TRAFIL
A Tool for Enhancing Simulation TRAce FILes Processing

Christos Bouras[1,2], Savvas Charalambides[1,2] , Georgios Kioumourtzis[4] and Kostas Stamos[1,2,3]
[1]Computer Technology Institute and Press, N. Kazantzaki Str., University Campus 26504, Rio, Greece
[2]Computer Engineering and Informatics Dept., University of Patras, Patras, Greece
[3]Technological Educational Institute of Patras, Patras, Greece
[4]Center for Security Studies, P. Kanellopoulou 4, 10177, Athens, Greece
{bouras, stamos}@cti.gr, charalampi@ceid.upatras.gr, gkioumou@gmail.com

Keywords: Simulation Analysis Tool, Trace File, NS-2.

Abstract: NS-2 (Network Simulator) is one of the most popular discrete event simulators used for network simulation. Trace files produced by NS-2 provide very useful information for post simulation analysis. This paper presents the architecture and development considerations for a TRAce FILe analysis tool, which intends to simplify the management of trace files generated during network simulations. The tool is focused on NS-2 trace files, but can be extended to handle a variety of simulation trace files formats. Its purpose is to make the execution of a large number of network simulations faster, and the extraction of results from a large amount of data more flexible and productive. TRAFIL introduces a novel way of interpreting, parsing, reading and eventually utilizing NS-2 trace files by using "metafiles" and "sub-metafiles" during the trace file recognition and process procedures, making the overall operations more abstract, substantially efficient and faster than alternative approaches. Furthermore, TRAFIL facilitates the whole trace file analysis task, offering the opportunity to store each trace file as well as every measurement produced for each trace file. The tool aims to aid the analysis of simulation results offering features that other tools in this area have been missing.

1 INTRODUCTION

The simulation of networks (Breslau, 2000; Fujimoto, 2003) in telecommunications and computer networks is a technique where the behaviour of a real network is modelled using software. Often a simulation of the network's functionality is represented by calculating the interaction of its pieces and then analyzing the performance.

Network simulation tools (Weingärtner, 2009) are widely used for simulating physical networks with great ease. A very popular network simulation tool is NS-2 (Breslau, 2000; Network Simulator NS-2; Cicconetti, 2006), which is a discrete event simulator used for research in computer networks. It can be used for simulating a variety of protocols such as multicast, TCP and routing in different kinds of networks.

Trace files (NS-2 Trace Formats) are considered some of the most valuable information a network simulator like NS-2 has to offer, they describe all the events that have taken place during a simulation and based on these events users retrieve information about the network's behaviour.

There are similar tools to TRAFIL that produce statistics of a simulated network's behaviour, but some of these projects have integrated NS-2, unlike our tool which is capable to only use NS-2 trace files to produce the requested measurements (Network Simulation Cradle Homepage; NS2Measure Homepage). There are also tools like JTrana(Qian, 2008), Trace Graph(Malek, 2003; Trace Graph Source Code) and NS-2 Trace Analyzer (Salleh, 2006; Trace Analyzer Source Code) which offer the opportunity to analyze NS-2 trace files and produce numerous statistics, metrics and charts.

All of the above tools provide useful information about a certain simulation scenario at each time and therefore all the measurements and results refer to only one trace file. In addition, these tools do not provide the user with the opportunity to store the trace file he has analyzed, for instance in a database, so that he can reuse it without having to reopen it. In terms of performance when it comes to loading a trace file for analysis the earlier mentioned tools behave well for small sized trace files. However, when it comes to serious simulations (Ben-El-Kezadri, 2008) that produce trace files in the orders of MB the performance deteriorates significantly.

The motivation for writing this paper is to present a new post simulation analysis tool that aims to make the extraction of information from trace files faster, productive and more flexible. This tool named TRAFIL gives the opportunity to process as well as store simulation trace files and then conduct various calculations and produce measurements based on their data.

The remainder of the paper is organized as follows: In section 2 we describe the architecture of TRAFIL with description of each component individually. Section 3 presents use cases and results of TRAFIL usage and finally Section 4 concludes the paper.

2 TRAFIL ARCHITECTURE

In this section TRAFIL's architecture is presented along with an explanation of every module that it consists of. Figure 1 depicts the tool's architecture. TRAFIL is the main module and it consists of 3 layers, the Presentation layer, the Business Layer and the Database access layer.

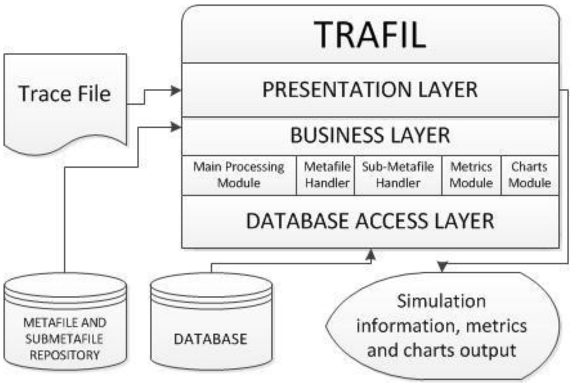

Figure 1: TRAFIL Architecture.

MySQL is used as the local database. Every trace file is stored in its own table depending on the metafile and sub-metafiles it was matched to, because the column names and data type of each column are extracted from them.

The metafile and sub-metafile repository is the component of TRAFIL that contains all the metafiles and their sub-metafiles. They are stored in a hierarchical order. Each metafile's sub-metafiles are stored in a unique folder named after the metafile they belong to. This way the loading of the sub-metafiles for a specific metafile becomes simpler.

Metafiles are unique files that describe each of the three main trace formats (NS-2 Trace Formats) that NS-2 produces which are the Normal trace format, the Old wireless and the New Wireless. They are used to encode the structure of a specific trace file format and to give information regarding the procedure of generating the metrics for a specific trace format, namely the names of columns whose data will be used. Using the metafiles TRAFIL can match any trace file that belongs to each of the 3 aforementioned categories. Furthermore, metafiles can be easily extended in order to support any type of trace file originating either from NS-2 or from another simulator, thus the use of metafiles renders TRAFIL a robust tool in regard to coping with changes of the input’s format e.g. newer versions of NS-2.

NS-2 trace files can have extra information depending on the packet types used in the simulation. This extra information comes in the form of additional header fields. This way a trace file may also contain different kinds of information in various lines. So we could say that each of these 3 main trace file types also has some sub types depending on the packets that the simulation uses. Normal trace files can log header information about TCP packets and Satellite packets. Old Wireless trace files can log information about ARP, CBR, TORA, AODV, DSR, TCP, IP, IEMP and RCA packet types. New Wireless can log information about ARP, CBR, TORA, AODV, DSR, TCP, IP and IEMP packets.

The combinations that can appear in a trace file cannot be known in advance since each simulation is different from another and the traffic used in each simulation can be anything the user wants. The number of header fields in each line is different in some cases from other lines in the same trace file and in some other cases so is the order that the header fields appear. It becomes obvious that the trace file recognition process is a very complex one. Each line in a trace file must be carefully examined to establish what types of header fields are present in order to handle the information it contains. The solution our tool uses to tackle this task is the use of “sub-metafiles”. Sub-metafiles are extensions of the metafiles that were introduced earlier. For each metafile we have additional sub-metafiles that encode the format for every different kind of header a trace file of a specific format can log. Therefore, the metafile used to match Normal trace files has two sub-metafiles, the metafile used to match an Old wireless trace file has nine sub-metafiles and the metafile used to match a New Wireless trace file has eight sub-metafiles.

3 TRAFIL USAGE AND RESULTS

3.1 TRAFIL Use Cases

In this section we illustrate an example usage of TRAFIL where we open a new trace file and present the information that the tool can extract. The scenario that was used to create the sample trace file involved 4 nodes connected in the topology shown in Figure 2. The simulation lasted 80 seconds and the communication was between nodes 0-3 and nodes 1-3. This scenario is a small one and its sole purpose is to present TRAFIL rather than validate or infer conclusions about the simulation itself.

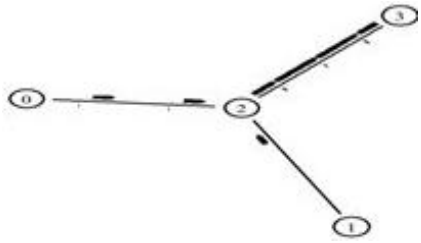

Figure 2: Sample Topology.

When the trace file has been analyzed and processed its contents are visible via the tool from a table, a feature that is unique in TRAFIL and its purpose is to enable users to have all the information they could possibly need centralized and ready to use. The columns of the table are created based on the metafile and sub metafiles that were used to process the trace file. TRAFIL can also produce specific QoS information about the simulation. The QoS parameters are Packet Delivery Rate in packets/second, Throughput in bits/second, minimum End to End Delay, maximum End to End Delay, average End to End Delay, Delay Jitter, minimum Delay Jitter, maximum Delay Jitter, average Delay Jitter and Packet Loss Ratio. All of the above QoS parameters are calculated by selecting two nodes of the simulation and the level at which the calculation of the parameters will take place. For a wired scenario the choices are Link Layer or Network Layer and for Wireless scenarios the choices depend on the trace levels present on the trace file and are a subset of AGT, RTR or MAC layer. The calculation of Throughput is performed by dividing the number of bits that were successfully sent and received between the selected nodes at the time of the arrival of the first and last bit between the nodes. The End to End Delay for each packet between the two nodes is calculated by finding the difference in the arrival time and transmission time of every packet that was exchanged from the sender node to the receiver node. The Jitter related parameters are calculated based on the RFC 3550 (Schulzrinne, 2003) for RTP packets. Finally the Packet Loss Ratio is calculated by finding all the packets that were sent and received between the two nodes, dividing their difference by the packets sent and multiplying by 100. Finally, TRAFIL enables a user to create charts regarding Packet Delivery Rate in packets/second, Throughput in bits/second, Delay Jitter and End to End Delay for each packet send between two selected nodes. There are two options for the time interval in which the sampling of values takes place in order to create a chart and it can be set to 1 second or 5 seconds. We believe that these two options are adequate to create charts with an acceptable level of detail.

3.2 Results

In this section we present some results regarding the processing time of TRAFIL when it comes to opening and storing new trace files of various sizes to the local database. The results are presented in Table 1 and they show that, even for very large scale simulations, TRAFIL manages to minimize the processing time. It is obvious though that the increase of processing time is near liner to the trace file sizes as one would expect.

Another observation extracted from the results is that for the trace files with sizes 11,413KB and 16,724KB compared to the ones with 25,251KB and 30,774KB sizes respectively, the latter ones have smaller processing time. This is because the first pair of trace files have Normal trace format and the number of sub metafiles used to process them is less than the number of the ones that would be used if they had Old Wireless or New Wireless trace formats. Hence, the execution time, to some extent, not only depends on the trace file size but also on the trace format.

Table 1: Processing Time results.

Trace File Size(KB)	Execution Time(ms)
2,524	2516
11,413	16453
16,724	13243
25,251	30751
30,774	21885
42,735	48427
54,462	59974
66,286	73092
115,335	127808

4 CONCLUSION-FUTURE WORK

We presented in this work a new tool to enhance simulations conducted with NS-2. This new tool named TRAFIL aims at offering an easy to use interface in which users can load and store simulation trace files in order to facilitate the analysis procedure. We used simulation trace files to produce measurements, statistics and charts in an effort to help students and researchers when running simulations with NS-2. TRAFIL was written in the JAVA programming language and is platform independent. The main benefit of TRAFIL apart from the production of ready to use metrics and charts is the reduced time it consumes for the whole trace file analysis procedure; firstly by minimizing the time to open and use a trace file and secondly by enabling the user to store each trace file in a database for easy access and future use. That was something that was missing all previous years in the NS-2 simulation environment. It also gives the opportunity to extract all the information that is stored or produced to a text or Excel file when it comes to the trace file, simulation metrics and information or jpeg for the charts. Finally, TRAFIL can be extended to support a variety of trace files and not only NS-2 trace files with great ease due to the use of Metafiles and Sub Metafiles.

In our future work we plan to extent TRAFIL with useful tools for simulating video transmission scenarios. The idea is to fully automate the whole simulation and analysis procedure. In addition to automating the simulation process we plan to offer the opportunity to perform SQL queries directly to the database and returning the results to the user. This way a user can retrieve all the different combinations of values contained in a trace file. TRAFIL is publicly available at (TRAFIL download).

REFERENCES

Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, Haobo Yu. 2000. “Advances in Network Simulation”, *IEEE Computer*,Vol. 33, No. 5, pp. 59 – 67.

R. Fujimoto, K. Perumalla, A. Park, H. Wu, M. Am-mar, and G. Riley, 2003. Large-scale network simulation: how big? how fast? Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. *11th IEEE/ACM International Symposium on*, pages 116-123, 2003.

E. Weingärtner, H. vom Lehn, and K. Wehrle, 2009. A performance comparison of recent network simulators. *In Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009)*, Dresden, Germany, IEEE.

Network Simulator NS-2: http://www.isi.edu/nsnam/ns/

Claudio Cicconetti, Enzo Mingozzi, Giovanni Stea, 2006. An Integrated Framework for Enabling Effective Data Collection and Statistical Analysis with ns-2. *In Proceedings of WNS2 '06; Proceeding from the 2006 workshop on ns-2: the IP network simulator*.

NS-2 Trace Formats: http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats

Network Simulation Cradle Homepage: http://www.wand.net.nz/~stj2/nsc/software.html

NS2Measure Hoepage: http://cng1.iet.unipi.it/wiki/index.php/Ns2measure

Qian, H. and Fang, W. (2008). Jtrana: A java-based ns2 wireless trace analyzer: http://sites.google.com/site/ns2trana/

J. Malek and K. Nowak, 2003. Trace graph-data presentation system for network simulator ns. *In Proceedings of the: Information Systems - Concepts, Tools and Applications* (ISAT 2003), Poland.

Trace Graph Source Code: http://www.angelfire.com/al4/esorkor/

Trace Analyzer Source Code: http://trace-analyzer.sourceforge.net/

Aliff Umair Salleh, Zulkifli Ishak, Norashida Md. Din, Md Zaini Jamaludin, 2006. Trace Analyzer for NS-2. *In Proceedings of the 4th Student Conference on Research and Development* (SCOReD 2006), Shah Alam, Selangor, MALAYSIA.

Ryad Ben-El-Kezadri, Farouk Kamoun, Guy Pujolle, 2008. XAV: a fast and flexible tracing framework for network simulation. *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, Vancouver, British Columbia, Canada

H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, 2003. “RTP: A Transport Protocol for Real-Time Applications”, RFC 3550, July 2003

TRAFIL download: http://ru6.cti.gr/ru6/research_tools.php#TRAFL