Operational Experience With a Virtual Networking Laboratory

Charlie Wiseman†, Ken Wong†, Tilman Wolf‡, and Sergey Gorinsky†

†Department of Computer Science and Engineering
Washington University in St. Louis
{cgw1,kenw,gorinsky}@arl.wustl.edu

‡Department of Electrical and Computer Engineering
University of Massachusetts Amherst
wolf@ecs.umass.edu

ABSTRACT

Virtual laboratories are a potential replacement for standard laboratory facilities that can reduce cost and maintenance overheads for teaching institutions while still ensuring that students have access to real equipment. Previous work indicates that students respond well to such environments, but one important operational aspect has been overlooked. In this work, we consider instructor overhead by comparing the amount of work required to teach courses with and without the use of a virtual laboratory. In particular, we examine two graduate computer networking courses, each taught with the standard software-only approach and then taught later with the *Open Network Laboratory*. Our data show that the effort required by the instructor to use a virtual laboratory is not much more than in a software-only environment, and that the increased interaction between student and instructor can be beneficial as the student questions are primarily focused on fundamental networking concepts.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education; C.2.0 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Human Factors

Keywords

Educational technology, virtual laboratory, assessment

1. INTRODUCTION

Laboratories are an important experience in computer science and engineering education which allow students to apply classroom knowledge to real software and systems, and to gain familiarity with state of the art technology and industry standards. The range of laboratory types that are employed in education spans software-based programming, simulation, and hardware-based labs. Recently, so-called "virtual laboratories" have been added to the repertoire of instructional technology.

A virtual laboratory is a facility where students can access real laboratory equipment remotely (e.g., via the Internet). By using a single centralized facility, many users can share equipment and thus reduce the financial burden of a lab setup. Using a suitable software interface, students can control and monitor experiments on the actual hardware. This allows for a learning experience that is nearly equivalent to being physically present in the lab. Reports on teaching experiences with virtual laboratories indicate a positive reception by instructions and students [4].

An important question that needs to be considered with the introduction of any new teaching technology is what level of additional effort such a system requires. As instructors are already very busy with other aspects of a course, a laboratory component should not add an unreasonable burden of work on the instructor. In this paper, we present an assessment study that addresses this very issue. In particular, our paper presents the following contributions:

- a discussion of the types of interactions that take place in a virtual laboratory between students, instructors, and laboratory administrators;
- a discussion of operational issues that arise when using and maintaining virtual laboratory technology;
- a detailed study of two courses to quantify how much effort is involved in using virtual laboratories (measured by the amount of email exchanged between all parties involved).

Our results show that virtual laboratories pose a set of novel challenges in terms of management and maintenance. Also, the comparison of a course using a traditional software-based laboratory (i.e., programming and simulation) and a course using a virtual laboratory shows an increase in the number of emails exchanged between students and the instructor. The effect is mostly due to an increase in the number of students who need help with this new technology. Further, we show how instructor experiences relate to the virtual laboratory technology.

The remainder of the paper is organized as follows. Section 2 briefly reviews the Open Network Laboratory (ONL) and then discusses the opportunities and operational challenges involved in using ONL. Section 3 looks at ONL from

SIGCSE'08, March 12–15, 2008, Portland, Oregon, USA.

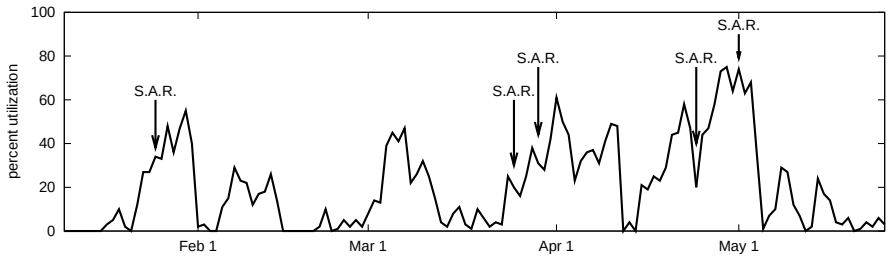

Figure 1: Utilization of the ONL Virtual Laboratory During the Spring 2007 Semester.

an instructor's perspective. Section 4 presents data on student-instructor-staff interactions associated with ONL assignments and discusses their implication. Section 5 summarizes and concludes this paper.

2. VIRTUAL NETWORK LABORATORY

In the context of computer networking, virtual laboratories typically contain some collection of networking equipment and standard PCs that are remotely accessible over the Internet and are shared (in time and space) by all parties using the laboratory. Examples of these types of systems include PlanetLab [1], Emulab [2], and WAIL [3]. The courses studied in this paper used the Open Network Laboratory (ONL) which was initially developed for high-speed networking research but has since been found to be a compelling educational tool [4].

ONL is currently composed of four extensible, high-performance routers connected to a large number of standard end systems [5]. Using this set of hardware, students can build network topologies and interact with the various network components through the *Remote Laboratory Interface* (RLI).

The RLI is an intuitive graphical user interface that provides users with the ability to configure and monitor many aspects of the network in real-time that are often hidden in simulation or inaccessible in standard networking equipment. In addition, students can inject *plugins* (customized packet processing code) into a router's packet processing path. Students can use plugins from a pre-written set of standard plugins (e.g., to delay packets) or write their own plugins.

2.1 Opportunities and Challenges

There are both opportunities and challenges in using ONL in courses. The opportunities lie in its many monitoring points, data display functions, control knobs (e.g., packet scheduling parameters, link rate), and code injection capability that provide an unprecedented ability to observe, control, and modify network behavior.

One challenge is structuring the presentation of these features so that students are not overwhelmed by the feature set. Student surveys have shown that most of ONL's features are reasonably straight-forward to use once the students are alerted to common misconceptions about the hardware. For example, when setting a link rate to 100 Mbps, the student may be surprised to find that the interpacket time of the first two back-to-back packets indicates that the link rate is 1 Gbps. This is because the physical link is a 1 Gbps link and the desired rate is modeled as a token bucket that allows a burst of two maximum-sized packets; i.e., the link model is a link *regulator*, not a link *emulator* that will delay every packet based on the link rate.

Another challenge is that the system is composed of real hardware and custom software that was built by academic researchers and is not supported by a world-wide user community as is the case with most off-the-shelf components. This means that under high load, components occasionally do not reset properly requiring students to restart an experiment or even worse, requiring the ONL staff to manually reset a component. Surprisingly, this has not been as large an issue as expected, but it does make the ONL staff nervous when all of the routers are being used almost continuously over a 72 hour period as assignment deadlines approach.

Figure 1 shows the daily utilization of the four ONL router clusters during the spring 2007 semester normalized such that a utilization of 100% would indicate that all four ONL router clusters were in use for the entire 24 hour period of that day. The peak periods correspond to the intervals preceding assignment due dates of the two courses using ONL. The points marked *S.A.R.* indicate times when the system required ONL staff attention. None of the S.A.R. points were significant except the next to last one at the end of April that caused the utilization to dip to around 25%. This event was due to a central software daemon lock up in the middle of the night that was not resolved until the next morning.

A third challenge is that each ONL router can only be used by one student at any given time. This is due to the necessity that each student's network be completely isolated from any other running experiments to ensure proper behavior. In essence, ONL supports space division through a resource reservation system that allows users to reserve resources much like diners reserve restaurant tables.

2.2 Course Support

The course support structure attempts to address the special challenges described above. The normal interaction between instructor (and any teaching assistants) and students is augmented by the ONL support staff who backstop the instructor by helping to answer questions that the instructor might be unsure of and by responding to strange system behavior reported by students. This approach relieves the instructors of many of the anxieties that might arise from using experimental equipment and has worked surprisingly well with the majority of questions and issues being handled by the instructor. Section 4.2 quantifies these interactions in detail, but a summary is shown in Figure 2 where the labelled arrows indicate how many emails were sent between students, instructors, and the ONL staff over the entire spring 2007 semester. Details about courses A

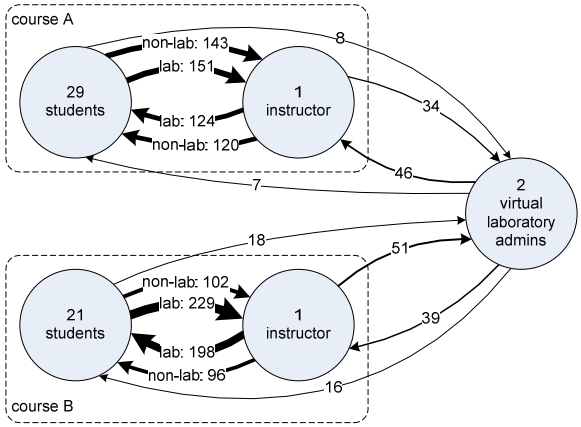

Figure 2: Email Interactions During Spring 2007.

and B are given in Section 4.1. All interaction is initially handled by email and goes through the instructor who is expected to answer most of the questions and only occasionally need help from the ONL staff. Students are allowed to contact the ONL staff directly when there is evidence of system failures.

3. INSTRUCTOR CONSIDERATIONS

From an instructor's perspective, some aspects of using ONL in a course are no different than using any other computer system, but there are also operational issues that are somewhat unique to ONL.

First, instructors need to work out solutions to their laboratory assignments before distributing them to students. Although this is generally true, it is a particularly important precept if the instructor is also new to the virtual laboratory. The typical problem is that the instructor's mental model disagrees with ONL's conceptual model in some subtle way. One example is the link rate parameter mentioned earlier. Although common misconceptions are documented in the ONL tutorial pages and summarized in FAQs, it is easy to glaze over the distinctions.

Second, each laboratory assignment should be preceded by paper-and-pencil exercises that emphasize the main concepts addressed by the assignment and develop approximation skills that can be used for sanity-checking experimental results. Trouble shooting exercises in which students are asked to postulate the source of unexpected measurement results and propose mini-experiments for verifying these postulates provide useful mental preparation.

Third, students should be counseled on efficient ways to conduct their experiments. For example, many of the initial RLI features can be explored without actually reserving hardware resources. A common mistake made by students in the first assignment is to reserve hardware and then spend the entire time going through the various RLI menus instead of actually using the hardware.

Fourth, there is a high level of paranoia that develops when more than one course is using the facility during the same period. "Hey, University XXX students are evil because they are using our resources" is a common complaint from students. The ONL staff has done a good job in coordinating assignment due dates to keep the actual contention level low, but instructors still need to pay attention and foster a cooperative attitude.

Table 1: Comparison of Courses That Use ONL.

	Course A	Course B
Department	ECE	CS
Only graduate students?	yes	no
Non-networking students?	yes	no
Number students	29	21
Number of labs	4	4
Labs requiring programming	0	2
Programming difficulty	n/a	basic; any language

Finally, there were a small number of system failures during high-usage periods. Most hardware failures are almost transparent since there is an automated repair system but it can become an issue if one of the four routers has a hard failure that automated procedures fail to properly correct. In practice, such failures have been almost non-existent. The only major failure was the one aforementioned instance when the central software daemon locked up.

4. COURSE EFFORT

4.1 Comparison of Courses

Our observations in this section are based on two spring 2007 courses located at different universities, Course A and Course B, as shown in Table 1. The main differences between these courses are:

- Course A is aimed specifically at giving breadth to graduate students that may not be majoring in networking.
- Course B requires students to write basic socket programs to transmit UDP packets. The course survey indicated that the programming was not a significant hurdle because they were allowed to use any language.

4.2 Interaction Data

We collected data on the amount of email interaction between students, instructors, and ONL staff as indicators of the amount of work accompanying laboratory assignments for instructors and support staff. The data is normalized to a class of 29 students, the size of Course A.

Figure 3 shows some of the data for Course A during the spring 2007 semester. It shows the number of daily emails from the instructor to students and ONL staff (lab staff) with a distinction between emails that were and were not related to a lab assignment. One observation is that there was very little interaction with ONL staff except near the beginning of the first assignment, i.e., the instructor handled most of the questions. The increased interaction with ONL staff during the week of March 1 was due to an overly aggressive security policy that resulted in the blocking of connections from student hosts that fit the profile of attack activity.

It is interesting to note that there was little interaction in the third assignment. The assignment involved exploring network behavior due to the competition between TCP and UDP flows and designing/performing an experiment to prove or disprove a hypothesis (e.g., UDP achieves a higher

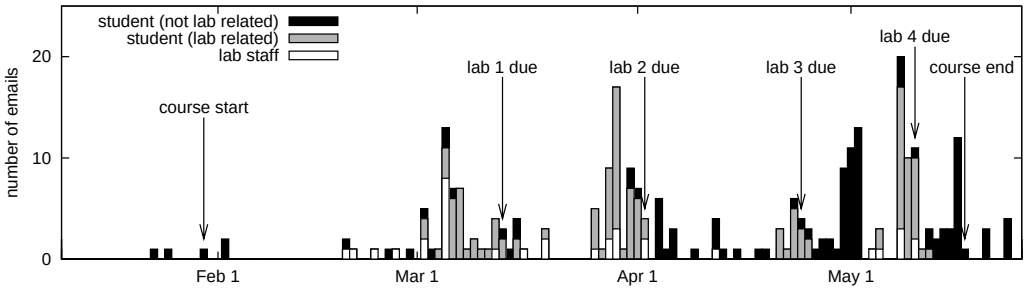

Figure 3: Email Counts in Course A (With ONL).

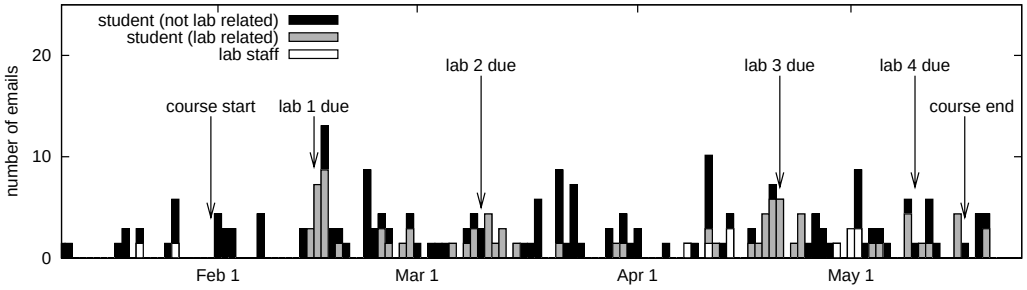

Figure 4: Email Counts in Course A (Without ONL).

utilization on the bottleneck link than TCP). In fact, this assignment generated the most student interest: they were excited and pursued the problem on their own (without checking with the instructor).

To give some perspective to Figure 3, Figure 4 shows the same type of email interaction data for the same course in spring 2005 when assignments involved socket programming and ns-2 [6] simulations instead of ONL. In this figure, *lab staff* refers to the local computer support staff, not the ONL staff. At first glance, the data indicates that instructors can expect to be interrupted by students less than if they were using ONL. Actually, this observation seems reasonable for two reasons:

- There were fewer operational issues. Students asked fewer questions on how to write socket programs. With ONL, there were operational issues; e.g., using `ssh` (Secure SHell) to access ONL resources.

- There were fewer questions regarding what to do in socket programming assignments. With ONL, there were more questions regarding the precise meaning of an assignment's requirements and how to meet those requirements.

The second point is worth elaborating. In the non-ONL course, there may have been more improper collaboration between students. Since students did not have to log into the system and set up their own experiment, they may have solved projects jointly and thus had peers to help and answer questions. In the ONL version of the course, peers could not really provide much help when something did not seem to behave as expected and had to work through a problem until they required expert advice.

Informal discussions with the teaching assistant in Course B corroborated this observation. Furthermore, there was a clear evolution in the type of student-instructor interactions as the semester progressed. Interactions at the beginning of the course concerned more operational ONL/RLI issues as the students were familiarizing themselves with a foreign environment. Towards the end of the course, the student questions evolved into discussions focused more on fundamental networking concepts than on operational issues.

Another important statistic is that the volume of email shown in Figure 3 (the ONL case) that is above that shown in Figure 4 (the non-ONL case) is not particularly significant. The benefit of increased student engagement and thinking precision far outweighs the small increase in student-instructor interaction time.

Figure 5 shows the interaction data for Course B. The most significant difference with Course A is that there is interaction on almost every day during the last two assignments. Furthermore, there is more interaction with ONL staff than in Course A. This increase in interaction can be explained by noting that these two assignments involved developing router plugins.

Developing a router plugin is almost equivalent to writing a module in C for an embedded processor where the normal debugging facilities are absent. The plugin environment delivers the packets to the user plugin in network byte order with attached packet headers. Errant code can hang the plugin processor requiring the student to abandon an experiment and restart – a time consuming process. As such, writing a plugin is initially daunting to students who are used to interactive debugging tools, simple data formats, and rapid retries.

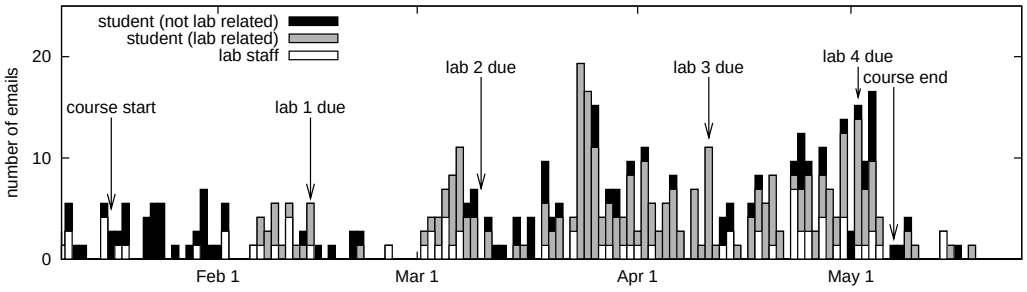

Figure 5: Email Counts in Course B (With ONL).

4.3 Observations

Below are some observations from discussions with the course instructors, teaching assistants, and the ONL staff:

- A shortcoming of our email counting approach is that we did not distinguish between those emails that were caused by ONL problems and those that were just about the assignment.
- It would be interesting to collect data in classes that are dominated by undergraduate students since graduate students represent the better students and cope better with non-ideal situations. Work in this direction is proceeding this fall.
- It is clear that good ONL assignments focus graduate students on fundamental networking principles. In fact, laboratory assignments, give all students an opportunity to reflect on fundamental concepts and to see them in practical terms.
- It appears that with graduate students, the ONL assignments have not promoted peer-to-peer interaction. One teaching assistant said: "Many graduate students don't do peer-to-peer instruction. They want to figure it out themselves. And if they can't, they're going straight to an expert." Peer-to-peer interaction is difficult to quantify although cooperative approaches to learning appear to correlate with improved student satisfaction and superior outcomes [7]. However, there is nothing preventing an instructor from using an ONL experiment running in real time to initiate a discussion about a fundamental concept.

5. SUMMARY AND FUTURE WORK

We have described our operational experience with using the ONL virtual networking laboratory, presented data on student-instructor-staff email interaction, and discussed the implications of the data. ONL assignments seem to increase student-instructor email interaction but primarily these interactions are more focused on learning fundamental networking concepts. An interesting direction might be to see how ONL assignments can improve peer-to-peer interaction.

Progress is being made in the development of tools to improve student-instructor interaction and in the assessment of the impact of ONL on learning in networking. In the near future, ONL will have tools for session-sharing and session recording/playback. The session-sharing feature will make it possible for a student who is having difficulty to seek help from an instructor/TA while they are working on a laboratory assignment. The session record/playback feature will allow students to document their ONL sessions and will allow an instructor/TA to study how a student interacts with ONL to get a better understanding of where they encounter difficulties.

Work is in progress to assess the impact of ONL along with the beginnings of a networking test bank that focuses on core networking concepts. By coincidence, the work appears to have many parallels to the idea of a concept inventory [8].

Soon, fourteen router clusters will be added to the existing four router clusters which should improve resource availability. Although the technology will be based on network processors, the conceptual model embodied by the RLI will not change, and the use of network processor technology will make it possible to inject code into a packet path that can potentially run at gigabit speeds.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. CNS-0551651, DRL-0632580 and DRL-0632586.

6. REFERENCES

[1] http://www.planet-lab.org

[2] Brian White, Jay Lepreau, Leigh Stoller, et. al., "An Integrated Experimental Environment for Distributed Systems and Networks", *Proc. 5th Symp. on Op. Sys. Design & Implementation*, Dec. 2002, pp. 255-270.

[3] Wisconsin Advanced Internet Laboratory, http://www.schooner.wail.wisc.edu

[4] Ken Wong, Tilman Wolf, Sergey Gorinsky, and Jonathan Turner, "Teaching Experiences With a Virtual Network Laboratory", *Proc. SIGCSE*, March 2007.

[5] John DeHart, Fred Kuhns, Jyoti Parwatikar, Jonathan Turner, Charlie Wiseman and Ken Wong, "The Open Network Laboratory", *Proc. SIGCSE*, March 2006.

[6] The Network Simulator ns-2, http://www.isi.edu/nsnam/ns/

[7] Richard M. Felder, "What Matters in College", *Chem. Engr. Education*, Vol. 27, No. 4, pp 194-195, Fall 1993.

[8] Vicki L. Almstrum, et. al., "Concept Inventories in Computer Science for the Topic Discrete Mathematics", *Proc. ITiCSE*, pp. 132-145, June 2006.