A Workbench for Internet Traffic Analysis

Philipp M. Eittenberger and Udo R. Krieger

Faculty of Information Systems and Applied Computer Science
Otto-Friedrich University Bamberg, Germany
philipp.eittenberger@uni-bamberg.de

Abstract. The specification and development of models for Internet traffic is mainly conducted by measurement-driven performance modeling based upon statistical analysis. Yet, this type of analysis can be a challenging task, due to the complexity and especially, with large sample sizes, the sheer quantity of the data. For this reason, preliminary data examination by graphical means and appropriate visualization techniques can be of great value. In this paper, we present the recent developments of the network traffic analyzer Atheris. Atheris has been designed specifically for measuring network traffic and for the visualization of its inherent properties. As a part of Atheris' functionality, it performs traffic measurements at the packet layer, data extraction, flow analysis and enables the visual inspection of statistical characteristics.

1 Introduction

Measurement-driven traffic modeling typically involves trace-based analysis of captured packet streams, to detect, identify and quantify the intrinsic characteristics. This quest of a statistical analysis is a multi-layered endeavor. For the purpose of performance modeling, graphical methods are intuitive and appealing ways for a preliminary examination of the data, especially when the sample size is massive. As an example, the initial process of distribution selection is usually a combination of using visual inspection and summary statistics. Thereby, after the packet capture has been finished, the usual procedure for performance modeling is the following: At first, the metrics of interest, (such as flow or session features, the packet size distribution etc.) are exported to a text file. Subsequently, the data file will be imported for further statistical analysis to the standard software of choice, e.g. *R* or Octave. Of course, it would be beneficial to get a first glance upon common criteria without this media discontinuity. Since such a statistical analysis is an explorative process that is normally conducted iteratively, the repetition of exporting and re-importing data sets can be an annoying and time-consuming task. Despite numerous measurement studies (e.g. [9] or [6] among many others), publicly available measurement tools, which are tailored to analyze the traffic features, are rarely provided yet. To summarize, for the purpose of performance modeling, visual inspection can be an integral complement to summary statistics and there is no solution available that provides powerful visualization techniques. In order to overcome this deficiency, we have developed

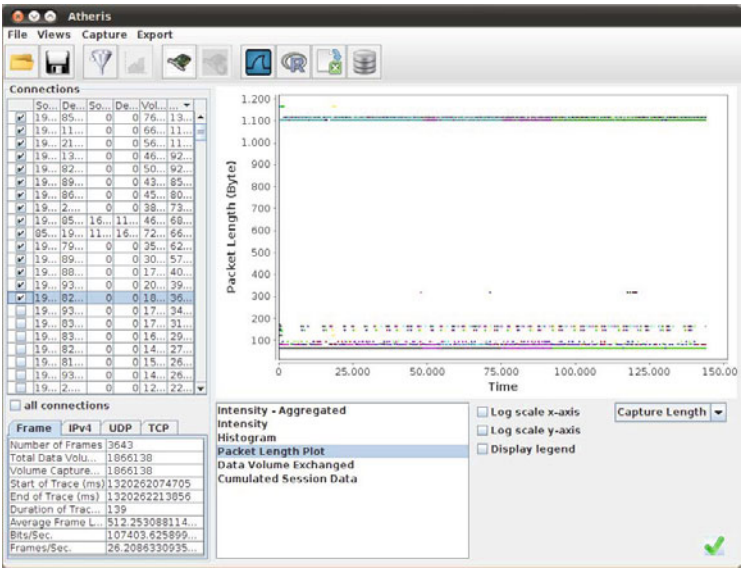

Fig. 1. Visualization Window of Atheris

the traffic analyzer Atheris based on our teletraffic analysis concept [10]. The target of this development was not only to enable straight through processing, but also to lay the foundations for a unified measurement workbench. We presented a demonstration of the first version of Atheris at the poster session at the P2P'2010 conference [8] and a more detailed description of the first version was presented at the PDP'2011 conference [7]. In this paper we illustrate the recent developments, which extend and enrich the functionality of Atheris.

2 Atheris

Atheris is developed in Java 1.6 due to Java's platform independence and the availability of suitable libraries. The first version of Atheris used Jpcap [3] as a Java wrapper for the libpcap/WinPcap library. Libpcap [4] and its Windows counterpart WinPcap [5] intercept the packets from the kernel space and copy them to the user space, so that traffic analyzers can dissect and store the packets. Libpcap respectively WinPcap are, of course, written in C, therefore, one needs a wrapper to get the packets into Java. Since the development of Jpcap has been discontinued and we encountered serious limitations and bugs, we migrated Atheris to jNetPcap [2], which serves now as the wrapper for the libpcap library. After the completion of the migration to the jNetPcap wrapper, the open source version of Atheris has been made publicly available [1]. The new version of Atheris includes also the following new functionality: We redesigned its GUI from scratch (see Figure 1 for the new visualization window) and included drag and drop functionality to ease its usability. Its workbench functionality has been

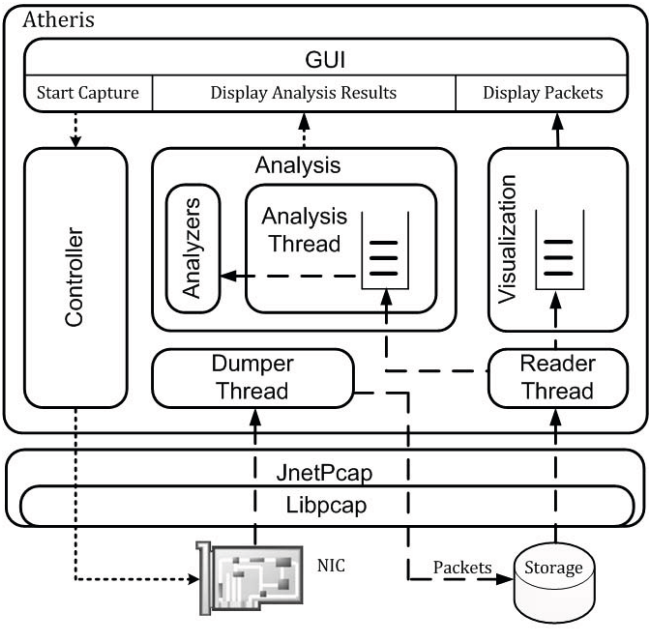

Fig. 2. Multithreaded Capture Engine of Atheris

extended, i.e. it is now possible to export the packets of selected flows or conversations to Wireshark or to CSV files. As a part of the ongoing work this export functionality will be extended to incorporate the export to SQL databases and the statistical software *R*. Finally, the addition of new plots to Atheris has been simplified. Due to its modular architecture, existing plots can be easily extended and to add a new plot, only one interface needs to be implemented.

3 Multi-threaded Packet Analysis

We have now implemented a fully multi-threaded approach to exploit the power of the modern generation of CPUs. Special attention had to be paid to the synchronization of the method calls to the underlying libpcap library. Since libpcap is single-threaded and thus, not reentrant, every call to the libpcap library must be properly synchronized. Without the synchronization of the method invocations, each parallel function call to libpcap can lead to a crash of libpcap and due to that, a crash of the Java virtual machine. Figure 2 illustrates the core architecture of the capture engine. Upon start of a capture session, one thread per interface is used to capture the packets and dump them to the storage device. Of course, as Atheris relies on the libpcap/WinPcap APIs, the usual filter expressions can be applied too, e.g. to capture only the traffic of one IP address. The process of dumping the packets on the storage device is due to avoid

excessive memory consumption and to have the possibility to split large traces (e.g. over 4GB) into several smaller files. After dumping a packet, another dedicated thread is used to read the packets from the packet dump and to put them into two queues. One queue is used to display the packets in the GUI. The other queue is part of a producer-consumer pattern: The analysis thread "consumes" the packets and "feeds" them to the various analyzers. These analyzers can be separated into two distinct groups: The *layer* analyzers collect layer specific information, like the total number of packets and bytes seen up to a given time, time stamps, port numbers etc., in regard to the protocol of the particular layer (e.g. IP or TCP/UDP). The *connection* analyzers collect the same information, but for a distinct connection, which could be a flow or a conversation (bi-flow). In addition, the evaluation of statistical session information is carried out by this thread too. For links with high speed, e.g. GBit/s links, there exists also the possibility, just to analyze the packets without storing them. This is beneficial to avoid storage exhaustion and to circumvent that the capture throughput is dependent on the read-write speed of the storage device.

4 Conclusion

In this paper, we presented the recent development of the open source traffic analyzer Atheris. We reported about its new functionality and its multi-threaded packet processing design. To the best of our knowledge Atheris is the first publicly available multi-threaded implementation of a graphical traffic analyzer. In future releases we plan to incorporate P2P functionality to enable the monitoring of applications in a fully distributed manner.

References

1. Atheris, `https://sourceforge.net/projects/atheris/`
2. jnetpcap, `http://jnetpcap.com/`
3. Jpcap, `http://netresearch.ics.uci.edu/kfujii/jpcap/doc`
4. Libpcap, `http://www.tcpdump.org`
5. Winpcap, `http://www.winpcap.org`
6. Baset, S., Schulzrinne, H.: An analysis of the Skype peer-to-peer internet telephony protocol. In: INFOCOM 2006, pp. 1–11 (2006)
7. Eittenberger, P.M., Krieger, U.: Atheris: A first step towards a unified peer-to-peer traffic measurement framework. In: 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP 2011), pp. 574–581 (2011)
8. Eittenberger, P.M., Krieger, U., Biernacki, A., Markovich, N.: Integrated measurement and analysis of peer-to-peer streaming traffic by the java tool Atheris. In: P2P 2010, pp. 155–157 (2010)
9. Hei, X., Liang, C., Liang, J., Liu, Y., Ross, K.W.: Insights into PPLive: A measurement study of a large-scale P2P IPTV system. In: Proc. of IPTV Workshop, International World Wide Web Conference (2006)
10. Markovich, N.M., Biernacki, A., Eittenberger, P., Krieger, U.R.: Integrated Measurement and Analysis of Peer-to-Peer Traffic. In: Osipov, E., Kassler, A., Bohnert, T.M., Masip-Bruin, X. (eds.) WWIC 2010. LNCS, vol. 6074, pp. 302–314. Springer, Heidelberg (2010)