

Supporting the Delivery of Learning-Contents with Laboratory Activities in Sakai

Work-in-Progress report

Muhammad WANNOUS, *IEEE Student Member*

Graduate School of Science and Technology
Kumamoto University
Japan
muhammad.wannous@ieee.org

Hiroshi NAKANO

Center for Multimedia and Information Technologies
Kumamoto University
Japan
nakano@cc.kumamoto-u.ac.jp

Abstract—Educational and training-providing institutions are increasingly depending on the web-based *Learning Management System (LMS)* for delivering course-contents in both on-campus and distance classes. This system offers an easily-accessible environment with various tools for managing learners and courses, and for producing rich contents. However, beside the rich contents, certain fields of study require that courses be accompanied with/ followed by laboratory activities which help learners deepen their understanding of the new concepts and develop skills on applying them in close-to-real-world context. More emphasize has been put on these activities when laboratory-based courses appeared and proved to be functioning well in delivering the information. Several stand-alone laboratories have been constructed to support delivery of such activities on-line, but laboratory-tools are not available within the LMS. To address this point, our research presents an approach towards integrating one stand-alone laboratory (NVLab) into Sakai LMS to enable the use of laboratory activities within the course context. The study suggests integrating parts of the original system into Sakai, and then using the LMS's "linktool" to connect to the other part. So far, an initial version of the new tool has been accomplished and we are in the testing phase of the new tool on a demo version of Sakai before putting it into service within the campus-wide system.

Keywords- K3.2 Computer and Information Science Education; K3.1.d Distance Learning; Web-based Laboratories; Learning Management Systems

I. INTRODUCTION

Dependence on web-based *Learning Management Systems (LMS)* is increasing in educational and training-delivery institutions. These systems are used not only with online-courses, but also with those delivered on-campus.

Web-based LMS is easily-accessible and incorporates tools for managing courses, learners, grades...etc, and other tools for supporting the learning process. The later are designed to assist in producing rich contents and to facilitate the interaction among learners and between them and their instructors.

In many cases, what the LMS offers is enough for building an innovating class structure; however, in certain fields of study more is required for successfully delivering the information and building the learners' capacity for real world context. This is certainly true for *Information and Communication Technologies (ICT)* courses where laboratory

activities are necessary for deepening the understanding and building skills for applying the concepts being introduced [1]. Instructors put further emphasis on such activities when they successfully introduced on-campus laboratory-based courses [2].

Since current LMSs lack the capability to offer experience-building tools, and to address the need for incorporating laboratory-facilities within online courses, stand-alone web-based laboratories came to existence in a wide range of engineering fields-of-study. Examples of such systems include those which used on-campus facilities equipped with access-networks [3] [4], and others that used modern technologies for constructing the laboratory [5] [6] [7]. This arrangement worked well for classes structured in the conventional way where students get the information first and later apply what they have learnt in the laboratory.

But in contexts where carefully-designed dynamic-contents/practical-activities were combined with the course static-contents it was proven that learning-performance was better in engineering and medical sciences [8] [9]. This, in addition to the success of laboratory-based courses, indicates the importance of building a tool for either integrating practice-acquiring facilities into the online-course, or linking the stand-alone laboratory with the LMS.

• Paper contribution

In this paper, we demonstrate our technical approach towards integrating a stand-alone web-based laboratory (NVLab) that we already built for supporting courses concerning *Computer Networks* into Sakai LMS. We suggest:

- Integrating the tool the learners used for producing a network-design in NVLab into Sakai. and
- Using the LMS's *linktool* to enable them access the networks corresponding to their designs which the system constructs on separate servers.

The rest of the paper is organized in sections as follows: section II provides brief background about NVLab and Sakai, and section III presents technical description of the new system development process and requirements. Then in section IV we introduce the task-flow of a simple exercise that we designed for testing the new system within Sakai. Finally, we end with our conclusions in section V.

II. BACKGROUND

NVLab [10] is a web-based laboratory that we designed for supporting online courses concerning *Computer Networks*. It introduces to the learners a tool for producing a network drawing, then uses *Virtualization (Xen)* [11] technology to realize this network on a single server by instantiating *Virtual Machines (VMs)* that correspond to the devices included in the network diagram. Learners are able to remotely access the VMs' terminals via the *Virtual Network Computing (VNC)* [12] [13] module (vnc-server) running on the server.

Students are able to use NVLab for constructing networks, trying to apply what they have learned, and finding out what problems may rise with their designs. Instructors, on the other hand, use it for producing properly/improperly configured network designs and ask the learners to verify/troubleshoot the settings of the devices included in these networks.

We introduced a sample exercise on IP-addressing to a group of learners and asked them to use NVLab to complete it and provide feedback on their experiences using the system. The participants in this experiment completed the exercises successfully and commented positively on the system [10]. Figure 1 shows a screenshot for a learner's machine while connected to NVLab.

As mentioned before, our goal is to have a new laboratory-tool to be used with online courses. For this, we decided to use NVLab and Sakai LMS [14]. Sakai has been selected for some specific reasons: 1) it is an open and freely available LMS that is kept by a community on the Internet; 2) it is evolving all the time and the number of organizations implementing it is increasing; 3) Sakai is written in Java language which is the same used in NVLab and this would make our progress faster.

Courses in Sakai are created using *Sites*, and each *Site* is a workspace that has a number of tools available for all participants. Figure 2 shows one course-*Site* titled *ComNet* which uses Sakai's *web-contents* tool for introducing some information related to *Networking in Java* from Sun website [15].

Tools available in Sakai are of wide range to cover the students' and faculty staff needs. These include: general collaboration tools (like *resources*), teaching and learning tools (like *syllabus*), portfolio tools (like *reports*), and administrative tools (like *users*). The instructor has the right to include the tools he/she sees necessary for conducting the various tasks within the course from Sakai's rich tool-pool.

III. TECHNICAL DESCRIPTION

Incorporating the laboratory system as new tool in the LMS will enable us:

- Leave all administrative tasks, previously considered in NVLab, to Sakai.
- Save information necessary for tracking the learner's advancement while progressing through the course.
- Replace the VM-management solution, *virt-manager* in NVLab, which learners found difficult to use with another application that introduces a web-based interface.

• System Description

NVLab (Figure 3) comprises 3 main parts: 1) the *virtual network*, which the learners create, 2) tools for producing the network (*Designer* and *Builder*), and 3) tools for accessing/managing the network (*vnc-server* and *virt-manager*).

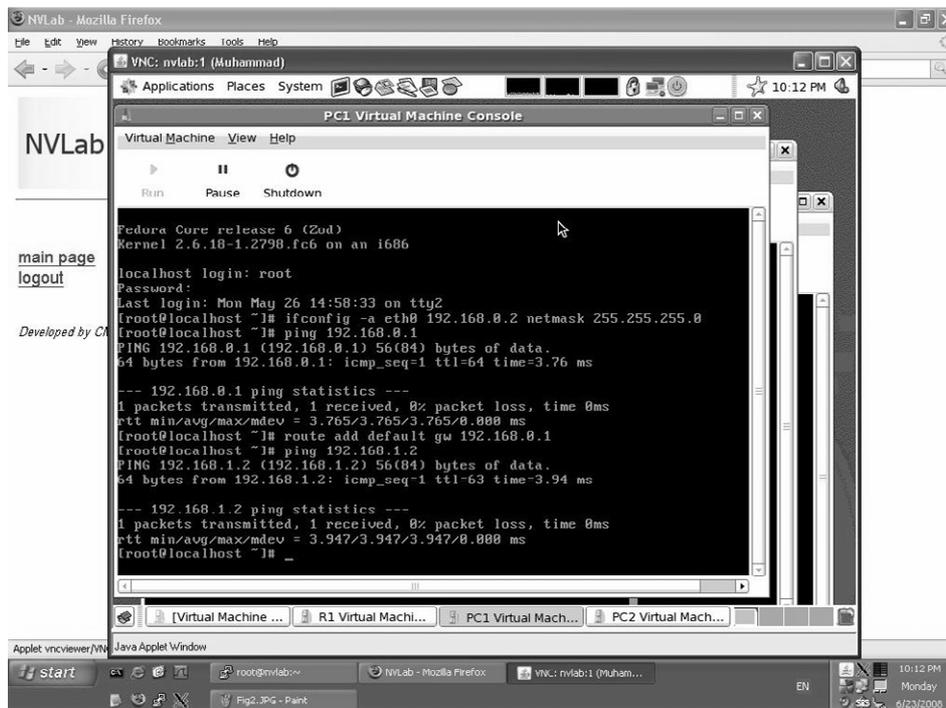


Figure 1. Learner's machine when logging into NVLab

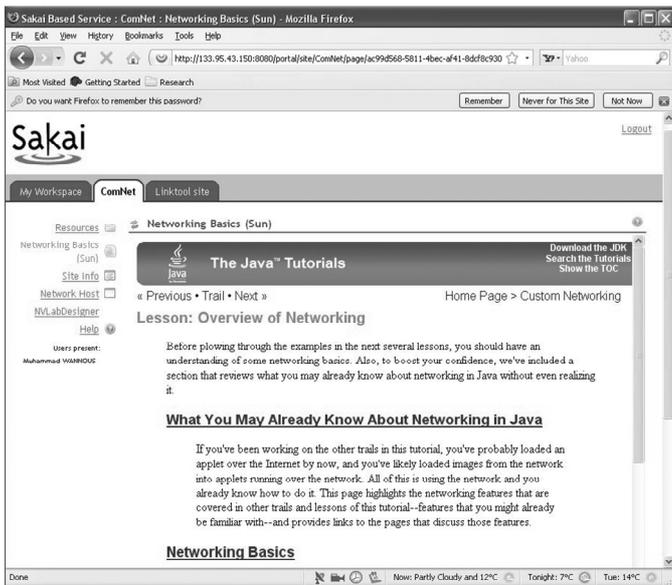


Figure 2. One course-Site in Sakai

We introduced two new tools within NVLab: *Designer* (Java Applet) and *Builder* (Java Servlet). A learner used the former tool to draw a network diagram and to save this diagram on the server side; then he/she invoked the later tool which used prototype VMs to create the network on the server. When the first phase is completed, a learner was able to connect to the machine hosting his/her network using the web-browser, and then used *virt-manager* to manage the VMs and start working on their configurations.

The new system has almost the same structure except that parts of it have been integrated into Sakai as shown in Figure 4. In the new arrangement, the laboratory comprises 3 main parts also: Sakai, *NVLabManager*, and a pool of *NetworkManager* servers. Network-designs are drawn in Sakai, while the system creates each of these designs on one *Networkmanager* server. *NVLabManager* works as a check point and server-pool controller /access mediator.

On Sakai side learners use *Nvlabdesigner* to draw a network diagram and save it as an xml-file in the course-site *resources*. This is a tool with GUI interface that enables producing a graphical representation of a network as shown in Figure 5. Copies of the network-diagrams generated by the students are also saved in a hidden space where only the instructor can access them for monitoring the learners' progress. When invoked from *NVLabDesigner* webpage, *createNetwork* Servlet reads the newest xml-file from the *resources* and sends it to *NVLabManager*.

NVLabManager keeps records of the users holding permission to use the laboratory and the servers available for the learners. It assigns one free *NetworkManager* server to the student when it receives a new diagram xml-file, and frees this server when the learner has finished his/her work. In this web-application, *serverManager* Servlet receives the xml-file, verify the user permissions, assign a free server to the user, and then redirect the xml-file to that server in order to create the network contained in it. *NVLabManager* also works as a

gateway for accessing each *NetworkManager* through Sakai *linktool*.

Actual work is done on *NetworkManager* where the *Virtual Network* resides. The *netCreator* creates the *Virtual Network* by instantiating VMs from a VM-pool available on each server, and then starts them in preparation for incoming connection from the user side. Creating each VM involves: 1) moving a file that acts as a storage medium for the OS running inside the machine, 2) writing a configuration file that defines the resources available to it (sample is shown in Figure 6.), and 3) writing an html file that contains the definition of a VNC Applet that is used to view the VM's terminal window. The Applet takes one parameter that indicates the TCP port number to which the terminal window is bound (sample html file is shown in Figure 7).

The other Servlet called *netManager* receives the connection initiated from Sakai *linktool* included in the course *Site*. This Servlet is partially based on a web-based Xen VM management tool [16] that we previously introduced. *netManager* provides a simple web-interface for managing the VMs in the *Virtual Network* as shown in Figure 8. In this interface, links to the html files previously created by *netCreator* are included. When accessed via these links, the learner will get a view like the one shown in Figure 9.

- *Development Environment*

Current work has been done and tested using: Sakai 2.5.4 demo version [17], the famous web-application container Tomcat Apache 5.5.25 [18], Eclipse IDE [19], and NetBeans 6.5.1 IDE [20].

Sakai developers considered that many instructors may wish to build tools that better meets achieving certain functions in the course, and therefore introduced many useful information/software pieces that simplify this task. One simple Sakai tool is considered to be a web application with special configuration file that instructs Sakai to include this application within its pool. To develop *NVLabDesigner*, we used *AppBuilder* available from Sakai developers' community [21].

IV. TASK FLOW IN A SAMPLE EXERCISE

Considering a course on *Computer Networks* where learners are supposed to gain complete understanding of IP

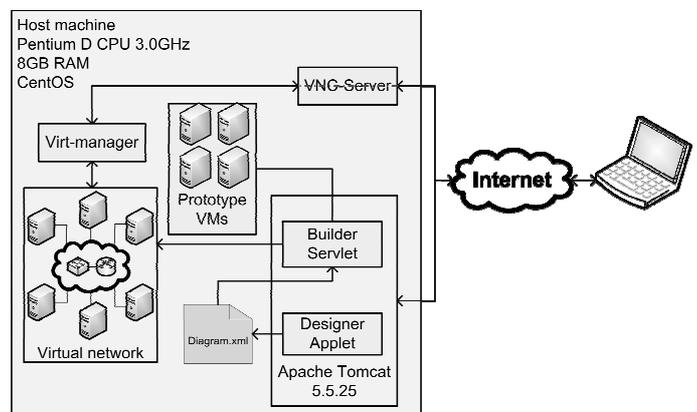


Figure 3. NVLab structure

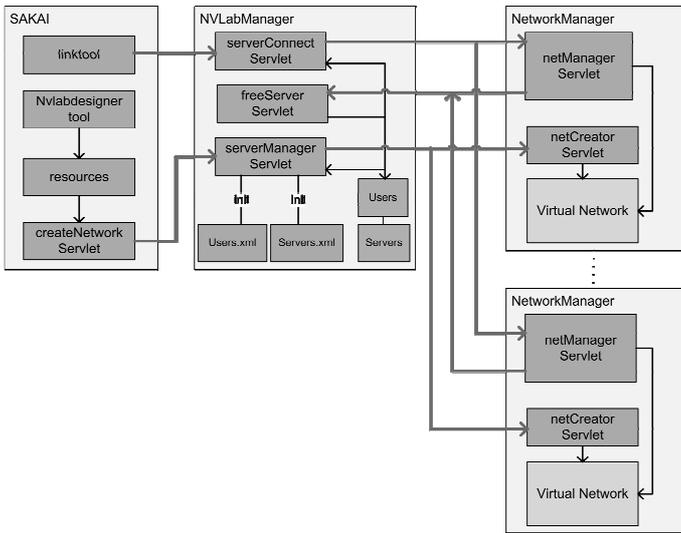


Figure 4. The new system structure

addressing concepts, the new tool has been configured and included in the course to introduce a simple exercise to the learners. The suggested exercise involves assigning the correct IP addresses to two computer machines which are connected to one switch and test the connectivity between them using the “ping” command. The original contents of the course provided the necessary background information on IP address structure and sub-net, and the student was asked to use the laboratory to apply what he/she has learnt.

The learner starts by drawing a network similar to the one shown in Figure 5 and then saves this network-diagram in the *resources* by selecting the proper option from *Nvlabdesigner* Applet’s menu (save). Next, by clicking on the link *Create Network* the learner sends the diagram to *NVLabManeger* which will assign a free server to the user and redirect the diagram information to it.

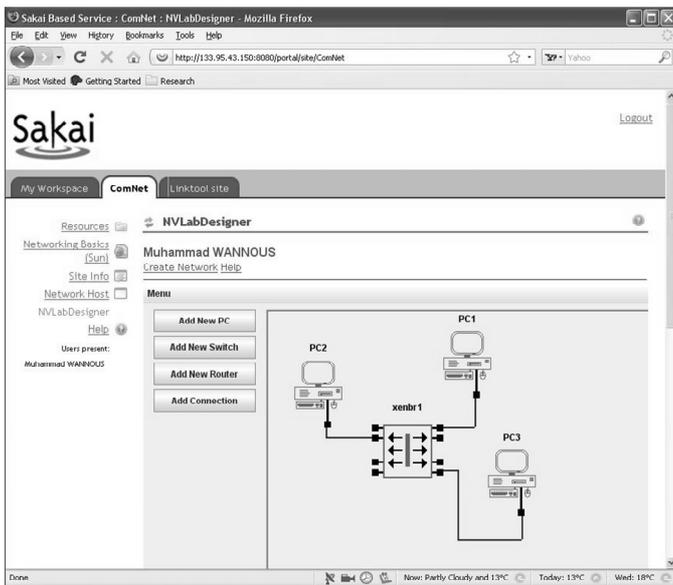


Figure 5. NVLabDesigner GUI

```

name = "PC1"
maxmem = 64
memory = 64
vcpus = 1
builder = "hvm"
kernel = "/usr/lib/xen/boot/hvmloader"
boot = "c"
pae = 1
acpi = 1
apic = 1
localtime = 0
on_poweroff = "restart"
on_reboot = "restart"
on_crash = "restart"
device_model = "/usr/lib/xen/bin/qemu-dm"
sdl = 0
vnc = 1
vncunused = 0
vnclisten = "xx.xx.xx.xx"
vncdisplay = "5"
vncpasswd = "*****"
keymap = "en-us"
disk = [ "file:/var/lib/xen/images/PC1.img,hda,w", ]
vif = [ "mac=00:16:3e:78:d9:00,bridge=xenbr1,type=ioemu" ]
serial = "pty"

```

Figure 6. Sample VM configuration file

Upon receiving this diagram, *NetworkManager* will create a *Virtual Network* that corresponds to the diagram, start the machines included in it, prepare the necessary files for accessing the VMs' terminals, and send a confirmation to the learner side indicating success. The learner is then able to use Sakai *linktool*, renamed as *Network Host* in the course *Site*, to connect to management interface residing on the server he/she has been assigned. This interface will show a list of the VMs running on the server and provide links to their terminals.

For the current exercise, we selected Fedora Core 6 to be run inside the VMs. To complete the task, any two IP addresses belonging to the same subnet will work and the learner decides to use the private network address 192.168.0.x/24, then configures eth0 interface on both machines by entering the command:

```
ifconfig eth0 192.168.0.X netmask 255.255.255.0
```

where X is 1 in the first machine and 2 in the second. Finally, to check connectivity between the two machines the learner enters the command `ping 192.168.0.2` in the terminal of first machine to get the result as shown in Figure 9.

Having completed the basic task required in the exercise, the learner may wish to move on and configure a third machine or try to make changes to the IP plan and see what happens. And when done with the network, the learner can go back and continue navigating through the other course-contents.

```

<HTML>
<HEAD>
<TITLE>
TightVNC desktop PC1
</TITLE>
</HEAD>
<BODY>
<APPLET CODE=VncViewer.class ARCHIVE=VncViewer.jar WIDTH=800
HEIGHT=632>
<PARAM name="port" value="5905">
</APPLET>
</BODY>
</HTML>

```

Figure 7. Sample html file for viewing a VM's terminal

REFERENCES

- [1] N.I. Sarkar, "Teaching Computer Networking Fundamentals Using Practical Laboratory Exercises," in *IEEE Trans. Edu.*, vol. 49, no. 2, pp. 285-291, May 2006.
- [2] P. Mateti, "A Laboratory-Based Course on Internet Security," in *Proc. of the 34th SIGCSE Tech. Symp. on Computer Science Education*, Reno, NV, 2003, pp.252-256.
- [3] C. S. Tzafestas and M. Alifragis, "Virtual and Remote Robotic Laboratory: Comparative Experimental Evolution," in *IEEE Trans. Edu.*, vol. 49, no. 3, pp. 360-369, Aug 2006.
- [4] H. A. Lahoud, Xin Tang "Information Security Labs in IDS/IPS for Distance Education," in *Proc. 7th Conf. on Information Technology Education*, Minneapolis, MN, USA, 2006, pp 47-52.
- [5] J. Nieh and C Vaill "Experiences Teaching Operating Systems Using Virtual Platforms and Linux," in *ACM SIGOPS Operating Systems Review*, vol. 40, no.2, pp. 100-104, Apr. 2006.
- [6] C. Border, "The Development and Deployment of Multi-User, Remote Access Virtualization System for Networking, Security, and System Administration Classes," in *ACM SIGCSE Bulletin*, vol. 39, no.1, pp. 576-580, Mar. 2007.
- [7] M. Anisetti, V. Bellandi, A. Colombo, M. Cremonini, E. Damiani, F. Frati, J. T. Hounson D. Rebecani, "Learning Computer Networking on Open Paravirtual Laboratories," in *IEEE Trans. Edu.*, vol. 50, no. 4, pp. 302-311, Nov 2007.
- [8] Holzinger A., Kickmeier-Rust M., Wassertheurer S., Hessinger M., "Learning Performance with Interactive Simulations in Medical education: Lessons learned from results of learning complex physiological models with the HAEMOdynamics SIMulator" in *Computers & Education*, vol.52 no.2, pp.292-301, February, 2009.
- [9] Holzinger A., Kickmeier-Rust M., Albert D., "Dynamic Media in Computer Science Education; Content Complexity and Learning Performance: Is Less More?" in *Educational Technology & Society*, vol. 11 issue 1, pp. 279-290, 2008.
- [10] M. Wannous, H. Nakano, (10 Aug. 2009) "NVLab, A Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologies," *IEEE Trans. on Learning Tech., IEEE computer Society Digital Library*. [online]. Available <http://doi.ieeeecomputersociety.org/10.1109/TLT.2009.31>
- [11] B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, I. Pratt, A. Warfield, P. Barham, and R. Neugebauer, "Xen and the art of virtualization," in *Proc. ACM Symp. Operating Systems Principles*, Bolton Landing, NY, 2003, pp. 164-177.
- [12] T. Richardson, Q. Stafford-Fraser, K. R. Wood, A. Hopper "Virtual Network Computing," in *IEEE Internet Comput.* vol. 2, issue 1, p.p:33-38, Jan/Feb 1998.
- [13] Tristan Richardson (2009, Nov 24) "*The RFB Protocol*" [online]. Available: <http://www.realvnc.com/docs/rfbproto.pdf>
- [14] Sakai Foundation (2009, Dec 22) "*Product overview*" [online] Available: <http://sakaiproject.org/> (Accessed Nov. 2009)
- [15] Sun Microsystems, Inc. (2009, Dec 22) "*The Java™ Tutorials*" [online]. Available: <http://java.sun.com/docs/books/tutorial/networking/overview/index.html>
- [16] M. Wannous, D. Ludena, H. Nakano, "A Virtual-Network Educational Management Interface for Motivating Learners in Web-based ICT Laboratories," in *Int. Conf. for Media in Education 2009*, Seoul, South Korea, , Aug 2009, pp. 286-29.
- [17] Sakai Foundation (2009, June 1) "*Sakai*" [online]. Available: <http://sakaiproject.org/download-install>
- [18] The Apache Software Foundation (2009, Nov 1) "*Apache Tomcat*" [online]. Available: <http://archive.apache.org/dist/tomcat/tomcat-5/v5.5.25/bin/>
- [19] The Eclipse Foundation (2009, Nov 1) "*eclipse*" [online]. Available: <http://www.eclipse.org/org/>
- [20] Sun Microsystems, Inc. (2009, Nov 1) "*NetBeans*" [online]. Available: <http://www.netbeans.org/>
- [21] Sakai Foundation (2009, June 1) "*AppBuilder*" [online]. Available: <http://source.sakaiproject.org/appbuilder/updates>

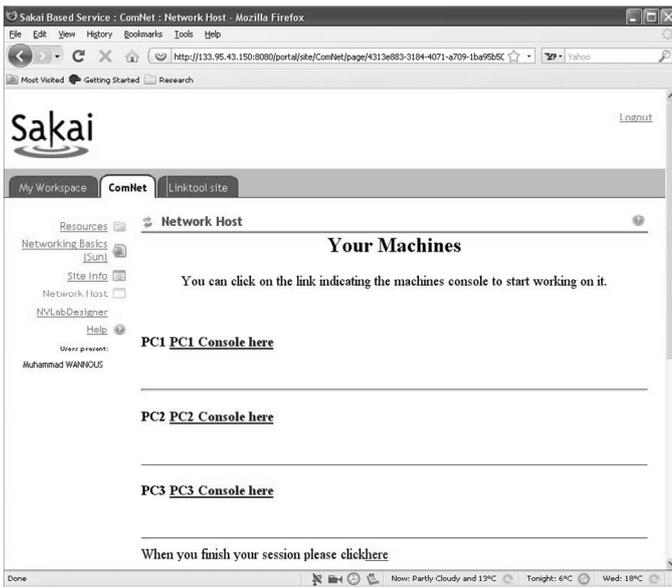


Figure 8. netManager's web-interface

V. CONCLUSIONS

In this paper, we presented a work-in-progress technical report that shows advancements in integrating a laboratory tool into the course context in Sakai LMS. The suggested system is based on NVLab which is a stand-alone web-based laboratory for *Computer Networks* online-courses. In the new system, part of the laboratory has been integrated as a new tool into Sakai, and the other part has been kept as a separate module since it requires managing a server-pool from which each learner is assigned one server to work on. New features that enable tracing the learners' progress and facilitate access to the workspace have been included.

The new system has been tested by introducing a simple exercise on basic IP addressing and connectivity check, and the learner was able to complete the exercise successfully.

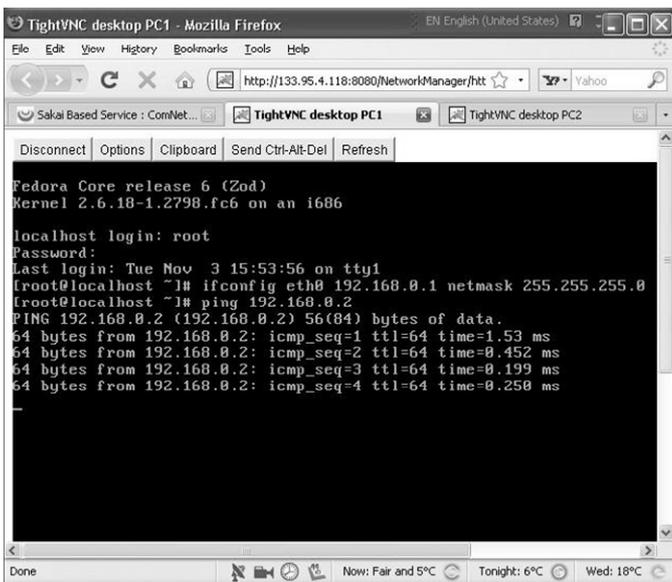


Figure 9. The VM terminal as shown in the learner's web browser

