From Learning to Researching
Ease the Shift through Testbeds

Guillaume Jourjon, Thierry Rakotoarivelo, and Max Ott

NICTA*
Australian Technology Park
Eveleigh, NSW, Australia
{guillaume.jourjon,thierry.rakotoarivelo,max.ott}@nicta.com.au

Abstract. This papers presents an e-learning platform that improves the current state of the art by successfully integrating four features.

Firstly, it provides a web interface incorporating lecture notes, labs instruction and results. This remote interface also allows the teacher to easily implement new experiments using a high level description language. Secondly, the proposed architecture will provide a low deployment cost without limiting the experimental scope. Thirdly, the new platform can take advantage of many existing and emerging testbeds. Finally, we introduce a new framework for teaching and learning network concepts. Thus a student using this new tool during an introductory course will embrace a less difficult path to perform more advanced studies on currently widely deployed testbed.

1 Introduction

Knowledge of networking concepts and technologies has become essential to most computer, software, and information technology engineers. Indeed, an ever increasing number of devices are now network-enabled, e.g. fridges emailing grocery lists, cars pulling information from highways, TV pre-fetching favourite shows. Thus many educational institutions have included networking subjects in the syllabus of most of their computer or electrical engineering courses.

When teaching introductory or advanced networking subjects, lecturers face the difficulty of illustrating both the concepts and technologies, and assessing students with various knowledge backgrounds. This illustration phase is usually implemented in the form of laboratory classes, where students use some software tools to experiment on various networking scenarios. While different solutions may be used for different course levels, it would be desirable that such tools allow the experimentations of both basic and advanced networking scenarios.

The use of simulation software (e.g. ns-3, OPNET [1,4]) is a possible approach to address both these illustration and assessment challenges. This solution offers the benefit of an easy installation, maintenance, and access to many different

* 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. Magedanz et al. (Eds.): TridentCom 2010, LNICST 46, pp. 496–505, 2011.

type of simulated resources, but may involve a steep learning curve, e.g. understanding the scripting interface to the simulator and unrealistic and simplified models.

Another possible approach is to use deployed experimental platforms (e.g. testbed) to both illustrate and assess the learning of networking knowledge. The main advantage of this approach is that it allows students to interact with real protocol implementations in realistic environments, thus potentially enhancing their learning experience. On the other hand, deploying and maintaining a testbed often involves significant financial costs and engineering overheads, which in turn limit the number and type of available experimental resources with which to experiment. A solution to these limits is to use existing open research testbeds such as ORBIT, PlanetLab, or Emulab [11,9,13]. However, similar to the simulation approach, the use of testbeds may also involve steep learning curves for students.

This paper presents a new alternate testbed-based approach, which combines the features of a previously introduced e-learning platform (Internet Remote Emulation Experiment Laboratory, IREEL [6]) with the experiment control and resource management services of a widely used testbed framework (cOntrol and Management Framework, OMF [10]). From a student perspective, this integrated tool addresses the learning curve issue associated with using testbeds. Indeed, it allows both beginner and advanced students to experiment with networking concepts through an intuitive web-based interface, or with their own protocols through detailed experiment descriptions and automated executions.

The remainder of this paper is organised as follows. Section 2 presents our e-learning platform vision that motivates the presented work. Section 3 reviews some of the related works.Section 4 presents the design of the proposed IREEL-OMF integration, and describes some of the key implementation decisions and their benefits. Section 5 discusses the contributions of the proposed IREEL-OMF integration in terms of new and enhanced learning methods and tools for networking students and future young researchers. Finally, Section 6 concludes this paper and presents some potential future works.

2 Motivation

We envision the future of network-course e-learning platform as depicted in Fig. 1. We believe this architecture provides four improvements compare to the state of the art.

Firstly, we think that lecture note and labs instruction must be accessible directly on the web interface that pilots the possible configurable experiments. This feature is necessary to allow student to work efficiently. Furthermore, the configuration of these experiments must be available in a high-level interface to allows student with no programming language skills to participate in these lectures and the teacher to add easily new experiments.

Secondly, the architecture should have a low-deployment cost and a large range of network configurations. This requirement leads us to consider emulation solution for the basic platform.

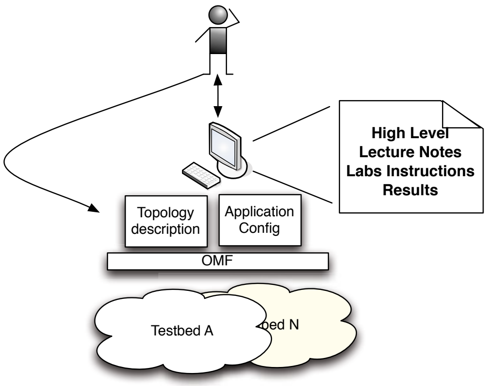

Fig. 1. General Architecture of the Future e-learning Environment

Thirdly, with the recent advancement in the worldwide deployment of network testbed, the future e-learning platform must facilitate the reproducibility of simple experiments over multiple testbed. In Fig. 1, this is illustrated by OMF's capability and current deployment over multiple testbed.

Finally, this new e-learning architecture should facilitate the shift from a beginner to advanced user of testbed as it is illustrated by the left arrow in Fig. 1.

3 Related Work

Few initiatives currently provide software services and tools to control experiment executions, access, and manage resources on networking platforms. Some examples of such software suites are the Emulab [13], the PlanetLab [9], and the OMF frameworks. These frameworks were primarily designed to address the experimental needs of the networking research community. Although, some of them could be used in a educational context, their usage requires a learning curve which is not suitable nor necessarily relevant to a semester-based academic course focusing on networking concepts and technologies.

Emulab [13] is a large network emulator. It 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 experimental environments: simulated, emulated, and wide area networks. It unifies all three environments under a common user interface. Emulab provides tools to describe a required experiment topology and map it to actual resources. Some control tools are also provided, but with minimal features. Emulab and OMF share many design principles with the differences primarily shaped by a focus on different hardware and a different user community.

PlanetLab [9] is a global research platform based on more than 1000 distributed computers, which are 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 contend distribution). PlanetLab provides a suite of software, which uses virtualisation tools to efficiently share the global resources among simultaneous short or long-lived experiments. Similar to Emulab, these tools are essentially focused on resource allocation and access, and only provide minimal supports in describing, controlling and measuring experiments. PlanetLab is also limited by its default offered layer-3 abstraction, i.e. it could not be used to illustrate layer-2 schemes (e.g. wireless MAC mechanisms) to networking students.

The Open Network Laboratory (ONL) [7] is a testbed-based educational resource, which has been used in the teaching of several academic networking courses. It consists of several computers interconnected by multiple extensible routers, which can be linked in various network topologies through a central virtual network switch. ONL allows the users to extend the routing functionalities through software plugins insertion. A user (e.g. a student) remotely access the platform through a stand-alone graphical interface, which easily allows the construction of various topologies, their configurations (e.g. route, bandwidth), and their monitoring. Similar to PlanetLab, the current ONL platform is limited to the illustration of layer-3 and above networking concepts.

Academic and industry research communities have developed many networking simulators, which provide inexpensive alternatives to testbed platforms as educational tools. Some of these simulators (e.g. OPNET [4]) have been successfully used in academic courses as the base of complementary laboratory activities [5]. Others (e.g. the ns-3 network simulator [1]) requires similar learning curves as the frameworks mentioned earlier, which may impede the learning of networking concepts. In any case, the inherent model assumptions within simulators limit the illustration of some networking aspects, e.g. there is no accurate simulation model for losses on wireless channels.

4 IREEL and OMF Integration: Design and Implementations

Facing the integration of an e-learning platform and a testbed management framework we have identified several challenges including the need to identify: a new communication scheme between the platform controller and the two computers that constitute the platform, a new measurement scheme based the a generic measurement library, and a new script generation for the experiment and the student.

4.1 New Architecture Overview

In the first version of the IREEL platform, communication between the platform controller and the two end-systems was done using remote calls with CORBA. This way of communication allows us to deploy the platform regardless of the operating system. On the counterpart, this scheme forced the teacher to follow strict rules for the deployment of new features.

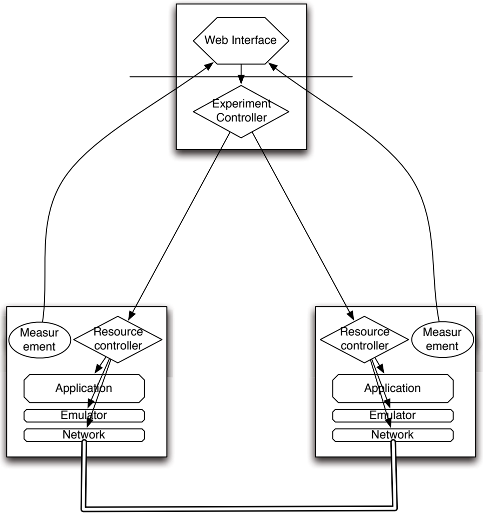

Fig. 2. New Architecture of the e-learning platform

In this new architecture, there is no longer a need for an intermediate computer thanks to the integration of a new emulation module inside OMF, as it will be detailed in the following of this section. Furthermore, as it is depicted in Fig. 2 we have added a new measurement channel using OML [12]. In the following of this section, we present the new communication channel using OMF, the measurement capability and finally the new emulation module in OMF.

4.2 New Communication and Measurement Scheme

In order to build the new architecture, we had to completely change the end-systems. This has been made easy thanks to the OMF application description. Then we had to decide how to start an application on these new end-systems. For that we changed all the application and the controller that was waiting for an XML file to decide which application to start. Therefore, we built ruby wrappers around all the legacy applications and remove the end-system controller to put instead the management framework.

The second challenge we had to resolve was the collection of measurement on the new end-system. Indeed, the management framework allows us to use a modular measurement library [12] that dynamically stores the results into a database on the platform controller. Previously, three kind of measurements were available on IREEL: network, application specific and application output such as a video over a network with impairments. For the moment we decided to use the measurement library to collect both network and application specific result and by-pass this library for the output of the video streaming. The answer to this challenge remains open for the moment.

4.3 Integration of an Emulation Module in OMF

Finally, in order to make this integration possible we had to develop a new module for OMF. As explained in Section 2, one of the main goal of this e-learning platform is to be easily deployable and therefore we need to maintain a low deployment cost. Thus, in order to provide a large scope of network configurations we decided to use an emulation solution. However, in the current state of OMF it was not yet possible to manage emulation tools but skeletons to implement such controls were available. Therefore we developed a new emulation module that allows us to firstly maintain a low deployment cost of the e-learning and secondly provide networking community an emulation enhancement to all testbeds that use OMF. As a side effect, this makes OMF even more suitable for wired networks and allows testing on more complex topology using existing testbed.

This emulation module has been make possible by the integration of a management procedure of Netem [8] on the end-systems. This integration has been possible through the enhancement of the topology description inside OMF and the interfacing of netem specific API. In OMF Experiment Description Language OEDL [3], it was possible to easily integrate this feature.

The addition of this emulation module inside OMF brings us two main benefits, an extension the topological abstraction and capacities of OMF and the student is given a theoretical graph representation of the configured network. Indeed, in addition to the configuration of the network, OMF's topology builder gives the experimenter a graph that represents the network. In this model, the network is represented as a directed graph $G(V, E)$, where V represents the set of vertices and E the set of edges.

This feature allows the student to verify if (s)he configured the network as (s)he was asked to do to finish his assignment. On the teacher side, this feature provides the opportunity to introduce the graph theory as a representation of the network. Furthermore, as we demonstrate in the following section, it also allows the teacher to point out the need of real experiments compare to pure theoretical approach to networking.

4.4 Integration Procedure: A More Flexible Approach

As explained in the previous section, we changed the way to initialise the application on the end-systems. This modification also facilitates the integration of new applications on both the end-system and the platform controller. In the first version of the platform developers have to follow a Java skeleton.

In the new architecture, the developer must simply install the new application on the end-systems and then write a ruby script that will be called by the management framework. This script is presented through an example in Listing 1. Starting the new application is therefore transparent for the developer. Furthermore, we plan to extend this application description to automatically generate the front end for the student. Thus, the teacher writes a script describing both the application to use and the different network impairments required for this experiment. This new feature can be illustrated in the second part of Listing 1.

```
defProperty(’delay1’, ’10ms’, ’between 1 and 2’)
defProperty(’bandwidth1’, ’1000kbit’, ’between 1 and 2’)
# Create an application representation from scratch
defApplication(’ireel :app:newApp’, ’newApp’) {|app|
  app.version(1, 1, 2)
  app.shortDescription = ”Illustration of the CS course”
  app.path = ”/usr/bin/newApp”
  app.defProperty(’numPackets’, ’number of packets’)
  app.defProperty(’timeout’, ’ping timeout in second’)
  app.defProperty(’size’, ’user data size’)
  app.defProperty(’output’, ’Output type binary or text’)
}
#Specify the link characteristics from node 1 to 2
defLink(’ireel :link:newLink’, ’newLink12’){|link|
  link.shortDescription = ”Configuration of the link characteristics ”
  link.setProperty(’bandwidth’, prop.bandwidth1)
  link.setProperty(’PLR’, 0.0)
  link.setProperty(’delay’, prop.delay1)
}
```

Listing 1. Example of a new application

The teacher then submitted this experiment description using a wiki-like interface. It is afterward processed by the system in order to generate an IREEL experiment interface with the desired network impairments and application configuration.

5 From Learning to Researching: Benefits of a Unified Solution

5.1 Introducing a New Learning Plane

As illustrated on Figure 3, the integration of IREEL with OMF provides a new *learning plane*, which serves as an interface between students/lecturers and the bare resources of a testbed platform. This plane provides a seamless access from a user-friendly graphical front-end to the full visualisation/edition of an OMF Experiment Description. This caters both for undergraduate basic courses and

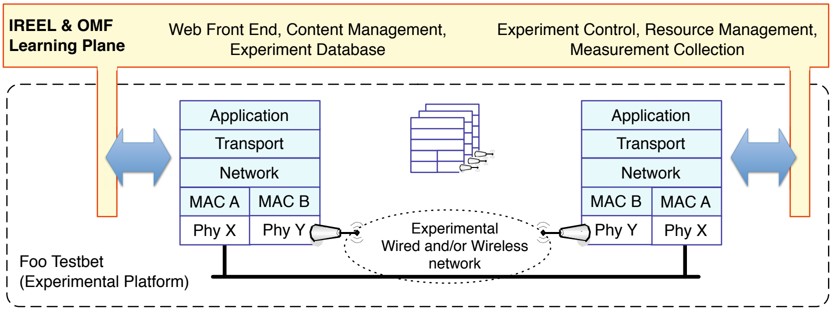
IREEL & OMF
Learning Plane
Web Front End, Content Management, Experiment Database
Experiment Control, Resource Management, Measurement Collection
Application
Transport
Network
MAC A
MAC B
Phy X
Phy Y
Experimental Wired and/or Wireless network
Application
Transport
Network
MAC B
MAC A
Phy Y
Phy X
Foo Testbet
(Experimental Platform)

Fig. 3. New Learning Plane

postgraduate advanced ones. Indeed, students of the former courses use graphical menus to develop, run, and analyse experiments, thus focusing on the networking concepts without the difficulty of a steep usage learning curve, nor the need for any programming skills.

Through the use of OMF's systematic descriptions and automatic executions of experiments, this learning plane further introduces new students to scientific concepts, such as reproducibility, statistical significance, or peer-verification. For example, students can easily cross verify their results, or run multiple trials of the same experiment with/without parameter modifications, and observe the effects on their result statistics. This feature also benefits advanced and research students, as they have now an uncomplicated tool to produce more accurate scientific results.

Another major benefit of this learning plane is that through the use of OMF as the interface with the underlying testbed, it potentially gives students and lecturers access to large number of heterogeneous resources within many different testbed platforms. Indeed, OMF is currently used to control experiments and managed various resources at many institutions. For example, the particularities of an existing protocol (e.g. TCP) can be illustrated on real PC-based resources connected via links with Internet-like characteristics, and then seamlessly evaluated on real mobile wireless resources connected via ad-hoc 802.11 links. Comparison between results from both environments would give students valuable insights into the studied protocol. An example of such a case is developed in the next section.

5.2 Demonstration of the Portability

In this section, we demonstrate through a simple example the benefit of the combined use of the new platform and a wireless testbed. In this scenario the student is first asked to configure the simple IREEL's network and then starts a traffic generator. Once this experiment is finished the student can now take the experiment script provided by the platform and execute it with the same topology on the wireless testbed. The aim of this simple scenario is to put in relief the physical characteristics between wireless and wired network even if the theoretical topology remains the same.

Execution on IREEL. In order to execute this experiment on IREEL, the student must first log on the IREEL portal. Once the student on his or her personal page, they may configure a simple scenario using iperf in its UDP mode and a topology with a bottleneck of 1 $Mbit/s$ and an RTT of 200 ms.

The student must then waits for the experiment to finish and analyzes the results provided by the platform. These results comprise: the output of iperf; a graph of the throughput perceived by both the sender and receiver; the file generated by TCPdump [2]; the newly introduced measurement database; and can be summarised by the Table 1.

Execution on a Wireless Testbed. Once the results have been analysed, the student will be in possession of an OMF Experiment Description. Advanced

Table 1. UDP Results on a Wired Network

Average Throughput	1010 $kbit/s$
Average Jitter	$2.21ms$
Packet Loss Rate	59.56%

student can now use this script to better understand the importance of the access medium.

We performed this simple experiment on the NICTA wireless testbed and able to observe results displayed in Table 2. In these results, slight differences are evident even if the exact same theoretical network has been configured. Based on these differences, the teacher can open the discussion on the importance of the access network and start a deeper explanation on this technology and its limitations.

Table 2. UDP Results on a Wireless Network

Average Throughput	987 $kbit/s$
Jitter (min/avg/max)	$5.43ms$
Packet Loss Rate	60.51%

6 Conclusion and Future Work

Based on feedbacks from students and teachers, we have presented a new e-learning platform that improves the state of the art on several aspects. Firstly, the proposed architecture provides both students and teachers with a rich web interface incorporating lecture notes, labs instructions and results of the experiments. Furthermore, this interface allows an easier integration of new experiments for the teacher through the use of a high level description language.

Our solution also provides a low deployment and maintenance cost in its basic architecture and facilitates the use of already deployed testbed. Finally, we introduces a new teaching plane for network course that ease the shift from beginners to advanced students.

On a technical aspect, we plan to extend this platform to support directly wireless experiments. Thus the new platform will be one of the first e-learning platform integrating a wireless component.

On the teaching counterpart aspect of this platform, we plan to provide it to Universities and follow its use. We also plan to evaluate the final product based on students and teachers evaluation.

Acknowledgements

This work was achieved in the context of the NaDa and Onelab2 projects 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. NS3 Network Simulator, http://www.nsnam.org/
2. Tcpdump/libpcap public repository, http://tcpdump.org
3. The OMF Testbed Control, Measurement and Management Framework, at: http://omf.mytestbed.net
4. The OPNET Network Simulator, http://www.opnet.com/
5. Aboelela, E.: Network Simulation Experiments Manual, 2nd edn. Morgan Kaufmann, San Francisco (2008)
6. Dairaine, L., Jourjon, G., Lochin, E., Ardon, S.: IREEL: Remote Experimentation with Real Protocols and Applications over Emulated Network. Inroads, the SIGCSE Bulletin 39(2), 92–96 (2007)
7. DeHart, J., Kuhns, F., Parwatikar, J., Turner, J., Wiseman, C., Wong, K.: The Open Network Laboratory. In: Proceedings of SIGCSE 2006 (March 2006)
8. Hemminger, S.: Network emulation with netem. In: Linux Conf. Au. (April 2005)
9. PlanetLab Consortium. Planetlab: An open platform for developing, deploying, and accessing planetary-scale services, http://www.planet-lab.org/
10. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: OMF: A Control and Management Framework for Networking Testbeds. In: Proc. of ROADS 2009: Fourth Workshop on Real Overlays and Distributed Systems (October 2009)
11. 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)
12. Singh, M., Ott, M., Seskar, I., Kamat, P.: ORBIT Measurements Framework and Library (OML): Motivations, Design, Implementation, and Features. In: Proceedings of IEEE Tridentcom 2005 (February 2005)
13. 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. In: Proc. of the Fifth Symposium on Operating Systems Design and Implementation, Boston, MA, pp. 255–270. USENIX Association (December 2002)