

MONASH UNIVERSITY
 DEPARTMENT OF ELECTRICAL & COMPUTER SYSTEMS ENGINEERING
ECE4045/ECE5045 Network Performance

Visualizing the Results with the OMNeT++ IDE

Alexander Senior

Y. Ahmet Şekercioğlu

1 Objectives

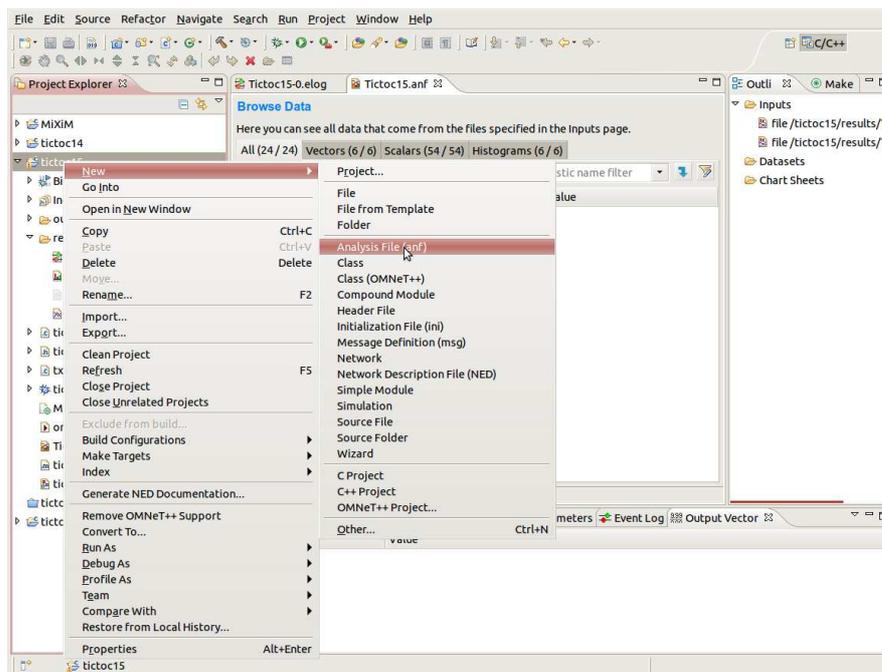
This short tutorial aims to help you for visualization and statistical analysis of simulation data through the IDE (Integrated Development Environment) of OMNeT++.

2 Introduction

The OMNeT++ 4.1 IDE has introduced a new way to visualize and analyze the statistics collected in simulations. The *settings* for creating charts are stored in `.anf` files; this means that charts can be easily re-created when new data becomes available - just replace the old results files with the new ones!

There are two ways of creating a new `.anf` file in the IDE:

1. You can do the following: Click on File → New → Analysis file. This action will start the analysis editor:

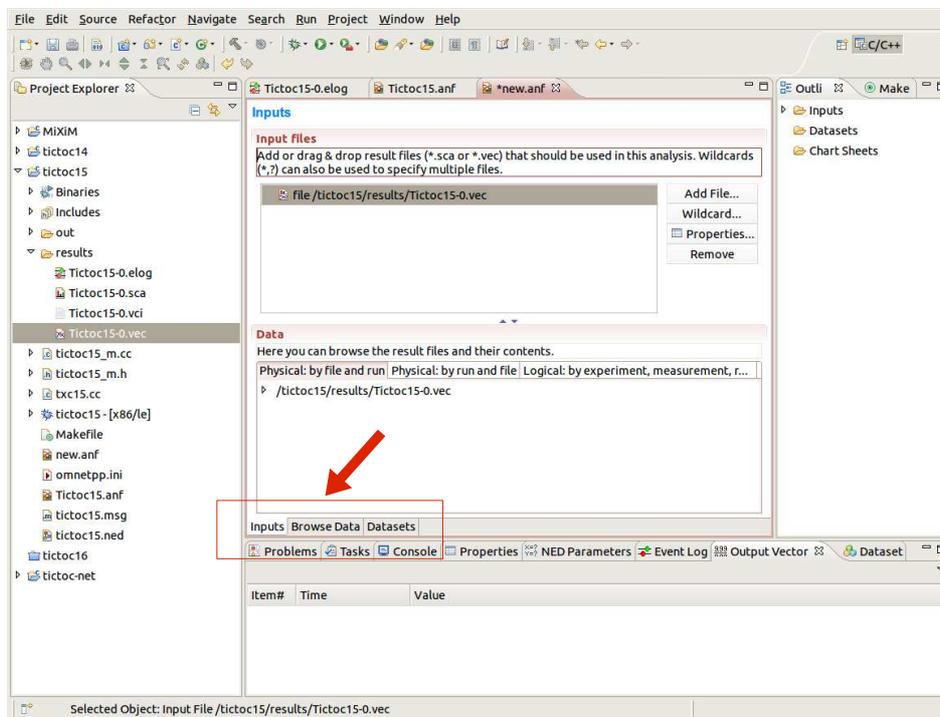


2. Double-click on a result file (a .vec or a .sca). All result files with the same prefix will be added to the analysis file (for example, if aloha-10.vec was double-clicked, then all files that match aloha-*.vec and aloha-*.sca will be added as well).

The analysis editor is a multi-stage tool, the stages roughly correspond to the following steps:

1. Select files to take as inputs.
2. Select what data to use from the files.
3. (Optional) Define any processing rules to apply.
4. Define what charts to create from the (processed) data.

Each .anf file has multiple *tabs* (or pages)



Their functionality of each page can be summarized as follows

Inputs: this is where the result files you want to process are selected, and is separated into two parts: the top part is where you either add files explicitly, or define rules with wildcards; the bottom half shows the files that match the selection rules.

Browse data: the data from the files you selected in the previous page is shown here; you can see all the results, or separate them into vectors, scalars and histograms. The results can be further filtered with the combo boxes you see in the default view, or you can create more advanced filters by clicking on the 'Switch to Advanced filter' button. The selected/filtered data can be plotted immediately, or remembered in named data sets for further processing (next page).

Datasets: The third page displays the datasets (top pane) and charts (bottom pane) that you can create from the data you've selected. A dataset is a set of input data, that has some processing applied to it and the charts generated from the processed data. Datasets are displayed as trees, and you can drag operations (such as adding data, removing bad or irrelevant data, processing and creating charts) onto them. Datasets can be exported as CSV's, Octave text files or MATLAB scripts¹. Charts can also be exported as SVG's.

It is time to try the analysis editor and generate some graphs.

3 Procedure: TicToc Example

We will attempt to recreate the graphs shown in Part 5 of the Tictoc tutorial². Note that, by now you should have completed Step 15 of the tutorial, and the result files of Tictoc15 simulation already created (the `.vec` and `.sca` files).

1. To begin, create the analysis file by double-clicking on one of the results files (located in the results folder of the OMNeT++ project). Save the `.anf` file as `Tictoc15.anf`. There should be two file filters in the top pane of the inputs page of the analysis file:

```
file /tictoc/results/Tictoc15-*.vec
file /tictoc/results/Tictoc15-*.sca
```

2. You can have a look at the data available in the Browse data page, but we don't need to apply any filtering for the moment so go to the Datasets page. It should be empty; remedy this by clicking on the button labelled **Dataset**. Call the dataset anything you like. Now add data by either clicking on the button labelled **Add**, or dragging it on top of the dataset you just created (most operations are created by dragging them onto an existing node in the dataset tree). If you dragged it, there will now be a box labelled **add scalars: all**; if you clicked it, select scalars in the data type box, and leave the filter field empty to add scalar data. Add all vectors and histograms using either method; if you're using the drag method, double-clicking on the node in the tree will allow you to edit what data type you're adding — you can also right-click and select properties.
3. To create the first chart in the example, drag the box labelled **Line Chart** onto the dataset; you will see a new node appear, called **line chart <unnamed>**. You can give the chart a name in a moment; first drag a **Select** module onto the line chart node — this will allow you to select what data will appear in the chart. Double-click on the select module; set the data type to vector, and put the following in the filter field

```
module(Tictoc15.tic[0..1])
```

It should be fairly apparent what this filter does. Double-clicking on the chart will make it visible in a separate page of the analysis file; you can customise it however you want by going back to the Dataset page and editing the properties of the chart.

4. The second chart in the example features a plot of the *mean* of the hop count for each model, and as such will include our first example of processing data. Drag a **Group** node onto the dataset, and give it a name. Now drag an **Apply** node onto the newly-created group node, and select the **mean** operation. Drag a line chart onto the group,

¹Note that histograms cannot be exported yet.

²<http://www.omnetpp.org/doc/omnetpp41/tictoc-tutorial/part5.html>

and again drag a selected node onto it; this time though, leave the filter box blank to select the vectors for each node. You can view and customise the graph as before.

5. The third chart is a bar chart, so naturally drag a **Bar Chart** node onto the dataset (be sure to add this to the dataset, and not the group you created for the previous graph). Drag a selected node onto the chart, select scalars for the data type, and type the following in the filter field:

`name("hop count:mean")` OR `name("hop count:max")`

again it should be fairly obvious what this filter is doing.

6. For the fourth chart, drag a **Histogram Chart** node onto the dataset, **Select** all histograms, and customise to your heart's content.