TraceMetrics

Guide

Tracemetrics Tutorial

Content

1 Introduction

The Tracemetrics tool is a trace file analyzer for the *ns-3* simulator. The purpose of this tool is to perform a practical analysis of the trace file produced by a simulation and, as result, show some interesting metrics on performance of the simulated scenario.

The goal of this tutorial is to introduce the tool and present the ways to perform the analysis and interpret the results.

2
Getting Started

2.1 Downloading Tracemetrics

In order to download the Tracemetrics you can access the project's SourceForge page: http://sourceforge.net/projects/tracemetrics/

Take some time to check the release sections news and supported protocols at the Tracemetrics website to see the new features and to see if the tool already support the protocols you will use.

To get the tool you have two options: get the JAR package or to get the source code.

Make sure you have the JRE or JDK version 5 or later.

2.1.1 Getting the JAR package

This way is the most simple. Just access the following link: http://sourceforge.net/projects/tracemetrics/ and get the latest version. There is also the MD5 checksum so you can verify the file integrity.

2.1.2 Getting the source code

Just access http://sourceforge.net/projects/tracemetrics/files/sourcecode/ and download the tarball.

2.1.2.1 Compiling the source code

The tool has developed using the Netbeans IDE and it uses the Ant tool to compile the code.

In the case, you do not want to use Netbeans IDE you can just use the Ant tool. Enter in the directory created after unzipping and type:

```
$ ant build.xml
```

3
Running Tracemetrics

3.1 Running the JAR

In order to run the JAR, enter the *dist* directory. Here you can find the JAR file. Now type:

```
$ java -jar TraceMetrics.jar&
```

The splash screen of figure 1 will be displayed, and the program will be loaded.

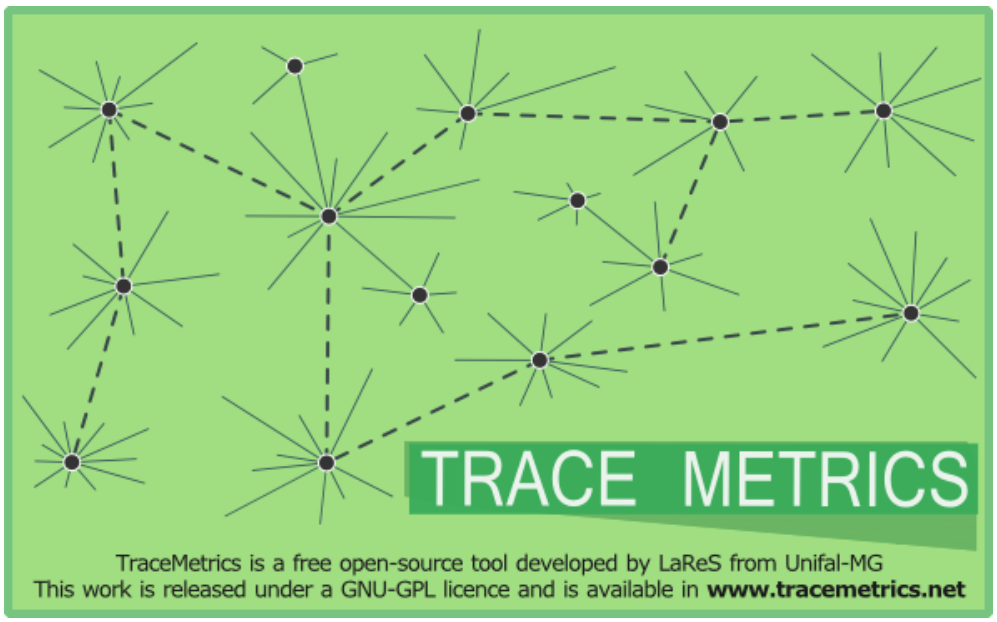

Figure 1: Splash screen of Tracemetrics

3.2 Command line options

3.2.1 Nimbus

In case you prefer a more beautiful GUI you can add the argument:

-Dswing.defaultlaf=javax.swing.plaf.nimbus.NimbusLookAndFeel

3.2.2 More memory

Sometimes the output trace is large and analyzing it will use more memory space. If you get the error*java.lang.OutOfMemoryError: Java heap space* you should consider using this option. You will enter with any number that you want. You should not pass more than the available memory.

-Xmx4096m

This option will set the maximum size of 4096MB or 4GB.

3.2.3 No splash screen at start-up

You can run the jar with option *--no-splash*or *--ns* to ignore splash screen and go directly to program.

4
Analyzing a trace

4.1 Choosing the trace file

After the main screen appear, you can choose the trace file by clicking on File > Choose file or you can press the shortcut key F2. This menu can be seen on figure 2.

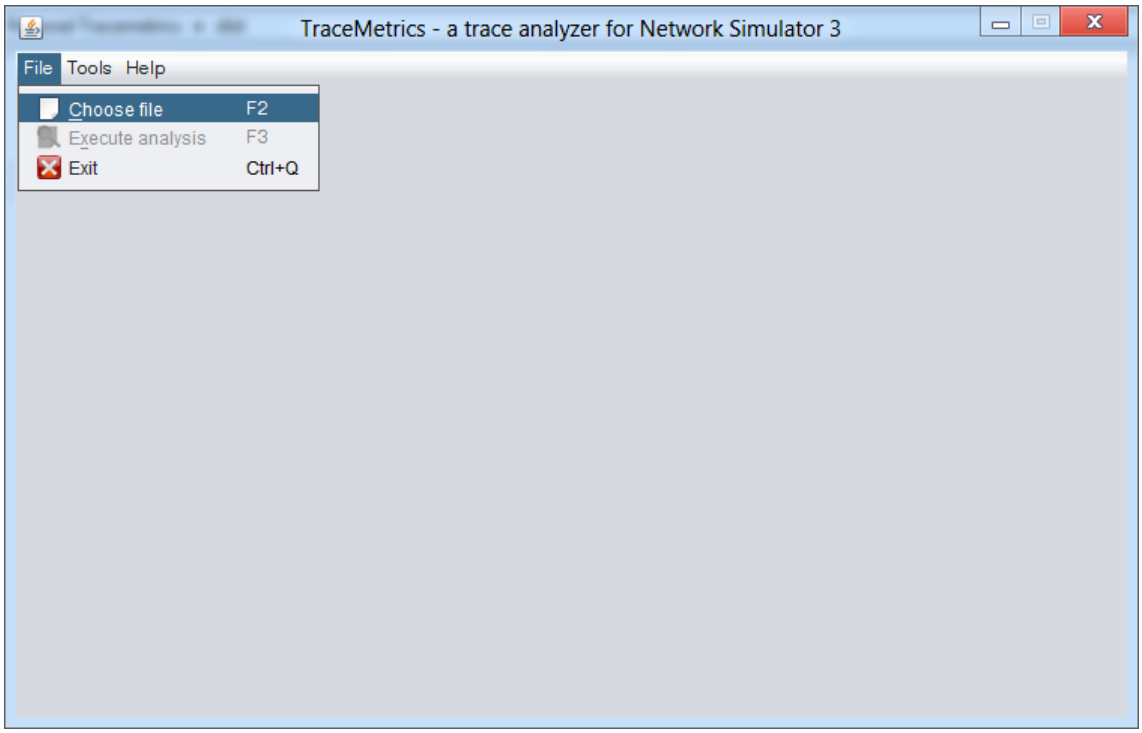

Figure 2: Choose file option

The window to choose the file will open in the execution path. You can freely go through the directories. After choosing the file, just click on the Open button. The choose window will close. The chooser windows will look like figure 3.

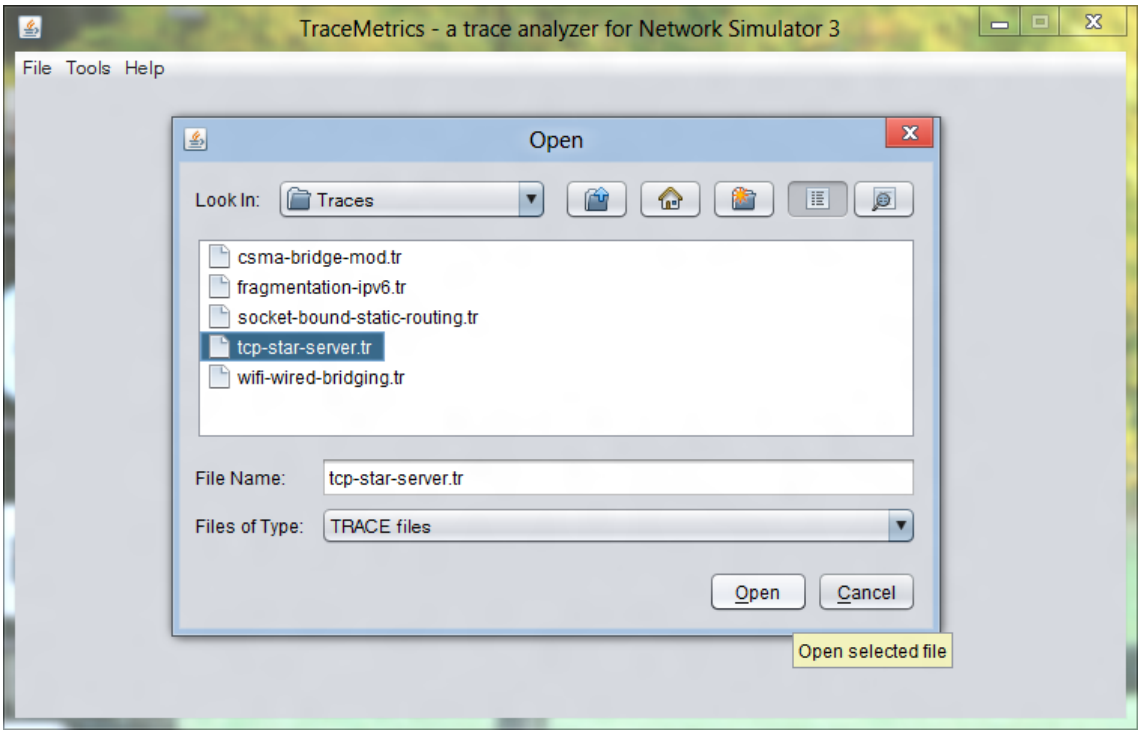

Figure 3: Choose file window

After open a file, the screen will show the file details, like figure 4a. If you already performed a previous analysis, an additional button you will be shown, giving to you the possibility to back to previously performed analysis, like figure 4b.

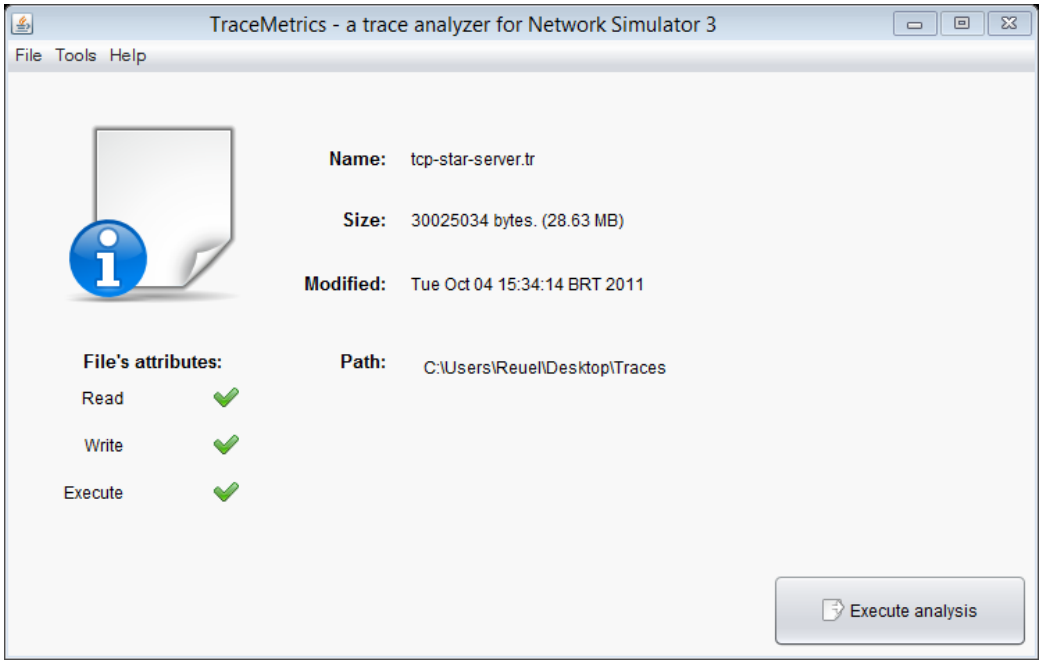

Figure 4a: File details window

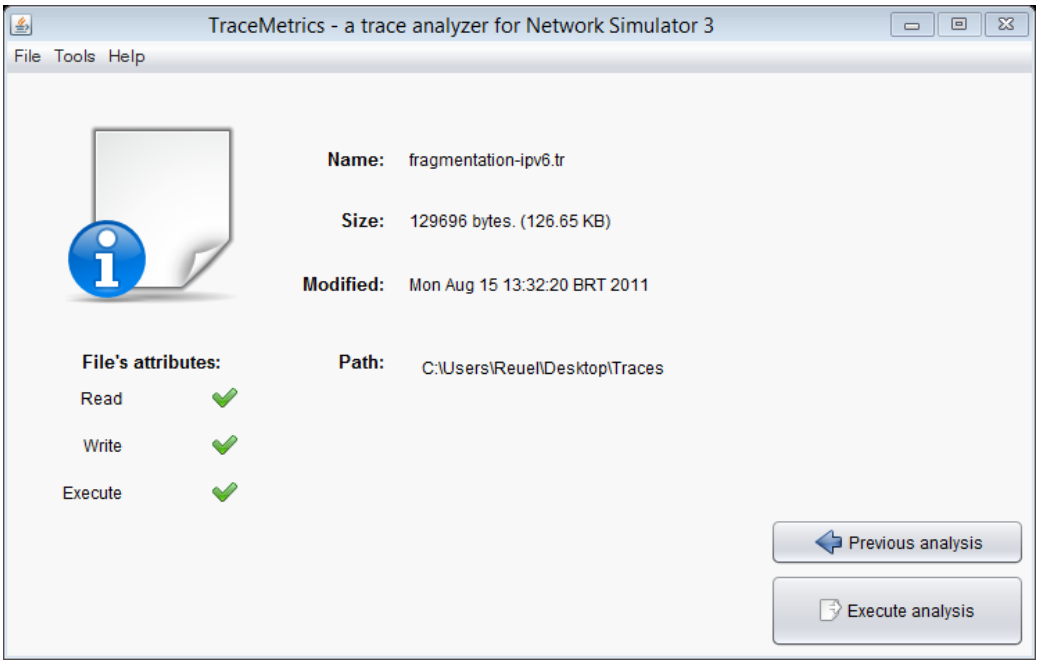

Figure 4b: File details window with previous button

4.2 Analyzing the trace file

After you have chosen the file, start the analysis through *File* >*Execute analysis* or press the shortcut key F3, or the button *Execute analysis*. A progress bar should appear, like figure 5. If it does not appear, something wrong happened. You can stop the analysis at any moment; the partial result will be shown, but the integrity of analysis is not guaranteed.

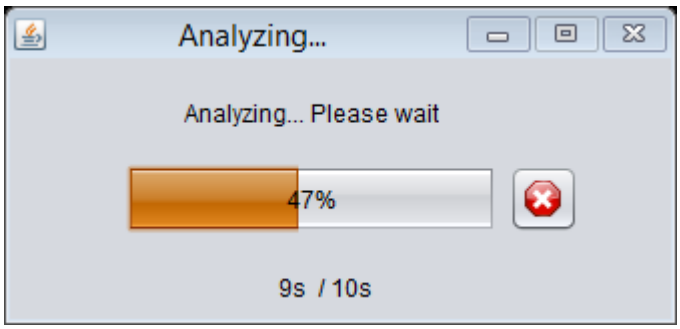

Figure 5: progress bar window

4.3 Tracefile format

Tracemetrics let us make some various types of files, and tracemetrics is be designed for some one specific output file.

On a .tr file, will be found some structure of data as follows in the next text box.

If some alterations that generate an invalid tracefile, certain will be catch by tracemetrics and will not be possible to continue on analysis.

If some new protocol supported by ns3 but not by tracemetrics are used for simulations, tracemetrics too will not accomplish analysis.

```
+ 1.00819 /NodeList/0/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
- 1.00819 /NodeList/0/DeviceList/0/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
+ 1.01029 /NodeList/4/DeviceList/1/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
- 1.01029 /NodeList/4/DeviceList/1/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
+ 1.01029 /NodeList/4/DeviceList/2/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
- 1.01029 /NodeList/4/DeviceList/2/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
+ 1.01029 /NodeList/4/DeviceList/3/$ns3::CsmaNetDevice/TxQueue/Enqueue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
- 1.01029 /NodeList/4/DeviceList/3/$ns3::CsmaNetDevice/TxQueue/Dequeue ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
r 1.01029 /NodeList/4/DeviceList/0/$ns3::CsmaNetDevice/MacRx ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
r 1.0124 /NodeList/1/DeviceList/0/$ns3::CsmaNetDevice/MacRx ns3::EthernetHeader ( length/type=0x806, source=00:00:00:00:00:01, destination=ff:ff:ff:ff:ff:ff) ns3::ArpHeader (request source mac: 00-06-00:00:00:00:00:01 source ipv4: 10.1.1.1 dest ipv4: 10.1.1.2) Payload (size=18) ns3::EthernetTrailer (fcs=0)
```

Format of a valid tracefile for tracemetrics.

4.4 Supported protocols

ArpHeader
PppHeader
TcpHeader
Icmpv6RS
Icmpv6RA
Icmpv6NS
Icmpv6NA
Icmpv6OptionLinkLayerAddress
Ipv6ExtensionLooseRoutingHeader
Icmpv6Echo
Ipv6Header
Ipv6ExtensionFragmentHeader
LlcSnapHeader
Icmpv4Header
Icmpv4DestinationUnreachable
WifiMacTrailer
AmsduSubframeHeader
Ipv4L3Protocol
WifiMacHeader
WifiMacHeaderMGT
WifiMacHeaderCTL_RTS
WifiMacHeaderCTL_CTS
WifiMacHeaderCTL_ACK
WifiMacHeaderCTL_BACKREQ
WifiMacHeaderCTL_BACKRESP
WifiMacHeaderDATA
WifiMacHeaderQOSDATA
WifiActionHeader
MgtAddBaRequestHeader
CtrlBAckRequestHeader
CtrlBAckResponseHeader
MgtProbeResponseHeader
MgtAssocRequestHeader
MgtAssocResponseHeader
MgtProbeRequestHeader

5
Analyzing the output

After having finished the analysis, the tabs will be filled with the results. This section will go through the tabs and explain the data shown.

5.1 The Simulation tab

This tab shows the general details about the analysis, such as the absolute path to the analyzed file, number of lines on the trace file, packet counters, simulation time and analysis time. An example is represented in Figure 6.

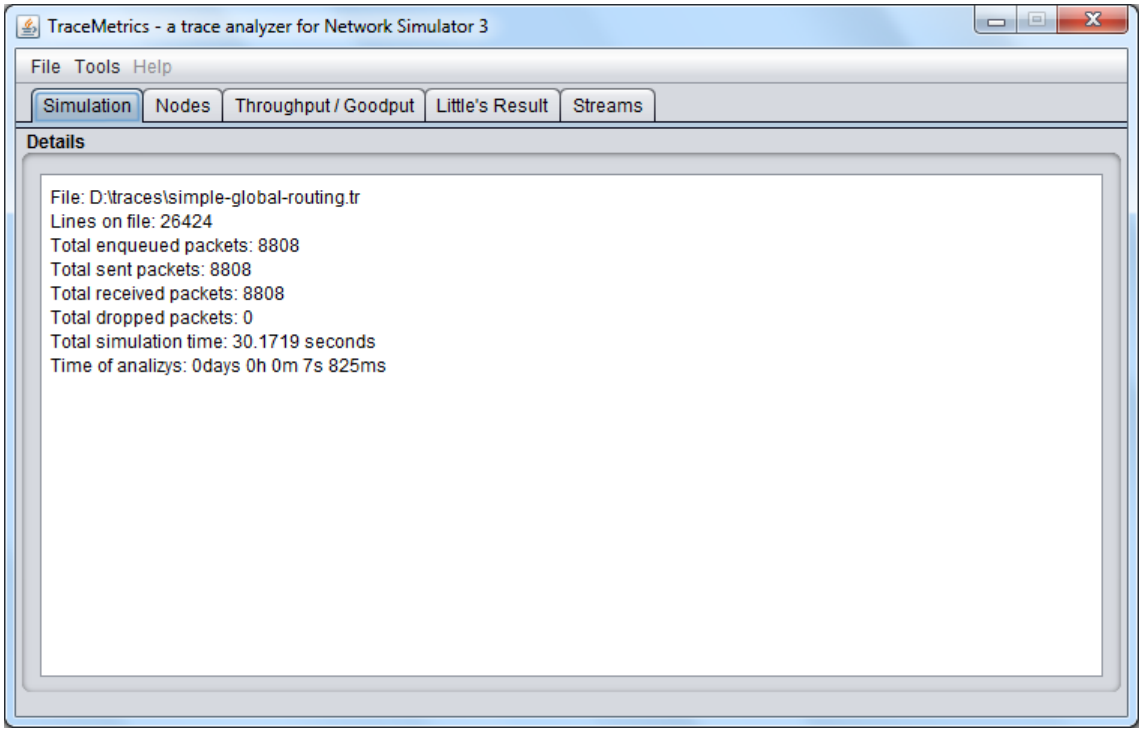

Figure 6: The Simulation tab

5.2 The Nodes tab

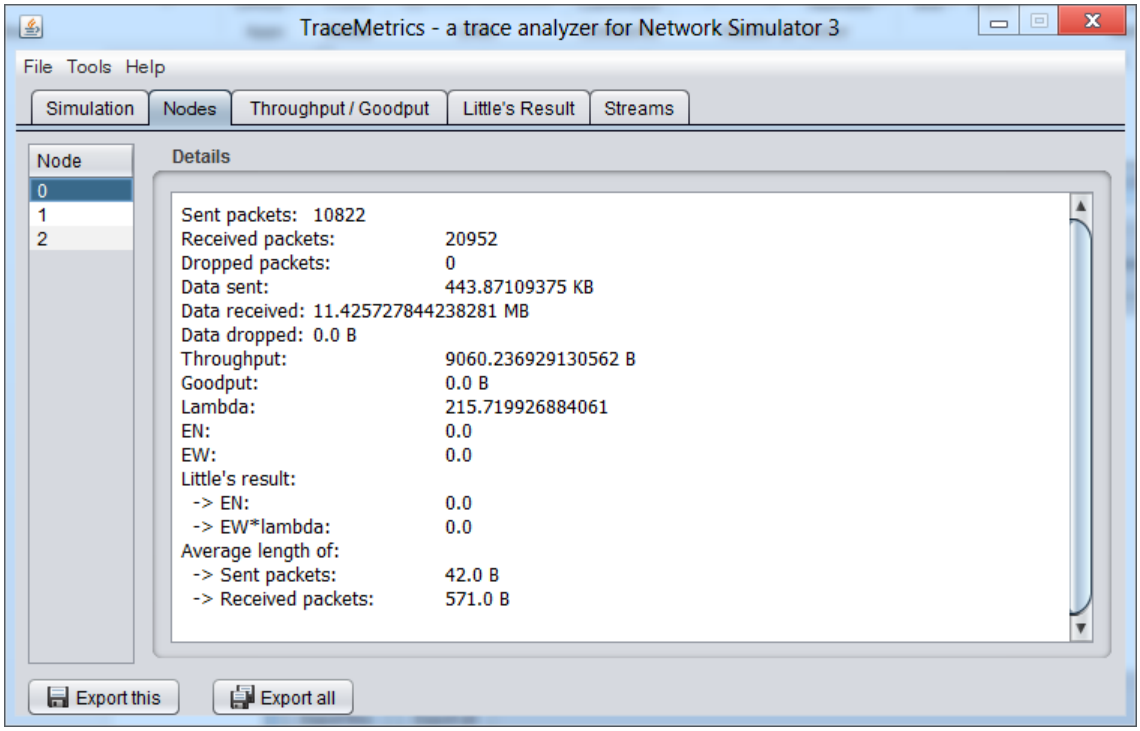

Figure 7: The Nodes tab

This tab has a list of nodes in the simulation. Clicking on one node will make its details appear in the text area. Theinformation go from the node number in the simulation to the mathematical validation by Little's Result.

The option to save the results is often very useful and it can be done using the Export buttons. The button 'Export this' will export the details of the current selected node. Clicking on 'Export all' will export the details of all nodes in the list.

5.3 The Throughput / Goodput tab

This tab has the purpose to make it easy to compare the throughput and goodput between the nodes in the simulation.

The data is presented in a table, as shown in Figure 8.

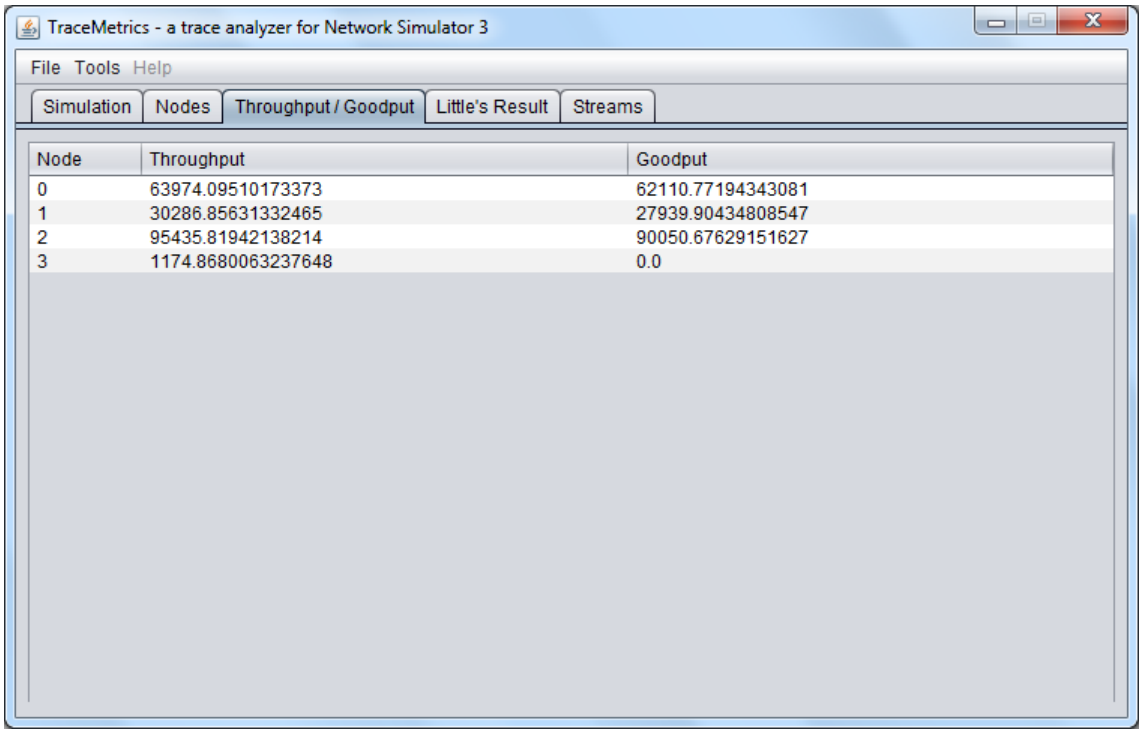

Figure 8: The Throughput / Goodput tab

5.4 The Little's Result tab

As the tab presented before, this tab has the purpose to ease the visualization. Figure 9presents this tab.

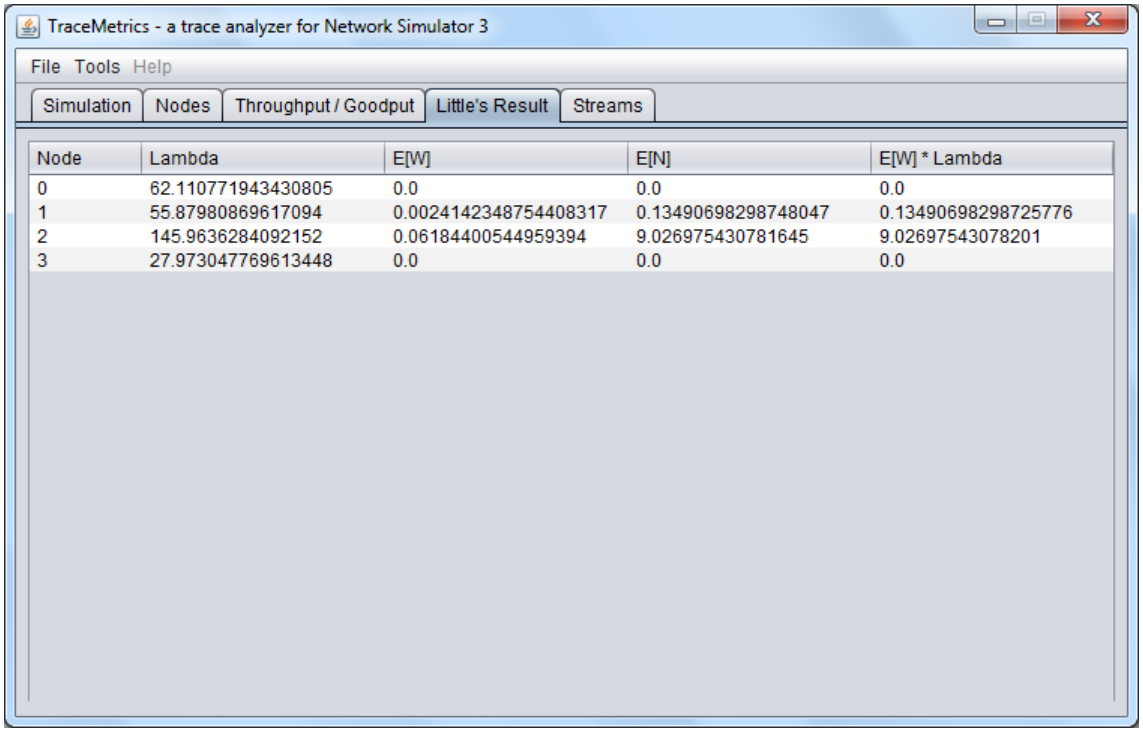

Figure 9: The Little's Result tab

In this tab, we can analyze three main things: the Little's validation, the average size of the queue (E[N]) and the average time spent by each packet in the node (E[W]). This information can be very useful to detect points of overload in the network. The Lambda column shows the average number of packets send by the node each second.

5.5 The Streams tab

This tab will present the results related with data streams like TCP or UDP. In the Figure 10a we can see this tab showing data about all TCP streams.

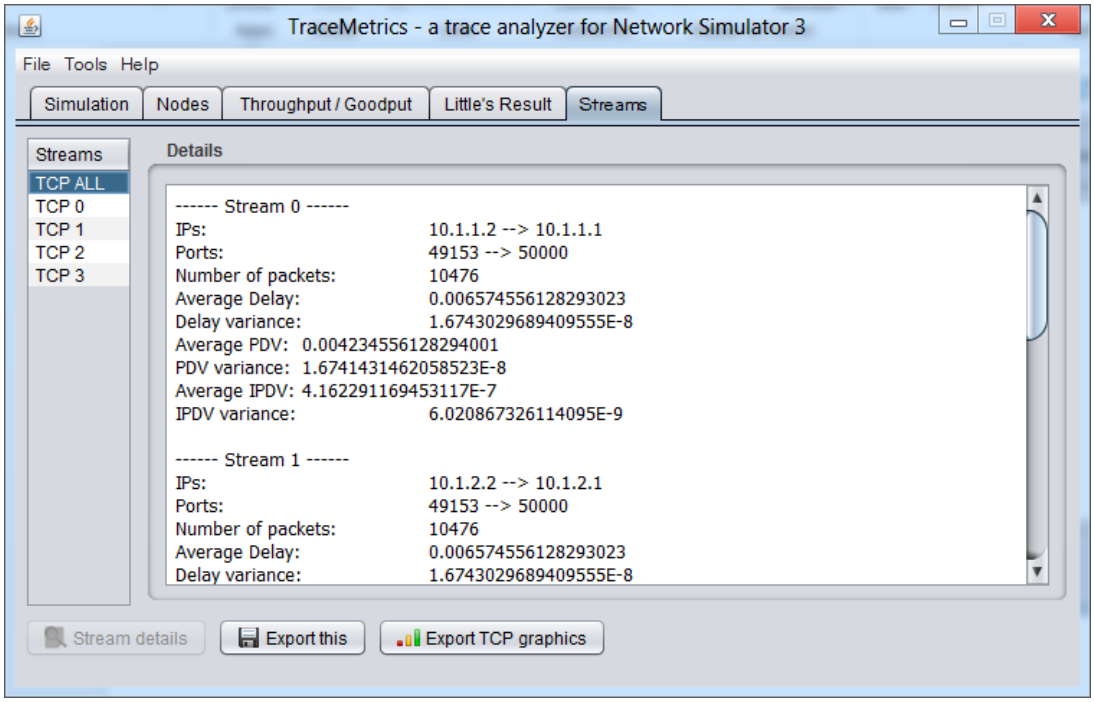

Figure 10a: The Little's Result tab - TCP

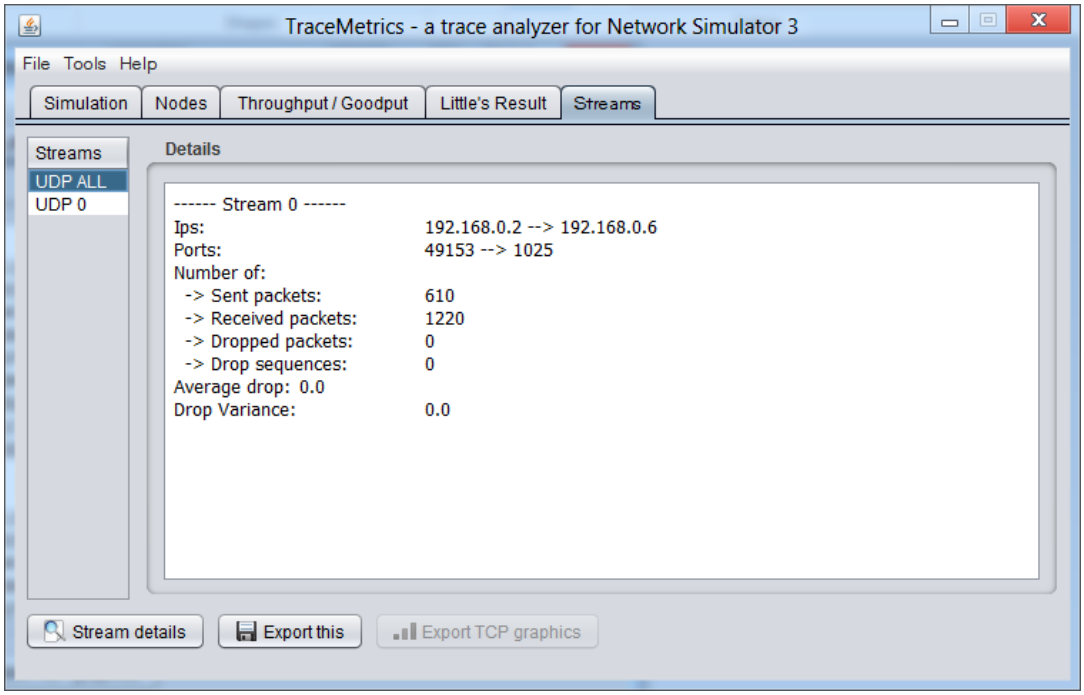

Figure 10b: The Little's Result tab - UDP

It is easy to notice that the information presented for the TCP streams are different of the ones for the UDP stream. The TCP streams have information mainly about delay and the UDP streams have mainly information about dropped packets.

This tab has the option to Export the results.

Thisfeature provided by the button Stream details threats separated the TCP and UDP streams.

This tab also has the button *Export TCP graphics* that will be explained on the chapter 6 – Export graphics. It is only used with TCP streams.

5.5.1 Stream details – TCP

The details presented here are presented in Figure 11a.

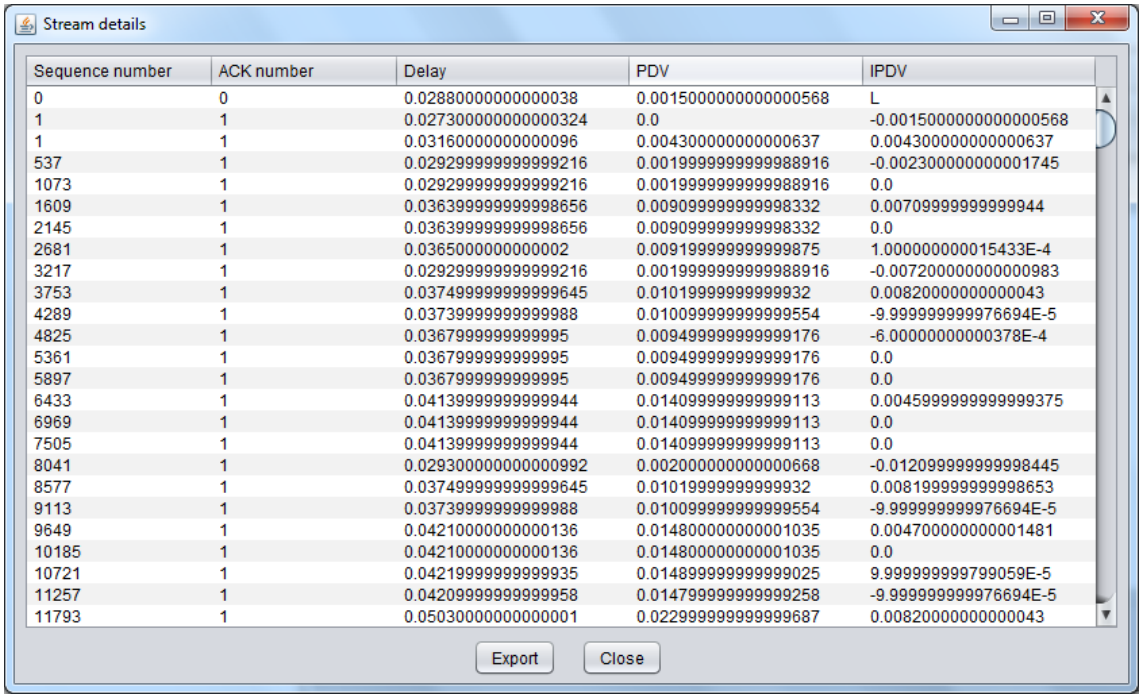

Figure 11a: The Stream details – TCP

This window presents the details of each packet sent from one node to other using TCP. The informations are: sequence number, ack number, end-to-end delay, PDV and IPDV (RFC 5481 Section 4).

5.5.2 Stream details – UCP

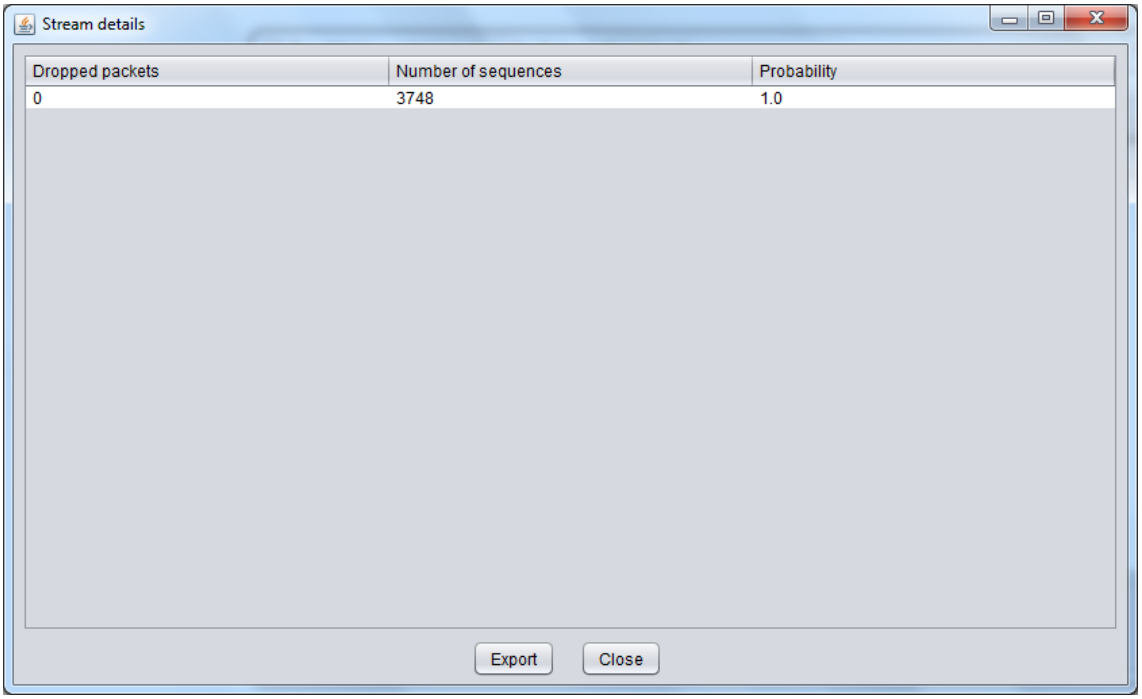

Figure 11b: The Stream details - UDP

The details here cover a probability distribution on drop packets. In this case, there was none packets dropped, so the probability of having 0 packets dropped is 1.0 or 100%. This window covers the number of dropped packets, the number of sequences that had the same quantity of drops in sequence and the probability of each one happen.

6
Export graphics

The easiest way to analyzethe data of set of each packet, such as delay, pdv or ipdv it's with graphics than only with the list of values. Tracemetrics do not generate any graphic alone, but make scripts that can be loaded on the gnuplot.

Tracemetrics will only build your script based on options chosen on export TCP graphics window.

Gnuplot is a portable command-line driven graphing utility, which can be downloaded on http://www.gnuplot.info/. Support windows, Linux, OSX, and many others systems.

6.1 Using the tracemetrics interface

The interface have four principal areas, General options, Type of graphic, Smooth-style and others options and make a view. The interface is shown on figure 12. The minimum size of interface is 930 x 660 pixels.

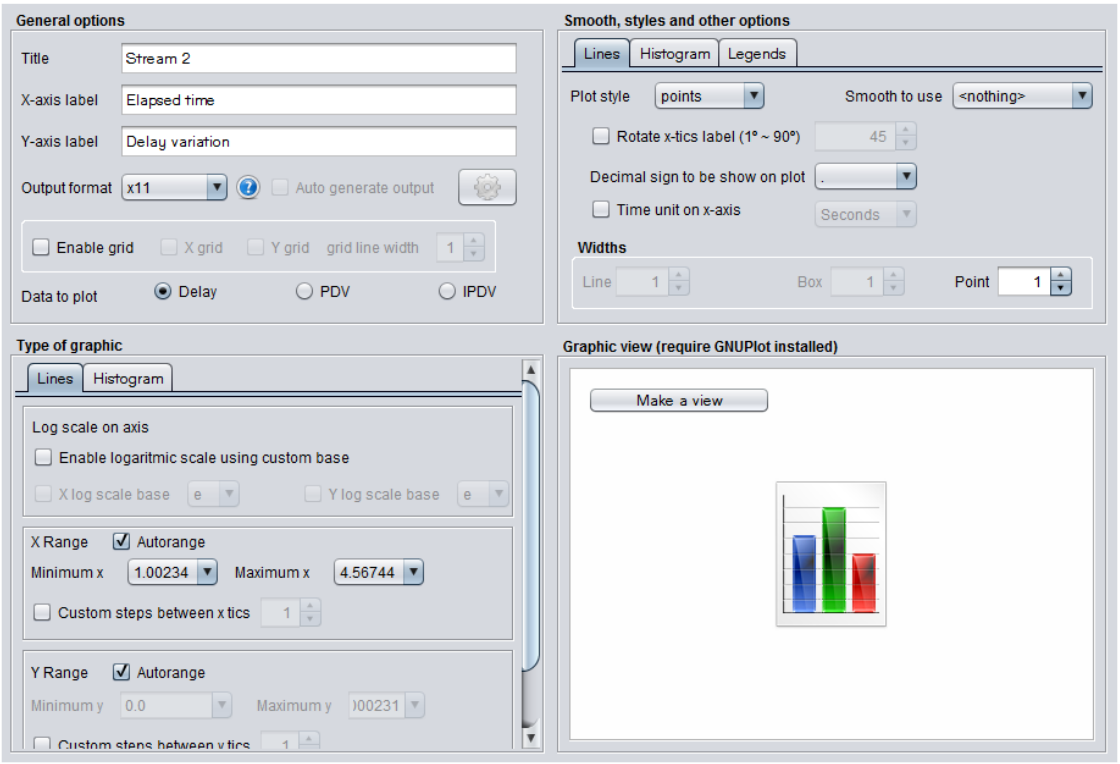

Figure 12: The interface to configure script

6.1.1 General options

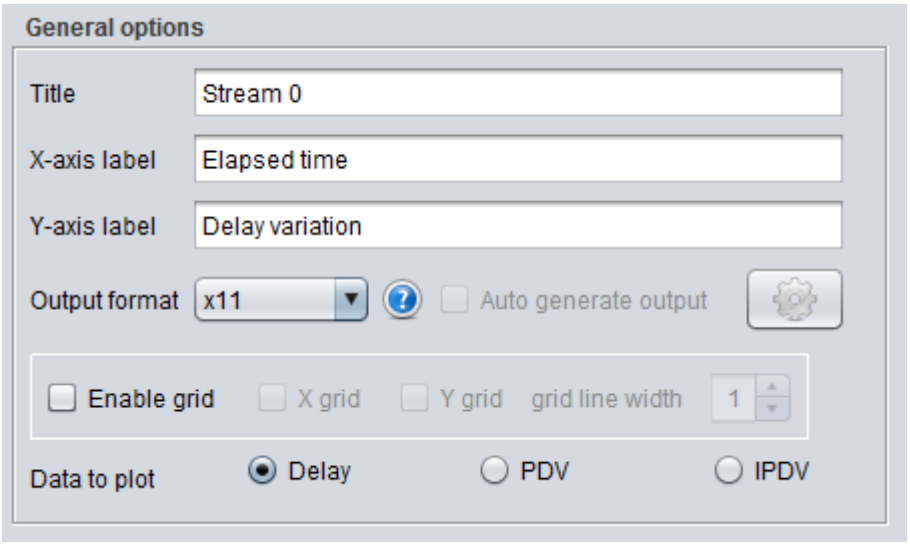

Figure 13: The "general options" area

On general options area, you configure the title for plot, the name to be shownon X-axis and Y-axis.

On "output format", gnuplot need to know where they will make a plot, orin an image or on a terminal. Gnuplot supports many terminals, but Tracemetrics offers only the main outputs, such terminals, images files and HTML output. If any terminal that generated a file as output, the "Auto generate output" will be enabled and can be selected; ifselected, when you save the file, Tracemetrics will try to auto load the script on gnuplot and gnuplot will make the output file with graphic image.

"Enable grid" leaves the graphic with lines that help on graphic analysis.

"Data to plot" indicates which of calculated metrics must be plotted. On image *11a of chapter 5*, after you click on the button "Stream details", a list with all calculated values is shown. All this values are used to make an image on gnuplot.

You only can plot one at time. If you selected "All streams" to export a graphic, one specific metric of each stream will participate of plot.

6.1.1.1 Changing output settings

The button with a gear, opens a new window pop-up that contains the set of configurations for each terminal based on a file.

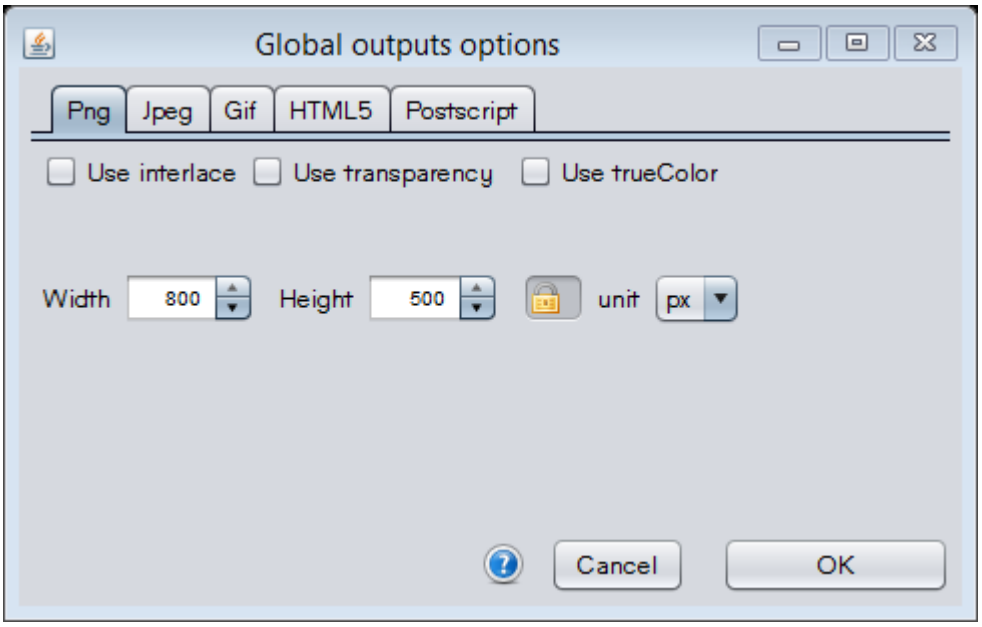

Figure 14: Pop-up window with configurations for terminals.

All options on this window are trivial and do not require explanation, and you can leave the mouse above some button or check box and get help with tooltip texts.

6.1.2 Type of graphic

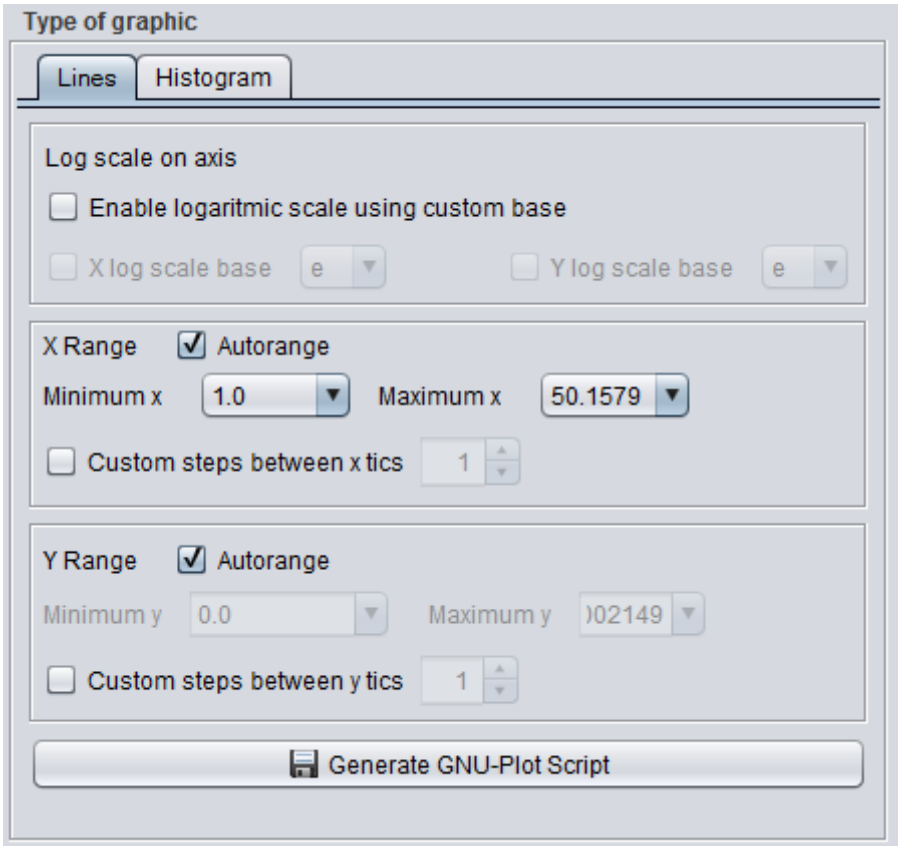

Figure 15: The "type of graphic" area with lines tab selected

Two types of plot are supported by Tracemetrics interface: lines and histograms. After choosingthe best set of options, when you click on "Generate GNU-Plot Script" a save file dialog will be displayed; Tracemetrics will try to make a name for file based on name of trace file opened, the actual metric and the actual stream used to generate script. After savingthe script, automatically, Tracemetrics will generate a file with same name, but with ".dat" extension; this file will contain all necessary data set for gnuplot.

6.1.2.1 Line

On graphic line, each value is generally represented by a dot, and is placed confirming the (x,y) tuple, where x is the inside time of simulation, and y is the calculated value.

In the figure 15, you can see the line interface; it is simply formed by three principal configurations, the possibility to use logarithmic scale[1] on x and y-axis, and the range on x and y-axis.

Log scale on axis

For logarithmic scale, you can select which axis will be treated, and a custom base for log scale. If the actual bases on Tracemetrics interface are not useful to you, open the script and edit the line:

set log {x | y} <base>;

replacing<base> by a convenient number, of defined variable on gnuplot environment, such *e*.

X Range

On second area configuration of line tab, "X Range", you will define how gnuplot will show graphic range on X; if you select auto range, gnuplot will start the x-axis on zero and will leave a space after *maximum x* value selected;otherwise, the maximum x value will be placed at right edge of image. All possible values on *minimum x* and *maximum x* are discrete values from simulation.

Y Range

On third area configuration, y auto range it's similar to x auto range, if you not select auto range, you can manual define the range; Tracemetrics will try to find the best values for these fields.

Tics

For both areas, "custom steps between tics" specifies the distance for each tic in the correspondent axis.

[1] A **logarithmic scale** is a scale of measurement that displays the value of a physical quantity using intervals corresponding to orders of magnitude, rather than a standard linear scale.

6.1.2.2 Histogram

On histogram graphics, the objective is to quantify inside from all calculated values, a specified range of values. To exemplify, imagine the current list values:

TIME	PDV
1	0.002
1.03	0.02
1.10	0.003
1.5	0.005
1.8	0.2
2	0.005
2.1	0.004
2.2	0.003

Table 1: Calculated PDV for all times

Suppose that you have to know how many values are inside the interval [0~0.003]; for this example, three values are inside and five are out, in other words, 37.5% of values are in the selected interval and 62.5 are outside from interval. The figure 16represent this in a graphic:

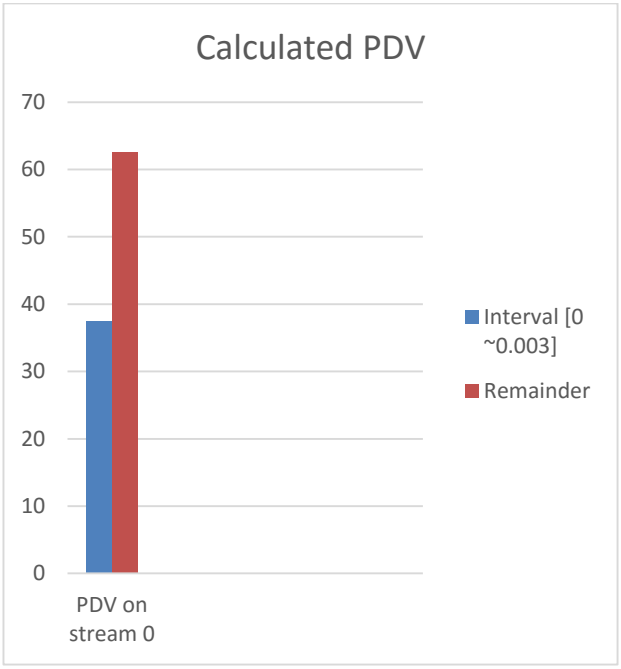

Figure 16: Histogram shown in the table above

The Tracemetrics interface for histogram is like figure 17below:

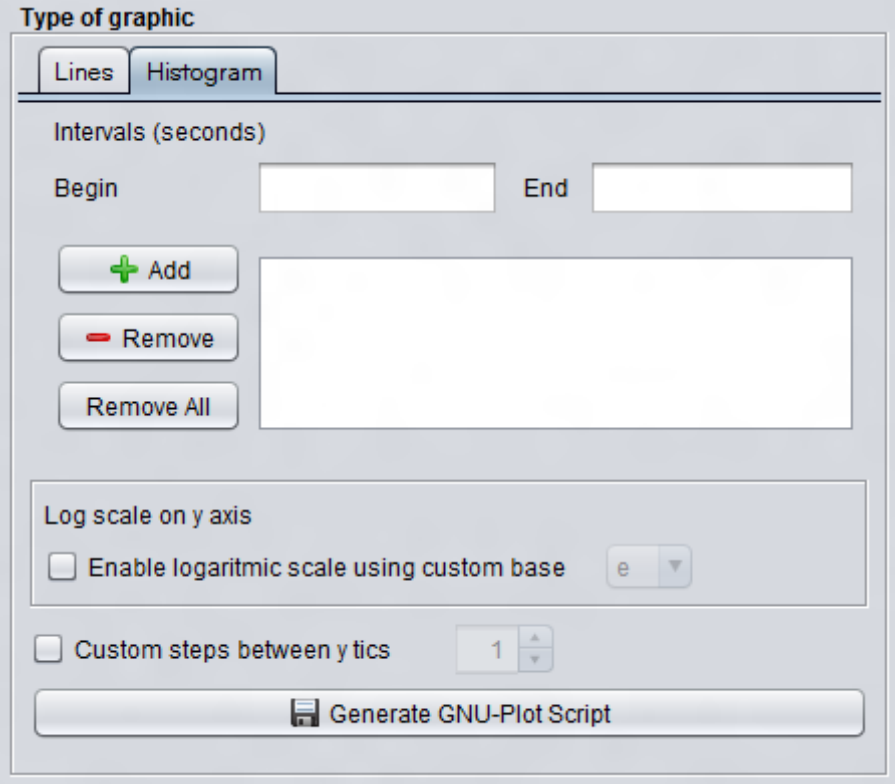

Figure 17: Histogram tab

You can add as many intervals as needed, but they must meet with the following criterion:

- Should not overlap
- The end of a range can be at most equal to the start of another interval

For example, the intervals [0 ~ 0.2] and [0.2 ~ 0.6] are valid, however, the intervals [0 ~ 0.2] and [0.1 ~ 0.6] are not valid for this type of analysis.

To verify if a value is inside of an interval, Tracemetrics verifies if value is greater or equal than begins and minor then end. In other words:

Begin <= Value < End

IMPORTANT: Tracemetrics does not forbids you to do this. Tracemetrics only requires that you insert a crescent interval, such [0 ~ 2]. The interval [2 ~ 1] is invalid and Tracemetrics will not allow this input.

Toput an interval, only writes the values on fields "Begin" and "End" and click on "Add"; all intervals are unique, if you try to input an interval that already exists, nothing happen. You can select a unique interval from list and remove with "Remove" or use "Remove all" to clear the list.

The explanation for logarithmic scale and custom steps between tics are the same of lines.

6.1.3 Smooth, styles and other options

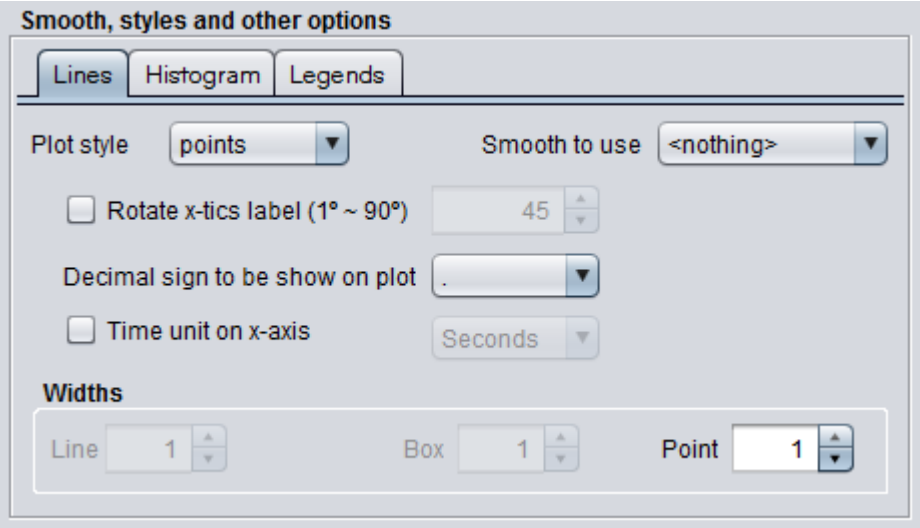

Figure 18: Lines configuration

This area contain three tabs, one for each type of graphic with specifics configuration and a tab for configure the legends, that are similar to lines and histogram graphics. Many options are self-explanatory.

6.1.3.1 Lines

A short explanation for each option field:

Plot style: how each value is represented on graphic; choose the most convenient style, if you have many values, dots can be the best option, otherwise, points or lines.

Smooth to use: Gnuplot can smooth your graphic with some math skills, but attention, at least three values are required for use this, if your stream has less than three values, gnuplot will disregard this.

Rotate x-tics: The name on x-axis can be rotate for most commodity.

Decimal sign to show: Specifies default decimal sign for show on plot; you can enter a custom char.

Time unit: How the data will be accommodated on the plot.

Widths: You can change the width of lines, boxesand point on your plot.

To see more details, please, check the Gnuplot documentation

6.1.3.2 Histogram

Figure 19: Histogram configurations

Histogram style: Grouped, where each bar is placed side by side; stacked, where each bar is stacked above of other. For this reason, you can't overlap intervals.

Gap size: When you plots a graphic for All Streams, each stream is a setof bars, and the distance between these setsare defined by "gap size".

Fill boxes: This option tells to gnuplot how it should fill the bars, whether it is with patterns or solid colors.

Box width: The width of border; is not considerate if "boxes without borders" is selected.

The other options are similar to line configurations.

6.1.3.3 Legends

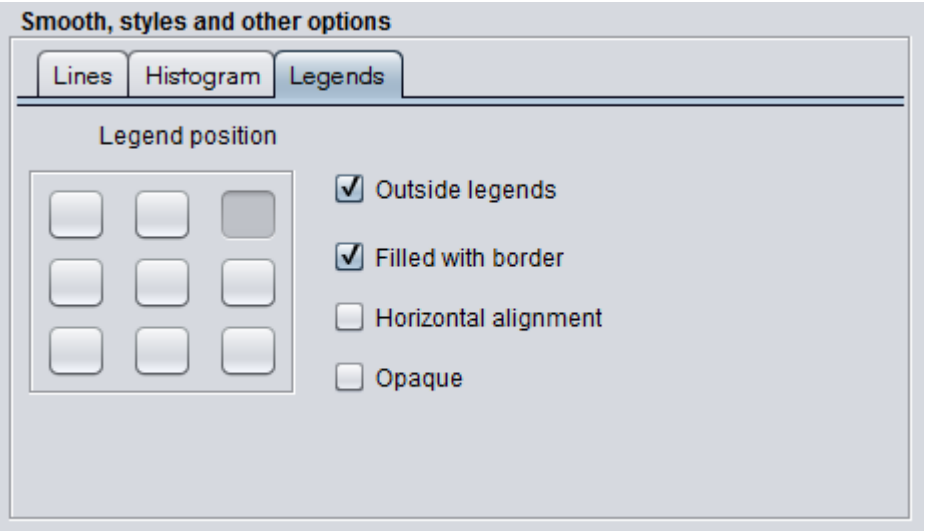

Figure 20: Caption Settings

Subtitles are common to two graphics, but, when you change from lines to histogram or vice-versa, the defaults values for these graphics are auto marked.

For lines, all checkboxes are deselected, for histogram, "outside subtitles" and "field with border" are selected, like figure 20.

Caption positions: Select a position for the field with subtitles

Outside captions: Make legends outside from area of graphic

Field with border: Fill or not fill the field containing subtitles with borders

Horizontal alignment: Determines how the set of subtitles are placed.

Opaque: If selected, gnuplot will draw the subtitles after graphic, overlapping the graphic with the subtitles.

6.1.4 Graphic View

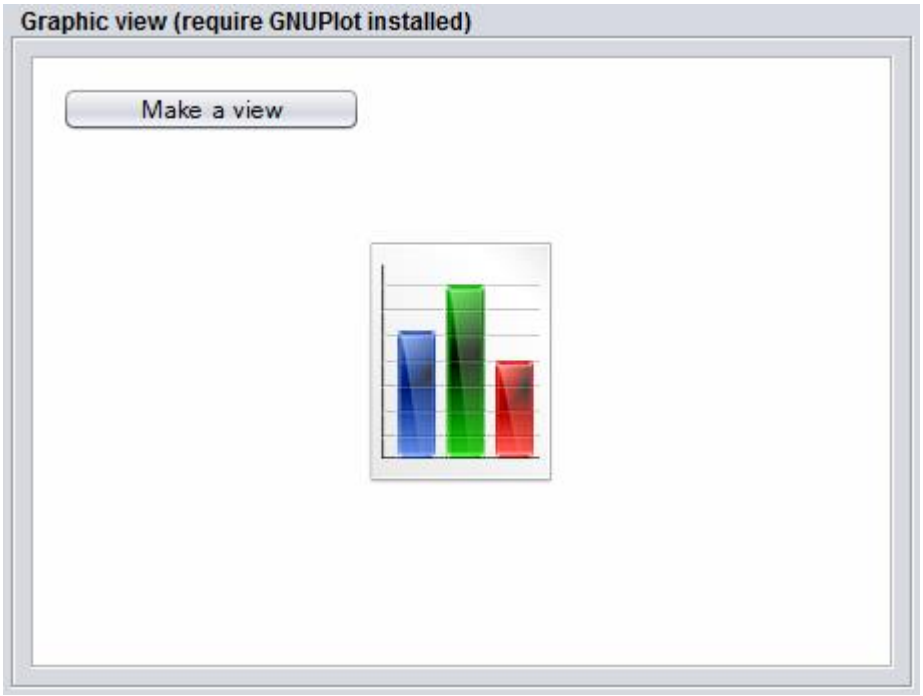

Figure 21: graphic view (initial state)

This last area shows the graphic plotted with set of configurations chosenon interface. Tomake the graphic for the first time, click on "Make a view".

IMPORTANT: Make sure that you have gnuplot installed on your computer, with support for jpg, png or gif terminal. If nothing of these terminals are installed, this feature will not work, showing an error image – see figure 22.

Other problems can show the figure 22on visualization area, besides the listed problems on the figure, as well as if you try to make a view for histogram without insert any interval.

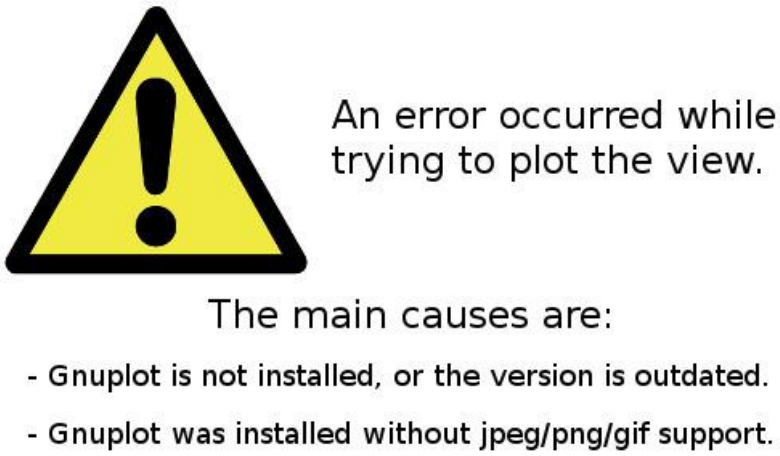

Figure 22: Error image displayed if some error occurred.

After to beget a first view, you can update any configuration and click on the refresh button to generate a new graphic. See figure 23.

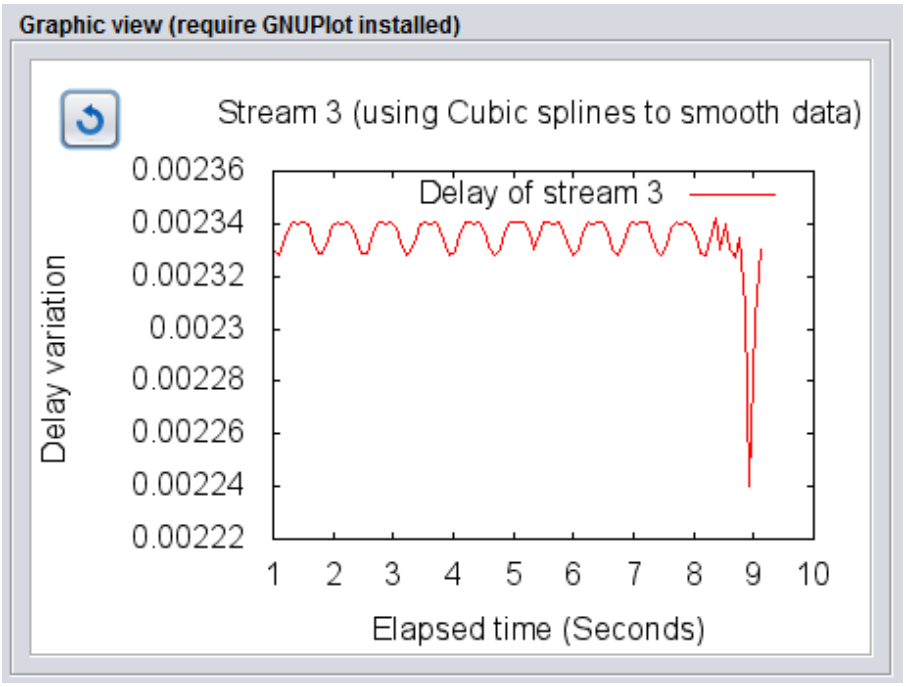

Figure 23: A view of line graphic (using smooth)

If you pass the mouse above the image area, the cursor will turn in a magnifying glass. 

When you click, a new window will be displayed with a large view of the graphic; don't let to be mistaken, sometimes the large view is a little different of the tiny view – this is because gnuplot try to accommodate the graphic as the best way possible. If you think thatyou graphic is not good, try to view a large picture first.

The view area will be resized according to the resizing of the window.

6.2 Saving your script

After choose the necessary configurations for your graphic, click on the button "Generate GNU-Plot Script". As explained, Tracemetrics will adjust name of output file accordingly with trace file, stream and metric to generate the graphic; when yousave the script, another file is created with same name, but with the extension ".dat".

OBS: If you choose HTML 5 as output, an additional directory will be created with all necessary java scripts, images and cascading style sheet (css files) necessaries to show the web page accordingly. This directory can be used for many different HTML 5 output files, saving space in disk and leaving organized your directory.

6.3 Loading your script on gnuplot

Inside of script, you will find the necessary instruction for load the script on gnuplot. See some ways to make this:

1. Open gnuplot by line command on the same directory of script;
   type *load "name_of_script.txt"*.
2. Open gnuplot in any directory;
   type *cd "complete_path_to_script_directory"*;
   make the step 1.
3. In a line command terminal, you can type *gnuplot "name_of_script.txt"* to open gnuplot and auto load the script, but attention: if the output was configured not generate a file, the gnuplot will be immediately closed, otherwise, the file defined on script will be saved on the same directory.
4. If you are using a graphic interface, you can load the file by menu.

6.4 Knowing the script

```
#Instruction: to plot the graphic, type load "Stream 3_(tcp-star-server)_Delay.txt" in the same directory.
#You can type 'cd "C:\Users\Reuel\Desktop"' on GNU-plot to reach it.
#Make sure that the file "Stream 3_(tcp-star-server)_Delay.dat" is in the same directory.

#This part configures the GNUplot environment to be adjusted conform your configuration.
#Feel free to alter it.

#As there may be incompatibilities with older versions of the script GNUPlot,
#which might not accept some of the explicitsettings, just comment out the line by placing a # at the beginning.

reset;
e = 2.7182818284590452354;
set datafile missing "NaN";
set terminal x11;
set output;
set title  "Stream 3";
set xlabel "Elapsed time (Seconds)";
set ylabel "Delay variation";
#The inline command to be used on GNUPlot.
plot "Stream 3_(tcp-star-server)_Delay.dat" every 1:1:0:0:5410:0 using 1:2 title "Delay of stream 3" with points
```

A typical script for lines plot

The script tries to be the most minimalist possible, to ensure the compatibility with old versions of gnuplot. When defaults values are selected on interface, they are not written on file.

IMPORTANT: The script is based on the newest 4.6 version of gnuplot to date.

Each line started by '#' is a commentedline; it makes no difference for gnuplot

First line after commented lines returns gnuplot environment to default values.

Second line define a local variable e (sometimes does not exist by default);

Third line set the 'NaN' as data file missing value, if encountered on data file .dat

```
#Instruction: to plot the graphic, type load "Stream 3_(tcp-star-server)_Delay.txt" in the same directory.
#You can type 'cd "C:\Users\Reuel\Desktop"' on GNU-plot to reach it.
#Make sure that the file "Stream 3_(tcp-star-server)_Delay.dat" is in the same directory.

#This part configures the GNUplot environment to be adjusted conform your configuration.
#Feel free to alter it.

#As there may be incompatibilities with older versions of the script GNUPlot,
#which might not accept some of the explicit settings, just comment out the line by placing a # at the beginning.

reset;
e = 2.7182818284590452354;
set datafile missing "NaN";
set terminal x11;
set output;
set key outside  box ;
set style data histogram;
set style histogram clustered gap 2.0;
set style fill pattern 1 border lt -1;
set title  "Stream 3";
set xlabel "Selected stream";
set ylabel "Percents of use for each interval (%)";
#The inline command to be used on GNUPlot.
plot "Stream 3_(tcp-star-server)_IPDV.dat" using 2 title "[0.0 : 0.003]",  "" using 3 title "[0.003 : 0.006]",  "" using
4:xtic(1) title "Remainder"
```

Fourth and fifth line defines the terminal and the output.

The next three lines are intuitive, and the last command is not explained here.

A typical script for histogram plot

Note: The two scripts are very similar!

This script contain additional lines such

```
set key outside box ;
set style data histogram;
set style histogram clustered gap 2.0;
set style fill pattern 1 border lt -1;
```

Those lines configure only some styles on gnuplot to work properly.

Again, the last command isn't explained here.

For more information about other commands and flags of gnuplot, you can find all necessary support on http://gnuplot.sourceforge.net/.