A Portal to Support Rigorous Experimental Methodology in Networking Research

Guillaume Jourjon, Thierry Rakotoarivelo, and Max Ott

NICTA*
Australian Technology Park
Eveleigh, NSW, Australia
firstname.lastname@nicta.com.au

Abstract. Whilst dealing with topics that are more and more influenced by physical properties of the underlying media, the networking community still lacks a culture of rigorous result verification. Indeed, as opposed to most of the science and engineering fields there are very few benchmarks to test protocols against. Furthermore, in most publications the authors do not give the community access to the raw results or details of the performed experimental procedures. Therefore it is impossible to accurately reproduce their experiments. We propose to solve this problem by extending the state of the art experiment tool OMF with a public portal. This portal, while providing the experimenter with access to experimental resources, also provides the community with a system for comprehensive experiment description and result verification. The collection of both the measurement set and the experiment's description is done in a transparent manner for the experimenter, who can decide to publish them via the portal once the research is mature enough.

Keywords: scientific method, data quality, measurement, testbeds, OMF.

1 Introduction

Research in computer networking has steadily increased for the past twenty years, as highlighted by the increased number of submissions to an increasing number of conferences and publications, e.g. submissions to some leading conferences quadrupled over that period [2]. Similar to other scientific fields, experiments are a cornerstone of research in networking. However the experimental approaches used by the networking community often suffer from shortcomings such as inconsistency with widely adopted scientific methods (e.g. hypothetico-deductive), lack of proper replication, lack of reproducibility, or lack of data/result quality analysis. While these issues do not prevent technical innovations, a research initiative in computer networking needs to address them to be able

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

T. Korakis et al. (Eds.): TridentCom 2011, LNICST 90, pp. 223–238, 2012.

to claim any scientifically sound contribution. Adopting a scientifically rigorous approach to experimentation is often challenging due to the cost, complexity, and scale of the involved experimental resources, and some potential limitations in the training of the research investigators.

There is a continuous range of experimentation tools available to networking researchers. On one end, simulators (e.g. NS3 [3] and OMNet++ [4]) provide relatively inexpensive, yet valuable results on the performance of a new scheme, model, or technology. However, they inherently make simplifying assumptions impacting on the accuracy of the produced results. On the opposite end, prototype deployments in real-world trials offer comprehensive and realistic results, but at often prohibitive costs. Networking testbeds such as PlanetLab [5], Orbit [18], or Emulab [20] provide an effective alternative, where new technologies are evaluated in controlled realistic environments and scales. However, they are often deployed and used in an ad-hoc and independent manner. In some cases, they are even solely built and used for a specific study, and are seldom maintained or re-used after its completion. The cOntrol and Management Framework and its Measurement Library (OMF [17], OML [21]) provide tools and services to support and optimise the usage of testbeds on a global scale, through systematic experiment description and execution, resource virtualisation, and testbed federation.

This article presents a web-based Portal, which offers a complete environment for networking researchers to design, run, track, and record experiments on OMF-managed testbeds. These experiments are in most cases not single events, but part of series of experiments within wider research studies. The goal of this Portal is to facilitate and promote the adoption of a rigorous approach to these experiment series. To achieve this goal, the Portal leverages the tools provided by OMF/OML to provide services supporting researchers (i.e. the Portal's users) at each step of a scientifically sound experimentation methodology. In that regard, the Portal is a akin to a digital laboratory notebook and assistant. For example, it provides a wiki-based interface to describe and version experiments, a reservation and batch-processing system to automate their executions on OMF-managed testbeds (e.g. find available resources conforming with requested experiment topology and run it), a repository to record execution contexts (e.g. used resources, application versions, etc...) and collected measurements, and an interface to various tools for result analyses. These features enable scientifically sound studies involving replications of reproducible experiments with data quality information. The Portal is independent of any specific testbed and different third parties could host Portal instances interfacing with different types of testbeds. Its current implementation is compatible with any testbed with deployed OMF/OML installations (e.g. Orbit, NITLab, NICTA [7]).

This remainder of this paper is as follows. Section 2 describes the four issues related to scientific experiments which motivate the introduction of the proposed Portal. Section 3 then presents the general architecture of the Portal and its different features and the components that support researchers in addressing

this experimentation issues. The main related works are presented in Section 4. Finally, Section 5 concludes this article and presents some future work.

2 Motivation

An experiment is a key component shared by different scientific methodologies, such as the induction method or the hypothetico-deductive method [16]. The latter method is outlined in Fig. 1. A scientific study starts with a set of *observations*, e.g. traffic traces from routers in a network, performance of a networking protocol in a given setting, popularity/usage of content in a Video-On-Demand service. Researchers then construct different *models* to explain the observations and in some cases to alter their values, e.g. classify traffic into elephant vs. mice flows, improve protocol throughput under same setting, explain content popularity and improve pre-caching. From these models, they then derive *predictions*, which are *hypotheses* (H_A) and their null counterparts (H_0), e.g. traffic pattern will not be different for similar network, changing algorithm X will improve protocol throughput under same setting, content popularity follow a distribution Y. A series of *experiments* are then realised to test these hypotheses through the generation of new sets of *observations*, e.g. collect new router traces from other similar networks, measure the performance of the modified protocol, follow and collect popularity metadata of new video contents. Such a series includes experiment re-runs under same but also different conditions to remove confounding factors and increase precision as described below. Finally the analysis of these new observations will allow researchers to falsify and/or refine some of their models, the new observations are added to the set of existing ones, and the all process is repeated with the refined models and/or new ones. This scientific

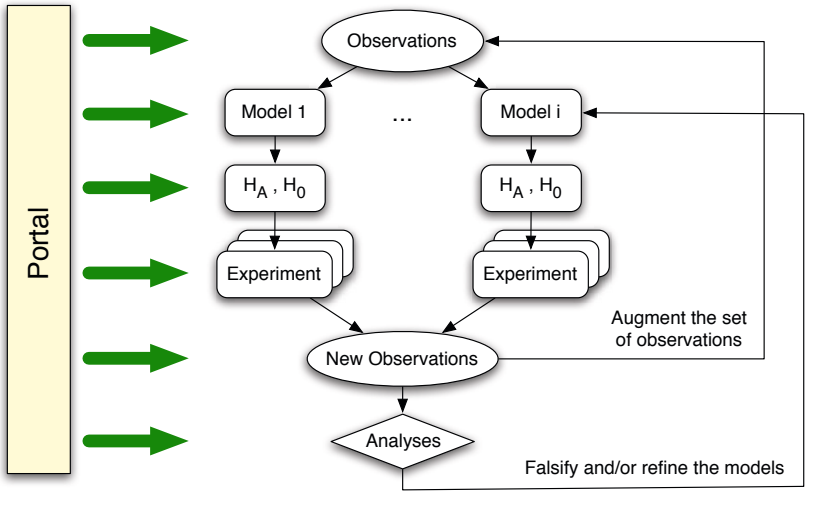

Fig. 1. The hypothetico-deductive scientific method and its support by the Portal

approach has dominated research and knowledge advancements in many fields [19,14], but is unfortunately not widely adopted by the networking research community as illustrated later in this section. The Portal proposed in this paper aims at providing supports and services for networking researchers at each step of this scientific approach, as illustrated on Fig. 1 and described in details in Section 3.

When performing an experiment within this scientific framework, researchers are faced with many factors which have the potential to confound the experiment's results. For example, time could be a confounding factor when the experiment is made with a duration not relevant to the studied system, e.g. collecting popularity metadata for video contents over only N-first months may fail to capture later emerging trends. Similarly scale/space could also be a confounding factor, e.g. observed traffic patterns for only a given network size may fail to capture patterns operating at other sizes. The hardware devices used in the experiments may also be a confounding factor, e.g. the throughput performance of a new wireless ad-hoc protocol relying on information from the underlying network device may depend on how accurate the device is at measuring this information. The solution to mitigate these confounding factors is often to use an experiment replication strategy which is adequate to the studied system, e.g. repeat the experiment at different time, duration, space/scale, and/or with different hardware. The proposed Portal provides the researchers with tools and services to facilitate such replications, through the use of systematic experiment description and instrumentation with OMF/OML and automatic experiment deployment and measurement collection.

Other issues that need to be addressed in a scientific experiment is the accuracy and the precision of its produced results, with accuracy being the closeness of a variable's measurement to its agreed *true* value, and precision being the closeness to that *true* value achieved by repetitive measurements under similar conditions. Accuracy is often inherent to the measurement device, process, or software and can be controlled by adequate calibration procedures. Moreover, accuracy may be secondary for some studies, e.g. when comparing the delay performance of different ad-hoc protocols, even if a constant offset is introduced by the measurement stack (thus impacting the absolute delay accuracy), the subsequent analyses can still use the relative delay differences. Precision is a cornerstone attribute that needs to be quantified and reported for all experiment results. The analyses and conclusions based on a given result should not be considered valid if no information on its precision is provided, e.g. protocol A has a mean delay of x while protocol B has a mean delay of $y > x$, the claim "A is faster than B" is invalid if the (un)reported precision is greater than $|x - y|$. In most cases precision can be estimated by reproducing the same experiment under the same conditions for a given number of time. The systematic description/instrumentation and automatic deployment/measurement features of the proposed Portal assist the researchers in reproducing their experiments and estimating their result precisions.

A final point regarding scientific experiments is their need to be comprehensively and unambiguously described when reporting their results. An experiment

Table 1. Some shortcomings of the *Evaluation* section of 33 articles from the 2010 ACM SIGCOMM conference

Description	Number of article (%)
Use of a model without indications or tests of its validity	4 (12%)
Missing indications of adequate replication	7 (21%)
Missing indications of the precision of obtained results	19 (58%)
Missing descriptions of experiments (thus not peer-reproducible)	3 (9%)
None of the above, i.e. "ok" articles	10 (30%)

described in such a manner would then be reproducible by peer researchers. While such a practise is not common in the networking field, it is part of the peer-review approach used in other fields such as chemistry or physics. The proposed Portal would facilitate such practise in the networking community by offering a repository for systematic experiment descriptions and for archived results. Thus peer researchers could have an easy access to the original experiment description and context to reproduce it, and also an access to the original resulting data to compare their reproduced results with.

To illustrate the need for a more rigorous approach in the networking community and a tool supporting such improvement, we carefully analysed the specific "Evaluation" section of the 33 articles from the leading ACM SIGCOMM 2010 conference, and specifically checked for 4 experimental shortcomings related to the different points presented earlier in this section. Table 1 presents these shortcomings and their associated number of articles. We found that most of these articles failed to provide any indication on the precision of their results (i.e. 58%), and only almost a third of them did not present any of these shortcomings (i.e. 30%). These are initial illustrative counts with no generality claimed from them. A proper accurate and precise analysis of this issue would require different teams to assess the same set of articles (i.e. *accuracy* through agreements across team on "if paper x shows issue i"), and these articles being from both different and same conferences across time (i.e. *precision/confounding factors* through replication).

3 An Architecture to Facilitate Scientifically Rigorous Experiments

In a previous contribution we developed IREEL [11], an interface to the infrastructure of an OMF-managed testbed, which provides a versatile e-learning platform for networking courses at universities. This new platform has been recently used within a Computer Science course at the University of New South Wales, where it allowed the automatic execution of more than 9000 experiments in less than a month. We leveraged our experience in building IREEL to propose a new Portal framework, which will help researchers to address the experimentation issues described in Section 2, thus promoting scientifically sound research

contributions. Compared to the IREEL e-learning platform, the proposed Portal framework faces two new challenges. The first one is to provide the aforementioned digital lab notebook and assistant features. The second one is to capture in an exhaustive manner an experiment's execution context, including the state of the network. The latest is a critical component for the reproducibility of experiments.

In the remaider of this section, we will present the general architecture of the Portal and in particular the management and measurement framework that orchestrates the experiment run. Then, we will focus on the different experimentation issues mentioned in Section 2, and show how the proposed Portal architecture can be used to address these issues.

3.1 General Architecture

The first goal of the Portal is to provide a general and modular reservation interface to any OMF-managed testbeds. In this section we first give an overview of the testbed management framework OMF and its measurement companion library OML. Then, we present the Portal architecture and its features, which will promote the development of scientifically sound networking experimentation.

OMF. Experimental platforms (or testbeds) are instrumental for the evaluation of new network technologies. In many cases, these testbeds are solely built and used for a specific research project, and are often not maintained, re-used, or shared. This wasteful approach also limits the independent verification of experimental results by the community. This is a cornerstone of the scientific method, and is further hampered by the lack of an unambiguous way to describe an experiment and enable others to repeat it. The cOntrol and Management Framework (OMF) [7,17] is a suite of management, control, and measurement services and tools for networking testbeds. From a management perspective, OMF provides several software services to access, allocate and manage heterogeneous resources within a testbed. From an experimenter's perspective, it provides a high level domain-specific language to systematically describe an experiment (i.e. its used resources, its required measurements, and its tasks to perform), and a set of software tools to automatically deploy and realise this experiment on one or many testbeds.

Fig. 2 shows a simple overview of OMF's architecture from an experimenter's point of view. As described on this figure, the input to the OMF system is an Experiment Description, which is produced by the researcher (i.e. the user). OMF will then perform all the necessary operations to deploy, configure, and execute the corresponding experiment. While the experiment is running, various measurements are automatically collected through the use of the OML measurement library. OMF and OML have been deployed on several heterogeneous testbeds, and have been used by many researchers worldwide [7].

OML. OML [21] is the OMF Measurement Library, which was first introduced as an additional component of OMF, but is now a stand-alone software which

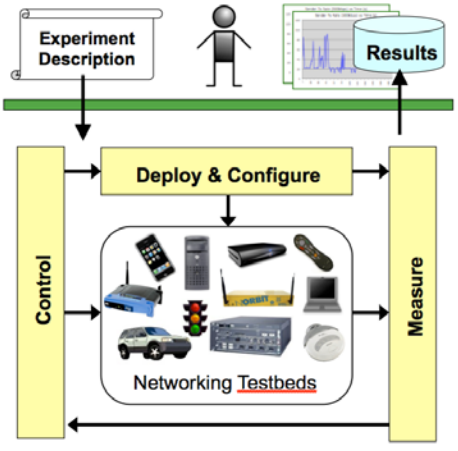

Fig. 2. Overview of OMF architeture from the user's perspective (source: [7])

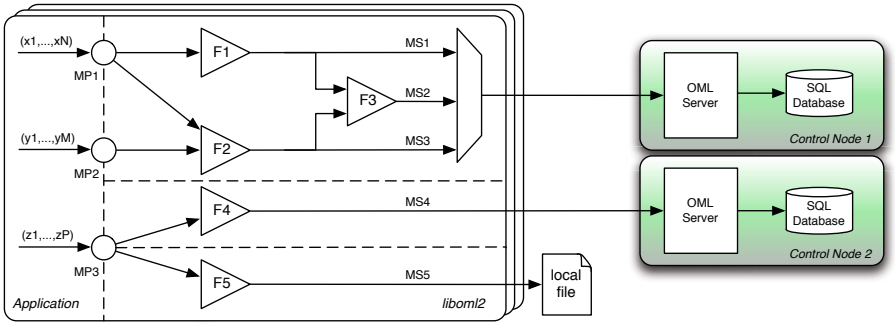

Fig. 3. Measurement data path in OML. The application illustrated defines three measurement points, and the user has configured the library to generate five measurement streams (source: [21]).

could be used to collect and store any type of measurements from any type of application. OML has three components that allow the user to automatically generate and store measurement streams as illustrated in Fig. 3. On the left side of that figure, the user defines some Measurement Points (MPs) within the application. While this application is running, these MPs generate some Measurement Streams (MSs) composed of tuples, which will be injected in the OML client library (i.e. liboml2). These streams are then processed by the client library (e.g. filtered, combined) as defined by the user. The resulting processed streams are stored in a local file or collected by one or many remote OML collection server, which stores them in a database.

OML has been integrated in many applications, such as traffic generators, passive network measurements, GPS coordinate loggers, and pressure/temperature sensor loggers [8]. In the proposed Portal, we developed a measurement analysis component as part of the lab-book support, which automatically generates simple graphs from the measurement database of a given experiment, and allow the import of these measurements into a wiki-based statistical analysis tool.

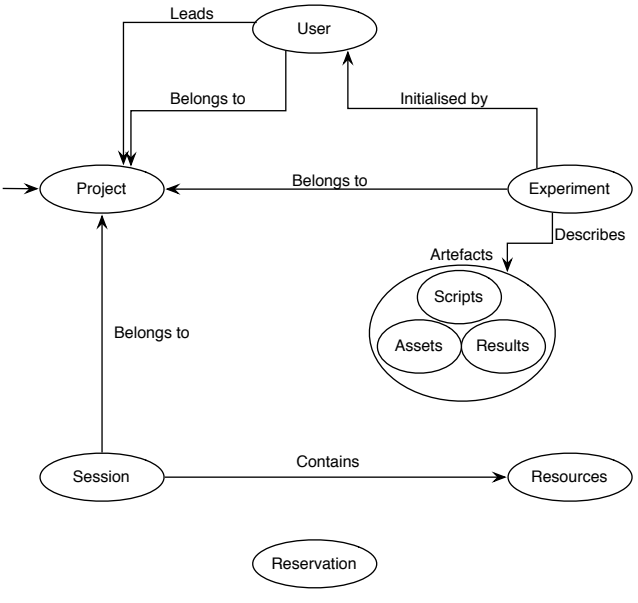

Fig. 4. Pseudo class diagram of the Portal

The Portal. Based on the two main blocks of OMF/OML and IREEL legacy front-end, we have developed a generic testbed portal, which offers basic reservation schemes and also provides a private space for users to record their experiments. Fig. 4 gives a general overview of the architecture of this Portal.

In this architecture, the project is the core component of the use of the testbed. We consider a project as a group of users agreeing to share information and their credential to the testbed or federation of testbed. In this context, when a user registers to the portal, she is given a private project by default. Based on this project, every user can create sub-projects in which she can invite other users.

On the project page on the portal, the different users will be able to contribute on a wiki to the different aspects of the description of the model. They are also able to edit the different experiment description using the OMF's domain-specific language (OEDL) [7]. The different versions of these descriptions are then saved and an historic of the changes is automatically maintained. Difference users can then select different versions of the experiments to execute.

In order to perform experiment runs, a user belonging to a project needs to first log in to the portal and then select which resources of the testbed to reserve. Indeed, the portal allows a user to select a subset of the testbed's resources in order to allow concurrent use. This functionality is particularly useful when the testbed's infrastructure is composed of disjoint sets of resources.

Finally, the portal allows the selective migration of a project's results, experiment descriptions and analyses between public and private wiki pages. This function allows the publication of selected parts of the project, thus simultaneously allowing community peer-review of published materials and protection of ongoing research items. All this process can be synthesised by the workflow diagram presented in Fig. 5.

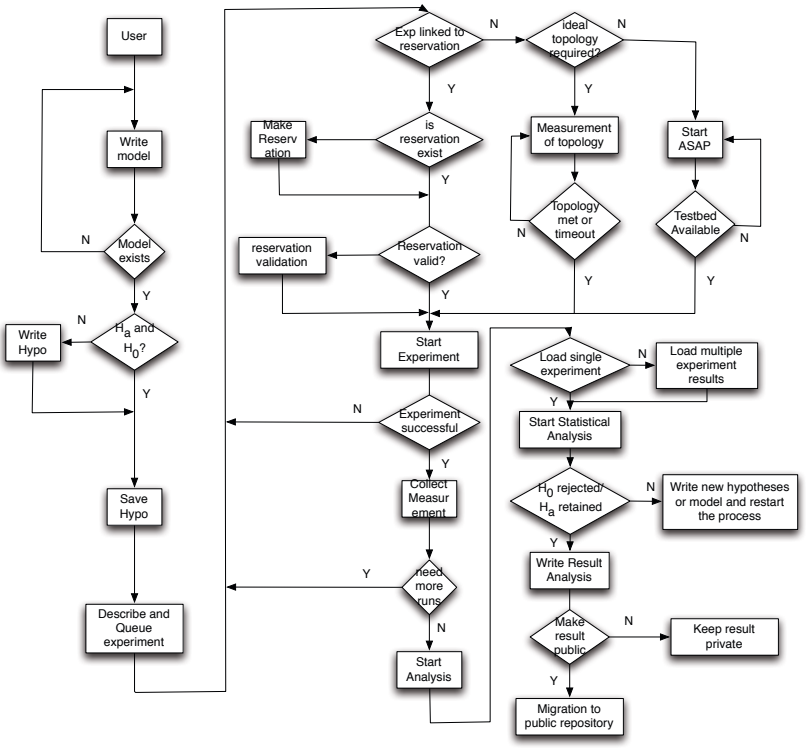

Fig. 5. Portal Workflow Diagram

In the following section, we discuss how this general portal architecture has been modified to embrace the scientific approach as described in Section 2.

3.2 Support for Model Description: A Wiki-Based Lab Notebook

The first step of the previously described scientific approach is the analysis of existing observations. In the context of networking, observations can range from basic delay/loss/bandwidth measurements on a given network to the monitoring of specific kernel or application-level variables.

In order to facilitate these observations in the context of experiment driven networking research and in conjunction with the general portal architecture depicted previously, we introduced a wiki-type lab notebook for each project. This lab notebook has the following characteristics:

- mathematical model and formal description for the hypothesis;
- experiment description using OEDL (c.f. Section 3.3) ;
- wiki-pages;
- post-run analysis (c.f. Section 3.6).

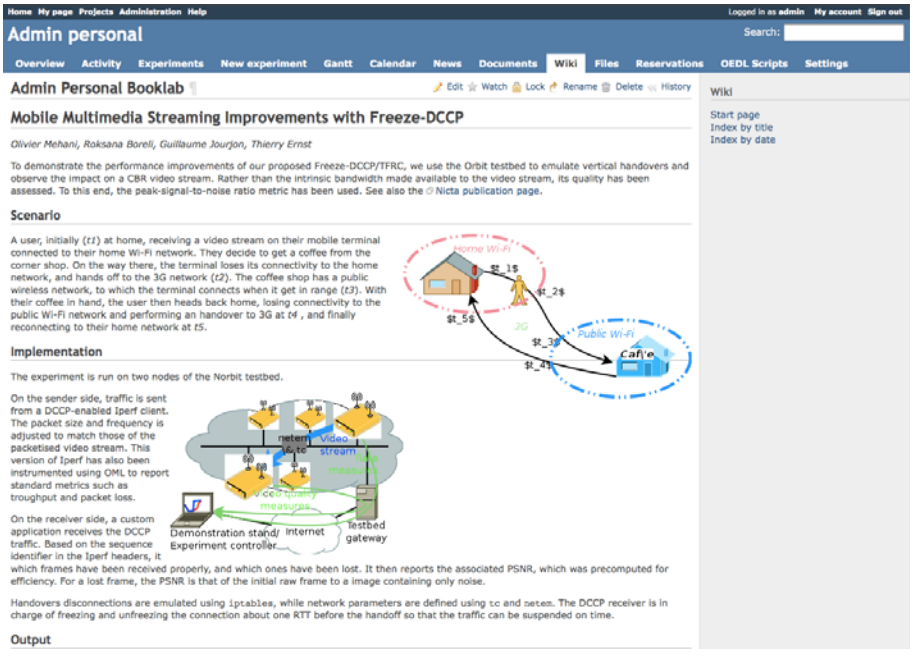

Fig. 6. Presentation of the different component of the portal

Fig. 6 presents the different components of the current User Interface (UI) of the portal. This figure shows an example of a personal project in which we describe the potential evaluation of some DCCP [12] modifications that integrate a freeze mechanism. As a result, the Freeze-DCCP mechanism [15] is supposed to be more efficient compared to DCCP for transferring media streaming content. In the current front-end, the user has access to all the components of his experimentation; the scripts that describe the experiment (*OEDL script* tab), the

reservation system (*Reservation* tab), the experiment queue and results (*New Experiment* and *Experiments* tabs) and the lab notebook extended (*wiki* tab) with the pre and post run analysis capabilities.

3.3 Support for Hypothesis Testing: Experiments and Versioning

Once the first step is accomplished, the user can describe the potential experiments that will help test the hypotheses. These experiment descriptions are in the form of OEDL scripts [7], which can be uploaded by the user via he web front-end as shown in Fig. 6. During this process the user will associate a name to each experiment before uploading the frist version of the corresponding script.

On the backend, this action will initialise an historic for the different versions of the OEDL script. The user can then edit and update this first version directly via the portal's front-end. Every subsequent edition is versioned using a Revision Control System, such as Git [1]. Furthermore, as explained in the general architecture description, the different versions of a script can be migrated to either another private or public project to allow peer verification of results.

Finally, in order to support the testing of different hypotheses (as depicted in Fig. 1) we have extended the OEDL language and in particular the topology description. Indeed, the former OEDL version was only allowing the definition of a static topology, which had to be loaded at the start of the experiment run. Therefore, we have added the support for dynamic topology modifications in OEDL, which will be released with the new OMF 5.4. In addition to this new topology feature, we also integrated a continuous monitoring of the testbed's resources through OML to the portal. This allows the user to retrieve statistical characteristics of the testbeds (e.g. link quality) to better interpret experiment results, and to describe "ideal" topologies to be mapped to the existing resources.

Listing 1 shows a topology example using the former and the "ideal" descriptions. The main difference between these two topology descriptions consists in the anonymising of the node in the "ideal" topology. We explain in the following section how this newly introduced feature is used by the portal to start experiments and subsequently report results.

3.4 Experiment Execution

We have described how the proposed Portal supports the user in performing the first two steps of a scientific approach to a research study, namely the description of models and resulting hypotheses, and the development of experiments to test them. The Portal offers three automatic methods to start the execution of these experiments, as well as supporting the original OMF command-line method:

- *as soon as possible*, i.e. start the experiment as soon as resources are available;
- *as per reservation*, i.e. start when the user's resource reservation period begins;
- *as per topology*, i.e. start when the described topology can be realised.

```
# Topology description using Netem
defTopology('mainTopology') { |topo|
        topo.addNode("myNode_1", prop.node1)
        topo.addNode("myNode_2", prop.node2)
        # We describe the characteristics of the links between node 1 and 2
        topo.addLink("myNode_2","myNode_1",:emulationTool => :netem, :asymmetric => true,
         :ruleID => 3, :delay => '200ms', :bw => '1Mbits', :bwBuffer => 12000,
         :bwLimit => 15000, :loss => '10%' )
        topo.addLink("myNode_1","myNode_2",:emulationTool => :netem, :asymmetric => true,
         :ruleID => 3, :delay => '200ms', :bw => '1Mbits', :bwBuffer => 12000,
         :bwLimit => 15000, :loss => '15%')

        topo.saveGraphToFile()
}
# Ideal topology
defIdealTopology('mainTopology') { |topo|
        # We require two Wireless nodes (WNodes)
        topo.addWNode("n1")
        topo.addWNode("n2")
        # We describe the characteristics of the links between node 1 and 2
        topo.addLink("n2","n1", :RSSI => '< 20db' )
        topo.addLink("n1","n2", :PLR => '> 5%')

        topo.saveGraphToFile()
}
```

Listing 1. New topology and "ideal" topology description

To execute an experiment, the user will first have to configure its initial parameters under the *New experiment* tab as shown on the Fig. 6. In this part of the portal, every registered user of a project can have access to all the experiment descriptions related to that project. The user can then select the experiment to use and also which specific version to execute. After that selection, a configuration page is automatically generated to let the user configure the initial parameters as described in the experiment's OEDL script.

Once the experiment is configured, the user may let the Portal execute it according to one of the three aforementioned automatic methods. For the *as per topology* method, the Portal uses a mapping-problem heuristic, which takes as inputs the monitored link quality information (as described in section 3.3), and evaluates the solutions to the requested "ideal" topology. When an adequate solution is found, the Portal execute the experiment with that topology. If no solution meets the requirements after a certain time, then the Portal executes the experiment with the *best* found mapping and return the observed link information to the user. If link information stability holds, we plan to use topology solutions from previous experiment executions as a parameter to future executions, e.g. for result verification by peers.

3.5 Providing New Observations: Measurements and Their Collections

In order to provide a comprehensive context description of the experiment's execution, we have developed and deployed on our testbed an OML-instrumented application to passively measure wireless link characteristics. As mentioned previously, these characteristics are also used in the experiment scheduler when the

user wants to start their application based on the ideal topology definition. The portal will use this monitoring application at the beginning and the end of the experiment, and will provide the generated map of the testbed as part of the results to the user. Fig. 7 shows an example of such a snapshot of the testbed characteristics. This snapshot provides more information to the user on the potential confounding impact of the environment on the experiment's results.

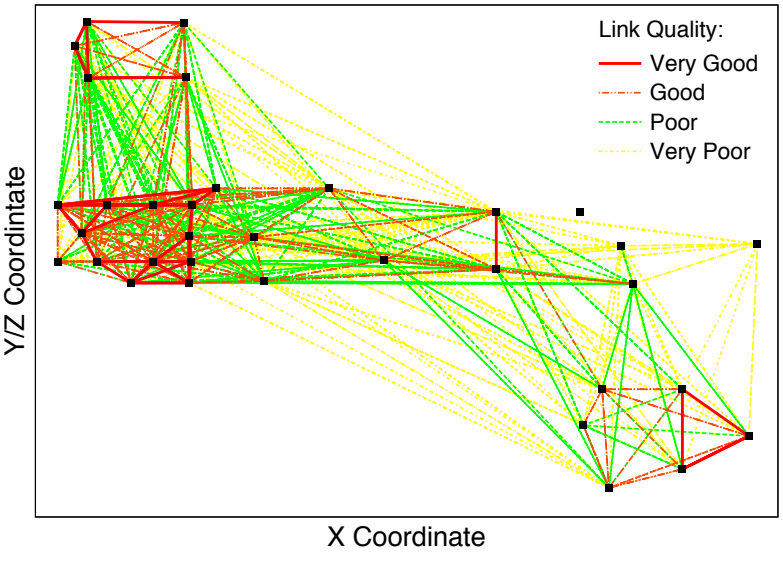

Fig. 7. Observed Testbed Characteristics (RSSI values)

The portal also collects the user-generated measurement through the use of OML [21]. As described in Section 3.1, OML allows a user to extend their applications to create measurement streams that are collected in a server and stored in a database in a transparent way. On the portal side, these measurements are available on a *Result* page once the experiment is finished. This page is accessible through the *Experiments* tabs shown in Fig. 6. In addition, this page also presents a set of automatically generated graphs of each measured experiment variable against its collection timestamp. Finally the whole set of measurement is made available in the lab notebook for analysis.

3.6 Testing the Hypotheses: Result Analysis

The last steps in one iteration of the life cycle of a scientific study (as detailed in section 2) is the analysis of the experiment results and the validation or rejection of the predictions (i.e. hypotheses and null counterparts) based on this analysis. Similar to other science areas, statistical tools are the main instruments used to analyse results in networking. Nevertheless, as outlined in Section 2, even some high quality publications fail to use adequate statistical tools and provide a large

set of quantitative graphs instead of a concise and precise analysis. For example in some articles, in order to prove correlation between two data sets the authors would plot them on a graph instead of using statistical tools to determine the effect size and significance of the correlation.

In order to facilitate the use of statistical tools, we are integrating within the Portal's lab notebook an extension to support the R environment [9]. This integration allows the user to load a measurement set[1] in an emulated R terminal. Once the measurements are loaded, all the R analytic tools are available to the user, who can then perform analyses, run tests and create output graphs. Thus, we believe that this capability will facilitate the generation of a proper statistical analysis of the observations. Finally, the user can link to these analysis results directly from the Portal's lab notebook pages of their project, and selectively mark them as private or public for sharing with peers.

4 Related Works

Few initiatives currently provide a software interface to support researchers in their use of networking testbeds for scientific studies.

MyEmulab [20] is a web-based portal interface to the Emulab network emulator testbed. Emulab provides experimenters with a set of computers, which can be configured into various topologies through emulated network links. The Emulab control framework supports three different environments: simulated, emulated, and wide area networks; and unifies them under a common user interface. The MyEmulab portal provides services to upload an experiment description, acquire and configure the required resources and automatically execute the experiment. Furthermore, it offers a graphical interface to build experimental network topologies and built-in wiki, chat, and versioning tools to allow collaboration between members of a given project. However, MyEmulab does not provide any component to store, access, and archive the measurements produced by an experiment run, or any services to analyse these measurements. MyEmulab and our proposed Portal share many features and the goal of supporting researchers in their scientific studies, with the differences primarily shaped by a focus on differently managed testbeds (Emulab and OMF/OML) and user communities.

MyPLC [5] is the web-based interface to the PlanetLab global research platform, which includes more than 1000 distributed computers hosted by independent organisations. It is the primary large-scale testbed used for experimental overlay and service oriented systems (e.g. distributed storage, peer-to-peer systems). It is limited by its offered default layer-3 abstraction, which does not allow experiments on underlying layers (e.g. new L2 schemes). It uses virtualisation tools to efficiently share the global resources among simultaneous short or long-lived experiments. MyPLC provides a user interface to these tools, and allows researchers to create virtual resources and associate them to their projects (or *slice*). While additional third-party tools are available to facilitate experiment description and execution on PlanetLab (e.g. Plush [10]), the MyPLC portal

[1] This set can result from a single or numerous experiments.

does not interface with them, nor does it provide tools to facilitate the versioning of experiments and their results, the recording of execution contexts, or the analysis of collected measurements.

Similar to MyPLC, the existing NITLab and ORBIT schedulers [6,18] provide web-based interfaces to acquire (or reserve in these particular cases) resources on a given OMF-managed testbed and associate them to a particular experiment. However, these interfaces do not provide any feature which would directly support researchers at the different stages of scientific study involving networking experiments.

Finally, other contributions proposed alternative frameworks to OMF/OML to describe, execute and instrument networking experiments, such as the Network Experiment Programming Interface (NEPI) [13]. NEPI proposes a framework based on a unified object model to describe a networking experiment which could subsequently be executed on different environments (e.g. simulations, emulations, testbeds). This framework is still at a prototyping stage, and does not provide services to help researchers address the different issues related to a scientific study as described in Section 2.

5 Conclusions

In this article we presented a modular testbed portal and its integration with the OMF & OML software suite. In addition to the generic services provided by other similar portals (e.g. resource discovery and reservation), this portal provides a complete environment supporting researchers in the design, execution, tracking and result analysis of series of networking experiments. As such, the objective of this Portal is to promote the adoption of a more rigorous scientific approach to networking research and to facilitate the production of high quality research data. The proposed solution does not claim to be perfect but rather contributes to a long term discussion within our community on how researchers should design, perform, and analyse experiments; and on how a scientific approach could be systematically applied to studies in computer science. Our solution is not final and remains open to discussion within the community.

The implementation of this ambitious objective has been facilitated by our experience in developing the IREEL e-learning platform, which also uses OMF. Indeed, it is the integration of new features (e.g. lab notebook, advanced experiment scheduling, passive testbed measurements) to existing IREEL capabilities, which enables the Portal to support the different steps in the life cycle of a scientific study.

In our future work, we plan to extend the Portal interface inside the OMF experiment controller in order to version the experiment and save its context directly after its completion. Furthermore, we plan to investigate new methods for automatic experiment execution based on previous experiments results analysis or the detection of acute topology disruptions.

Acknowledgements. This work was performed in the context of the Onelab2 project funded by the E.U. 7^{th} Framework Program, and the GENI (Global Environment for Network Innovations) initiative funded by the U.S. National Science Foundation.

References

1. git - The Fast Control System, http://git-scm.com/
2. Networking Conferences Statistics, http://www.cs.ucsb.edu/~almeroth/conf/stats/
3. NS3 Network Simulator, http://www.nsnam.org/
4. OMNet++ Simulator, http://www.omnetpp.org/
5. PlanetLab Testbed, http://www.planet-lab.org/
6. The NITLab Scheduler em, http://nitlab.inf.uth.gr/NITlab/
7. The OMF Testbed Control, Measurement and Management Framework, http://omf.mytestbed.net
8. The Orbit Measurment Library, http://oml.mytestbed.net
9. The R Project for Statistical Computing, www.r-project.org/
10. Albrecht, J., Tuttle, C., Snoeren, A.C., Vahdat, A.: PlanetLab Application Management Using Plush. ACM Operating Systems Review (2006)
11. Jourjon, G., Rakotoarivelo, T., Ott, M.: From Learning to Researching, Ease the Shift through Testbeds. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 496–505. Springer, Heidelberg (2011)
12. Kohler, E., Handley, M., Floyd, S.: Designing DCCP: Congestion control without reliability. SIGCOMM Computer Communication Review 36(4), 27–38 (2006)
13. Lacage, M., Ferrari, M., Hansen, M., Turletti, T.: NEPI: Using Independent Simulators, Emulators, and Testbeds for Easy Experimentation. In: Workshop on Real Overlays and Distributed Systems (ROADS) (2009)
14. Lewis, R.W.: Biology: A hypothetico-deductive science. The American Biology Teacher 50, 362–366 (1988)
15. Mehani, O., Boreli, R., Ernst, T.: Analysis of TFRC in disconnected scenarios and performance improvements with Freeze-DCCP. In: Ott, J., Tan, K. (eds.) 4th International Workshop on Mobility in the Evolving Internet Architecture, MobiArch 2009. ACM, New York (2009)
16. Popper, K.R.: The Logic of Scientific Discovery. Hutchinson London (1968)
17. Rakotoarivelo, T., Ott, M., Seskar, I., Jourjon, G.: OMF: a control and management framework for networking testbeds. In: SOSP Workshop on Real Overlays and Distributed Systems (ROADS 2009), Big Sky, USA, p. 6 (October 2009)
18. Raychaudhuri, D., et al.: Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols. In: Proc. IEEE Wireless Communications and Networking Conference, WCNC (2005)
19. Underwood, A.: Experiments in ecology and management: their logics, functions and interpretations. Australian Journal of Ecology 15, 365–389 (1990)
20. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. SIGOPS Oper. Syst. Rev. 36(SI), 255–270 (2002)
21. White, J., Jourjon, G., Rakatoarivelo, T., Ott, M.: Measurement Architectures for Network Experiments with Disconnected Mobile Nodes. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 315–330. Springer, Heidelberg (2011)