		Section 4.1
NCOOL_CYCLE	3	Number of cooling-off periods after a signal extension or recall Section 8.4
NSEED	800	Random number seed for all random events other than the generation of random arrival headways (if QARRIV=T). Section 4. The number varies automatically with day counter as: NSEED=NSEED+100*NDAY. Section 11
NEED2	4321	Random number seed for generating vehicle arrival headways. Section 4. It varies with day counter if QDEMAND=T as: NSEED2=NSEED2+100*NDAY. Section 11
PBUS	0	Proportion(%) of total demand (in number of vehicles/hr) made up by buses, i.e. PBUS=10, 10% of total demand are buses. Section 6.1.4
PFASTBRD	0	Proportion (%) of total passengers hold seasonal bus tickets.
PHGV	0	Proportion of total number of trips are made by HGVs Section 6.1.4
PISA	0	Percentage vehicles fleet under ISA control Section 8.4
PLGV	0	Proportion of total number of trips are made by LGVs Section 6.1.4
PPK	0.0	Pence per km – used to convert distances into generalised costs. Section 9.
PPM	1.0	Pence per minute, used to convert times into generalised costs. Section 9.
PTAXI	0	Proportion (%) of total number of trips are made up by taxis Section 6.1.4
PTRAJ	0	Percentage of vehicles' trajectories to be output. Example: PTRAJ=10 for 10%.
QCARL	F	If T, display vehicle-length in proportion to the link-length and crow-fly distance. Section 4.2.2.2

QCHRG	F	TRUE if to calculate congestion road pricing Section 8.5
QDET_FULL	F	If T output full detection report into file **.TXD**
QEXPERT	F	If T the level of print out in .TXS file generally is such that only an "expert" would fully appreciate it.
QMAINSIM	F	T if to terminate the simulation at the end of the main period. Section 6.1.1
QPT_PRIORITY	F	T if to carry out extension or early-recall type of priority signal controls Section 8.4
QTRAJ	F	True if to output vehicle trajectories in **.TRJ** file Section 9.1
RB_GAP	3.0	Acceptable gap (sec) used upon entering roundabouts Section 6.1.6
RB_VISIBILITY	50	Distance on roundabout which the entry traffic would look back for potential conflicts (m) Section 6.1.6
RDCOLOUR	7	Road colour displayed. Section 3.7.2.2
RECALL	5	The amount of recall time of a red signal (sec) Section 8.4
START_HOUR	7	Clock's starting hour to be displayed
START_MIN	50	Clock's starting minute to be displayed. This correspond to the default warm-up period.
TIMECHRG	1.0	Charge rate for time-based pricing (pence/min) Section 8.5
TAPPRO_JNCT	18.0	Estimated time (sec) before approaching a junction from when a vehicle begins to react to the junction control. Estimated based on the current cruising speed. Sections 6.1, and XAPPRO_JNCT
TBOARD1	4.0	Boarding time per passenger (including purchasing a ticket) (sec/person) Section 5.2
TBOARD2	1.0	Boarding time for seasonal ticket holders (sec/person) Section 5.2

TDOOR	5.0	Time takes for a bus to open and close door (sec) Section 5.2
TCOOL	10	Cooling off period (minutes). Section 6.1.1.
TINLANE_BUS	30	Time (seconds) a bus keeps in the lane it has just changed into before making another lane-changing attempt.
TINLANE_CAR	15	Ditto for cars
TINLANE_HGV	30	Ditto for large goods vehicles
TMAIN	30	Simulation time period (minutes) Section 6.1.1
TWARM	10	Warmup period (minutes) Section 6.1.1.
TOUTPUT	10	Time interval (min) for outputs Section 9.1
VZTPM	10	*Speed (km/hr) below which a Z-type turning movement and major flow traffic will take in turns to merge into exit link* *Section 4.2.1.5 - under development*
XAPPRO_JNCT	150.0	Distance (metres per 30kph) upstream of a stop-line where a vehicle starts to react to junction control. Sections 6.1.7 and 6.1.8, and TAPPRO_JNCT
XVIEW_JNCT	6.0	Distance upstream of a stop line where a vehicle starts to look for gaps in the approaching junction Section 6.1.8

APPENDIX C: The DRACULA-MARS menus

Menu	Command	Function
File	Load network	Load in simulation network
	Load BMP map	*Load in bitmap background map in .bmp format.*
	Exit	Close graphics window first by clicking the OK button, before exit
	Screen	Change screen colours
Display	Network info.	Display network information
	Demand info.	Display demand description
	Zone number	Toggles to display or not the zone numbers
	Node number	Toggles between display or not display node numbers
	bus-stop number	As above for bus-stop numbers
	Background map	*As above for background map*
	GIS data	As above for GIS data
View	Full network	Fit the full network on screen
	Full background	Fit the bitmap according to either the original coordinates specified in .XYB file or to align the image to bottom-left corner of screen
	Windows	Select to change either network, bitmap or both.
	Box	Use left-hand muse button to select the area for viewing
	Zoom	Zoom in
	Pan	Zoom out
	Left	Move the current viewing box to the left
	Right	.. to the right

	Up	.. upwards
	Down	.. downwards
Simulation	Run	To start the simulation
	Animation	Toggles between with and without graphical animation
	Display vehicle by….	
	type	Display vehicle by vehicle-type
	Junction turning	… by vehicle's downstream junction turning movement
	Route	… by routes (for limited number of routes only)
	Type & randomly	…By vehicle-type, and randomly select colours for cars
	Animation by….	
	vehicle	Display individual vehicles
	link delay	Display link delay
	wider	When display link delay, make the link/bandwidth wider
	narrower	… make it narrower

APPENDIX D.1: Changes in DRACULA 2.0

DRACULA 2.0 was released in January 2003 and is the first fully 32-bit version.

I. MODELLING

- Time-dependent demand with fixed assignment modelled as step-functions (§4.3.1.2).

- Time-dependent demand with variable assignment from **SATURN** multiple-time period modelling (**SATTPX/SATSUMA**) (§ 4.3.3.3)

- New junction lane-choice model giving greater opportunity for vehicles to choose a downstream lane that takes into consideration its next junction turn - a "looking-ahead" factor. It also considers vehicle types: slower-moving vehicles made to choose near-side lane and vice versa.

- Improved overtaking model for long (motorway) links: after overtaking, vehicles prefer to move back.

- Modelling yellow-box junctions (Section 6.2).

- Combining rural two-lane overtaking using opposite road space with ordinary overtaking over multiple lanes in the same direction (§8.5).

- Definition of vehicle-character distributions is extended to include, not only means and variances, but also upper and lower limits (§6.1).

- New DRACULA-style demand input in **.TRP** files: (a) with a new data field to represent user class using the route, and (b) time-dependent route flows to be represented in different **.TRP** files (§4.3.1).

- Distinction between user class and vehicle type (§4.3.1.1).

- New vehicle detector record and new responsive signal control logic. The detector input includes a new record which specifies the node number of the junction which is linked to the detection. It is no longer the downstream node of the link on which the detector is located that is to be controlled. It is now possible that a detector can be used to control a junction several links downstream, or a junction not even on any downstream routes (§8.1).

- New definition of supply and performance measures for network, routes and links. Outputs in **.OUT** file (§9.1)

- Link-based (or approach-based) gap acceptance values (§6.2.1)

- Roundabout-specific circulating speed (§6.2)

- Simplified coding for two-way bus route and bus route in reverse order (§5.2.1).

II. TECHNICAL

- Generally, more checks and report on coding errors.

- New SATURN-style data display program **SPATULA** which takes **DRACULA** simulation outputs (in **.SPA** file) and produces a **.UFD** file which can be loaded to the SATURN program **P1X** for display (§9.2).

- Representation of curved links, via SATURN-style GIS data input (§10.1).

- Check for non-connectivity and banned turns in input routes including bus routes.

- Automatically convert **SATURN** bus route information (in card 66666) into a **DRACULA**-formatted data file in **.BUS** (§5.2.1).

- The "name" of a bus route can be alpha-numeric (e.g. a string of characters and/or numbers) of up to 5 characters (§5.2.1).

- If all or part of a bus route follows a straight line (more or less), it is not necessary to define every node along the line, only the nodes at the bends. This applies to the routes that are manually created in a **.TRP** file.

- New bus service record in **.BUS** file: does not require input of origin/destination zones. This is in recognition the fact that bus services may not always begin/end in a conventional zone. The old format, where origin and destination zones for a bus service are coded, can still be used by setting parameter QOLDBUS=T in **.PAR** file (§5.2.1).

- A population/vehicle list created by **DRACPREP** in **.VEH** file which contains all vehicles to be simulated and their departure times and characteristics. This can be used as a fixed population from one simulation to another (§4.3.2).

- Automatically convert SATURN junction-based gap acceptance parameters (GAP, GAPR, GAPM) to DRACULA-style link-based gap acceptance values (§6.2).

- Read in SATURN network coordinates using the DUTCH format.

- Options in Display menus of **DRACULA-MARS** Window to change colour schemes used in simulation, including those to colour a bus lane and a blocked lane (§3.7.2, Appendix B).

- Check coding on lane usage: whether all lanes being used for turns.

- All additional network descriptions are to be coded in file **.ADD**. Additional coding currently includes: junction-dependent gap acceptance parameters (§6.2.1), roundabout circulating speed (§6.2), rural overtaking links (§8.5), and ISA speeds (§8.3)

- New **SATPIG** (with **SATURN 10.3**) outputs to **DRACULA** required **.TRP** format (§4.3.3).

- Definition of input file names via parameters in file **.PAR** (Sections 2.2.3, 4.2.1 and 4.2.2).

- Complete parameter lists with cross-reference to the relevant sections in Manual (Appendix A and B).

- Displayed clock's starting time (hour and minute) can be specified via parameters in **.PAR** file (Appendix B).

- Option for setting all link-lengths to that of crow-fly distances between upstream and downstream nodes, via parameter QCROWFLY in .PAR file (§4.2.2.5).

- For SATURN-format network coding, various defaults were set in case of implausible coding. These include setting link-length to the crow-fly distance and link free-flow speed to 40 km/hr if they were coded as zeros (§4.2.2.5)

- A windows-based front end **DRACWIN** from where the DRACULA programs and the relevant SATURN programs can be accessed.

APPENDIX D.2: Changes in DRACULA 2.1

DRACULA 2.1 was released in October 2003.

I. MODELLING

- Improved bus lane-changing algorithm to consider the need of overtaking a dwelling bus in order to reach a bus stop in front of.

- Global parameters TINLANE_BUS, etc to control the "cooling" period a vehicle keeps after completing one lane-changing movement before attempting another one. §6.1.7.

- New junction models for merge, motorway slip road, and turning movements with clear exit (§4.2.1.5).

- New type of junction control: always give-way to traffic from left (or right) for driving on the left (or right). This is a type of junction control frequently seen in Dutch networks and some networks in Belgium. §4.2.4.

II. TECHNICAL

- Options to load bitmap background map in .BMP format, similar to that used in SATURN. §10.2.

- From Version 2.1, SATURN-format GIS files can be loaded directly into DRACULA without any modification. In previous versions, the corners of a polygon, poly-line or a curved link have to be specified manually in *.gis* file. See §10.1.

- Bitmap and GIS file names can be specified via parameters in either the network file (.net) or the parameter file (.par). Alternatively, a GIS file can also be picked up by **DRACSIM** automatically if it shares the same name as the network name (§2.2.3). The bitmap file can be loaded via command line input (§3.6) or via **DRACSIM** file menu (Appendix C).

- Include SATURN program **SATPIG** within DRACWIN front-end for the convenience of DRACULA users. SATPIG extracts route assignment results from SATURN in a format directly importable to DRACULA. §3.

- Allow removal of routes which have very small flows. §4.3.3.4.

APPENDIX D.3: Changes in DRACULA 2.2

DRACULA 2.2 was released in December 2004.

I. MODELLING

- Allow mixture of two-lane "rural" roads (where overtaking is allowed in opposite road space) and "normal" roads (where vehicles are not allowed to move over to the other side of the road) in one network. §8.5.

- New type of junction control: give-way to on-coming traffic. Associated is a new link type: links with just one lane and a narrower than normal lane. This is to represent narrow roads, such as a narrow bridge, where one direction of traffic has to give way to traffic from the opposite direction. §4.2.1 and §4.2.5.

- Model of traffic calming measures. §6.2.

- Removed constraints on how many links can be in any one stages. This applies to both the DRACULA-format network coding (§4.2.1.2) and the loading of SATURN networks.

- The default vehicle type LGV should be called Light goods vehicle instead of Large goods vehicle. They include 2-axil transit vans. §6.1.2.

- Allow user-specification of the speeds at which vehicles enter the network. By default, vehicles start with a zero speed at entry. The car-following model will determine the speed of the vehicle at the next time increment. Through the new parameters, INIT_SPEED and INIT_OLD_SPEED, users can specify the global initial speeds (in kph) or, if one of the two new parameters is set to –1, the local free-flow speed of the entry link will be assigned to the vehicle. This is coupled with parameter NENTRY to allow user selection of the entry speed of the vehicles. Appendix B.

- Add coefficient of variation in passenger arrival distributions.

II. TECHNICAL

- Remove SATURN bus routes which have zero service frequency. A warning message is given in .TXP file.

- An error message will be given in .TXP file, and the program DRACPREP terminated abnormally, if a traffic signal stage is coded with more than two links. §4.2.1.2 and 4.2.2.3.

- Auto-load BMP and GIS file names specified via SATURN parameters BMPFIL and GISFIL to DRACULA. This function will also work for a cordoned SATURN network – as long as the BMPFIL and GISFIL are passed to the cordoned network file.

- Output error message and terminate DRACPREP process if the number of A- and C- nodes specified for traffic signal stage does not match the number required.

- Corrected display of signal aspects for turns.

- Corrected an error in displaying curved links for driving-on-the-right situation.

- Allow easy specification of a single speed-limit on a link by setting the end position of the speed-limit as –1. §8.3.

- Pass new SATURN parameters IFGIS(n), n=1,2,..7, to DRACULA as options to switch certain GIS features on/off. Defaults are False. See section §4.2.2.1. Also introduce them as DRACULA parameters which can be specified in .par files (Appendix B) and add options within DRACSIM windows menu to allow GIS features switched off/on (§3.8.2.2).

- Correction made on the discrepancy between what documented in Section 8.2 and what programmed the marker for the kerb lane in coding detectors. What documented suggests that the kerb lane should be coded as 1. What programmed had treated 0 as the kerb lane. Correction is made in the programme.

- Include in the DRACULA-format network coding, specification of yellow-box junctions.

- Reset all network limits based on the licensed number of nodes, in DRACPREP.

- Set NGDIGIT to allow user-specified digit in outputs (.spa).

APPENDIX D.4: Changes in DRACULA 2.3

DRACULA 2.3 was released in December 2005.

I. MODELLING

- Improved models of roundabout with better management of conflicts at roundabout approaches.

- Add a global variable TAPPRO_JNCT to represent the time before approaching a junction when a vehicle begins to react to junction control. This replaces the old XAPPRO_JNCT, which is now used to represent the absolute distance upstream of the junction where the vehicle starts to react to junction control. §6.1.

- Make the equivalent of TAPPRO_JNCT and XAPPRO_JNCT link-specific, to be included in the additional data input file .ADD. §6.2.

- Start to generate passenger flows to the network ***after*** the warm-up period. This is to ensure that there is sufficient number of buses serving the network and picking up the passengers when the main simulation period begins, and in doing so, to eliminate the potential problem of excess waiting by passengers at the end of a long bus route.

- Introduce turn priority marker (TPM) "C" and "S" to indicate a clear turn to the near and far side exit respectively (§4.2.1.5).

- The vehicle characteristics are drawn from truncated normal distributions with the values of mean, variance, minimum and maximum of the distributions specified by the users (§6.1.3).

- The default simulation break-out time is now set as three times of the main demand period (§6.1.1).

II. TECHNICAL

- Provide warning messages in .TXP file if a point along a curved link falls inside the junction (§10.1.5).

- When a curved link is specified in .GIS file more than once, only the first specification is used and a warning message is written to .TXP file (§10.1.5).

- Two menu options to display points along a curved link and their A- and B-node numbers are provided in the DRACSIM window (§3.8.2).

- Examples of network and other coding and practical tips are highlighted with graphical symbols[*]

---

[*] The author wishes to acknowledge the contribution of Yaron Hollander who created the symbols for examples and practical tips.

- Option to do a simulation run for a pre-defined time period (§6.1.1).
- Improved display of stopline marks along curved links and on approaches to roundabouts.
- Improved display of bus-lane and flared approached on curved links.
- Allow space in direction and file names.
- Improved display of bus-layby; they are now drawn offside of the main carriageway.
- Check on route connectivity extends to bus routes.