

ov2txt
User's manual

Miguel Hernandez University ¹

April 19, 2011

¹Copyright (c) 2008 P. Pablo Garrido Abenza. All rights reserved.



Abstract

In this manual the installation and usage of the *ov2txt* program is described, which is included within the *simul_XXXset* of programs. It has been developed by P. Pablo Garrido Abenza (pgarrido@umh.es), as a member of the Arquitectura y Tecnología de COMPUTADORES Group (GATCOM) at the Miguel Hernandez University.

Contents

1	Introduction	3
2	Installation	5
2.1	Requeriments	5
2.2	Windows	5
2.3	Linux	6
3	Usage of the program	7
3.1	Syntax	7
3.2	Description	7
3.3	Files	9
3.3.1	Input files	9
3.3.2	Output files	9
3.4	Examples	10

Chapter 1

Introduction

When OPNET Modeler [1] run a simulation, results are written to two kind of files: vectorial data (.ov) and scalar data (.os). These files have a binary format, and the typical way for extracting the data in order to be processed by external tools is by using the mouse within the OPNET Graphical User Interface. This is a tedious task, specially when there are a great number of statistics and simulations.

This tool is intended for extracting data from the binary vectorial file (.ov) generated by OPNET Modeler as a result of every simulation run, without any user interaction with the OPNET graphical interface. The data are written to a column separated text file, suitable for being processed by R [2] or MS-Excel. In addition, the program can compute scalar data from the vectorial data processed, without process the .os file. Finally, this program can be executed in a cluster environment like Condor [3].

In this way, it is possible to automate the run of a set of simulations and extract the corresponding data with a simple script, invoking first to *op_runsim* for running the simulation, and then to *ov2txt* for extracting the data.

The set of programs *simul_xxx*, contains the following command-line tools:

- *simul_geoloc*: used for generating network scenarios automatically, computing the partition grade according to the transmission range.
- *simul_ah2r*: used for converting the binary animation file (.ah) to a text format (.as).
- *simul_run*: used for program sets of simulations with OPNET Modeler without using the Graphical User Interface.
- *simul_data*: used for extracting the data obtained with the simulations. This program calls to *ov2txt* for each of the simulations run.
- *ov2txt*: invoked from the *simul_data* for extracting data from the binary vectorial files (.ov). It can also be invoked for the user for extracting data for a particular simulation (as described in this manual).
- *simul_graphs*: used for generating graphs from the extracted data (R or gnuplot).

- *simul_reports*: used for generating a report containing the previously generated data and graph.

These programs have been developed using the C programming language, so it is easy to migrate them to any platform; currently Windows and Unix-like systems as Linux.

Chapter 2

Installation

For the installation of the *ov2txt* program, you only have to copy the binary to any directory and set some environment variables (`PATH` and `LD_LIBRARY_PATH`). In the following sections, how to set these environment variables is explained.

2.1 Requeriments

The program *ov2txt* requires that OPNET Modeler be installed, as it makes use of the API External Model Access (EMA). Moreover, the OPNET Modeler version installed should match with the one used for compiling this program, which is specified in the name of the executable (e.g. *ov2txt-...-16.0.A.bin*).

2.2 Windows

Let us suppose that the directory choosen for copying the *ov2txt* program is:

```
C:\Program files\utils
```

It is recommended to modify the `PATH` environment path. This can be do as explained in the following:

- Windows 98 or previous: edit the `C:\AUTOEXEC.BAT` file, adding the following line at the end. It would be possible that the path needs to be enclosed within quotes if it is more than 8 characters length or it contains blanck spaces:

```
SET PATH=%PATH%;C:$\Program files\utils
```

- Windows NT, 2000, XP, ...: within the Control Panel, choose System > Advanced > press the button [Environment variables]. Then the path could be added to the `PATH` environment variable, separated by a semicolon ';'. Alternatively, the `Autoexec.NT` file can be modified in a similar way as explained for the `Autoexec.bat` file.

2.3 Linux

Let us suppose that the directory choosen for copying the *ov2txt* program is:

```
$HOME/bin
```

Under Linux it is necessary to set the `PATH` and `LD_LIBRARY_PATH` environment variables. In the later we should to add the path for the OPNET libraries, as it is necessary in order to run programs compiled with the API EMA (built with `op_mkema`). This path depends on the home directory for OPNET and version. The way to do that depends on the shell or command-line interpreter used (bash, ksh, ...):

- Linux (shells: bash, ksh, zsh, sh): `$HOME/.bash_profile` or `$HOME/.profile`:

```
PATH=.:$PATH:$HOME/bin
LD_LIBRARY_PATH=/usr/opnet/16.0.A/sys/pc_intel_linux/lib
export PATH
export LD_LIBRARY_PATH
```

- Linux (shells: csh, tcsh): `$HOME/.login`:

```
setenv PATH .:$PATH:$HOME/bin
setenv LD_LIBRARY_PATH /usr/opnet/16.0.A/sys/pc_intel_linux/lib
```

Some Linux distributions have problems with the `LD_LIBRARY_PATH`, and user cannot set the value. This is why it is necessary to execute that each time that either you are going to execute the program or you open the console.

In order to simplify the process, you can write a little script called `ov2txt.sh` similar to the following, and put it together to the binary. In order to pass the arguments from the script to the executable, you can use `"$@"` (with the quotes) instead of `$*` too.

```
#!/bin/sh
export LD_LIBRARY_PATH=/usr/opnet/16.0.A/sys/pc_intel_linux/lib
exec ov2txt-en-16.0.A-x86.bin "$@"
```

Chapter 3

Usage of the program

In this chapter, the usage of the *ov2txt* program is explained. It is supposed that we have written the *ov2txt.sh* script explained in the previous chapter.

3.1 Syntax

```
ov2txt.sh -?  
ov2txt.sh -m input-file [-i input-path] [-o output-path] [-w working-path]  
                    [-v] [-q] [-x data-to-export] [-f output-format] [-s suffix]
```

3.2 Description

ov2txt is used to extract data from the *.ov* binary files. It extracts vectorial data, that is, complete columns corresponding to the selected statistic variables. Moreover, it also can compute scalar values from the vectorial data, namely: last value, arithmetic mean, average or temporal mean, variance (sample), minimum value, maximum value, and sum.

The program does not run interactively. Instead, the program begins to extract data immediately upon invocation and goes on until the process finishes, without interruption. Because of this, it is necessary to specify some options when invoking the program. These are the options to *ov2txt*:

-?: shows this help.

-v: verbose mode.

-q: force to generating the list of statistics out file (*.std*).
It is also generated automatically when no data is selected to be exported. By default, the *.std* file is not generated when exporting data (-x option) in order to reduce the number of files generated, unless the -q option be used.

-i <dir>: input directory for the *.ov* file.
If this option is not used, the *.ov* file will be searched in the paths specified in the 'mod_dirs' OPNET preference, in the

same order. The use of the `-i` option has the advantage that the `.ov` processed will be the desired one, as it could exist a `.ov` file with the same name but in different path previously defined in the `'mod_dirs'` preference.

`-o <dir>`: output directory.

Is the destination directory for output files (`.std` and `.data`). This directory can be relative to the current directory (e.g. `./data`) or absolute (e.g. `/home/user/op_models/data`). If this option is not used, the current path or the working directory (specified with the `-w` option) will be considered. The options `-o` and `-w` are equivalents (i.e. both produce the same results). The difference between them is that `-w` change temporarily the directory, and with `-o`, the path is used when opening the files.

`-w <dir>`: change the working directory.

Is the destination directory for output files (`.std` and `.data`). This directory can be relative to the current directory (e.g. `./data`) or absolute (e.g. `/home/user/op_models/data`). By default, the working directory is the current directory from the program `ov2txt` has been executed. It is equivalent to specify the output directory (`-o` option). The `-w` option is necessary when executing `ov2txt` from a cluster.

`-s <suffix>`: suffix to be appended to the output file names

`-f <format>`: format for the output data file, where `<format>` could be:

- `-f e[h]`: Excel.
- `-f r[h]`: R package (default).
- `-f c[h]`: to be read from a C program.

If the `'h'` is specified, a header with the name of the variables will be exported too (first line for vectorial variables and first column for scalar data).

The default format is `'-f rh'`, that is, R and with headers.

`-x <data>`: statistic variables to be exported to the data file.

Data will be written in the format specified with `-f` (R by default).

The statistic variables can be specified by mean of 3 different ways:

[Method 1]: by a indexes list (`>=0`).

> Example: `-x 0,14,17,20`

> The corresponding index to each variable can be obtained from the statistics output file by executing once the program specifying only the input file (`-m <file>`) without any other option (also with `-q`).

> The index 0 is the simulation time (abscissa axis (x) in graphs)

> In addition, you can use several functions that allow to obtain scalar data similar to the `.os` files generated by OPNET, but computed by the `ov2txt` program from the vectorial data (`.ov`):

- `LAST(n)` --> Last value column '`n`'
- `MEAN(n)` --> Arithmetic mean column '`n`'
- `AVG(n)` --> Temporal mean column '`n`'
- `VAR(n)` --> Variance (sample) column '`n`'
- `MIN(n)` --> Minimum value column '`n`'
- `MAX(n)` --> Maximum value column '`n`'
- `SUM(n)` --> Sum column '`n`'
- `ALL(n)` --> All scalar data column '`n`'

> Example: `-x 0,14,17,20,MEAN(14),MAX(17)`

[Method 2]: by a file containing the index of each selected variable (one per line). This method is very similar to the previous one. This could be useful when the statistics to be exported are determined with other program, which could write the indexes to a text file.

The way to specify this method is with the 'K' character preceding the file name specified.

> Example: -x Kvars.txt

The 'vars.txt' file contains the indexes of the statistics to be exported, one per line.

> With this method, the previous functions can also be used.

[Method 3]: by a file containing the name completely referenced of the selected variables (one per line). The names can be obtained from the output file containing the listed statistics or from the graphical interface within OPNET Modeler.

The way to specify this method is with the '@' character preceding the file name specified.

> Example: -x @vars.txt

The 'vars.txt' file contains the names of the statistics to be exported, one per line.

3.3 Files

3.3.1 Input files

- The input-file is the name of an OPNET generated .ov file (mandatory).
- The file name should be specified without the extension '.ov' at the end.
- If it contains blank spaces, it should be specified within quotation marks.
- The absolute path should be included in `mod_dirs` preference within OPNET. The .ov file will be searched by following the same order as in `mod_dirs`. It is possible to specify a path for the specific .ov to consider (-i) which sets (temporarily) the `mod_dirs` preference to that path.

3.3.2 Output files

- Two text files are generated in the working path:
 1. Indexed list of all variables (statistics) included in the input .ov file:
file.ov \Rightarrow file.std
 2. Selected data extracted from the .ov file:
file.ov \Rightarrow file.xxx
where 'xxx' = suffix according to the selected output format (-f)
Excel = xls; R = data; C = csv
- If an optional suffix is specified with the -s option, it will appear at the end of both output file names (e.g. <file (sss).std>).

3.4 Examples

The following examples show how one can extract data from the .ov files. Let us suppose that the input file is 'mymodel.ov'. Usually, the first thing one should do is to know the name of the statistics variables that the .ov file contains. So, the following command should be issued:

```
ov2txt.sh -m mymodel
```

Please, note that the .ov suffix should not be specified at the end, because we are not specifying the file name but the corresponding name model. By default, the final .ov file used is determined by searching through the paths listed in the `mod_dirs` preference within OPNET Modeler, exactly in that order. However, you can force the file to be used by using the `-i` option, in order to specify the unique directory where to look for the .ov file. This directory can be either an absolute or relative path to the working directory. Let us suppose that you have run two sets of simulations with the same network model but with different parameters each one, and the resulting .ov files are in the directories `./result/v1` and `./result/v2`, respectively. So, there should exist two files exactly with the same name but in different directories. If we want to extract data from the last simulation (`./result/v2`), we can modify the `mod_dirs` preference, or specify the `-i` option:

```
ov2txt.sh -m mymodel -i ./result/v2
```

By executing the program without the `-x` option only a text file is generated containing the name of all the statistics variables included in the 'mymodel.ov' file, but no data are extracted. In this way, we can take a look to that file and choose the statistics to export either by index or by name. Then, we can execute a second time in order to extract data by specifying the `-x` option. For example:

```
ov2txt.sh -m mymodel -x 0,1,2,3
ov2txt.sh -m mymodel -x @names.txt
```

In the first example we are using the indexes of the statistics as shown in the previously generated text file. In the second one, we are using a text file called 'names.txt' with the full referenced names of the statistics (one per line), extracted from the same previously generated text file or from the OPNET Modeler graphical interface. The second method is used when the character '@' is found at the beginning of the file name, next to the `-x` option. In both cases, the same 'mymodel.data' file is generated. The `-q` option causes to force the statistics test file be generated too, which is rarely useful. The file 'names.txt' is a plain text file that could be similar to the following:

```
time (sec.)
IP.Number of Hops
Wireless LAN.Load APP (bits/sec)
Wireless LAN.Throughput APP (bits/sec)
Wireless LAN.Delay (sec)
Wireless LAN.Data Dropped (Buffer Overflow) (bits/sec)
```

By default, the output file(s) is written in the current directory, that is, the directory from the *ov2txt* program was invoked. The output directory can be changed with either the `-o` or `-w` options. Both are equivalent although they work in a different way. So, the following commands are similar in practice:

```
ov2txt.sh -m mymodel -x 0,1,2,3 -i ./result/v2 -o ./data/v2
ov2txt.sh -m mymodel -x 0,1,2,3 -i ./result/v2 -w ./data/v2
```

In this way, we can export the result data `v1` and `v2` to different directories. Another option could be to adding a suffix to the end of the output file name with the `-s` option:

```
ov2txt.sh -m mymodel -x 0,1,2,3 -i ./result/v1 -s v1
ov2txt.sh -m mymodel -x 0,1,2,3 -i ./result/v2 -s v2
```

In these cases, all the generated data files will be written in the same directory, but the names will be different: `'mymodel (v1).data'` and `'mymodel (v2).data'`, respectively.

By default, the format of the output file is suitable for the R package (`.data`). That can be changed with the `-f` option. For example, if we are going to process the data with MS-Excel (`.xls`) we can execute the following command:

```
ov2txt.sh -m mymodel -x 0,1,2,3 -f eh
```

Now, the generated output data file will be `'mymodel.xls'`, and will include a header as a first line.

Bibliography

- [1] I. OPNET Technologies, "Opnet modeler." [Online]. Available: <http://www.opnet.com/>
- [2] R Development Core Team, "R: A language and environment for statistical computing," 2007, ISBN 3-900051-07-0. [Online]. Available: <http://www.R-project.org>
- [3] T. C. Team, "Condor." [Online]. Available: <http://www.cs.wisc.edu/condor/>