Traffic simulation

User manual (version 1.6.0)

Laboratory for Data Technologies
Faculty of Computer and Information Science
University of Ljubljana

October 2012

Contents

1 Traffic simulation

1.1 Traffic simulator

Description Desktop application for traffic simulation (see Figure 1) that allows some further control over the simulation (e.g., addition of collisions, congestions, vehicles). When selected, current state of traffic simulator is also logged to MySQL database (see Section 1.4).

Platform PC, Linux or Mac (preferred).

Requirements JRE 6 and MySQL database (when selected).

Execution Execute `java -Dfile.encoding=UTF8 -jar Traffic-1.6.0/Traffic.jar -Xmx1g` in a console or double-click `Traffic-1.6.0/Traffic.jar` (see Figure 2).

Configuration Set MySQL database host, port, user name, password and schema in `Traffic-1.6.0/parameters.xml` appropriately (when selected).

Support Slovenia (preferred), Europe and toy.

Currently User desktop.

Comment For best performance, traffic simulator and MySQL database (see Section 1.4) should reside on the same physical machine (when selected).

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*).

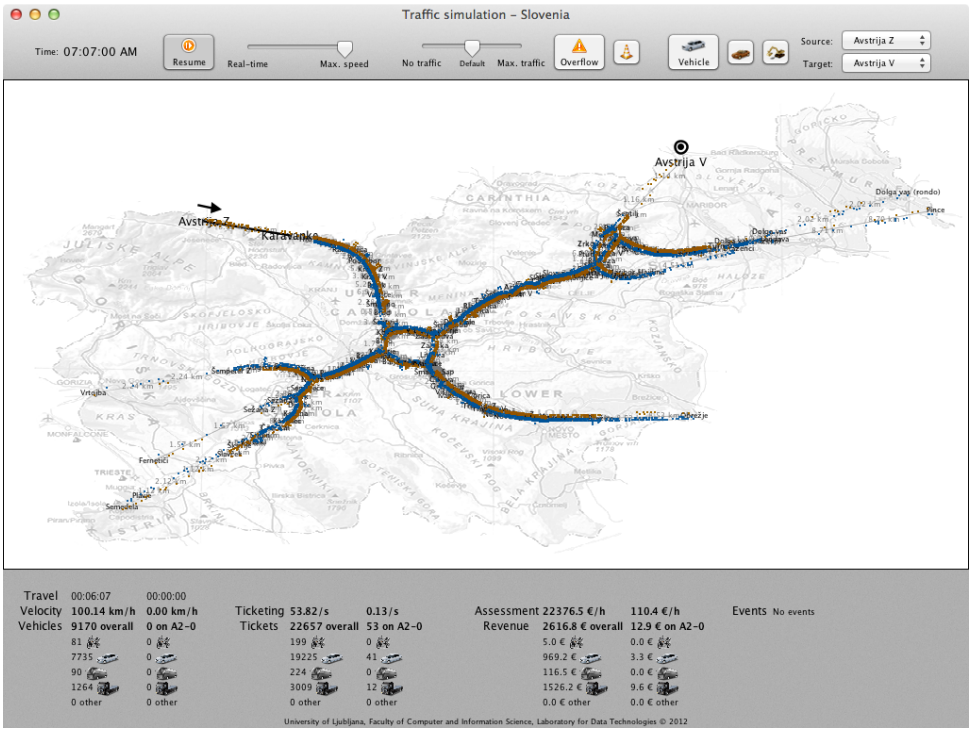
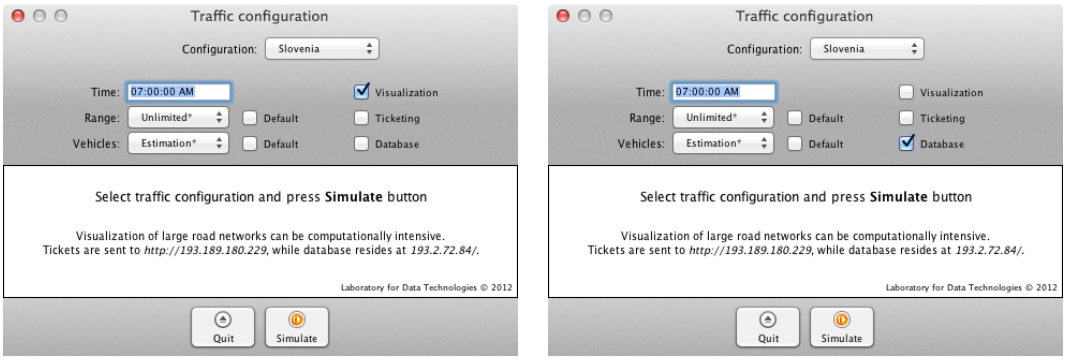


Figure 1: Traffic simulator application.



(a) Basic configuration. (b) Configuration with database logging.

Figure 2: Traffic simulator configuration.

1.2 Traffic server

Description TCP socket and HTTP server for traffic simulation that return detailed travel information for a designated vehicle (due to source and destination location, time, speed limits etc.). When selected, travel details are also cached to MySQL database (see Section 1.4).

Platform PC, Linux (preferred) or Mac.

Requirements JRE 6 and MySQL database (when selected).

Execution Execute `java -Dfile.encoding=UTF8 -jar Traffic-1.6.0/Traffic-server.jar -Xmx512m` in a console or double-click `Traffic-1.6.0/Traffic-server.jar`.

Configuration Open ports `65505` and `65535` for TCP traffic, and set MySQL database host, port, user name, password and schema in `Traffic-1.6.0/parameters.xml` appropriately (when selected).

Support Slovenia (preferred), Europe and toy.

Currently `193.2.72.83` and `193.2.72.84`.

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*).

1.3 Traffic visualization

Description Desktop application for visualization of traffic simulation (see Figure 3). Application visualizes current state of traffic simulator (see Section 1.1) that is logged to MySQL database (see Section 1.4).

Platform PC (preferred), Linux or Mac.

Requirements JRE 6 and MySQL database.

Execution Execute `java -Dfile.encoding=UTF8 -jar Traffic-1.6.0/apps/visual/TSv1.0.jar -Xmx1g` in a console or double-click `Traffic-1.6.0/apps/visual/TSv1.0.jar`.

Configuration Set MySQL database host, port, user name, password and schema in `Traffic-1.6.0/apps/visual/config.txt` appropriately.

Support Slovenia.

Currently User desktop.

Comment For best performance, traffic visualization and MySQL database (see Section 1.4) should reside on the same physical machine.

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*) and Ernest Beličič (*ernest.belicic@gmail.com*).

Figure 3: Traffic visualization application.

1.4 Traffic database

Description MySQL database for traffic simulation that logs current state of traffic simulator (see Section 1.1) and caches detailed travel information for traffic server(s) (see Section 1.2).

Platform PC, Linux (preferred) or Mac.

Requirements MySQL database.

Configuration Create MySQL database using script `Traffic-1.6.0/sql/traffic.sql` and fill it using script `Traffic-1.6.0/Slovenia/Slovenia.sql` (when selected).

Support Slovenia.

Currently `193.2.72.84`.

Comment For best performance, MySQL database, traffic simulator (see Section 1.1) and traffic visualization (see Section 1.3) should reside on the same physical machine.

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*) and Milutin Spasič (*milutinspasic92@gmail.com*).

2 Additional

2.1 *Prevozi Slovenije*

Description Android application that shows scheduled travels at `http://prevoz.org` (see Figure 4) with detailed travel information retrieved from TCP traffic server (see Section 1.2).

Platform Android.

Requirements TCP traffic server.

Execution Execute application `Prevozi Slovenije` on a tablet.

Configuration Set TCP traffic server host and port in `Traffic-1.6.0/apps/prevoz.org/assets/Domains.txt` appropriately, and install application on a tablet using `Traffic-1.6.0/apps/prevoz.org/bin/Prevozi Slovenije.apk`.

Support Slovenia.

Currently *ASUS Eee Pad Transformer* tablet.

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*) and Nejc Gašperin (*nejc.gasperin@gmail.com*).

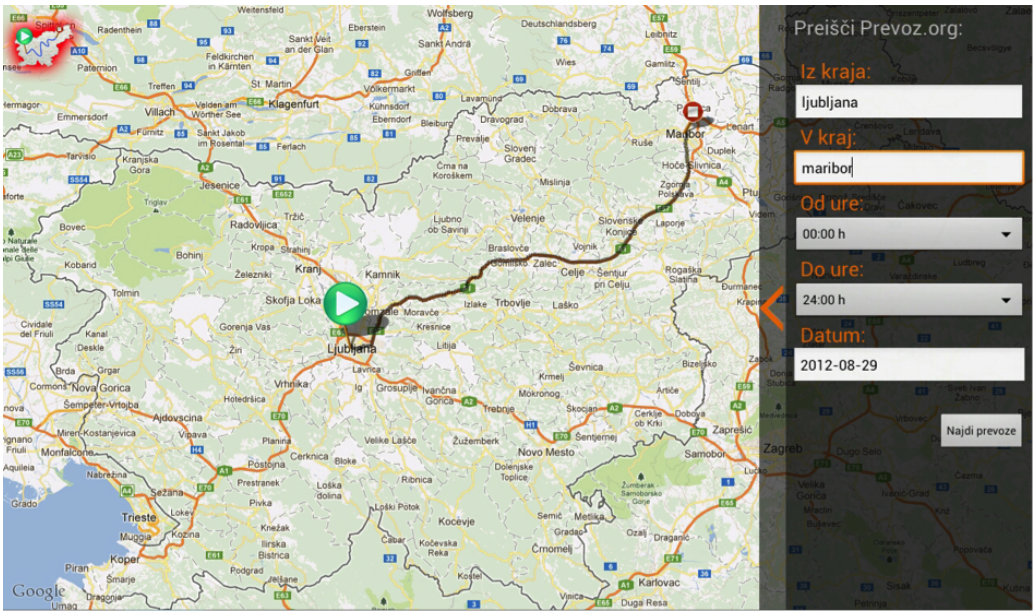


Figure 4: Android application *Prevozi Slovenije*.

2.2 Travel information

Description Web application that plots detailed travel information for a designated vehicle (see Figure 5) retrieved from HTTP traffic server(s) (see Section 1.2).

Platform PC, Linux or Mac (preferred).

Requirements Browser with enabled JavaScript and HTTP traffic server(s).

Execution Open `Traffic-1.6.0/apps/travel/travel.html` in a browser or double-click `Traffic-1.6.0/apps/travel/travel.html`.

Configuration Set HTTP traffic server host(s) and port in `Traffic-1.6.0/apps/travel/travel.html` appropriately.

Support Slovenia (preferred), Europe and toy.

Currently `http://lovro.lpt.fri.uni-lj.si/travel.html`.

Contact Lovro Šubelj (*lovro.subelj@fri.uni-lj.si*).

Figure 5: Travel information web application.

Traffic simulator

Version 1.2.1

Laboratory for data technologies
Faculty of computer and information science
University of Ljubljana

May, 2012

Contents

1 Design specification

1.1 Road networks

Traffic simulation road network is defined as a set of *roads* of some *category*. Road category can be either `HIGHWAY`, `MAIN_ROAD`, `ROAD`, `STREET` or `TRANSFER` (see below). Each road consists of consecutive directed *sections* defined through source and destination *locations*. Each section consists of multiple parallel *lanes* that are connected through *paths*. Paths define possible connections between consecutive section lanes. Each lane is further partitioned into equidistant *cells* that can be occupied by at most one vehicle (see Section 1.2). For visual examples see Figure 1 and Figure 2.

Roads, sections, locations and lanes are defined through traffic simulation configuration files, while `TRANSFER` roads and sections, paths and cells are determined automatically based on traffic simulation parameters (see Section 3.2). Transfer roads and sections are artificially added at each location and serve for vehicle inflow and outflow from the network.

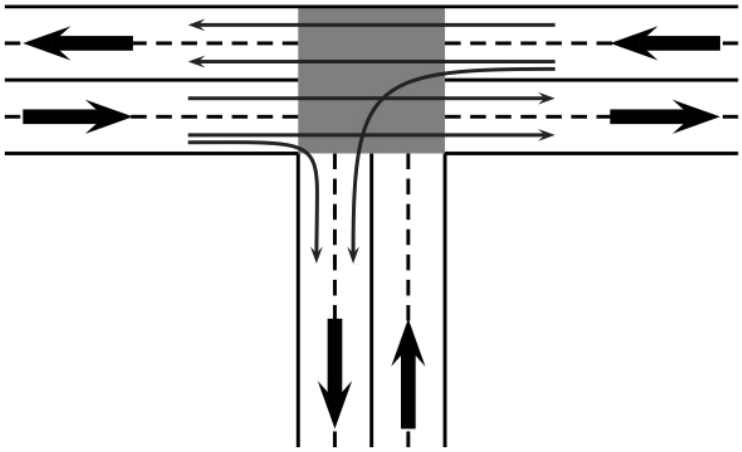


Figure 1: Example road network containing two roads (vertical and horizontal) with six directed sections (thick arrows) between four locations. Each section consists of two lanes that are connected through six paths (narrow arrows). (Figure is adopted from De Gier et al. (2011).)

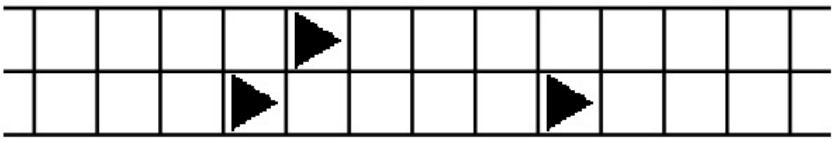


Figure 2: Example road section with two lanes partitioned to equidistant cells containing three vehicles (triangles). (Figure is adopted from De Gier et al. (2011).)

1.2 Simulated vehicles

Simulated *vehicles* on some road network can be of different *types*, where vehicle type can be either `MOTOR`, `CAR`, `BUS`, `TRUCK` or `OTHER` (see below). State of the vehicle is defined through its current *speed* and *position* within the road network, while all other characteristics (e.g., type) are determined automatically based on traffic simulation parameters (see Section 3.2).

Vehicles can be further divided between *standard* (simulated) vehicles that are added by the traffic simulator, and user *designated* vehicles that are added through graphical user interface

(see Section 3.4). Standard vehicle type is randomly selected based on the vehicle type probability distribution (see Section 3.2), while all designated vehicles are of type `OTHER`. Designated vehicles can either respect road speed limits, or adopt fast or slow speed limits defined in traffic simulation parameters (see Section 3.2).

1.3 Traffic dynamics

Traffic simulator implements different realistic models of vehicle dynamics on an arbitrary road network. Position and speed of all vehicles is updated at the end of each time second, while the time step is not a parameter of the traffic simulator (due to simplicity).

Vehicle dynamics on a single section lane are modeled using classical Nagel-Schreckenberg model of highway traffic (see Nagel & Schreckenberg (1992)). The main rationale behind the model is that each vehicle would always attempt to accelerate, as long as it remains in one of the allowed states (e.g., no collision among vehicles). Dynamics between parallel section lanes are modeled using topological and dynamical line changes (see De Gier et al. (2011)). Topological line changes model dynamics that are needed for a vehicle to reach its destination (e.g., vehicle outflow), while dynamical changes model other reasonable movements that a vehicle can make (e.g., vehicle overtaking). Finally, vehicle dynamics between consecutive section lanes are determined using section traffic loads at the corresponding time (see Section 3.2). The rationale behind the latter is that vehicle inflow must always equal the outflow at each location. Models of vehicle dynamics can be controlled through traffic simulation configurations and parameters (see Section 3.2).

Standard vehicle dynamics is based on road section traffic loads (see above), while designated vehicles with specified destination always adopt the shortest path across the road network (see Section 1.2). Shortest path is defined as a sequence of sections towards destination with the shortest travel time considering road speed limits.

2 System architecture

Traffic simulator is a standalone computer application that requires no installation or additional resources. The application is developed in *Java* 1.6 programming language and is logically partitioned into three modules, namely, domain, simulation and visualization modules. Domain module contains all road network entities and corresponding parsers. Simulation module consist of traffic simulator, road network navigator, ticketing and logging functionality. Visualization module contains graphical user interface that also enables certain control over traffic simulator. High-level view of the application can be seen in Figure 3.

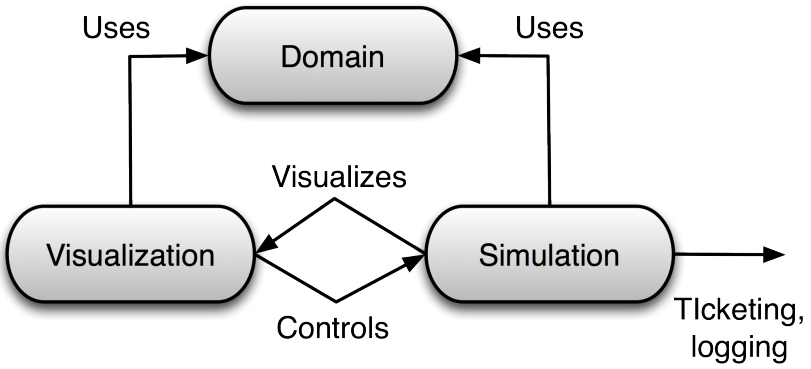


Figure 3: High-level view of the traffic simulator application.

Domain module is implemented in `si.uni.lj.fri.lpt.traffic.domain` package, simulation module is implemented in `si.uni.lj.fri.lpt.traffic.simulation` package and visualization module is implemented in `si.uni.lj.fri.lpt.traffic.visualization` package. For further details see *Javadoc* documentation (see Section 3.1).

3 System documentation

3.1 File system structure

File structure of root folder `Traffic-1.2.1` is given in Table 1, while detailed description of different configuration and output files is given in Section 3.2 and Section 3.3, respectively.

File or folder	Description
`Traffic.jar`	Traffic simulator executable code as *JAR* archive.
`Traffic-source.jar`	Traffic simulator source code as *JAR* archive.
`Traffic.sh`	Traffic simulator executable as Unix *sh* script (preferred).
`Traffic.bat`	Traffic simulator executable as Windows *bat* script (preferred).
`parameters.xml`	Traffic simulation parameters (see Section 3.2).
`stdout`	Traffic simulation output as plain file (see Section 3.3.1).
`Traffic-lib`	Libraries for traffic simulator as *JAR* archives.
`icons`	Traffic simulator icons as *PNG* graphics.
`docs`	Traffic simulator documentation as *TeX* files.
`javadoc`	Traffic simulator *Javadoc* documentation as *HTML* files.
`data`	Traffic simulation configurations as folders (see Section 3.2).
`logs`	Traffic simulation logs as *TAB* files (see Section 3.3.3).

Table 1: Description of the file structure of root folder `Traffic-1.2.1`.

Comments.

`Traffic.jar` Double-click executes the traffic simulator, however, the usage of `Traffic.sh` and `Traffic.bat` scripts is preferred.

`Traffic.sh` Script executes `Traffic.jar` using UTF-8 encoding and 1 GB of reserved memory resources (preferred). Note that values should be set according to limitations of the computer architecture or the operating system! Script can be run from the console on Unix-based systems, whereas `java` command must be included in the class path.

`Traffic.bat` Script executes `Traffic.jar` using UTF-8 encoding and 1 GB of reserved memory resources (preferred). Note that values should be set according to limitations of the computer architecture or the operating system! Script can be run from the console on Windows-based systems, whereas `java` command must be included in the class path.

`data` Traffic simulation configurations consist of files `roads`, `traffic`, `ticketing` and `locations`, while only the first two are mandatory (see Section 3.1). Background for road network visualization can be defined as graphics file `background.png`.

3.2 System configuration

Traffic simulation is configured through file `parameters.xml` and graphical user interface (see Section 3.4), while simulation for some road network is configured through files `roads`, `traffic`, `ticketing` and `locations` (only the first two are mandatory).

3.2.1 File `parameters.xml`

Description. Configuration file `parameters.xml` contains traffic simulation parameters. Default `parameters.xml` file can be seen in Listing 1, while detailed description is given in Table 2.

Parameter	Value	Default	Description
configuration[1]	*Slovenia* etc.	*Slovenia*	Traffic simulation configuration.
cell	> 0.0 km	0.005 km	Road section cell length.
transfer	> 0.0 km	2.0 km	Road section transfer length.
propagation	*true* or *false*	*true*	Road section directions propagation.
time[1]	≥ 0 s	10800 s	Initial traffic simulation time.
timeRange[1]	> 0 s or -1	-1	Traffic simulation range (-1 for unlimited).
trafficRange	> 0 s	3600 s	Road section traffic loads estimation time.
loggingRange	> 0 s	15 s	Traffic simulation logging time interval.
vehicles[1]	≥ 0 or -1	-1	Initial vehicles number (-1 for estimated).
initialNumber	> 0	1000	First vehicle identification number.
limitNumber	> 0	100000	Vehicle identification number limit.
fastLimit	> 0.0 km/h	250.0 km/h	Fast designated vehicle speed limit.
slowLimit	> 0.0 km/h	50.0 km/h	Slow designated vehicle speed limit.
breakingProbability	$[0, 1]$	0.2	Vehicle random breaking probability.
changingProbability	$[0, 1]$	0.95	Vehicle line changing probability.
logging[1]	*true* or *false*	*false*	Traffic simulation logging.
ticketing[1]	*true* or *false*	*false*	Traffic simulation ticketing.
ticketingServer	URL	*http://.../*	Traffic ticketing server URL.
ticketingService	String	...	Traffic ticketing service name.
ticketingResponse	*true* or *false*	*true*	Traffic ticketing response check.
ticketingThreads	> 0	1000	Traffic ticketing threads limit.
ticketingExtra	String	...	Traffic ticketing extra string.
visualization[1]	*true* or *false*	*true*	Traffic simulation visualization.
scaling	$(0, 1]$	1.0	Road network visualization scaling.
iterations	> 0	5000	Network layout algorithm iterations.
code	String	*A*, *H*, *C* etc.	Unique road category codes.
limit	> 0.0 km/h	130.0 km/h etc.	Road category speed limits.
assessment	≥ 0.0 €/km	0.01 €/km etc.	Road categories, vehicle types assessments.
probability	$[0, 1]$	0.84, 0.14 etc.	Vehicle type probability distribution.
axles	≥ 0	2, 3, 4	Vehicle type axles number.
ticketed	*true* or *false*	*true*	Vehicle type ticketing.

Table 2: Description of configuration file `parameters.xml`.

Comments.

configuration Traffic configuration must be one of the defined configurations (see Section 3.1).

cell According to De Gier et al. (2011), typical space occupied by a vehicle in a jam is 7.5 meters. Thus, road section cell length should be below this value (see Section 1.1).

[1]Parameter can also be adjusted through graphical user interface (see Section 3.4).

transfer Road section transfer length should be large enough to prevent congestion of vehicles on section source locations (see Section 1.1).

loggingRange To reduce log file size, simulation logging time interval defines the time between two logging events (see Section 3.3.3).

limitNumber*, *limitNumber First vehicle identification number should be larger than the maximum possible number of designated vehicles, while identification number limit should be small enough to allow vehicle repetitions with high probability (see Section 1.2). Standard vehicle identification numbers are randomly selected from the range between *initialNumber* and *limitNumber*, while designated vehicles have consecutive identification numbers from the range between 1 and *initialNumber*.

breakingProbability Random vehicles deceleration is crucial for realistic traffic simulation (see Section 1.3). According to De Gier et al. (2011), vehicle random breaking probability should be around 0.2.

changingProbability Vehicle line changing probability should be below 1 to prevent oscillations of vehicles on consequtive time steps (see Section 1.3).

logging Whether the state of the traffic simulation is logged to file (see Section 3.3.3).

ticketing Whether vehicle traffic tickets are sent during traffic simulation (see Section 3.3.2).

ticketingThreads Traffic ticketing threads limit corresponds to the maximum number of threads that can be designated for sending tickets (see Section 3.3.2). Value should be below the limitations of the computer architecture or the operating system!

ticketingExtra Traffic ticketing extra string is only for testing purposes and has no actual semantic meaning (see Section 3.3.2).

visualization Whether traffic simulation is visualized as a road network (see Section 3.4).

assessment Vehicle type assessment value for a road of certain category (see Section 3.3.3). Parameters are not mandatory, while the default value is 0.0.

probability Vehicle types are randomly selected from the defined probability distribution (see Section 1.2). When the values do not represent probability distribution, they are normalized accordingly.

ticketed Ticketed vehicle types, while no tickets are send for other types (see Section 1.2).

Listing 1: Default configuration file `parameters.xml`.

```
<parameters>

  <configuration>Slovenia</configuration>
  <cell>0.005</cell>
  <transfer>2.0</transfer>
  <propagation>true</propagation>

  <time>10800</time>
  <timeRange>-1</timeRange>
  <trafficRange>3600</trafficRange>
  <loggingRange>15</loggingRange>

  <vehicles>-1</vehicles>
  <initialNumber>1000</initialNumber>
```

```
15  <limitNumber>100000</limitNumber>
16  <fastLimit>250.0</fastLimit>
17  <slowLimit>50.0</slowLimit>
18  <breakingProbability>0.2</breakingProbability>
19  <changingProbability>0.95</changingProbability>
20
21  <logging>false</logging>
22  <ticketing>false</ticketing>
23  <ticketingServer>http://193.189.180.229/</ticketingServer>
24  <ticketingService>AccessModuleContinental/postMessage</ticketingService>
25  <ticketingResponse>true</ticketingResponse>
26  <ticketingThreads>1000</ticketingThreads>
27  <ticketingExtra></ticketingExtra>
28
29  <visualization>true</visualization>
30  <scaling>1.0</scaling>
31  <iterations>5000</iterations>
32
33  <code>
34    <category>HIGHWAY</category>
35    <value>A</value>
36  </code>
37  <code>
38    <category>MAIN_ROAD</category>
39    <value>H</value>
40  </code>
41  <code>
42    <category>ROAD</category>
43    <value></value>
44  </code>
45  <code>
46    <category>STREET</category>
47    <value>C</value>
48  </code>
49  <code>
50    <category>TRANSFER</category>
51    <value>T</value>
52  </code>
53
54  <limit>
55    <category>HIGHWAY</category>
56    <value>130.0</value>
57  </limit>
58  <limit>
59    <category>MAIN_ROAD</category>
60    <value>100.0</value>
61  </limit>
62  <limit>
63    <category>ROAD</category>
64    <value>90.0</value>
65  </limit>
66  <limit>
67    <category>TRANSFER</category>
68    <value>60.0</value>
69  </limit>
70  <limit>
71    <category>STREET</category>
72    <value>50.0</value>
73  </limit>
74
75  <probability>
76    <type>MOTOR</type>
77    <value>0.01</value>
78  </probability>
79  <probability>
80    <type>CAR</type>
81    <value>0.84</value>
82  </probability>
83  <probability>
84    <type>BUS</type>
85    <value>0.01</value>
86  </probability>
87  <probability>
88    <type>TRUCK</type>
```

```
    <value>0.14</value>
  </probability>
  <probability>
    <type>OTHER</type>
    <value>0.0</value>
  </probability>

   <ticketed>
    <type>MOTOR</type>
    <value>true</value>
  </ticketed>
  <ticketed>
    <type>CAR</type>
    <value>true</value>
  </ticketed>
  <ticketed>
    <type>BUS</type>
    <value>true</value>
  </ticketed>
  <ticketed>
    <type>TRUCK</type>
    <value>true</value>
  </ticketed>
  <ticketed>
    <type>OTHER</type>
    <value>true</value>
  </ticketed>

  <axles>
    <type>MOTOR</type>
    <value>2</value>
  </axles>
  <axles>
    <type>CAR</type>
    <value>2</value>
  </axles>
  <axles>
    <type>BUS</type>
    <value>3</value>
  </axles>
  <axles>
    <type>TRUCK</type>
    <value>4</value>
  </axles>
  <axles>
    <type>OTHER</type>
    <value>2</value>
  </axles>

  <assessment>
    <category>HIGHWAY</category>
    <values>
      <entry>
        <type>TRUCK</type>
        <double>0.1</double>
      </entry>
      <entry>
        <type>OTHER</type>
        <double>1.0</double>
      </entry>
      <entry>
        <type>MOTOR</type>
        <double>0.005</double>
      </entry>
      <entry>
        <type>CAR</type>
        <double>0.01</double>
      </entry>
      <entry>
        <type>BUS</type>
        <double>0.1</double>
      </entry>
    </values>
  </assessment>
```

```
163   <assessment>
164     <category>MAIN_ROAD</category>
165     <values>
166       <entry>
167         <type>TRUCK</type>
168         <double>0.1</double>
169       </entry>
170       <entry>
171         <type>OTHER</type>
172         <double>1.0</double>
173       </entry>
174       <entry>
175         <type>MOTOR</type>
176         <double>0.005</double>
177       </entry>
178       <entry>
179         <type>CAR</type>
180         <double>0.01</double>
181       </entry>
182       <entry>
183         <type>BUS</type>
184         <double>0.1</double>
185       </entry>
186     </values>
187   </assessment>
188
189 </parameters>
```

3.2.2 File `roads`

Description. Configuration file `roads` contains a list of sections within a road network (see Section 1.1). Example of `roads` file can be seen in Listing 2, while detailed description is given in Table 3.

Listing 2: Example configuration file `roads`.

```
1  A1      30      Avstrija V      Šentilj         2       1.137   1
2  A1      630     Šentilj         Avstrija V      2       1.163
3  A1      31      Šentilj         Pesnica         2       9.761
4  A1      631     Pesnica         Šentilj         2       9.725
5  ...
6  A2      0       Avstrija Z      Karavanke       2       8.019   1
7  A2      600     Karavanke       Avstrija Z      2       8.019
8  ...
9  H2      32      Pesnica         Meljska         2       4.516   1
10 H2      632     Meljska         Pesnica         2       4.513
11 ...
12 430     9000    Ptujska         Slivnica        1       6       1
13 430     9001    Slivnica        Ptujska         1       6
```

Index	Attribute	Value	Example	Description
1	*road*	String	*A1*	Road identification as category and unique number.
2	*section*	≥ 0	30	Road section identification as unique number.
3	*source*	String	*Avstrija V*	Section source location as unique string.
4	*destination*	String	*Šentilj*	Section destination location as unique string.
5	*lanes*	> 0	2	Number of parallel lanes on the section.
6	*length*	> 0.0 km	1.137 km	Actual length of the section.
7	*direction*	0 or 1	1	Direction of the section.

Table 3: Description of configuration file `roads`.

Comments.

road Road category must be one of categories defined in file `parameters.xml` (parameter *code*).

section Section identification number must be unique within the entire road network, and not only within the corresponding road.

direction Attribute is not mandatory, while the default value is 0. Section directions are needed only for the visualization purposes and can be propagated along the corresponding roads as defined in file `parameters.xml` (parameter *propagation*).

3.2.3 File `traffic`

Description. Configuration file `traffic` contains a list of road section traffic loads (see Section 1.1). Traffic load is the average number of vehicles on a section, while traffic estimation time is defined in file `parameters.xml` (parameter *trafficRange*). Traffic loads must be defined for at least one section, and are imputated with the average for all the other sections. Example of `traffic` file can be seen in Listing 3, while detailed description is given in Table 4.

Listing 3: Example configuration file `traffic`.

```
1 A1    30    94.59615385    80.94230769    71.13461538    ...
2 A1    630   75.61538462    60.13461538    53.25          ...
3 A1    31    112.2884615    94.36538462    85.05769231    ...
4 A1    631   88.65384615    70.34615385    58.59615385    ...
5 ...
```

Index	Attribute	Value	Example	Description
1	*road*	String	*A1*	Road identification as category and unique number.
2	*section*	≥ 0	30	Road section identification as unique number.
$i \geq 3$	*loads*	≥ 0.0	94.59615385	Section traffic load over i-th estimation time.

Table 4: Description of configuration file `traffic`.

Comments.

road Network road must be one of the roads defined in file `roads`.

section Road section must be one of the sections defined in file `roads`.

loads Number of traffic loads must be the same for all the sections.

3.2.4 File `ticketing`

Description. Configuration file `ticketing` contains a list of ticketed road sections (see Section 1.1), whereas no tickets are generated for other sections. Example of `ticketing` file can be seen in Listing 4, while detailed description is given in Table 5.

Listing 4: Example configuration file `ticketing`.

```
1 A1        31        1967
2 A1        631       1750
3 ...
4 A2        0         1663
5 A2        600       2026
6 ...
```

Index	Attribute	Value	Example	Description
1	*road*	String	*A1*	Road identification as category and unique number.
2	*section*	≥ 0	31	Road section identification as unique number.
3	*code*	≥ 0	1967	Road section ETS code as unique number.

Table 5: Description of configuration file `ticketing`.

Comments.

road Network road must be one of the roads defined in file `roads`.

section Road section must be one of the sections defined in file `roads`.

code Section ETS code must be unique within the entire road network, and not only within the corresponding road.

3.2.5 File `locations`

Description. Configuration file `locations` contains a list of road section locations (see Section 1.1), while other locations are determined using network layout algorithm. Example of `locations` file can be seen in Listing 5, while detailed description is given in Table 6.

Listing 5: Example configuration file `locations`.

```
1 Avstrija V      0.6963  0.1585
2 Avstrija Z      0.2041  0.2805
3 Vrtojba         0.0879  0.6296
4 Plavje          0.1387  0.8399
5 ...
```

Index	Attribute	Value	Example	Description
1	*location*	String	*Avstrija V*	Section location identification as unique string.
2	x	$[0, 1]$	0.2401	Location x coordinate in a unit plane.
3	y	$[0, 1]$	0.2805	Location y coordinate in a unit plane.

Table 6: Description of configuration file `locations`.

Comments.

location Section location must be one of the locations defined in file `roads`.

x Location x coordinate is horizontal location on the screen ($x = 0$ and $x = 1$ corresponds to left-most and right-most part of the screen, respectively).

y Location x coordinate is vertical location on the screen ($y = 0$ and $y = 1$ corresponds to top and bottom of the screen, respectively).

3.3 System output

Standard traffic simulator output is written to file `stdout`. Furthermore, as defined in file `parameters.xml` (parameters *ticketing* and *logging*), traffic tickets are sent to designated ticketing server (parameters *ticketingServer* and *ticketingService*), while the complete state of all vehicles is also logged to folder `logs` with respect to some time step (parameter *loggingRange*). For further details see Section 3.1 and Section 3.2.

3.3.1 File `stdout`

Most error messages are reported through user dialogs windows (see Section 3.4). However, error messages that result from computer architecture or operating system limitations (e.g., lack of memory resources or exceeded maximum number of concurrent threads) are redirected to file `stdout` (see Section 3.1). In the case of abnormal behavior of traffic simulator application, file `stdout` should be considered for possible error messages! Most of such problems can be eliminated by simply adjusting traffic simulator parameters in file `parameters.xml` appropriately (see Section 3.2).

3.3.2 Ticketing

Description. Traffic tickets are sent in JSON format to designated ticketing server for vehicles at the end of some section (see Section 3.2). Note that only tickets for ticketed vehicles and sections are actually sent, whereas server response can also be checked for possible errors in transmission (see Section 3.2). Example of a ticket can be seen in Listing 6, while detailed description is given in Table 7.

Listing 6: Example tickets in JSON format.

```
1 {"ticketA":[1,"00000",1,12345,32352,1930,1,10800000,10800000,0,1702,0,2,0,0,""]}
2 {"ticketA":[2,"00000",1,12345,83556,1964,1,10801000,10801000,0,1908,0,2,0,0,""]}
3 {"ticketA":[3,"00000",1,12345,13012,1924,1,10801000,10801000,0,1809,0,2,0,0,""]}
4 {"ticketA":[4,"00000",1,12345,61707,1913,1,10801000,10801000,0,1830,0,2,0,0,""]}
5 ...
```

Index	Attribute	Value	Example	Description
1	*ticket*	≥ 1	1	Sequential ticket identification number.
2	*obu*	00000	00000	OBU identification number (irrelevant).
3	*version*	1	1	Context version number (irrelevant).
4	*serial*	12345	12345	Ticket serial number (irrelevant).
5	*vehicle*	≥ 1	32352	Vehicle identification number.
6	*section*	≥ 1	1930	Current road section ETS code.
7	*context*	1	1	Context identification number (irrelevant).
8	*time*	≥ 0 ms	10800000 ms	Current vehicle simulation time.
9	*start*	≥ 0 ms	10800000 ms	Vehicle simulation start time (equal to *time*).
10	*enter*	≥ 1	0	Entrance road section ETS code (0 for missing).
11	*exit*	≥ 1	1702	Exit road section ETS code (0 for missing).
12	*units*	0	0	Vehicle unit count (irrelevant).
13	*axles*	≥ 0	2	Vehicle number of axles.
14	*prop1*	0	0	First additional property (irrelevant).
15	*prop2*	0	0	Second additional property (irrelevant).
16	*extra*	String	...	Vehicle ticket extra string.

Table 7: Description of tickets in JSON format.

Comments.

section Current road section is given as section ETS code (see Section 1.1).

enter Previous road section is given as section ETS code, or 0 for vehicle that has no previous section (see Section 1.1).

exit Next road section is given as section ETS code, or 0 for vehicle leaving road network (see Section 1.1).

extra Ticket extra string is identical for all vehicles (see Section 3.2).

3.3.3 Logging

Description. Log file *log-x.tab* contains a complete state of all vehicles throughout the traffic simulation (see Section 1.2), where parameter x is a sequential log number. Vehicles are logged at predefined time steps (see Section 3.2) and also at the end of each section (when tickets are sent). Example of *log-x.tab* file can be seen in Listing 7, while detailed description is given in Table 8.

Listing 7: Example log file *log-x.tab*.

```
1 10800  78697  10800  CAR    126.0  T55      0.0050  1  A2-600  null  null  0.081
2 10800  34672  10800  TRUCK  126.0  A2-14    0.4     1  null    A2-15 null  0.0
3 10815  3649   10800  CAR    108.0  A1-646   2.715   1  null    A1-645 null 0.0
4 10815  35534  10800  BUS    126.0  A4-3691  2.18    1  null    T141  null  0.0
5 ...
```

Index	Attribute	Value	Example	Description
1	*time*	≥ 0 s	10815 s	Current vehicle simulation time.
2	*vehicle*	> 0	35534	Vehicle identification number.
3	*start*	≥ 0 s	10800 s	Vehicle simulation start time.
4	*type*	`CAR`, `TRUCK` etc.	`BUS`	Type of the vehicle.
5	*speed*	≥ 0.0 km/h	126.0 km/h	Current vehicle speed.
6	*section*	String or *null*	*A4-3691*	Current vehicle road section (*null* for pending).
7	*position*	≥ 0.0 km	2.18 km	Current vehicle position within the section.
8	*lane*	≥ 1 or -1	1	Current vehicle section lane (-1 for pending).
9	*previous*	String or *null*	*null*	Previous vehicle road section (*null* for missing).
10	*next*	String or *null*	*null*	Next vehicle road section (*null* for leaving).
11	*destination*	String or *null*	*null*	Vehicle destination location (*null* for standard).
12	*assessment*	≥ 0.0 €	0.0 €	Current vehicle assessment value.

Table 8: Description of log file *log-x.tab*.

Comments.

section Current road section is given as road and section identification numbers, or *null* for pending vehicle that is yet not within the road network (see Section 1.1).

position Vehicle position is measured from the beginning of road section (see Section 1.1).

previous Previous road section is given as road and section identification numbers, or *null* for vehicle that has no previous section (see Section 1.1).

next Next road section is given as road and section identification numbers, or *null* for vehicle leaving the road network (see Section 1.1).

destination Designated vehicle destination location or *null* for standard vehicle (see Section 1.2).

assessment Vehicle assessment value is only reported at the end of each section (see Section 3.2) and corresponds to the assessment for the entire section (see Section 1.2).

3.4 Graphical user interface

Graphical user interface consists of traffic configuration and traffic simulation windows (see Section 3.2 and Figure 5, respectively). The former allows basic configuration of simulation parameters, while the latter shows the state of traffic simulation and allows basic control over simulation.

3.4.1 Traffic simulation configuration window

Description. Traffic configuration window allows basic configuration of traffic simulation parameters (see Section 3.2). Print-screen of the configuration window can be seen in Figure 4, while description of different control components is given in Table 9.

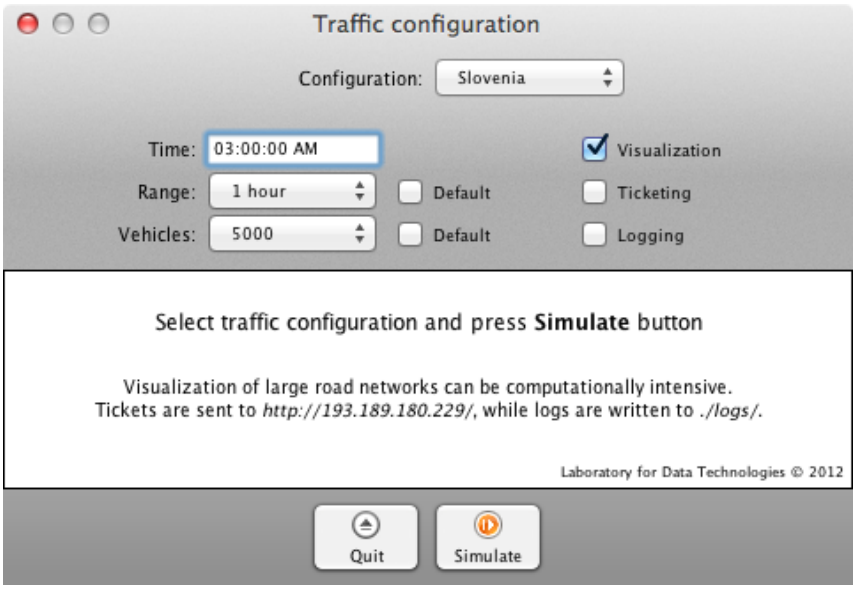


Figure 4: Traffic simulation configuration window on OS X.

Component	Parameter	Description
Configuration	*configuration*	Traffic simulation configuration.
Time	*time*	Initial traffic simulation time.
Range	*timeRange*	Traffic simulation time range.
Vehicles	*vehicles*	Initial vehicles number.
Visualization	*visualization*	Traffic simulation visualization.
Ticketing	*ticketing*	Traffic simulation ticketing.
Logging	*logging*	Traffic simulation logging.
Quit	/	Quits the traffic simulation.
Simulate	/	Starts the traffic simulation.

Table 9: Description of traffic simulation configuration window controls (from left to right, top to bottom). Field parameter refers to a corresponding parameter in the configuration file `parameters.xml` (see Section 3.2).

Comments.

Range *Default* refers to a configuration file parameter value (see Section 3.2).

Vehicles *Default* refers to a configuration file parameter value (see Section 3.2).

Visualization Whether traffic simulation is visualized as a road network.

Ticketing Whether vehicle tickets are sent to server during traffic simulation.

Logging Whether vehicle states are logged to file during traffic simulation.

3.4.2 Traffic simulation visualization window

Description. Traffic visualization window shows the current state of traffic simulation and allows basic control over the execution (see Section 3.2). Print-screen of the configuration window can be seen in Figure 5, while description of different control components is given in Table 10.

Traffic visualization window is partitioned into three parts. Top part allows basic control over traffic simulation, middle part visualizes traffic simulation as a road network and bottom part shows some basic statistics for the simulation.

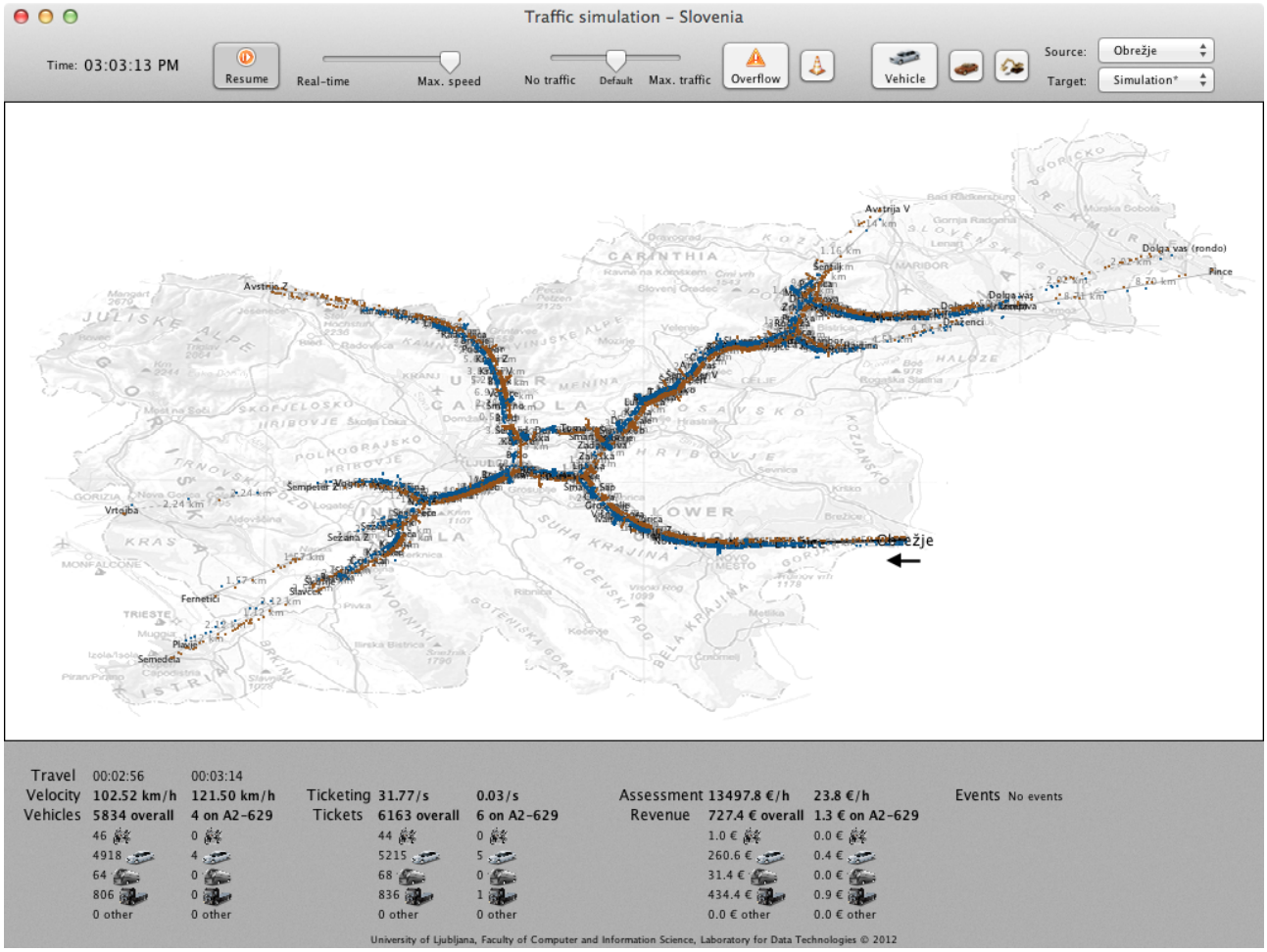


Figure 5: Traffic simulation visualization window on OS X.

Component	Description
Time	Current traffic simulation time.
Start/Stop/Resume	Starts, stops or resumes traffic simulation.
Simulation speed	Controls traffic simulation speed.
Vehicle inflow	Controls traffic simulation vehicles inflow.
Vehicle overflow	Adds overflow of vehicles on a selected road section.
Vehicle collision	Adds vehicle collision on a selected road section.
Vehicle	Adds designated vehicle on a selected road section.
Fast vehicle	Adds fast designated vehicle on a selected road section.
Slow vehicle	Adds slow designated vehicle on a selected road section.
Source	Selects source section location for special events.
Target	Selects destination section location for special events.

Table 10: Description of traffic simulation visualization window controls (from left to right).

Comments.

Vehicle inflow When slider as in the middle, vehicles are added according to section traffic loads (see Section 3.2). When slider is in the left-most position, no new vehicles are added, while, when slider is in the right-most position, a new vehicle is added to each section at each time step (see Section 1.2).

Vehicle overflow Overflow of vehicles adds a new vehicle on each lane of the selected road section at each time step. Selected road section is defined through *Source* and *Target* parameters.

Vehicle collision Vehicle collision adds a collision over all lanes of the selected road section. Selected road section is defined through *Source* and *Target* parameters.

Vehicle Adds designated vehicle that adopts road section speed limits (see Section 3.2). Selected road section is defined through *Source* and *Target* parameters.

Fast vehicle Adds designated vehicle that adopts fast vehicle speed limit (see Section 3.2). Selected road section is defined through *Source* and *Target* parameters.

Slow vehicle Adds designated vehicle that adopts slow vehicle speed limit (see Section 3.2). Selected road section is defined through *Source* and *Target* parameters.

Source Source section location for special events. The latter can be selected also by clicking on road network vertex and pressing key 'S'.

Target Destination section location for special events. The latter can be selected also by clicking on road network vertex and pressing key 'D'.

References

De Gier, J., Garoni, T. M., & Rojas, O. (2011). Traffic flow on realistic road networks with adaptive traffic lights. *Journal of Statistical Mechanics: Theory and Experiment*, *2011*(04), P04008.

Nagel, K., & Schreckenberg, M. (1992). A cellular automaton model for freeway traffic. *Journal de Physique*, *2*(12), 2221–2229.