VISUALIZATION TECHNIQUES FOR THE ANALYSIS OF NETWORK SIMULATION RESULTS

by

Christopher Main

A Proposal Submitted to the Honors Council

For Honors in the Department of Computer Science

September 18, 2012

Approved: ______________________

Luiz Felipe Perrone
Thesis Advisor

____

Stephen Guattery
Chair, Department of Computer Science

1 Introduction

The use of graphics to explore interesting characteristics of data has exploded into popularity in the past fifty years. To a great extent, this increased attention is due to the work of the pioneers of exploratory data analysis, most notably John Tukey [1] and William Cleveland [2]. Both of these statisticians showed that graphics were a very effective tool in analyzing large datasets. The other major contributing factor to the increase in use of data graphics was the widespread adoption of the personal computer. With computers accelerating the amount of data being produced, collected, and stored, data visualization became a necessity in gleaning knowledge from a sea of numbers.

Network simulation is one area of scientific research that creates massive amounts of data [3]. Simulation has become a popular tool in network research because physical networks are often expensive to build and difficult to test with precision. Simulation allows researchers to conduct experiments with high-quality models of real world entities and observe how various changes to a network affect its operating characteristics. Simulation has the added benefits of making certain metrics easier to observe and giving the experimenter precise control over the experimental scenario. In many cases the complexity of the simulation model leads to numerous potentially interesting streams of data which, depending on the length of the simulation and the sampling interval, can produce millions of data points.

In making sense of the large amount of data produced by network simulations, graphical exploratory data analysis can be a very effective tool. Modern data visualization tools allow users to explore data in interactive ways that were simply not possible fifteen years ago. The goal of my research is to survey various modern visualization techniques and to evaluate their suitability in the study of network simulation datasets. To support this goal, I will construct a tool that will allow researchers to explore interesting characteristics of their simulation data and then produce statistically valid, publication-quality graphics.

2 Motivation

Numerous credibility problems exist in much of current network simulation methodology and output analysis. Researchers often publish simulation results that lack repeatability and are statistically biased. Furthermore, many of the published results in network simulation that utilize plots fail to include confidence intervals. These publications have made their way to some of the most prestigious conference venues, including the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) [4].

While frameworks such as Akaroa2 [5], STARS [6], and SimProcTC [7] address some of the above problems, none of them have made statistically rigorous and presentable simulation experiments available to users who are not experienced in network simulation. Perrone et al. [8] argue that network simulation is a powerful educational tool for both graduate and undergraduate students, yet much of a student's time is spent learning the inner-workings of a particular simulator. A tool that makes credible simulation experiments accessible at the undergraduate level would be invaluable in both networking and simulation courses. Such a tool would also allow researchers in the field of network simulation to publish statistically credible and repeatable results.

A crucial part of reporting experimental results is making them understandable and meaningful. Years of research have shown that presenting information visually greatly improves the interpretability of results [1]. There is a serious need in the network research community for a tool that allows users to generate such graphics in an intuitive manner.

3 Previous Work

The Simulation Automation Framework for Experiments (SAFE) [9] addresses many of the aforementioned issues in network simulation research. It automates much of the network simulation workflow, allowing users to focus on science rather than workflow while ensuring that credible results are obtained. SAFE is built to support the open source and widely used ns-3 network simulator. This allows many of ns-3's contributors to also be involved with the development of SAFE.

SAFE supports two types of users, defined as follows. A power user is defined as an individual who is comfortable with the inner-workings of ns-3 and experienced with UNIX system applications such as `ssh` and `sftp`. A novice user, on the other hand, may have little to no knowledge of ns-3, and will access the system via a web browser. While a graduate student would likely fall into the power user category, an undergraduate would likely fall into the novice user category. SAFE is designed to serve the two types of users with different interfaces supported by a common backend. Figure 1 shows the gateway for both novice users and power users into the framework. Power users can access SAFE through both the web user interface (WUI) and a command line interface (CLI). Novice users will want to access SAFE solely through the WUI.

4 Project Description

While much work has gone into the behind-the-scenes functionality of SAFE, the front-end and output processing portion of the framework still needs to be developed. The visualization tool will integrate tightly into to core of the SAFE framework, providing static plots to power users and both static and interactive plots to novice users.

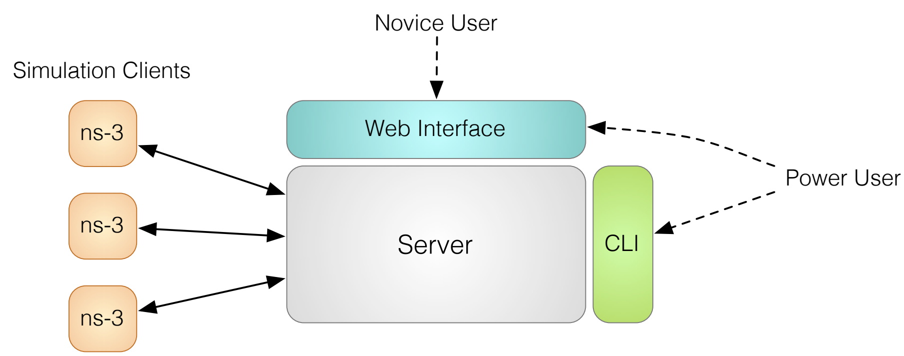

Figure 1: High level view of SAFE's architecture.

Meeting the visualization goals of SAFE's diverse user-base requires a flexible visualization tool. There are two main components to this visualization tool: an interactive view into simulation data and a straightforward interface for generating customizable, publication-quality graphics. Having these two components allows both researchers and students to easily extract the information that they require from simulation results. The development of these components will follow guidelines defined in data visualization literature.

The field of interactive data visualization is relatively young and still experiencing rapid growth, and thus many undocumented problems often present themselves in developing tools in this area. Network simulations can easily produce millions of data points, any of which could be of interest to the experimenter. For this type of data, a micro-macro visualization technique is particularly useful [10]. Such a visualization gives the viewer an overall picture of the data, while maintaining the ability to zoom into any particular data point. Ben Shneiderman's Visual Information Seeking Mantra directly applies micro-macro visualizations to modern, interactive computer graphics: "Overview first, zoom and filter, then details-on-demand" [11].

The interactive view of simulation results will consist of two main displays. The first display will be a micro-macro time-series plot of the entire experiment execution. The user will be able to select multiple metrics through an intuitive drag-and-drop interface, and have them plotted on the same grid. An overview plot with a visual slider will be placed directly below the main plot, a technique commonly referred to as overview plus detail, or context plus focus [11]. Such a plot allows the user to see how a metric evolved over the course of the experiment, compare it to another metric, and then zoom into to as little as a single data point from any metric.

The static view of simulation results will give users a variety of options in generating customizable static plots. The user will be presented with a web form that offers a variety of options in generating static plots. Plot types will include scatterplots, box-plots, time-series plots, and a variety of plots for distribution comparison. The tool will ensure that best practices as identified by data scientists such as Cleveland [12; 2], Tukey [1], and Tufte [10; 13] are followed for each plot type. For example, Cleveland [2] notes that residual data can interfere with the perceived visual fit of data plotted on a scatterplot. For this reason it is much more effective to fit or smooth the data using a technique such as locally weighted scatterplot smoothing (LOWESS).

5 Methods

SAFE's visualization tool will utilize two libraries for the majority of plot creation. The first library, D3.js, is open source and JavaScript-based, developed after years of research into best practices in visualization [14]. The second library, ggplot2, is also open source but is based around the R statistical analysis platform. The ggplot2 package is based on Leland

Wilkinson's *The Grammar of Graphics* [15], giving it a strong theoretical foundation and the ability to create any meaningful plot imaginable [16]. I will also investigate additional libraries or abstractions of the above libraries, such as Rickshaw [17] when developing the tool.

One major advantage of D3.js is that it utilizes Scalable Vector Graphics (SVG), which can be resized and zoomed to any resolution without noticeable quality loss [14]. SVG based plots also have the advantage of having a well defined structure that can be easily converted into other commonly used vector formats. Both of these factors are significant advantages of SVG-based plotting libraries over ones that utilize the HTML5 canvas element.

To streamline the integration of R and ggplot2 with the Python-based web backend, I will use RPy2, an R interface for Python. Working in tandem, ggplot2 and RPy2 make the production of easily downloadable, publication-quality plots nearly straightforward. The use of RPy2 also keeps SAFE's codebase consistent in its use of Python, which leads to more maintainable, understandable code. I will employ the same principles when integrating the WUI of SAFE with the core backend.

Django is an open source, Python based web development framework that follows the model, template, and view (MTV) design pattern. I chose to utilize Django in building SAFE's WUI because it is the most powerful, general-purpose web framework available in Python. A Django based WUI will integrate seamlessly with the core backend of SAFE which is written in Twisted, a Python based networking engine. I will develop this part of SAFE after the core of the visualization tool is complete, for which progress should mirror the following schedule.

The first step in building the visualization tool will be designing and constructing the

WUI in which it will reside. The next step will be constructing the interactive component of the visualization tool. The interactive component will be built up over time, starting with static plots, and then adding an interactive layer on top of them. This goal is projected to be complete by late fall. The final component will be constructing the static visualizations to support both novice and power users. This part of the project is projected to be complete by early winter. Finishing touches on the usability and design of the interface as a whole will be the final steps of development, which is projected to be complete by early to mid-winter. After the tool is complete, I will investigate how it performs in the workflow of common experiments.

6 Conclusion

The aim of data visualization goes far beyond beautiful design. Data visualization reveals patterns and knowledge that would be nearly impossible to quickly discern with rote data analysis. The amount of data generated by network simulations make data visualization a powerful tool in analysis of results. Coupling a strong visualization tool with built-in techniques for reporting credible results would be a strong contribution to the network simulation community.

The work to be conducted falls under the umbrella of many disciplines including statistics, network simulation, graphic design, and cognition. SAFE's visualization tool will be constructed after careful study of the research literature in these fields. The ultimate goal is create a tool that lets the data speak for itself, effortlessly revealing its knowledge to all interested observers, from researchers to students. I expect that this work will be a significant contribution to the advancement of SAFE, which will benefit the ns-3 community.

References

[1] J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley Series in Behavioral Science, 1st ed. (Addison-Wesley Pub. Co., Reading, Mass., 1977).

[2] W. S. Cleveland, *Visualizing Data*, 1st ed. (AT&T Bell Laboratories, Murray Hill, N.J., 1993).

[3] G. F. Riley and M. H. Aremar, "Simulating large networks - how big is big enough?" in "Proceedings of the 1st International Conference on Grand Challenges for Modeling and Simulation," (2002).

[4] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: The incredibles," ACM SIGMOBILE Mobile Computing and Communications Review, **9**, 50 (2005).

[5] A. Koepke and H. Woesner, "The ns-2/Akaroa2 project," Tech. Rep. TKN-01-008, Telecommunication Networks Group, Technical University of Berlin, Berlin, Germany (2001).

[6] E. Millman, D. Arora, and S. Neville, "STARS: A framework for statistically rigorous simulation-based network research," in "Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA)," pp. 733–739 (2011).

[7] T. Dreibholz, E. P. Rathgeb, and X. Zhou, "SimProcTC: The design and realization of a powerful tool-chain for OMNeT++ simulations," in "Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools '09)," pp. 75:1–75:8 (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2009).

[8] L. F. Perrone, C. Cicconetti, G. Stea, and B. C. Ward, "On the automation of computer network simulators," in "Proceedings of the 2nd International Conference on Simulation Tools and Techniques," SIMUTools '09, pp. 49:1–49:10 (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2009).

[9] B. Ward, "A framework for the automation of discrete-event simulation experiments," Bucknell University Undergraduate Honors Thesis. (2011).

[10] E. R. Tufte, *Envisioning Information*, 1st ed. (Graphics Press, Cheshire, Conn., 1990).

[11] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in "Proceedings of the 1996 IEEE Symposium on Visual Languages.", pp. 336–343 (1996).

[12] W. S. Cleveland and R. McGill, “Graphical perception: Theory, experimentation, and application to the development of graphical methods,” Journal of the American Statistical Association, **79** **387**, 531 (1984).

[13] E. R. Tufte, *The Visual Display of Quantitative Information*, 2nd ed. (Graphics Press, Cheshire, Conn., 2001).

[14] M. Bostock, V. Ogievetsky, and J. Heer, “D3: Data-driven documents,” IEEE Transactions on Visualization and Computer Graphics, **17** **12**, 2301 (2011).

[15] L. Wilkinson and G. Wills, *The Grammar of Graphics*, 2nd ed. (Springer, New York, 2005).

[16] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*, Use R!, 1st ed. (Springer, New York, 2009).

[17] “Rickshaw: A JavaScript toolkit for creating interactive time-series graphs,” Availabe at `http://code.shutterstock.com/rickshaw/` [Online; accessed 30-June-2012] (2012).