

# Experimental Methodology For Real Overlays

Diego Dujovne, Thierry Turletti, Walid Dabbous

Projet Planète, INRIA  
2004 Route des Lucioles, BP93  
Sophia Antipolis, France

{diego.dujovne, turletti, dabbous}@sophia.inria.fr

## 1. INTRODUCTION

The validation of new network protocols is a complex task requiring mathematical analysis, simulation and experimentation with networking platforms. Recently, two types of experimentation platforms with different objectives were developed : emulation testbeds offering controlled conditions and experimentation overlays like PlanetLab and OneLab, offering real networking conditions. In this paper, we propose an experimentation methodology to validate network experiments on real overlay platforms. This methodology aims to enable researchers to follow a concrete structure, defining a common set of experimental steps, which involves the definition, execution, processing, analysis and storage of the statistics obtained by experimentation.

As stated before, experimentation platforms can be classified in two types : *Emulation testbeds*, which offer controlled conditions for the experiments [1], [2] and *Real Overlays* aiming to do experimentation on top of production networks, which offer access through virtualization to real working conditions [3], [4].

In real overlays, the traffic generated by an experiment shares real paths and restrictions with other users' traffic. Since networking conditions vary with time, experimental results obtained will also vary. To take into account these varying conditions, it is important to use a common methodology to first establish the experimental setup, executing the experiment, capturing, processing and storing the results. Provided that a strict methodology has been followed, researchers are able to compare between experiments which were executed under the same setup from one run to the following. For example, if the load increases, it is interesting to be able to check whether the protocol or application under test will react (or not) to this change. In order to analyze this possible reaction, the traffic variation load must be registered.

The methodology we propose is based on the main objec-

tive of networking experimentations: the validation of an algorithm, protocol or application under realistic conditions. Unlike emulation testbeds, whose aim is controlling network conditions so that one is able to reproduce the same experiment, real overlays provide no control on network conditions. Basically real overlays will allow to repeat experiments with the same setup in a different time or place and to analyze in detail the effect of varying network conditions on the performance on the protocol under study.

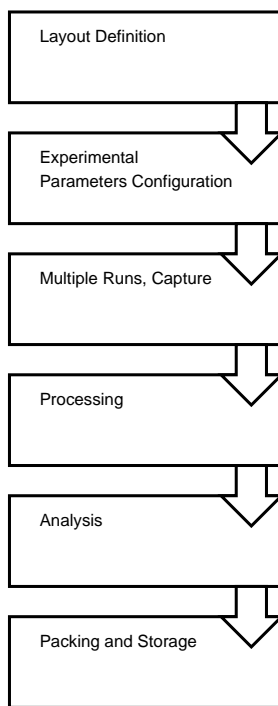
From this analysis, several important items arise. The layout (or the setup) of the experiment must be well defined; this is critical if part of the experiment is executed within a wireless network since not only traffic load or delay can change with time, but also transmission conditions due to environmental factors. All the participating devices (including stations, access points, routers, switches, links) have specific configuration which must be saved in order to reproduce these conditions. During the execution of the experiment, the network variations must be monitored and registered too, in order to correlate them with the captured data. Another important item is the data capture process: on the one hand, this process describes all algorithms and statistical functions to obtain the results; on the other hand, raw data processing extracts parameters which can be used to compare the different runs of the same experiment between them. Finally, all these items must be classified and stored in an efficient way to enable easy access to the data. In a later stage, this stored data can be used to extract new results, and to build new experiments based on the same setup.

## 2. METHODOLOGY DESCRIPTION

We propose the following methodology, made up of a sequence of six steps for each proposed experiment, as illustrated in fig 1.

The first step, called the *Layout Definition*, includes the description of the environment, the hardware and software components used, and the configuration of each of the participating devices, such as nodes, routers, access points and links. In order to achieve reproductibility, the user must be able to rebuild the same experimentation layout in another place or at another moment by loading and executing the same group of steps.

The second step is called *Parameter Configuration*. With all the devices already configured and tested, the experimen-



**Figure 1: Experimentation Methodology**

tion sequence must be defined in detail: when to capture, when to generate traffic and which kind of traffic pattern, how many times the experiment must be repeated, and at what timescale. The timescale is an important item since the networking conditions can change according to time, for example.

The third step consists of the experiment itself; it is called *Multiple runs and Capture*. This stage executes the runs as many times as defined in the former step, following the predefined sequence. All the devices must have their time bases synchronized, to execute the tasks within schedule and to timestamp the captured packets as they traverse the network. During this stage, raw data from the network is acquired and stored to be further processed. Realtime monitoring is used here to check the evolution of key parameters like traffic load and packet loss during the experiment, so as to discover anomalies or divergences before the processing stage.

The fourth step, called *Processing*, aims to handle offline statistics. This stage comprises synchronizing the packet timestamps from the packet traces, correlating and detecting missing packets, filtering and extracting parameters from the stored statistics.

The fifth step, called *Analysis*, aims to present the experimental results. Once the data has been processed, the results must take a human readable form as a graphical representation of the data. Timelines, flowcharts, bar graphs and curves are typical examples of these representations.

The sixth step is called *Packing and Storage*: Here the data is classified, organized and stored in an easily recoverable

way. The package includes the raw data, network layout, system configuration setup and processed results. This will enable researchers to configure the same layout and setup to execute a new experiment, and also to have the whole data for the experiment definition.

### 3. EXPERIMENTATION TOOL PROPOSAL

To implement this methodology, we propose to build an integrated experimentation tool, supporting each of the former steps. Basically, there are two related objectives; the first one is to keep a simple interface for researchers, so they can concentrate on the experiments themselves. The second one, is to build a tool that integrates the definition, method, processing and storage under the same umbrella. To achieve this goal, we define a *run* as the atomic unit, and an *experiment* as a group of runs. On the same experiment, common parameters between runs are defined as fixed or variable. For each run, these parameters are kept fixed in order to analyze the influence on the uncontrolled environment. Runs are executed on top of a real overlay, like Planetlab or Onelab, in possible different times, under the influence of potentially varying network conditions (including varying load, delay and packet loss).

There are several items to consider when designing such a tool, especially those related to the processing and analysis stages. In particular, it is necessary to synchronize stations to schedule events on the network. Packet timestamps synchronization can be achieved through NTP and through beacon timestamps on wireless nodes. Also, the list of required parameters must be available on the packet traces.

Current proposed experimental environments on top of PlanetLab such as Plush [5] or Weevil [6] do not include the processing and analysis stages which are indispensable to display the obtained results [7]. On the other side, we have developed Wismon, a wireless probe that supports monitoring, data processing and analysis on wireless experimentation platforms [8]. The experience we acquired with WisMon will make easier the development of the new tool that will implement the experimentation methodology. Another important feature to add for this upcoming tool is realtime monitoring during each run, so that a run may be aborted when necessary (e.g. after hardware failures or unexpected instabilities). To enable realtime monitoring, only a few key parameters along with their bounds must be supervised in order not to add a too much measurement overhead on the overlay network.

Within the same tool, at the end of the experiment all the raw and processed data including the layout and configuration information should be stored and packed. This stored data can be used afterwards in different ways: 1/ to verify again the validity of results; 2/ to repeat the same setup in a new experiment and obtain new packet logs from the stations; and 3/ to analyze the processed data using different criteria.

This tool will work on top of a real overlay like PlanetLab or Onelab for wide area representability. The overlay also provides the infrastructure to deploy operating system images, support monitoring and clock synchronization, transfer packet traces and allow remote access to the stations.

We believe that the networking community will highly benefit from a tool with the described features. For this purpose, we target the design of a freely available tool. Its widespread use should allow the exchange of compatible data between researchers to cross-validate the results. Hopefully, this will also create a user community which will contribute to build a repository for networking environments and layouts.

#### 4. RELATED WORK

Specific workflow management tools exist for grid applications, as it was shown by Jia Yu et al. [9]. Several experimentation management tools exist in the context of the Emulab project [10], which are designed for controlled experimentation environments and they cover experiment definition using NS2 compatible scripting, control and data storage. On the “real overlays” side, there are few proposals, like Plush [5], and Weevil [6] both oriented to deployment, remote execution and monitoring of distributed applications on overlays. To the best of our knowledge, there is not yet a proposal of a full methodology including the missing steps for processing, analysis and storage.

#### 5. CONCLUSION

We have presented a new methodology for experimentation in real overlay networks, followed by a list of features required to support this methodology. We believe that such a methodology is crucial to characterize the traffic, execute experiments under common methods and promote the exchange experimentation data between research teams. We have put emphasis on the fact that the processing and analysis stages will be of great help to researchers by simplifying their task so that they concentrate on the experiment itself.

#### 6. REFERENCES

[1] <http://www.emulab.net/>[Last visited: June 2007]

[2] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu and M. Singh, “Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols”, WCNC’05, March 2005

[3] <http://www.planet-lab.org/>[Last visited: June 2007]

[4] <http://one-lab.org/wiki/view/OneLab>[Last visited: June 2007]

[5] Albrecht, J., Tuttle, C., Snoeren, A. C., and Vahdat, A. 2006. “PlanetLab application management using plush”. SIGOPS Oper. Syst. Rev. 40, 1, Jan. 2006, pp. 33-40.

[6] Wang, Y., Rutherford, M. J., Carzaniga, A., and Wolf, A. L. 2005. “Automating experimentation on distributed testbeds”. In Proceedings of the 20th IEEE/ACM international Conference on Automated Software Engineering (Long Beach, CA, USA, November 07 - 11, 2005). ASE ’05. ACM Press, New York, NY, 164-173.

[7] D. Andersen, N. Feamster, “Challenges and Opportunities in Internet Data Mining”. Parallel Data Laboratory, Carnegie Mellon University, Research Report CMU-PDL-06-102, Jan. 2006.

[8] <http://www-sop.inria.fr/planete/software/WisMon/index.html> [last visited: June 2007]

[9] Jia Yu and Rajkumar Buyya, “A Taxonomy of Workflow Management Systems for Grid Computing”, Technical Report, GRIDS-TR-2005-1, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, March 10, 2005.

[10] E. Eide, L. Stoller, J. Lepreau: “An Experimentation Workbench for Replayable Networking Research”, Fourth USENIX Symposium on Networked Systems Design and Implementation (NSDI ’07), Cambridge, MA, Apr. 2007.