Using a real Internet-scale environment for protocol testing in undergraduate courses: students' behaviour and feedback

Joan Manuel Marquès, Angel A. Juan, Alejandra Pérez-Bonilla, Thanasis Daradoumis, Daniel Lázaro and Rubén Mondéjar
Computer Science, Multimedia and Telecommunication Studies, Open University of Catalonia, Barcelona, Spain
E-mail: ajuanp@uoc.edu

Abstract This paper presents a five-semester experience regarding the use of a real Internet-scale environment for testing protocols and services under realistic conditions. The paper discusses the convenience of using such real environments over other alternatives, i.e. emulated or simulated environments. Since most currently published work focuses primarily on technical specifications of academic testing environments, the paper also argues for the need to carry out studies which analyse students' opinions, performance, and behaviours while using these environments. Accordingly, an exhaustive survey analysis is included. This first analysis is complemented with a second one regarding the way students interact with the system. From both studies, several conclusions and recommendations are derived.

Keywords computer-supported collaborative learning; educational data analysis; higher education; Internet-based education; Internet-scale systems; protocol-testing environments; telecommunications engineering

Protocol- and service-testing environments allow students of telecommunications engineering and computer science to perform hands-on, practical tasks with real or virtual hardware systems. These experiences support the building of skills through experimentation and promote a proactive, exploratory and self-discovery learning paradigm. They also contribute to filling the gap between theoretical and practical education. Internet protocols and services have traditionally been tested using simulators and emulators. However, as some authors point out[1], it is difficult to model or to emulate the real conditions of a wide area network.

Fortunately, during the last few years virtualisation of computational resources has enabled the emergence of real testing environments distributed over the Internet. These environments offer an ideal framework to test Internet protocols and services under realistic conditions. One such environment is PlanetLab (www.planet-lab.org), 'an open platform for developing, deploying, and accessing planetary-scale services.' PlanetLab is an international consortium that aggregates virtualised computational resources from its member institutions, which are scattered around the world, and connects them through the Internet. Thus, members of these institutions can deploy and test Internet protocols and services on a set of geographically dispersed hosts connected using a real network. As of September 2011, PlanetLab consisted of 1,079 nodes at 529 sites.

As stated before, the main advantage of using environments like PlanetLab, instead of using simulated or emulated environments, is that the former offer a real distributed environment – with realistic working conditions – to experiment with. Specifically, Internet-scale protocol or service implementations have to deal with hidden complexities which cannot be easily modelled in simulators or emulators, e.g. real latencies, concurrent access to local data, unexpected failures and breakdowns, bandwidth variations, package collisions, and many other possible events associated with any real telecommunication process. Additionally, in the case of simulators the protocol or service implementation has to be adapted to the simulation framework, which is not a trivial task to perform. Therefore, environments such as PlanetLab suggest new learning opportunities by deploying and testing protocols and services in real Internet-scale systems working under realistic conditions.[2] This paper describes a long-term teaching experience in the use of PlanetLab to deploy and test protocols and services in an undergraduate course on Internet-scale systems. This course is offered to Telecommunications Engineering and Computer Science students at the Open University of Catalonia (UOC). The paper focuses on the more educational aspects of the PlanetLab@UOC experience, i.e.: (a) how students interact with the system, including their habits and strategies; and (b) how students evaluate the experience, including their overall satisfaction, the main difficulties they have found during the completion of their assessments, etc. In our opinion, these aspects are relevant for other instructors interested in integrating similar learning environments over the Internet in either distance-learning or face-to-face courses. For a detailed description of the PlanetLab@UOC technical characteristics, the reader is addressed to our previous work[3].

The paper is structured as follows: First, we present a brief literature review on related work. Then, a description of the specific UOC learning context and learning activities is provided. After that, the results of a five-semester survey among our students are discussed. Next, a section presents and analyses the observed traces of students' interaction with the system. Finally, we summarise the main findings of our study and highlight how it can be useful for other instructors.

Background and related work

To the best of our knowledge, analysis and evaluation of students' interaction, behaviour and performance in Internet-scale testing environments has scarcely been done. According to some authors[1,4], it is not a trivial task to design and implement a distributed application over the Internet which provides specific facilities to support the teaching process and to promote students' active participation. Interaction is the key to achieving successful outcomes in finding the right mix of human and technical elements for an efficient learning process. Special attention should also be directed to adult, online or part-time students who have to integrate learning activities with working and family duties[5]. Different frameworks have been suggested for the evaluation of Internet-based leaning environments. Most of them focus on two aspects: the first concerns the evaluation of the learning environment, and the second deals with the evaluation of the students' performance. Benigno and Trentin[6] considered factors such as student characteristics, student–student interaction, effective support, learning materials, learning environment, and information technology. Although some previous works regarding the analysis and assessment of courses on Internet-scale systems can be found[7,8], they focus on the general description of system functionalities and do not analyze student-system interactions or students' feedback.

Other approaches[9] discuss technical aspects of scalable emulation environments. One example could be ModelNet, a configurable Internet-based environment that allows deployment of unmodified software prototypes subjected to emulated faults and varying network conditions. This environment has been evaluated as regards its architecture and implementation. However, no evaluation of users' interactions or users' opinions has ever been performed. Another similar environment is Emulab (http://www.emulab.net/), a network test bed that has been widely used by researchers in the fields of networking and Internet-scale systems. Emulab is also designed to support education in the telecommunications engineering and computer science fields. In fact, it provides a virtual laboratory in which students can have complete control of a contained environment, which encourages exploration and experimentation. Though its use is widespread, no systematic analysis and evaluation of these experiences has so far been reported. Finally, Netbed[10], a descendant of Emulab, provides an experimentation facility that allows researchers to configure and access networks composed of emulated, simulated, and wide-area nodes and links. Again, as far as we know, no evaluation of Netbed users' interactions or users' opinions has been carried out so far.

Accordingly, this paper takes the above research one step further by describing a long-term experience regarding the use of PlanetLab, a real Internet-scale environment which allows students to test protocols and services under more realistic conditions than the ones artificially generated by simulated or emulated environments. Moreover, the paper focuses on students' opinions, behaviour, and performance while using PlanetLab. In our opinion, the results of this study can be helpful for the increasing number of instructors interested in using realistic testing environments distributed over the Internet.

Learning context and activities

The experience described in this paper was carried out at the Open University of Catalonia (UOC), a distance-based university which has offered online degrees since 1995. The UOC course on Internet-scale systems is offered to students of both telecommunications engineering and computer science, and it is typically enrolled by approximately 100 students per semester. Students on this course are required to carry out some assignments regarding the extension of several distributed protocols and services partially developed by the course instructors. The code provided to students already implements those parts which are not directly related with the course learning goals. In order to perform the assignment, students must perform the following tasks, either individually or collaboratively in small groups: (a) to implement a set of well-known distributed protocols and algorithms; (b) to test these implementations on a single computer; and (c) to test these implementations over a real Internet-scale environment called PlanetLab@UOC, which involves distributed execution of these implementations using several PlanetLab hosts. Notice that in the last step students' protocol implementations will have to interact with the framework or template previously developed by the instructors and already deployed over PlanetLab.

To test a protocol or service in PlanetLab many steps have to be done: register to the system, select a set of hosts, install and configure each host, deploy the protocol or service in each node, coordinate the execution, collect results, etc. Apart from being time consuming, some of these activities fall outside the scope of a typical course covering Internet communication protocols. To overcome these nuisances we developed PlanetLab@UOC, a platform based on PlanetLab to automatically deploy protocols or services in a set of hosts, coordinate their execution, perform predefined tests, collect the results, assess them and provide feedback to students. We have already used it during several semesters to teach an undergraduate course on Internet-scale systems at our university. Using PlanetLab@UOC, students execute and test their Internet protocol or service under realistic working conditions, including real network latencies, unexpected node failures, bandwidth variations, etc. More specifically, we have organized and implemented the learning scenario illustrated in Fig. 1. In this scenario, students collaborate online in small groups to implement a distributed protocol or service. Then, students run their implementation on the PlanetLab@UOC environment, so that it interacts with a framework previously implemented and deployed by instructors in PlanetLab. During this interaction process, several programmed tests are carried out in order to validate the behaviour and performance of the students' implementation. Finally, results of each test

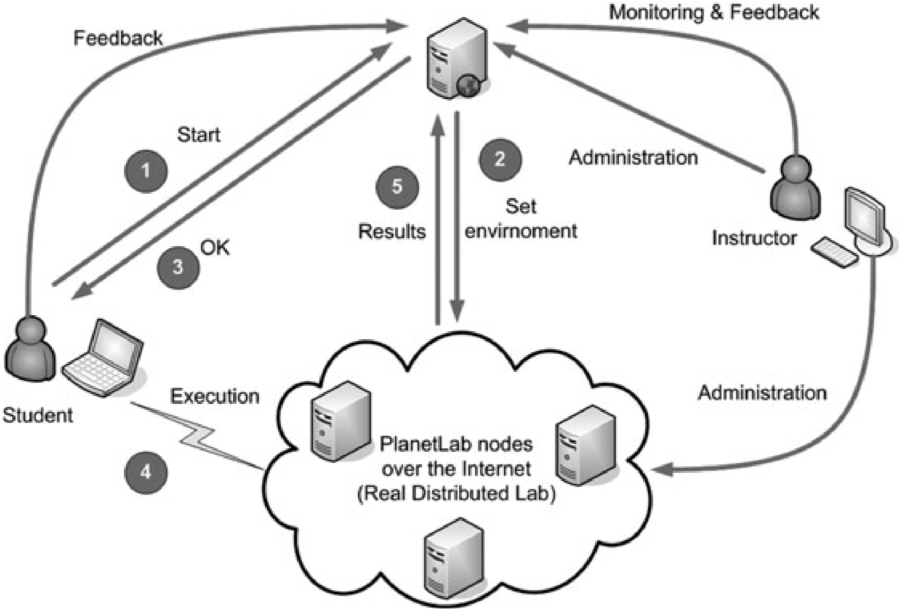

Fig. 1 *Using a real lab over the Internet in our distance-learning courses.*

are stored in a server and made available to students and instructors through a web site.

Thus, an example of an assignment could be to implement a consensus protocol in an Internet-scale environment. While implementing their solution, students could locally test their application using the script for local execution provided by the instructors. Once completed, their solutions can be finally tested on a set of PlanetLab hosts using the script provided for this purpose by the instructors. At this moment, several tests are automatically performed on a large set of geographically distributed PlanetLab hosts. The results of these tests are collected and evaluated by the PlanetLab@UOC system. Different reports are then generated and made available to both students and instructors. These reports not only contain scores for each student or group of students, but also information about possible errors that might have occurred during the testing process, either due to implementation pitfalls or to unexpected problems associated with any involved real-life system. As pointed out by some authors[11], this kind of information can be especially useful in the case of blended and distance-learning education, where students and instructors do not have the opportunity to meet face-to-face in a regular basis. Figure 2 shows a typical student's work flow when developing and testing his/her implementation.

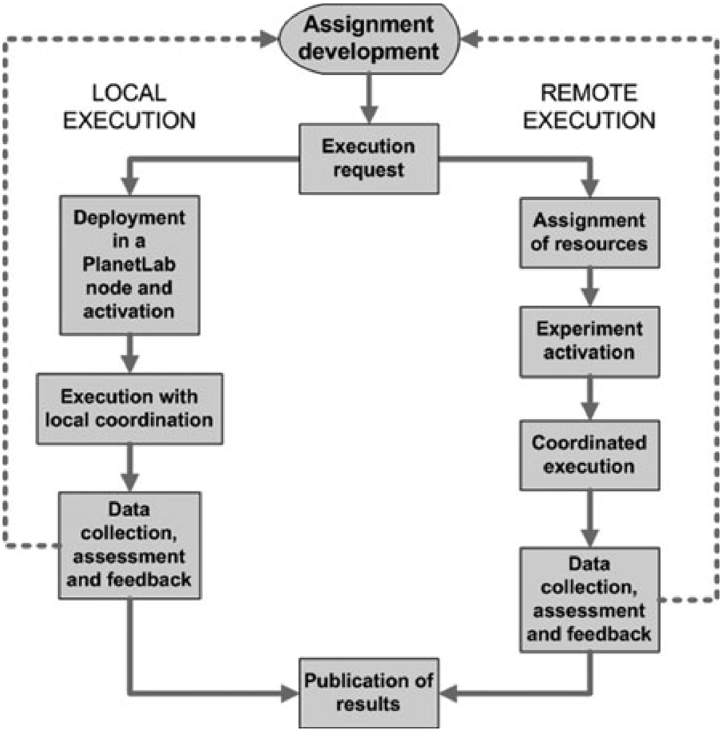
Assignment development
LOCAL EXECUTION
REMOTE EXECUTION
Execution request
Deployment in a PlanetLab node and activation
Execution with local coordination
Data collection, assessment and feedback
Assignment of resources
Experiment activation
Coordinated execution
Data collection, assessment and feedback
Publication of results

Fig. 2 *A typical student's work flow.*

Analysis of a five-semester students' survey

In order to evaluate students' opinions about several issues related with the course, we asked students to participate in a survey at the end of each semester. Table 1 shows the questionnaire we utilised during the five analysed semesters, from autumn 2008 to autumn 2010. Notice that while a simulated test environment was used

TABLE 1 *The 5-semester questionnaire*

#	Question	Autumn 2008	Spring 2009	Autumn 2009	Spring 2010	Autumn 2010
	Students registered	89	79	76	75	74
	Survey participants	29	28	24	19	13
	Survey rate	33%	35%	32%	25%	18%
1	Rate from 1 (lowest) to 3 (highest) the overall difficulty of implementing a distributed algorithm in a realistic distributed environment	2.4	2.5	2,6	2.3	2.6
2	Which are the main complexities of the assignment? (more than one option is allowed):					
	Understanding the problem	41%	36%	63%	45%	58%
	Understanding the algorithms	52%	46%	42%	5%	50%
	Implementing the algorithms	41%	39%	25%	15%	17%
	Developing an application that works as intended	59%	57%	67%	65%	25%
	Interacting with the template provided by the instructors	28%	32%	29%	50%	42%
	Interacting with instructors instances deployed in PlanetLab	N/A	75%	54%	15%	25%
	Other	0%	39%	63%	35%	25%
3	Splitting the activity in several phases helps me to manage the different tasks	89%	81%	72%	74%	82%
4	Splitting the activity in several phases allows me to organise and regulate my effort	75%	75%	67%	50%	62%
5	In case you have not fully completed all the phases, what was the reason? (more that one option is allowed)					
	Not interested in the course	0%	0%	0%	0%	0%
	Lack of time due to personal or professional issues	39%	31%	21%	43%	36%
	Excessive difficulty of some phases	39%	11%	5%	21%	0%
	Excessive time requirements	18%	31%	42%	29%	27%
	Interested in distributed systems but not in implementation	3%	3%	5%	0%	9%
	Other	0%	23%	26%	7%	27%
6	This practical assignment has been… (more that one option is allowed)					
	Useful to understand important issues on distributed systems	66%	89%	79%	74%	91%
	Useful to acquire practical knowledge on distributed systems	62%	61%	58%	63%	64%
	Helpful to understand how to implement real distributed systems	55%	50%	50%	89%	82%
	Other	0%	14%	17%	5%	18%

TABLE 1 *Continued*

#	Question	Autumn 2008	Spring 2009	Autumn 2009	Spring 2010	Autumn 2010
7	Is the assignment well related with the course goals? (the percentage of positive answers is shown on the right)	83%	75%	71%	79%	82%
8	The statement of the assignment was…					
	Clear and well detailed	21%	29%	13%	16%	8%
	Easy to follow but not detailed enough	55%	39%	21%	58%	50%
	Unclear and poorly detailed	24%	29%	58%	26%	42%
9	About the statement of the assignment…					
	I appreciated that it was closed and the algorithms to be used were given beforehand by instructors	66%	61%	75%	74%	42%
	It was too closed and I would have preferred a design phase to decide which algorithms to use	10%	11%	13%	11%	0%
10	Difficulty of phase 1 (1 = lowest; 3 = highest)	1.7	1.8	1.8	1.5	1.2
11	Difficulty of phase 2 (1 = lowest; 3 = highest)	2.4	2.6	2.7	2.3	2.3
12	Difficulty of phase 3 (1 = lowest; 3 = highest)	2.7	2.5	2.7	2.5	2.3
13	Do you prefer to execute your application in a realistic environment instead of using your PC? (the percentage of positive answers is showed on the right)	52%	70%	75%	79%	75%
14	Rate from 0 (completely unsatisfied) to 10 (completely satisfied) your overall satisfaction with the practical assignment	6.7	5.6	5.9	7.3	6.9

during autumn 2008, from spring 2009 we used a real Internet-scale test environment, PlanetLab@UOC. Therefore, significant methodological and technological changes took place between the first and the subsequent academic periods. Of course, we were especially interested in analysing to which extent, if any, these changes modified students' perceptions about our course.

The questionnaire was sent to each student individually the same day he/she received his/her final marks. From the survey results in Table 1, the following conclusions can be derived:

- The overall difficulty of the assignment is considered to be high in all the semesters.
- Regarding the main difficulties students found while completing their assignments, notice that in spring 2009 around 75% of participants considered 'interacting with instructors instances deployed in PlanetLab' as their main problem. This percentage decreased in the next semesters. Attention should also be paid to 'developing an application that works as intended' (high percentages in 3-out-of-5 semesters. These difficulties could be due to a combination of the following two facts: (a) by having to interact with instructors' implementation, students were forced to use a given interface and adapt their codes to it; and (b) when

tests did not work properly, distributed debugging is a hard issue and no hints were provided about which part of the code was generating the problem. This lack of feedback when interacting with instructors' instances was one of the main students' complaints during spring 2009. On the contrary, in autumn 2008 students executed their experiments in local mode and, therefore, executions were easy to track. Data from the last three semesters suggest that the improvements performed on the testing environment after spring 2009 contributed to improve students' perception on this 'lack-of-feedback' issue.

- Regarding the splitting of the assignment into different phases, students from all the semesters clearly agree that it helps them to manage their tasks and regulate their effort better.
- Notice also that most students agreed that the test-environment assignment was useful to understand important issues on Internet-scale systems, to acquire practical knowledge on the subject, and also to work with real-life distributed systems. This is a very encouraging result. Especially because, as discussed before, students encountered a lot of difficulties creating an application that runs over a real distributed setting.
- Students' opinions also show that the assignment is coherent with the course goals. However, it seems clear that improvements should be made in the assignment statement, since an important percentage of students did not consider it to be clear and detailed enough. Nevertheless, a considerable percentage of students appreciated that the statement was closed (quite specific) and that some of the algorithms to be used were already provided by instructors.
- Regarding the difficulty of each phase, both Phase 2 and Phase 3 are significantly more difficult than Phase 1. Notice that average rates for each phase are approximately equal across semesters.
- Only about 52% of autumn 2008 students preferred to execute their application over a real Internet-scale environment, while this percentage increased significantly in the following four semesters. This seems to suggest that students who have already worked with a real testing environment prefer using it over a virtual one.
- Finally, Fig. 3 shows a multiple box-plot representing students' overall satisfaction levels in each semester. Notice that the overall satisfaction decreased the first semester PlanetLab@UOC was used. This is probably due to the increase in difficulty associated with using a real Internet-scale environment instead of a local setting. However, as can be seen in Fig. 3, this overall satisfaction gradually returned to its initial level as we were able to improve the feedback capabilities of PlanetLab@UOC. In summary, after introducing some enhancements in the realistic testing environment, we were able to return to the high satisfaction levels we had when using a less realistic but simpler setting.

Analysis of students' assignment execution traces

Students' execution traces were obtained for the 2009 fall semester. A total of 1,187 observations were registered. These observations were generated by the interaction of 40 working groups with the distributed environment. The composition of these

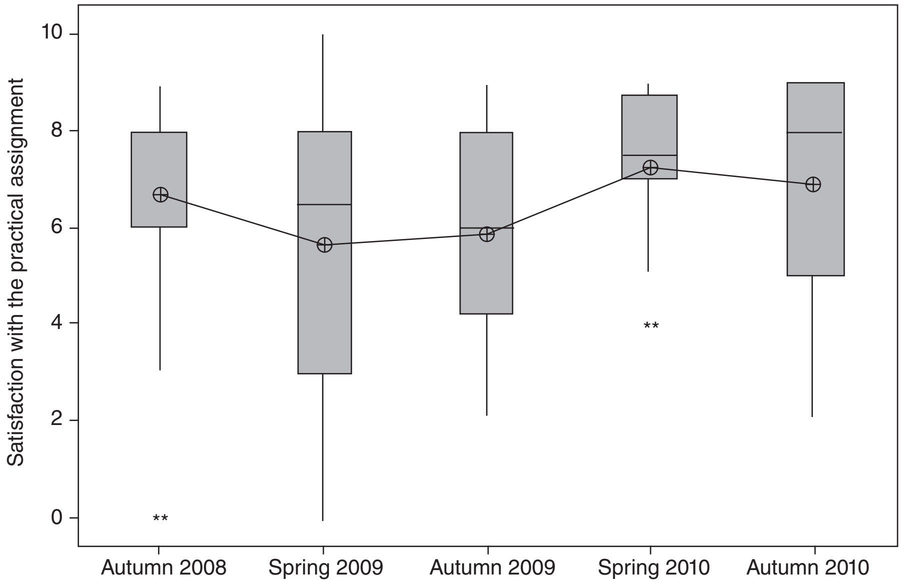

Fig. 3 *Multiple box-plot for overall satisfaction by semester.*

40 groups was as follows: 11 were composed of 2 students, while the remaining 29 were composed of individual students who decided to work alone. The traces obtained are described by the following variables:

- *Id:* Execution identifier; it is sequentially increased.
- *Id_Group:* Identifier of the group running the assignment.
- *Date:* Date of execution (dd.mm.yyyy).
- *Time:* Time of execution (hh:mm:ss).
- *Mode:* Mode in which the assignment was executed, i.e. Local (Mode = 0) or Remote (Mode = 1) (Fig. 2).
- *Results:* Number of tests, from 0 to 3, satisfactorily completed by a group in each run.

Students interacted with the system from 11.29.2009 to 12.21.2009, i.e. during 23 days. Executions took place anytime during the aforementioned period, 6 p.m. being the observed median, which suggests that most students interact with the system after their working hours. It should be noticed that most UOC students combine their studies with their professional activity and with their family duties (most of our students are between 25 and 45 years old), which explains this behaviour[12].

Approximately 50.9% of executions were run in local mode, while the rest (49.1%) were run in remote mode. About 45% of the times an execution was run in local mode, no test was successfully passed. Analogously, about 68% of the times an execution was run in remote mode, no test was successfully passed. This might suggest that the assignment is not a trivial activity for students, since they usually require several executions in order to succeed in at least one test (Fig. 4). Also, based on instructors' opinion, most students execute their assessments following a trial-and-error strategy. Notice, however, that whenever an execution has been able to pass one test, it is likely that it also passes other tests. In other words, there might be an 'initial barrier' during the development of the activity, but once this barrier is over, students are able to get good results within a few executions.

Concerning the number of executions completed by a group during the system interaction period, this variable shows a high variance, with values ranging from 6 to 96 executions depending on the group. Figure 5 shows the within-group differences between the number of local-mode execution and the number of remote-mode

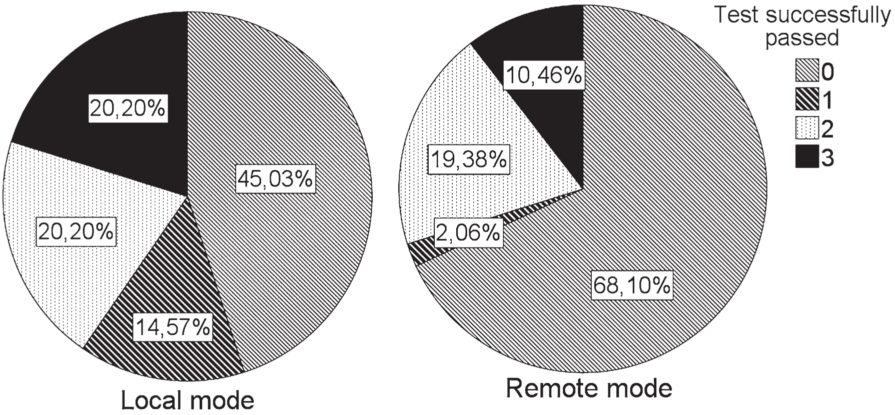

Fig. 4 *Percentage of tests that are successfully passed.*

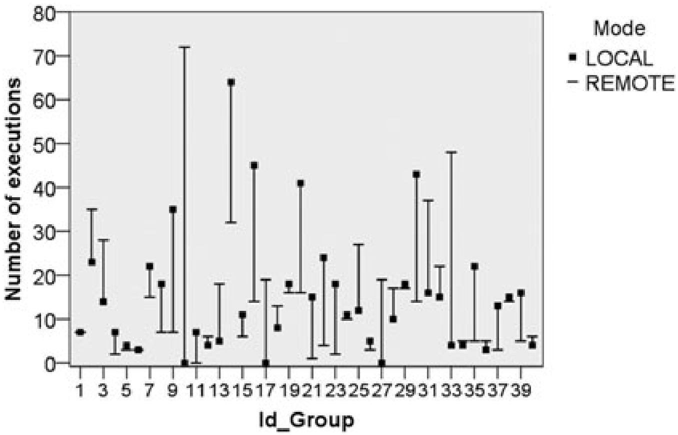

Fig. 5 *Number of executions by group in each mode.*

executions. Considering the first time each group interacts with the system, it is interesting to highlight that 14 out of 40 groups (i.e. 35% of the groups) perform executions in remote mode before performing runs in local mode – which is assumed to be somewhat easier. In other words, even when most groups tend to follow the more 'conservative' learning path (first try the easy mode, then the difficult one), there still is a perceptible percentage of groups which prefer to try the remote mode first.

The Pearson correlation coefficient associated with the variables 'number of executions by group' and 'group final marks' is 0.314 (p-value = 0.052). Therefore, we concluded that there was not a significant (linear) relationship between both variables. In other words, the number of times a group has interacted with the system is not strongly correlated (at least in linear terms) with its final marks.

Figure 6 shows the number of executions in each working day by execution mode. Similarly, Fig. 7 shows the number of groups performing runs for each working day. From both graphs, it can be derived that:

- During the first 5 or 6 days, there is a low activity level. This might be due to the fact that students need about one week to fully understand the task and the environment, as well as to complete their first code implementations.
- On the contrary, activity level is much higher during the last days. This is probably due to the fact that some students need to work hard during the last available days in order to complete their assignments.

Figure 8 shows daily system activity by group. A dot *(x, y)* in the plot represents the fact that the *y*th group completed one or more executions during the *x*th day. Notice that while some groups interact with the system from the very beginning (e.g. groups

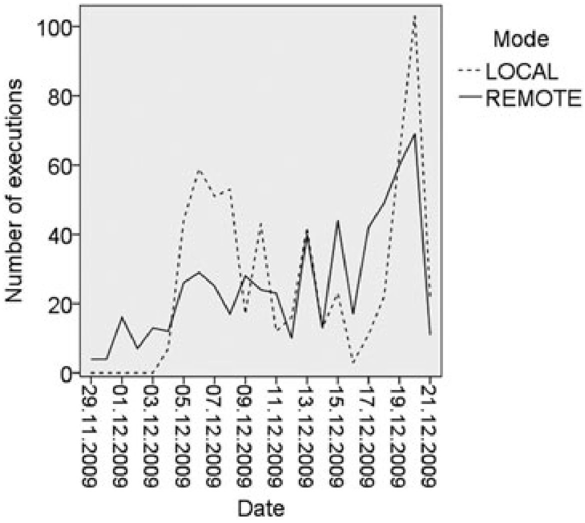

Fig. 6 *Number of executions by day and mode.*

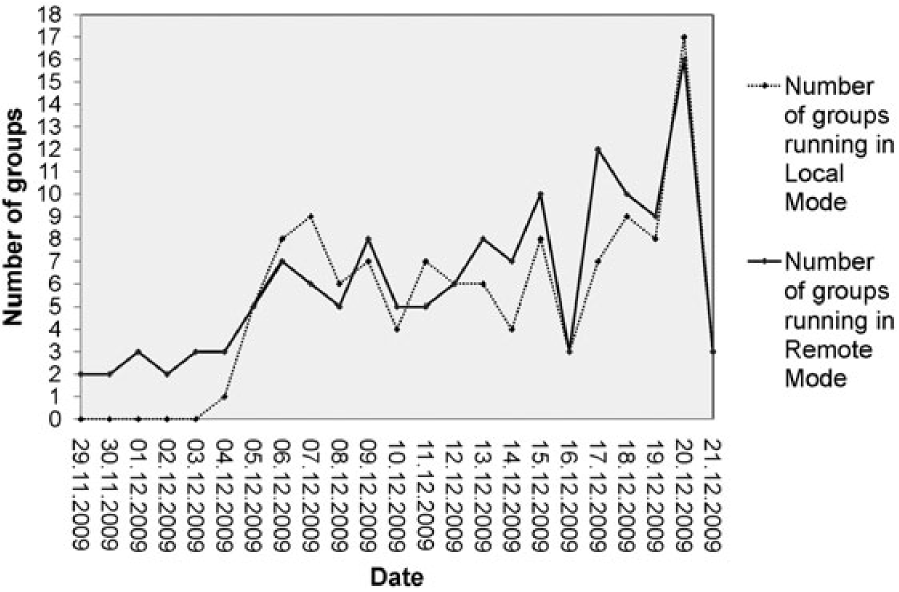

Fig. 7 *Number of active groups by day and mode.*

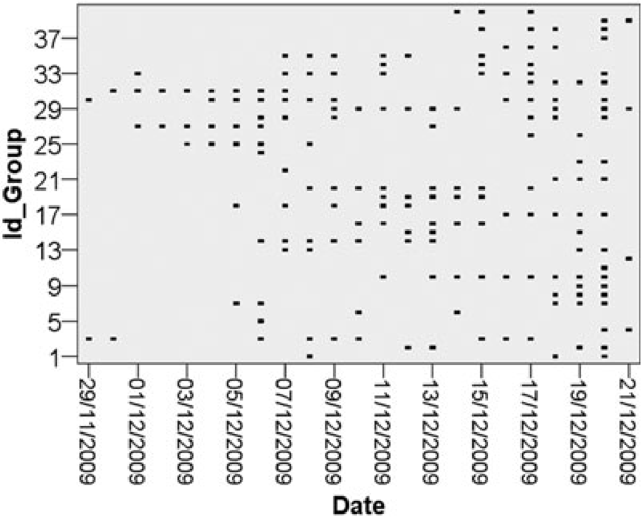

Fig. 8 *System daily activity by group.*

3, 30, 31 or 33), other groups do not interact with the system until the last working days (e.g. groups 38, 39 and 40). This type of graphics can be a powerful tool for monitoring groups' activity levels throughout the course[13]. Finally, Fig. 9 shows daily system activity by daily hour. A dot *(x, y)* in the plot represents the fact that

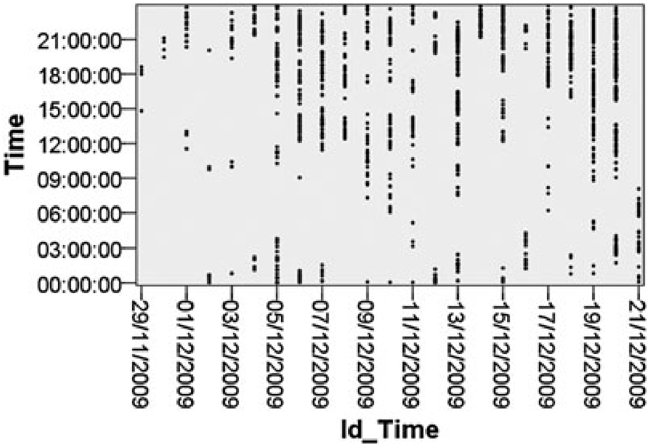

Fig. 9 *System daily activity by daily hour.*

executions have been running at time *y* of day *x*. As expected, the activity is low or almost non-existent between 3 a.m. and 7 a.m., while it seems to be more intense between 8 p.m. and 1 a.m. Observe that activity seems to increase somewhat on weekends and holidays (November 29; December 5, 6, 7, 8, 12, 13, 19 and 20).

Conclusions

This paper has presented a long-term distance-learning experience regarding the use of a real Internet-scale protocol-testing environment. In our study, we have collected, organised and analysed a variety of data that concern student-system interactions, students' behaviour and students' opinions regarding the use of our platform, PlanetLab@UOC. On the one hand, a students' survey allowed us to draw some useful conclusions which can be summarised as follows: (a) the main difficulties that students found when completing their practical assignments are, in this order, 'interacting with instructors instances deployed in PlanetLab', 'developing an application that works as intended', and 'understanding the algorithms'; (b) the main reasons why some students could not complete all phases of the project are, in this order, 'lack of time due to personal or professional issues' and 'excessive time requirements'. On the other hand, an analysis of the system logs provided the following findings: (a) most students interact with the system after their working hours; (b) there is an 'initial barrier' during the development of the activity, but once this barrier was overcome, it is expected that students get some good results within few executions; (c) most groups tend to follow the more 'conservative' learning path, i.e., from local to remote execution; (d) the number of times a group has interacted with the system is not by itself a reliable factor to estimate its final marks; (e) the activity level during the last days is much higher than the average activity level; (f) while some groups interact with the system from the very beginning, other groups do not interact with the system until the last working days; (g) the activity is low or almost non-existent between 3 a.m. and 7 a.m., while it seems to be more intense between 8 p.m. and 1 a.m. All in all, we expect that the experiences described in this paper can be helpful for other instructors interested in integrating real Internet-scale protocol-testing environments in their engineering courses.

Acknowledgments

This research work has been partially funded by the Catalan Government under grants 2009MQD00121 and 2009 CTP 00007, as well as by the CYTED2010-511RT0419 network (http://dpcs.uoc.edu).

References

1 I. Calvo, M. Marcos, D. Orive and I. Sarachaga, ‘Building complex remote learning laboratories’, *Computer Applications in Engineering Education*, **18** (2010), 53–66.

2 J. Albrecht, C. Tuttle, A. C. Snoeren and A. Vahdat, ‘PlanetLab application management using plush’, *SIGOPS Operating Systems Review*, **40** (1) (2006), 33–40.

3 J. M. Marquès, D. Lázaro, A. Juan, X. Vilajosana, M. Domingo and J. Jorba, ‘PlanetLab@UOC: A Real Lab over the Internet to experiment with Distributed Systems’, *Computer Applications in Engineering Education*, 2010 (doi: 10.1002/cae.20468).

4 A. Kayssi, S. Sharafeddine and H. Karaki, ‘Computer-based laboratory for data communications and computer networking’, *Computer Applications in Engineering Education*, **12** (2004), 84–97.

5 E. Martínez-Caro, ‘Factors affecting effectiveness in e-learning: An analysis in production management courses’, *Computer Applications in Engineering Education*, 2009 (doi: 10.1002/cae.20337).

6 V. Benigno and G. Trentin, ‘The evaluation of online courses’, *J. Computer Assisted Learning*, **16** (2000), 259–270.

7 L. Leonini, E. Rivière and P. Felber, ‘SPLAY: distributed systems evaluation made simple (or how to turn ideas into live systems in a breeze)’, in *Proc. 6th USENIX Symposium on Networked Systems Design and Implementation*, 2009, pp.185–198.

8 C. Ivica, J. Ridley and C. Shubert, ‘StarHPC - teaching parallel programming within elastic compute cloud’, in *Proc. 31st Int. Conf. on Information Technology Interfaces (ITI 2009)*, 2009, pp.353–356.

9 A. Vahdat, K. Yocum, P. Mahadevan, D. Kostic, J. Chase and D. Becker, ‘Scalability and accuracy in a large-scale network emulator’, in *Proc. 5th Symposium on Operating Systems Design and Implementation*, 2002, pp.271–284.

10 B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb and A. Joglekar, ‘Netbed: an integrated experimental environment’, *ACM SIGCOMM Computer Communication Review*, **32**(3) (2002), 27.

11 A. Juan, T. Daradoumis, J. Faulin and F. Xhafa, ‘A data analysis model based on control charts to monitor online learning processes’, *Int. J. Bus. Intell. Data Mining*, **4** (2009), 159–174.

12 C. Zirkle, ‘Distance education in career and technical education: A review of the research literature’, *J. Vocat. Educ. Res.*, **28** (2003), 151–171.

13 A. Juan, T. Daradoumis, J. Faulin and F. Xhafa, ‘SAMOS: A model for monitoring students’ and groups’ activity in collaborative e-learning’, *Int. J. Learning Technology*, **4**(1/2) (2009), 53–72.