**Proceedings of the ASME 2009 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30 - September 2, 2009, San Diego, California, USA**

DETC2009-87112

AN INFORMATION TECHNOLOGY INFRASTRUCTURE FOR INTERNET-ENABLED REMOTE AND PORTABLE LABORATORIES

J. Lane Thames
Distributed Networked Applications Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology, Savannah
210 Technology Circle
Savannah, Georgia 31407, USA
lane.thames@gatech.edu

Andrew Hyder
Systems Realization Laboratory
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology, Savannah
210 Technology Circle
Savannah, Georgia 31407, USA
ACHyder@gatech.edu

Robert Wellman
Information, Laboratory, and Distance Learning
Technology Support
Georgia Institute of Technology, Savannah
210 Technology Circle
Savannah, Georgia 31407, USA
bobby.wellman@gtsav.gatech.edu

Dirk Schaefer
Systems Realization Laboratory
G. W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology, Savannah
210 Technology Circle
Savannah, Georgia 31407, USA
dirk.schaefer@me.gatech.edu

ABSTRACT

With the proliferation of distributed and distance learning in higher education, there is a growing need for remote and portable laboratory design and deployment for the engineering, science, and technology education sectors. Amongst the current threads of research in this area, very little work has focused on solutions to the challenges, which are imposed by modern day information technology infrastructure, enterprise networks, and enterprise network security change management processes, that will be faced by ***large scale*** deployments of remote and portable labs. In this paper, the authors will discuss some of these challenges and will propose the use of the command and control communications architecture coupled with Web 2.0 as a solution to many of the deployment challenges.

1. INTRODUCTION

The last decade has given rise to many advances in Internet-enabled technology, and as a result of the "ubiquitous" Internet, the education sector has seen significant growth in the area of distributed education and distance learning. This growth is a result of institutions offering online coursework and providing remote campuses. For example, Georgia Tech provides several avenues for distance learning. First, it provides online coursework opportunities via its distance learning and professional education department, which offers online coursework for undergraduate, graduate, and professional education. Second, Georgia Tech provides distance learning via its flag-ship remote campus Georgia Tech Savannah (GTS), which is located in Georgia's city of Savannah. GTS is a highly information technology (IT) enabled campus that provides distance learning to students who are enrolled in the Georgia Tech Regional Engineering Program (GTREP), which is a collaboration between Georgia Tech, Georgia Southern University, Armstrong Atlantic State University, and Savannah State University. GTS students take their first two years of coursework with one of the collaborating universities where the students take their sciences, humanities, social sciences, mathematics, and pre-requisite engineering coursework. After the first two years of coursework have been completed, the students transfer into Georgia Tech to pursue their core engineering coursework at the Georgia Tech Savannah campus. GTS offers coursework via local Georgia Tech faculty and via Atlanta based faculty. IT enabled technology such as advanced video teleconferencing and tele-collaboration equipment is used to transport real-time classes both from Atlanta to Savannah and from Savannah to Atlanta. Third, Georgia Tech provides distance learning at the international level via Georgia Tech-Europe and Georgia Tech-Asia. Georgia Tech-Europe incorporates a remote campus in Metz, France and a research institute in Athlone, Ireland. Georgia Tech-Asia provides distance learning via collaboration with the National University of Singapore and with the Shanghai Jiao Tong University.

Even though many universities are offering online coursework and distance learning, with approximately eighty-five percent of US universities considering distance learning and online coursework as vital components of their educational strategies [1], there is a significant challenge faced when offering distance learning coursework in the engineering, science, and technology sector. This challenge is the ability to offer coursework that requires laboratory instruction. As a result of this challenge, much research in the area of remote and portable laboratories (RPL) is being conducted. We define RPL as the following: a remote and portable lab, in general, includes an apparatus that is operated through a computer, possibly interfaced with audio and video equipment for real-time feedback to the remote user, and is controllable from a remote location via computer networks. Further, a remote and portable lab incorporates laboratory units (the apparatus) that can, in certain cases, be moved, seamlessly, between different locations and computer networks. The apparatus in RPL is any networked device that implements a remote lab environment. For the rest of this paper, we will refer to the apparatus as a remote laboratory unit (RLU).

One can enumerate various reasons for deploying distance learning curricula and associated remote laboratories, such as:

- Students can experiment with different configuration settings, get results very quickly, and hence are encouraged to do more "what if" exploring than they would do in a traditional, physical laboratory.
- Students have the flexibility to log in, conduct an experiment and complete associated assignments from any place in the world and at any time they choose.
- Remote laboratories provide broader access to expensive and/or specialized equipment and thus foster the concept of "magnet schools".
- Remote laboratories help prepare students for the workplace of tomorrow, in which remote work and remote collaboration (Mass Collaboration [2]) will be the norm.

The two main reasons for developing distance learning curricula, however, are (1) that universities can reach out to students who, without distance learning technology, would otherwise not be able to obtain higher education and advanced degrees; and (2) significant revenue generation.

Currently, three primary threads of research and discussion are being pursued in the area of RPL:

- Design of RPL
- Pedagogical aspects of RPL
- IT infrastructure and architecture for RPL

The design of RPL is focused on the mapping of traditional laboratories into Internet-enabled environments. For example, designing a robot control based laboratory where the user (student) performs the lab exercise (i.e., controlling the robot with specialized software) over an Internet connection. The pedagogical aspect of RPL is the study of how well students learn and how instruction should adapt or evolve within RPL environments. In other words, it is the study of learning objectives and outcomes within RPL environments. Finally, the work with IT infrastructure and architecture for RPL is focused on the deployment issues of RPL. The primary focus of this paper is on the requirements and associated challenges of the IT infrastructure that is needed for large scale Internet-enabled remote and portable laboratory deployment. However, we also give an overview on using remote laboratories in education. The purpose of describing both aspects is a single paper is to give the reader a complete view of the ideas, concepts, and needs. We are interested in 1) the scholarship of education and 2) the scholarship of integration. We believe that in the years to come, researchers in education need to look at problems such as the ones described in this paper from both the educational and technical perspective in order to make significant progress.

2. RELATED WORKS

Since the mid 1990s, numerous works have been introduced in the design of remote laboratories. Table 1 provides a small sample of these previous works for the following disciplines: electrical and computer engineering (ECE), mechanical engineering (ME), civil engineering (CE), and physics.

Table 1. Previous works in the design of RPL.

Discipline	Description	References
ECE	semiconductor characterization	[3]
	electrical element characterization	[4]
	logic design	[4]
	control theory	[5]
	electric motors	[6]
	image acquisition and processing	[7]
	robotics	[8]
	telecommunications	[9]
	signals, systems, and controls	[10]
	Field Programmable Gate Arrays	[11,12]
	Programmable Logic Control	[13]
	optical circuits	[14]

	antennas	[15]
ME	motion and friction	[4]
	tensile testing	[10]
	speed of sound measurement	[10]
CE	tensile testing	[10]
Physics	speed of light measurement	[16]
	electron diffraction	[17]
	photoelectric effect	[17]
	voltage-current characteristics	[17]
	diffraction and interference	[17]
	photonics	[18]
	fluid mechanics	[19]

For the topic of pedagogical aspects of RPL, Nickerson et al. [20] provide a model that can be used to test the relative effectiveness of student learning in remote engineering laboratories. Their results show that remote laboratories achieve learning comparable to on-site, traditional laboratories. In [21], Esche et al. discuss their pedagogical approach for integrating remote experimentation into undergraduate laboratory classes and detail the observed benefits and shortcomings of their integration process. Lang et al. [22] describe pedagogical evaluations for remote laboratories in the eMerge project. Lang et al. performed their evaluation with a group of 84 students, 42 of which performed experiments using the remote lab and where the other 42 performed experiments physically within the lab. Their results show that: 1) Internet-based experimentation does not damage the positive learning effect of the remote students and 2) Students had no difficulties performing the remote laboratory experiments.

A few recent works have focused on software architectures for remote laboratories. In [12], the authors discuss their Remote Monitored and Controlled Laboratory (RMCLab). RMCLab is a client-server based architecture and includes a user interface which provides capabilities such as lab administration, hardware management, and instrument operation. The primary components of their architecture include a client, an instructor-client, an application server, a resource server, and the laboratory infrastructure. Their system has been in operation since March 2004 and provides educational services to the department of electrical and computer engineering of the University of Patras, Greece. Esche et al. [23] developed an architecture for multi-user remote laboratories with a goal of improving engineering and science education by developing a more diverse and competitive workforce. They describe various characteristics needed by remote laboratories, and these include modularity and expandability, scalability, usage of existing communication standards, and computer platform independence. Further, they posit that remote laboratories should possess the following attributes in order to be accepted by the academic community: the labs should correlate with curricular needs, should be compliant with ABET requirements, should be pedagogically sound, should be affordable, should be easy to use, and should be reliable. Esche et al.'s system is designed with the client-server network model. It provides laboratory connectivity by way of a Linux-based Web server. The server provides a graphical user interface (GUI) for the student, and it hosts a process queue along with associated input and output files for the laboratory experiments. The server is networked to individual data acquisition computers that are connected to laboratory equipment and process experiments with LabVIEW VI software. The authors claim that student performance while conducting remote laboratory experimentation is comparable to previous non-remote experimentation. Datta and Sass [11] introduced RBoot, which is a software infrastructure for remote field programmable gate array (FPGA) laboratories. RBoot is based on the client-server network model. With RBoot, the student accesses the server using a secure shell (ssh) client. Then, the student uses the authors' fpga-session software to establish a lab session with one of the available ML-310 FPGA boards. Troger et al. [24] introduce the concept of "experiment as a service" as an extension of service oriented architectures (SOA) for virtual remote laboratories. Their architecture is built with the client-server network model and provides experimentation as services, where the experimentation service is mapped into SOA web based standards. Once a user schedules a required service, the system sends job service information to an appropriate execution host using client-server communication techniques.

Other works have established criteria for the design of remote laboratories. Schaefer et al. provides the following desirable characteristics of remote labs [10]:

- The user interface should be easy to understand and easy to use.
- The user (client side) should not need any special hardware or software.
- Experiments should require minimal or no interaction from on-site lab personnel once the experiment is set up.
- The system in the experiment should neither be too fast nor too slow.
- The experiment should be interesting to observe.
- The lab experiment hardware should be controllable via a computer or computer controlled equipment.
- Experiment measurements should be able to be taken via electric sensors such as ammeters, voltmeters,

tachometers, electrical pressure sensors, electrical temperature sensors, etc.

- The experimental apparatus must lend itself to familiarization via videos or web tutorials.
- Access to the remote experiments is secure and protected from unauthorized users.

Another criteria, as described by Salzmann and Gillet, is that "*a remote experimentation solution is to make the student interaction with the distant system as close as possible to the actual work on the real equipment*" [25]. Salzmann and Gillet also enumerate various challenges that will be faced when moving from small scale remote laboratory settings, which is a common characteristic of most works in remote labs that have been previously discussed in the literature, to large scale professional quality remote laboratory settings. They enumerate the following challenges [25]:

- Physical equipment challenges
- Software challenges
- Maintenance challenges
- Deployment challenges
- Educational challenges
- Sustainability challenges

In this paper, we will discuss various requirements and challenges for RPL environments. But, before completely enumerating these, we claim that one primary criterion is that RPL systems ***must*** be scalable. In a recent workshop, Lindsay et al. [26] state "*Remote laboratories are an increasingly popular innovation in engineering education, but their development has for the large part been an ad hoc process—they are developed in isolation to address the needs of a particular laboratory class…*". The lack of a standardized infrastructure for remote laboratories is one reason for the "ad hoc process" of creating remote labs. Research in remote laboratories is still in its infancy. Most of the research in this area has been on the design of remote laboratories (mapping traditional labs into remote labs) and on the pedagogical aspects of remote labs. Work in infrastructure is starting to surface. However, most of the infrastructures that have been discussed are built on the traditional client-server model for Internet applications. Client-server technology is well founded. But, the client-server model alone is not sufficient for large scale remote lab deployment. Lindsay et al. claim that the current deployment model "*simply doesn't scale*" [26], and we concur. The client-server model can be a necessary condition for large scale remote lab deployment, but it is definitely not a sufficient condition.

3. EXAMPLES OF REMOTE AND PORTABLE LABORATORIES

In this section, we provide a few examples of RPL instantiations to motivate the remaining discussions of this paper.

3.1 Embedded Web Server Interface for Microcontroller Design

This remote laboratory consists of a Microchip PIC microcontroller system where students develop source code based on the particular lab assignment. For example, the students might be tasked with developing code that collects data from onboard temperature and pressure sensors. The students develop their code on their personal computers via Microchip's MPLAB integrated development environment. Once their code is ready for real-time testing and analysis, the students access the target microcontroller system via an embedded web server application, which runs on another lab-manager controlled microcontroller system and is electrically interfaced with the target system. The student accesses the system with a web browser and uploads his compiled source code to the target system. Once the target system has been configured, the student can access the target's output via the web interface. Figure 1 shows the target system and the web interface system.

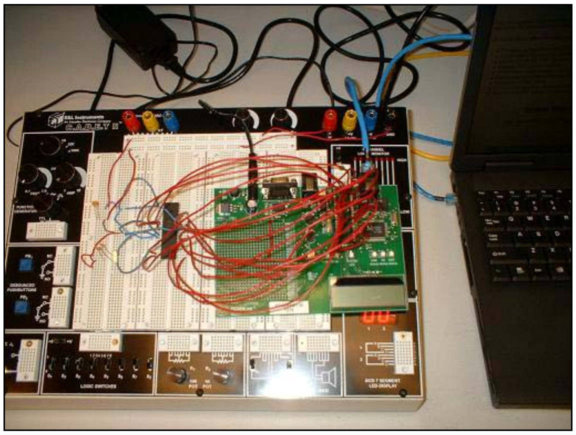
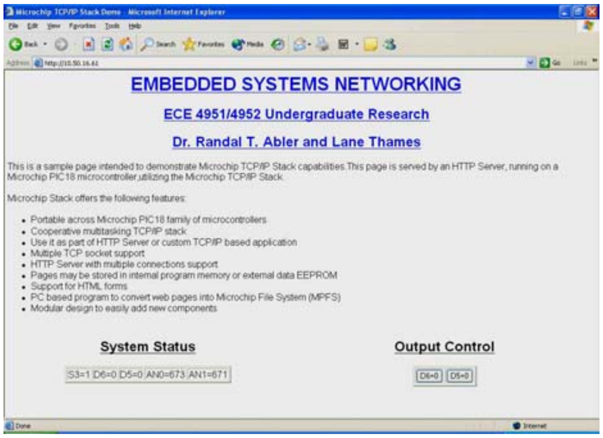


Figure 1. Microcontroller target system and embedded web server interface.


Microchip TCP/IP Stack Demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://10.50.16.61
Go Links
Done
Internet

EMBEDDED SYSTEMS NETWORKING

ECE 4951/4952 Undergraduate Research

Dr. Randal T. Abler and Lane Thames

This is a sample page intended to demonstrate Microchip TCP/IP Stack capabilities This page is served by an HTTP Server, running on a Microchip PIC18 microcontroller,utilizing the Microchip TCP/IP Stack

Microchip Stack offers the following features:

- Portable across Microchip PIC18 family of microcontrollers
- Cooperative multitasking TCP/IP stack
- Use it as part of HTTP Server or custom TCP/IP based application
- Multiple TCP socket support
- HTTP Server with multiple connections support
- Pages may be stored in internal program memory or external data EEPROM
- Support for HTML forms
- PC based program to convert web pages into Microchip File System (MPFS)
- Modular design to easily add new components

System Status

S3=1 D6=0 D5=0 AN0=673 AN1=671

Output Control

D6=0 D5=0

Figure 2. Target system output displayed on web interface.

In Figure 2, the web interface providing system feedback after the student uploads her code to the target system is shown. One interesting feature of a system such as this is that it is economically friendly. The web interface system can be built for less than USD $50. Further, it is easily reprogrammed, which enables the system to be re-purposed for various laboratory environments.

3.2 Remote Robotic Control—Collision Avoidance and Environmental Monitoring

This remote laboratory consists of a mobile robot that roams an area and performs environmental monitoring. The goal is to model wireless sensor network applications. In this scenario, the robot must perform its primary task while implementing self-preservation. Self-preservation includes tasks such as collision avoidance and power management. For this lab, the robot is connected to the Internet via wireless Ethernet (WiFi) hardware. The student develops control software in MATLAB or other development environments and communicates with the robot over web-based protocols such as Hyper Text Transport Protocol (HTTP). A wrapper program sends commands from the students control software to the robot and receives sensor feedback from the robot. Figure 3 shows the robot and its roaming (sensory) environment.

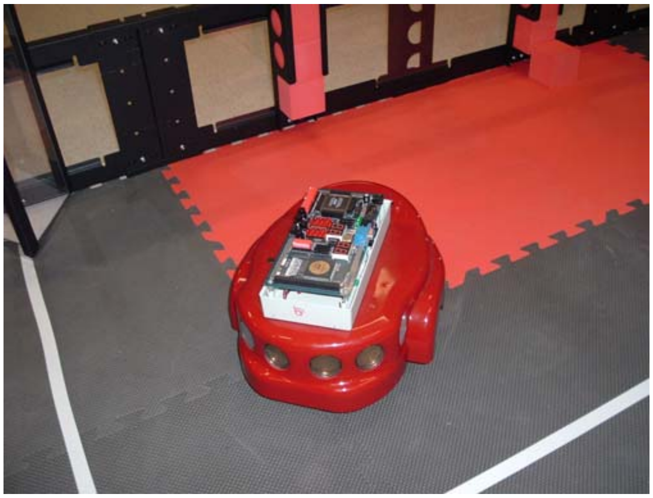
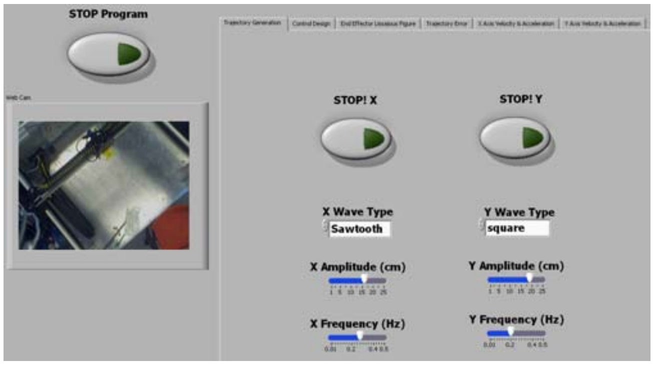


Figure 3. Wireless (WiFi) enabled robot.

3.3 Control Theory Concepts with a Multi-axis Robot

With this remote laboratory experiment, the remote student will gain an understanding of control theory in a real-world setting. A proportional-integral-derivative (PID) controller is a generic controller widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a reference input by calculating and then outputting a corrective action that can adjust the process accordingly. The PID controller involves three separate parameters: the proportional, the integral, and the derivative values. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via an actuator. By "tuning" the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. To perform this lab exercise, the student will login to the lab computer via remote desktop software. This computer is electrically interfaced to a bi-axial robot that can move in the X-Y plane. LabVIEW software was developed that provides a user interface and implements a PID controller to control the movement of the robot in the plane. The user interface is shown in Figure 4.



X Wave Type: Sawtooth

Y Wave Type: square

X Amplitude (cm)

Y Amplitude (cm)

X Frequency (Hz)

Y Frequency (Hz)

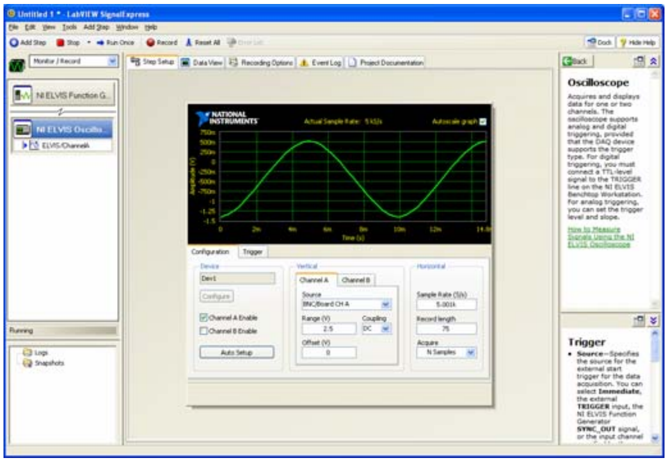
Figure 4. User interface for bi-axial robot control.

The user interface provides real-time audio and video feedback of the robotic system, as shown on the left-middle side of the interface. The student configures the desired robotic motion in the plane; the motion is based on sawtooth, sinusoidal, or square waveforms. Further, the student can configure the amplitude and frequency of the waveforms. Once the student applies the motion parameters and the robot goes into motion, the student can choose various options such as observing, in real-time, plots of the trajectory, both theoretical and real, and the robots trajectory error. Further, the student can tune the PID parameters and observe the effects of the tuning, in terms of the trajectory and trajectory error.

3.4 Remote Signal Measurement with Waveform Generators and Oscilloscopes

In this remote laboratory experiment, a personal computer (PC) in the physical lab is connected to a National Instrument's (NI) ELVIS development board. The NI ELVIS device along with LabVIEW software provides a comprehensive laboratory solution with a relatively small footprint. The ELVIS workstation is bench top device that consists of twelve integrated instruments, a prototyping board, a data acquisition (DAQ) card, and a PC. The instruments found on the ELVIS include: arbitrary waveform generator, bode analyzer, digital bus reader, digital bus writer, digital multimeter, dynamic signal analyzer, function generator, impedance analyzer, oscilloscope, two- and three-wire current-voltage analyzer, and variable power supplies.

All of the instruments are controlled via soft front panel (SFP) software. The SFP provides a virtual representation of the physical instrument, displayed on the PC monitor, for the user to operate manually. As the ELVIS is uniquely integrated with LabVIEW, the ELVIS virtual instruments (VI) can be added to and modified in any LabVIEW project. Leveraging the flexibility and functionality of National Instruments LabVIEW, the NI ELVIS is a very capable platform for laboratory courses which involve circuit design and analysis. Figure 5 shows the software's output of a measured sinusoidal signal as seen by a remote student.



Actual Sample Rate: 5 kS/s

Autoscale graph ☑

Dev1

☑ Channel A Enable

☐ Channel B Enable

Source: BNC/Board CH A

Range (V): 2.5

Coupling: DC

Offset (V): 0

Sample Rate (S/s): 5.001k

Record length: 75

Acquire: N Samples

Oscilloscope

Acquires and displays data for one or two channels. The oscilloscope supports analog and digital triggering, provided that the DAQ device supports the trigger type. For digital triggering, you must connect a TTL-level signal to the TRIGGER line on the NI ELVIS Benchtop Workstation. For analog triggering, you can set the trigger level and slope.

How to Measure Signals Using the NI ELVIS Oscilloscope

Trigger

- **Source**—Specifies the source for the external start trigger for the data acquisition. You can select **Immediate**, the external **TRIGGER** input, the NI ELVIS Function Generator **SYNC_OUT** signal, or the input channel

Figure 5. Oscilloscope measurements with LabVIEW.

To perform the laboratory exercise, the student accesses the lab PC via remote desktop application software. Once the student logs into the machine, she can use LabVIEW software to generate waveforms via the NI ELVIS development board and to measure the signals via real-time oscilloscope outputs. The only prerequisite for this type of remote lab experiment is for the laboratory personnel to connect cables from the signal generator channels to the measurement channels (channels used by the LabVIEW software to provide measurements with the oscilloscope, ammeter, voltmeter, etc.) before the laboratory session starts.

4. REQUIREMENTS AND CHALLENGES FOR LARGE SCALE REMOTE AND PORTABLE LABORATORY DEPLOYMENT

Various requirements and criteria have been previously proposed, some of which were discussed in section 2. Based on these previous observations, we define the following "minimal" set of requirements for *large scale* remote laboratory deployment. These requirements should coincide with the various requirements that have been previously suggested in the literature. Large scale remote labs should:

- Provide simple and easy to understand user interfaces
- Not require sophisticated communications software—it should build on standard Internet protocols
- Retain high levels of security for all entities involved in the remote lab architecture, which includes remote and local students, lab personnel, laboratory equipment, RLUs, etc.
- Retain abilities for high degrees of scalability, i.e. remote labs should be able to scale to large numbers of RLUs and associated devices and equipment
- Provide all necessary remote laboratory management functionality
- Provide sustainability

In order to meet these requirements, we foresee the following minimal set of information technology (IT) infrastructure challenges that must be addressed:

- Network security management
- RLU management
- RLU synchronization management
- RLU scheduling management
- RLU portability management

Network security management: Network security management (NSM) must provide two primary functions: 1) To allow seamless remote access from users on the Internet to RLUs that are located within highly complex and secure enterprise networking systems (such as a university network), and 2) To retain high-levels of security for the RPL environments. Striking a balance between these two functions is non-trivial in modern day enterprise networks.

RLU management: Within large scale RPL environments, many RLU devices will need to be managed. This management includes RLU inventory, RLU connectivity information, RLU hardware maintenance, and RLU software maintenance.

RLU synchronization management: With many RLU devices being operated by many different users, the issue of synchronization must be addressed. Certain environments will allow for multiple users to simultaneously access the RLU. But, certain environments will only allow for single access instances. A system must be in place to address these issues. This management function is tightly coupled with the scheduling management function.

RLU scheduling management: In order to provide synchronization between multiple users and multiple RLUs, a scheduling subsystem will need to be deployed. This function also assists the RLU management interface—when RLUs need to go offline due to maintenance, the scheduling software needs to be informed so it will not allocate the associated resources to students during the downtime.

RLU portability management: A system for seamless portability will need to be in place. If an RLU is moved from one location to another, connectivity information for the RLU within the new network location must be distributed to all entities involved in the RPL environment.

We believe that a centralized, Web 2.0 based management interface can be used to solve all of the above challenges except for network security management. Wikipedia defines

Web 2.0 as the following [29]: "*Web 2.0 refers to a perceived second generation of web development and design, that facilitates communication, secure information sharing, interoperability, and collaboration on the World Wide Web. Web 2.0 concepts have led to the development and evolution of web-based communities, hosted services, and applications; such as social-networking sites, video-sharing sites, wikis, blogs, and folksonomies*". Web 2.0 offers a very rich set of advanced communication and collaboration technologies. These technologies include seamless database integration, web-based remote desktop sharing applications, real-time chat and video teleconferencing, web logs (blogs), wikis, really simple syndication (RSS, which is effectively a web-based multi-cast information distribution technology), dynamic content generation, and much, much more. One of the defining features of Web 2.0 is its ability to perform "information mashup". Information mashup is the ability of web sites to dynamically generate web pages based on content that is relevant to the user. For example, when one logs into a Web 2.0 site, the page is generated dynamically based on information that is associated with the user's account. This provides significant functionality for one who is developing RPL environments and RPL user interfaces from a centralized location. However, Web 2.0 will not solve the network security management (NSM) challenge. Web 2.0 is based on client-server technology. But, pure client-server technology will not be scalable for RPL environments. To understand why this is true, one needs to understand the design of modern day enterprise networks, enterprise network security (such as implemented within most university networks), and the enterprise network security change management process. This is an area that has not been discussed in the literature when describing RPL deployments. But, as we will show, the enterprise network security change management process will be a major bottleneck for the deployment of large scale RPL environments—i.e., it will prevent scalability. One of the primary purposes of this paper is to describe the NSM challenge for RPL deployment. To do so, we now provide an overview of firewalls, enterprise network security, and enterprise network security change management processes. We will continue the discussion of Web 2.0 for RPL later in the paper.

5. CHALLENGES IN NETWORK SECURITY MANAGEMENT

To understand the underlying issues with enterprise network security and network security change management, one needs to have an understanding of firewall technology.

5.1 Firewalls and Enterprise Network Security

The Internet was originally designed as an "open" communications architecture. However, as the Internet grew and became commercialized, network security threats caused network administrators across the Internet to "close" their networks with firewalls. Firewalls are network devices that filter incoming and outgoing network traffic (computer communication connections) based on a network security policy. Firewalls are used to partition enterprise networks into secure regions. All traffic flowing between the regions will traverse the firewall. For each communication connection traversing the firewall, the firewall will use data contained within the communication unit (a network data packet) and compare the data against the firewall's security policy. If the policy is defined to "allow" the particular type of traffic, then the firewall will provision the traffic and allow it to continue its flow. Otherwise, the firewall will filter the traffic and not allow (deny) it to continue flowing—i.e., it prevents the traversal of the communication connection through the firewall.

Figure 6 provides a visual representation of a firewall. When designing firewall systems, the designer must understand the topology of the network and the firewall's orientation with respect to the topology. The idea of inside and outside interfaces is tightly coupled with the topological perspective. From Figure 6, the firewall provides a secure network on the inside interface. The outside interface connects to an external network—external with respect to the firewall and the network topology. Communication traffic flows from the inside network through the firewall to the outside network and vice versa. Common enterprise networks and their firewall systems normally allow most traffic to flow freely from the inside to the outside. However, the firewall acts as a modulated switch for traffic that is **initiated** from the outside to nodes (computers) on the inside. The "modulation" is provided by a security policy. An example security policy could be: "Allow all Internet connectivity to the web server located within the inside network." This policy is implemented via a firewall filtering rule, which is a translation of the high-level security policy into a filtering language understood by the firewall. The "switch" closes for traffic that matches the firewall's security policy and opens for traffic that does not match the policy.

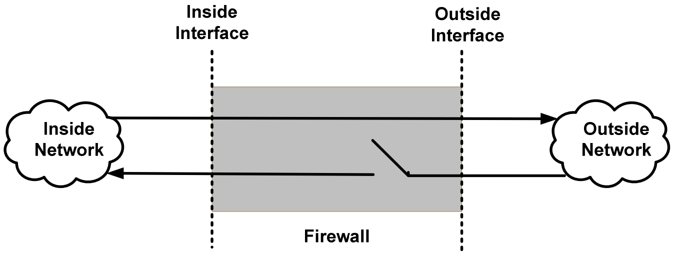


Figure 6. Firewall concept.

There are two types of firewall security policies: dynamic and static. Static policies are the formal policies defined by the enterprise. Dynamic policies are policies created by the firewall when an inside node **initiates** a communications connection to an outside node. Notice that when an inside node communicates with an outside node, the outside node will reply to the inside node. Hence, there will be traffic flowing from the outside to the inside. In order to allow reply connections to the internally initiated connections, the firewall employs stateful packet filtering technology. With stateful filtering, the firewall stores connection tracking state information for every internally initiated connection. Once the external node replies to the internal node, the firewall will inspect the connection tracking state table. If the reply connection matches information stored in the state table, then the connection is allowed to pass through the firewall.

5.2 The Enterprise Network Security Change Management Process

We now provide an example of the enterprise network security change management process in order to show the complexities that will be involved with large scale RPL deployments. Figure 7 illustrates an enterprise network topology. Note that the network is partitioned into various network regions by firewalls. In this case, the network regions correspond to different enterprise units. For example, the top level region named "GT" is separated by the second level regions such as CoE (college of engineering), CS (computer science department) and CoM (college of management). CoE is further divided into the following units: ME (mechanical engineering), ECE (electrical and computer engineering), and CEE (civil and environmental engineering). Finally, CEE has a separate lab region.

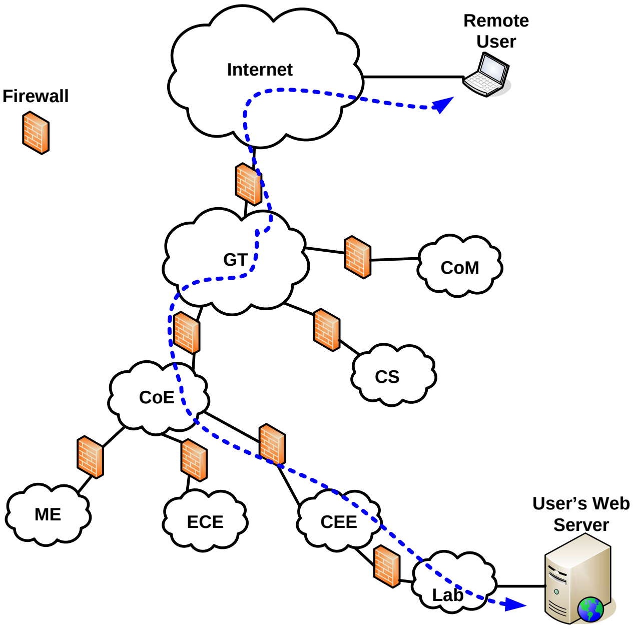


Figure 7. Example of an enterprise network topology.

Consider the case where a user in CEE's lab region wants to add a web server to the network and needs the "world" (i.e. anyone who can connect to the Internet) to be able to retrieve web pages served by the web server. This is represented by the dotted line in Figure 7. All of the firewalls on this network path must be configured with a security policy that allows network traffic that is destined to the user's web server. For this to happen, a collaborative effort between the user, system administrators, network administrators, and security administrators (and possibly others) will be required in order to satisfy this requirement. The following is a common set of steps required for this change management process:

- The user establishes a networking and/or security need
- The user communicates this need to a system administrator
- The system administrator works with user and then establishes the need with the security administrator
- The security administrator works with the system administrator and then establishes the need with the network administrator
- The network administrator implements the final process to solve the need

First, the user establishes a need, which in this case, is to make his/her web server available to the world. Second, the user communicates this need to a system administrator. Third, the system administrator installs the required software, configures the system, and applies any necessary security updates to the system. Fourth, the system administrator communicates the need to the security administrator. Fifth, the security administrator will perform vulnerability and risk assessment processes against the system (the web server) to ensure that it meets the enterprise's required security standards. This step is normally an iterative process between the security administrator and system administrator—i.e. the security administrator will communicate any security issues to the system administrator. Then, the system administrator will apply appropriate software/hardware updates to the system according to the security administrator's advice. This part of the process stops once the system meets the security standards and receives a "clean bill of health". Sixth, the security administrator will coordinate with network administrators so that appropriate security policies are updated on the necessary firewalls and other network security devices along the network path (the path illustrated by the dotted line in Figure 7) to the web server. Once the security policies have been updated, the change management process is complete and the system is available for users on the Internet.

Considering that RPL environments will contain many devices (RLUs) that require remote accessibility, one can see that enterprise network security change management will require significant amounts of resources for large scale deployment. For example, anytime a new RLU is added to or taken from the remote laboratory or moved from one network location to another, the change management process will need to be invoked. This is further compounded if the RLU (or system of RLUs) is moved from one enterprise to another (for example, if the lab is portable and used at multiple universities). In this case, the change management process must be a collaborative effort across multiple enterprise systems. This leads to the following questions: “How can remote labs be implemented on a large scale without requiring the costly change management process to be invoked?” and “How can remote labs retain high-levels of security?” Another question revolves around the issue of RLU connectivity. In Internet environments, computer to computer communication is done with Internet Protocol (IP) addresses. IP addresses are very dynamic. The addresses change if the node moves between different networks. Further, IP addresses change due to dynamic host configuration protocol (DHCP) and network address translation (NAT). So, another question is “How do we manage dynamic IP addresses in RPL environments with large numbers of RLUs and how do we distribute the connectivity information to the remote users in a seamless manner?” We propose the ***Command & Control*** (C&C) communications architecture as a solution to the problems implied by these questions.

6. THE COMMAND AND CONTROL COMMUNICATIONS ARCHITECTURE

With the C&C architecture, a centralized server, known as the command and control center (CCC), is used as a multifunctional unit that acts as an interface to all entities involved in the RPL environment, and these entities include, but are not limited to, remote students, instructors, teaching assistants, laboratory personnel, and the physical RLU. Figure 8 provides an illustration of the C&C architecture.

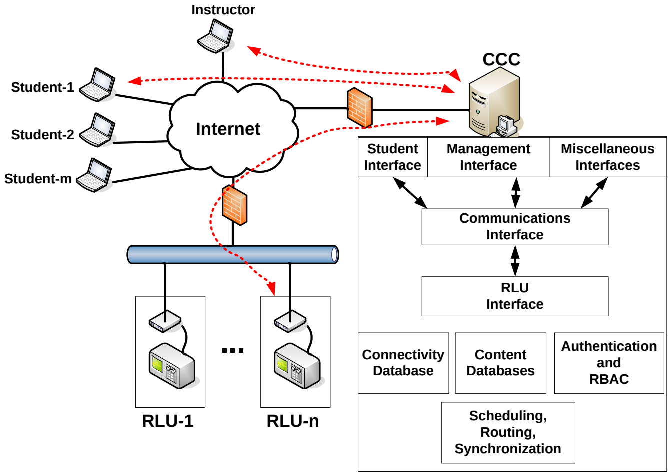


Figure 8. Command and control communications architecture.

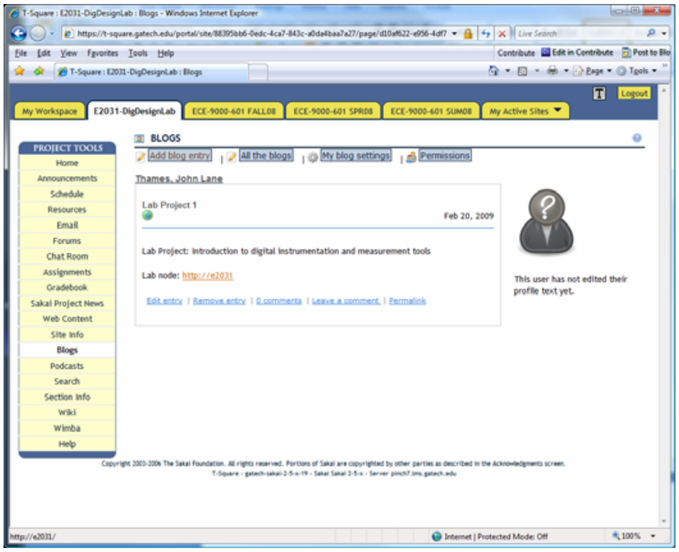
The features provided by the CCC include the following. The user interfaces are developed with Web 2.0 technology. The laboratory instructor can deploy RPL web sites seamlessly using the content databases and dynamic content generation features that are enabled with various Web 2.0 technologies. The CCC will host various functions that control RLU connectivity information (via the connectivity database); routing, scheduling, and synchronization; security features such as authentication and role-based access control (RBAC); and, most importantly, the communications and RLU interfaces.

In the C&C architecture, the CCC will go through the enterprise network security change management process and will be available (accessible) to any remote user connected to the Internet. The CCC possesses functionality that can solve the various management issues as previously enumerated. Recall that most common enterprise network systems allow connections that are initiated from the inside interface of the firewall to the outside interface. This is a key feature that makes the C&C architecture a feasible solution to the NSM challenge. The idea is as follows. Each RLU uses special command-control communications software. The RLU polls the CCC periodically—i.e. it initiates a connection ***to*** the CCC ***from*** its secure internal network. During the poll, the RLU “asks” the CCC for any commands that it needs to execute. Then, the CCC will reply appropriately (recall from the previous firewall discussion that the firewall will allow the reply communication to the RLU). This functionality is provided by the RLU and communications interface within the CCC.

The ultimate goal is to establish communications between the remote user (i.e., the student, instructor, lab personnel, etc.) and the RLU. To accomplish this, the remote user logs into the CCC by way of a Web 2.0-based user interface. Once the user logs in, he/she chooses the appropriate laboratory web site and associated RLU system. Then, the CCC acts as a virtual switch whereby all communications between the remote user and the RLU are managed via the CCC. This functionality is provided by the communications interface as shown in Figure 8. Since all communications are initiated ***from*** the RLU and remote users ***to*** the CCC, the architecture provides an ability to essentially bypass any firewall change management process for any RLU device.

We now provide a more detailed discussion of the C&C process. For this discussion, we assume that a remote student is taking a remote laboratory course in E2031, which is a digital design lab. First, the student logs into the web interface. Note that the login process enables security features such as authentication and role-based access control, which is implemented within the CCC that is hosting the web interface. Authentication is the process of proving one’s identity to a system. This is commonly implemented with username and password values. Role-based access control is the process of assigning fine-grained access control policies to an authenticated user based on the user’s role. For example, the user might have a student role or a teaching assistant role. A student role might be given only “read” access to an object whereas the assistant might be given “read and write” access to an object. Once the student logs into the site, he/she will choose the appropriate lab course from a set of class options. This is illustrated in Figure 9.

In this example, the student chooses the E2031-DigDesignLab tab. From this tab, the student has the option to choose an RLU node. This capability is provided by the "Lab node: http://e2031" hyperlink, which is shown in the middle of the page of Figure 9. Note that this is a pseudo-hyperlink. Only the underlying communications interface of the CCC knows how to link to the RLU. Once the student clicks the link, the CCC queries the routing/scheduling database, which stores node availability information. If a node is available, then the connectivity database will provide the connection information to the CCC's communication interface. Then, upon the RLU's next communication poll, the CCC will issue a "connection request" command whereby the RLU will establish a virtual connection with the remote student whereby all communication between the remote student and the RLU is virtually switched via the CCC's communication interface.



Thames, John Lane

Lab Project 1

Feb 20, 2009

Lab Project: Introduction to digital instrumentation and measurement tools

Lab node: http://e2031

This user has not edited their profile text yet.

Copyright 2003-2006 The Sakai Foundation. All rights reserved. Portions of Sakai are copyrighted by other parties as described in the Acknowledgments screen.
T-Square - gatech-sakai-2-5-x-19 - Sakai Sakai 2-5-x - Server pinch7.lms.gatech.edu

Figure 9. An example of the student interface.

The network timing diagram shown in Figure 10 will be used to provide a deeper explanation of the underlying C&C communications process. In Figure 10, the vertical axes represent time relative to each network node with time increasing in the downward direction. The angular lines connecting one node's time axis to another represents a communication flow. Note that the angular line starts from one node and terminates at the other node's axis at some delta-time later. This represents the time delay inherent in communication networks. For example, if the RLU sends a communication to the CCC at time t, then the CCC will receive the connection at some time t + Δt later. The angular lines are labeled with information representing the type of communication for that particular flow. For example, the top three lines from the RLU to the CCC represent the polling connections. In this example, the RLU is communicating with the CCC via the user datagram protocol (UDP). For each one of these three connections, the RLU is asking the CCC if it needs to execute any new commands. As an aside, recall that one of the scalability issues with RPL was concerned with dynamic IP addresses. So, how does the C&C architecture solve this? This is solved via a registration command. The CCC's IP address is assumed to be non-dynamic. In other words, the CCC does not relocate to new networks, does not use DHCP, and does not use dynamic NAT. The RLU's command-control software will monitor its own IP address. If the RLU's IP address changes, it will send a registration command to the CCC and inform it of the RLU's new IP address. The CCC will then update the connectivity database with the new information.

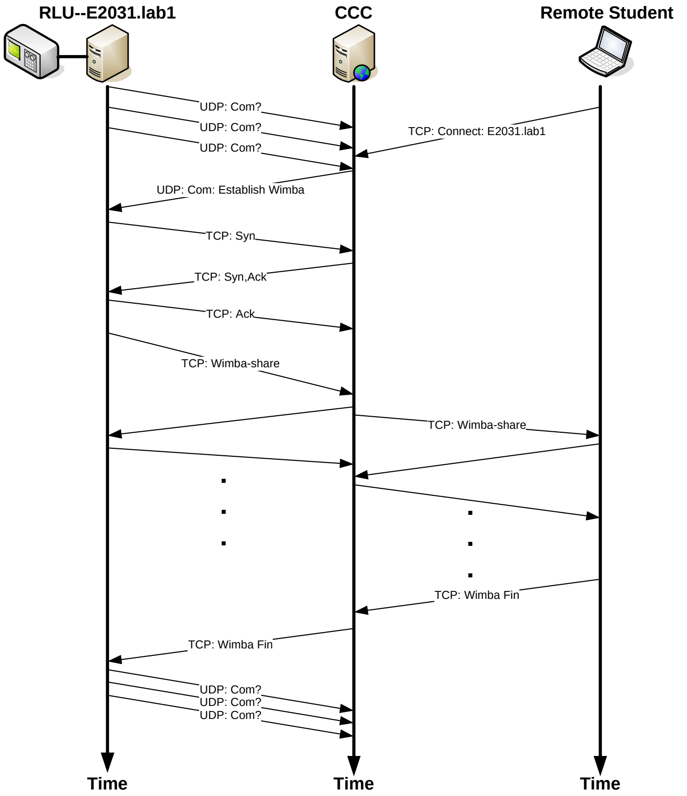
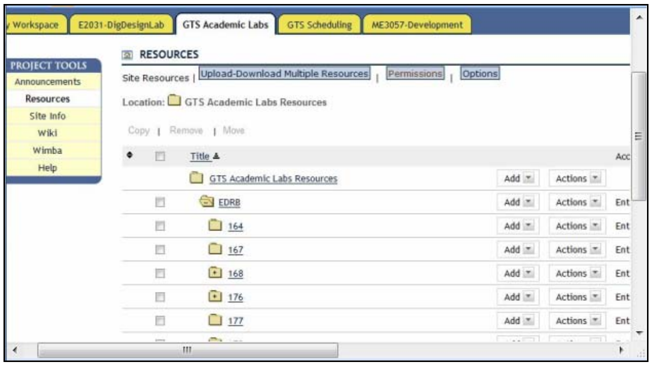


Figure 10. C&C communication network timing diagram.

We now continue with the original discussion. At some point in time between the RLU's second and third polling interval, the remote student clicks the pseudo-hyperlink in the lab's website (as shown in Figure 9). At this point, the CCC knows that the student needs to access the RLU. At the RLU's next poll, the CCC sends a reply command back to the RLU telling it to establish a transport control protocol (TCP) session with the CCC's "Wimba" [27] Web 2.0 interface. (Wimba is Web 2.0 collaboration software that has been deployed on this prototype system). Once the RLU receives the connection establishment command, it issues the TCP "3-way hand shake" to establish a logical TCP circuit with the CCC. This is shown with the TCP:SYN, TCP:SYN/ACK, and TCP:ACK connection streams between the RLU and the CCC. Once the 3-way handshake is complete, the RLU invokes the Wimba connection with the CCC. After the Wimba connection is established, the student will have a virtual connection with the RLU, and all communication will be handled via the CCC's communication interface. Once the student is finished with the lab exercise, he closes the Wimba session, which is shown in the timing diagram with the TCP:Wimba Fin connection. Then, the CCC sends a TCP:Wimba Fin command to the RLU and it tears down the TCP session. After the session is terminated, the RLU resumes its polling connections, as illustrated by the bottom three connection streams in the timing diagram.

7. PROTOTYPE DEMONSTRATION OF AN RPL

We now describe a prototype implementation that is developed with Web 2.0 and C&C technology. The discussion provides a view of the system from the user's perspective. The Web 2.0 interfaces are provided by the Sakai open source collaboration and courseware management framework [28]. This type of framework allows a lab instructor to deploy an RPL course web site in a matter of minutes. The instructor can add various Web 2.0 features (using Sakai's dynamic content generation tools) to the site such as chat, wikis, RSS feeds, video telecollaboration, blogs, etc. Figure 11 shows the instructor's web interface.



Location: GTS Academic Labs Resources

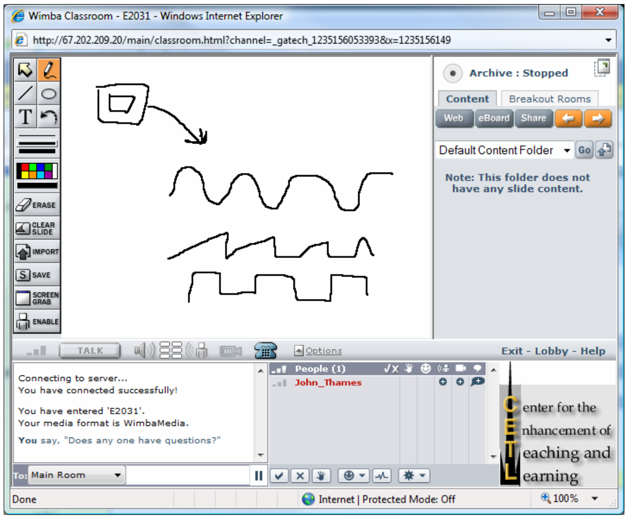
	Title ▲			Acc
	GTS Academic Labs Resources	Add	Actions	
□	EDRB	Add	Actions	Ent
□	164	Add	Actions	Ent
□	167	Add	Actions	Ent
□	168	Add	Actions	Ent
□	176	Add	Actions	Ent
□	177	Add	Actions	Ent

Figure 11. Instructor's web interface.

Notice from Figure 11 that the instructor has a lab interface where the various RPLs can be managed, an interface for scheduling, an interface for lab development, etc. The ability to dynamically generate and provide content is a Web 2.0 feature that can provide significant scalability for RPL deployments.

We now demonstrate a real-world remote laboratory instance. In this example, remote students are attending the E2031 Digital Design Laboratory and the students are preparing to execute their first laboratory assignment, which is an exercise in signal measurement. First, the students attend a synchronous pre-lab session. The students and instructor login to the web interface at a pre-determined time. Then, a web-based collaboration tool is used to discuss the pre-lab topics. The collaboration tool provides various features such as electronic whiteboard, video and audio interaction, and remote desktop sharing capabilities. These are shown in Figures 12, 13, and 14 respectively. In Figure 12, the instructor is explaining some different types of signals that will be measured in the lab exercise. In Figure 13, the instructor is communicating with real-time audio and video while discussing a sinusoidal signal. In Figure 14, the instructor is sharing his desktop where he has simulated a sinusoidal signal in MATLAB and is plotting the waveform.

Once the pre-lab session has ended, the student will return to the lab's web site during his/her scheduled time frame unless the lab is done on a first-come, first-serve basis. In either case, the CCC and web-based scheduling software will manage the user's ability to connect to an available RLU. The student will access the RLU using the interface shown in Figure 9.


Wimba Classroom - E2031 - Windows Internet Explorer
http://67.202.209.20/main/classroom.html?channel=_gatech_1235156053393&x=1235156149
ERASE
CLEAR SLIDE
IMPORT
SAVE
SCREEN GRAB
ENABLE
Archive : Stopped
Content
Breakout Rooms
Web
eBoard
Share
Default Content Folder
Go
TALK
Options
Exit - Lobby - Help
People (1)
John_Thames
Center for the Enhancement of Teaching and Learning
To: Main Room
Done
Internet | Protected Mode: Off
100%

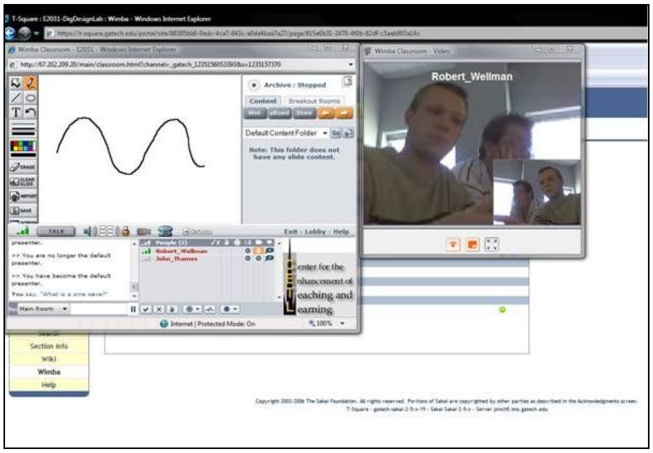
Note: This folder does not have any slide content.

Connecting to server...
You have connected successfully!

You have entered 'E2031'.
Your media format is WimbaMedia.

You say, "Does any one have questions?"

Figure 12. Electronic whiteboard.



Note: This folder does not have any slide content.

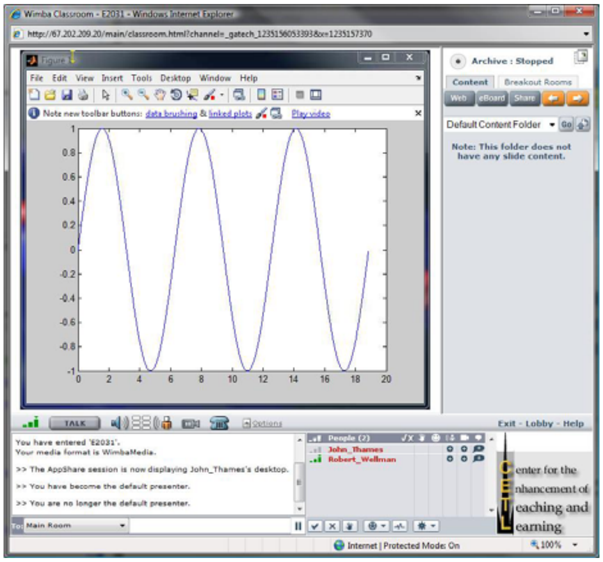
>> You are no longer the default presenter.

>> You have become the default presenter.

You say, "What is a sine wave?"

Figure 13. Real-time audio and video content with electronic whiteboard.

After the student clicks the pseudo-hyperlink and the CCC to RLU connection has been established, the student will access the RLU via the Wimba interface, which is deployed along with the Sakai framework in this example prototype. For this laboratory example, the student will measure waveform signals using LabVIEW software. The interface from the student's perspective is shown in Figure 15.


Wimba Classroom - E2031 - Windows Internet Explorer
http://67.202.209.20/main/classroom.html?channel=_gatech_1235156053393&x=1235157370
Figure
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
2
4
6
8
10
12
14
16
18
20
Archive : Stopped
Content
Breakout Rooms
Web
eBoard
Share
Default Content Folder
Go
TALK
Options
Exit - Lobby - Help
People (2)
John_Thames
Robert_Wellman
Center for the Enhancement of Teaching and Learning
Main Room
Internet | Protected Mode: On
100%

Note: This folder does not have any slide content.

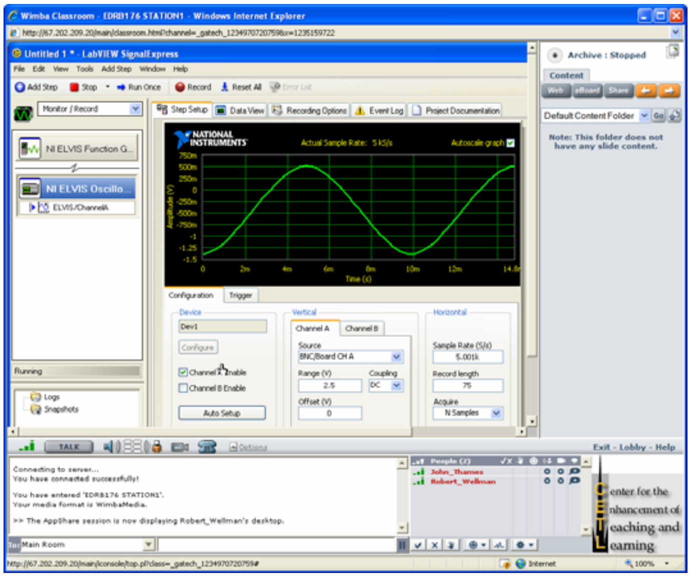
You have entered 'E2031'.
Your media format is WimbaMedia.

>> The AppShare session is now displaying John_Thames's desktop.

>> You have become the default presenter.

>> You are no longer the default presenter.

Figure 14. Web-based desktop sharing application.



Device: Dev1

☑ Channel A Enable

☐ Channel B Enable

Source: BNC/Board CH A

Range (V): 2.5

Coupling: DC

Offset (V): 0

Sample Rate (S/s): 5.001k

Record length: 75

Acquire: N Samples

Note: This folder does not have any slide content.

Connecting to server...
You have connected successfully!

You have entered 'EDRB176 STATION1'.
Your media format is WimbaMedia.

>> The AppShare session is now displaying Robert_Wellman's desktop.

Figure 15. LabVIEW software via the Wimba web interface.

Notice in Figure 15 that the remote desktop session is embedded within the Wimba web interface. From this point, the student can perform all of the laboratory exercises from this interface. Once the lab session is complete, the student will close the Wimba session and the CCC will instruct the RLU to terminate its Wimba connection and return to its polling state.

There are several benefits for this type of infrastructure. First, there is a single point (the CCC) that enables user functions such as student interfaces and lab management interfaces (i.e. for managing the large number of RLU devices). Second, the CCC can implement authentication and role-based access control. Third, the enterprise security change management process does not need to be invoked for any RLU device. The change management process will happen only once, and that occurs when the CCC is first placed on the network. Fourth, since the RLU devices are not accessible through the firewall to the Internet and only accept command instructions from the CCC, and since the CCC is providing authentication and role-based access control, the architecture retains a very high-level of remote laboratory security. Finally, the user interfaces within the CCC can be implemented with Web 2.0 technology such as the ones shown in this section. Web 2.0 provides many collaborative features such as web-based remote desktop sharing, really simple syndication (RSS) feeds, blogs, wikis, real-time chat and video teleconferencing, and many, many more. Technology such as this provides many opportunities for both instructors and students who are participating in RPL environments. We believe that Web 2.0 coupled with C&C architecture, as described in this paper, provides a very real and promising solution for large scale remote and portable laboratory deployment.

8. CLOSURE AND FUTURE WORK

Distance learning has proven to be a necessary service that will be offered by many higher education institutes. However, providing distance learning coursework for students in engineering, science, and technology is currently challenged due to the vast number of courses within these disciplines that require laboratory components. Current research in education that focuses on the concept of remote and portable laboratories is showing that this challenge can be solved, and recent results show that remote and portable laboratories retain good pedagogical outcomes. However, there is still much work remaining. The mapping of traditional laboratories into remote laboratories is making good progress. But, efforts are needed to develop cohesive sets of remote laboratories to accommodate entire educational programs so that the current ad hoc remote and portable laboratory design process can be replaced. Next, the challenges with IT infrastructure must be addressed so that remote and portable laboratories can be deployed in large scale. This paper described one such infrastructure. Our goal was to set in motion future discussions aimed at solving the IT challenges that were revealed in this paper. Our future work will focus on methods to streamline and standardize this infrastructure with hopes to create a system that can be used as a model when deploying remote and portable laboratory environments.

9. REFERENCES

[1] Kirkland, N., Vitanov, V., and Schaefer, D., 2008, "An Investigation into Using Current Information Technologies to Provide Engineering Education to Sub-Saharan Africa," International Journal of Engineering Education, 24(2), pp. 345-356.

[2] Rippel, M., Panchal, J. H., Schaefer, D., and Mistree, F., 2009, “Fostering Collaborative Learning and Educational Mass Customization in a Graduate Level Engineering Design Course,” International Journal of Engineering Education, in print.

[3] Shen, H., Zu, Z., Dalager, B., Kristiansen, V., Strom, O., Shur, M., Fjeldly, T., Lu, J., and Yterdal, T., 1999, “Conducting Laboratory Experiments over the Internet,” IEEE Transactions on Education, 42(3), pp. 180 -185.

[4] Hamza, K., Alhalabi, B., and Marcovitz, D., 2000, “Remote Labs!”, *Proceedings of the Society for Information Technology and Teacher Education International Conference*, C. Crawford et al., eds., Chesapeake, VA, pp. 155-161.

[5] Sanchez, J., Dormido, S., Pastor, R., and Morilla, F., 2004, “A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated with an Inverted Pendulum,” IEEE Transactions on Education, 47(3), pp. 321-329.

[6] Kikuchi, T., Fukada, S., Fukuzaki, A., Nagaoka, K., Kenjo, T., and Harris, D., 2004, “DVTS-Based Remote Laboratory Across the Pacific Over the Gigabit Network,” IEEE Transactions on Education, 47(1), pp. 26-32.

[7] Sebastian, J., Garcia, D., and Sanchez, F., 2003, “Remote-Access Education Based on Image Acquisition and Processing Through the Internet,” IEEE Transactions on Education, 46(1), pp. 142-148.

[8] Chang, G., Yeh, Z., Chang, H., and Pan, S., 2005, “Teaching Photonics Laboratory Using Remote-Control Web Technologies,” IEEE Transactions on Education, 48(4), pp. 642-651

[9] Scheets, G., Weisner, M., and Sharda, R., 2005, “Changing a Standard Telecommunications Laboratory to a Same-Time-Different-Place Virtual Laboratory Format: Techniques Utilized and Lessons Learned,” IEEE Transactions on Education, 48(4), pp. 713-718

[10] Schaefer, D., Scott, D., Molina, G., Al-Kalaani, Y., Murphy, T., Johnson, W., and Goeser P., 2008, “Integration of Distance Learning Technology into Traditional Engineering Physical Laboratory Exercises,” *Proceedings of the ASEE Southeast Section Conference*, Memphis, TN.

[11] Datta, K., Sass, R., 2007, “RBoot: Software Infrastructure for a Remote FPGA Laboratory,” *Proceedings of the FCCM 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, CA..

[12] Karadimas, D., and Efstathiou, K., 2007, “An Integrated Educational Platform Implementing Real, Remote Lab-Experiments for Electrical Engineering Courses,” Journal of Computers, 2(2), pp. 37- 44.

[13] Marques, R., 2008, “Design and Implementation of a Reconfigurable Remote Laboratory Using Oscilloscope/PLC Network for WWW Access,” IEEE Transactions on Industrial Electronics, 55(6), pp. 2425-2432.

[14] Gurkan, D., Mickelson, A., and Benhaddou, D., 2008, “Remote Laboratories for Optical Circuits,” IEEE Transactions on Education, 51(1), pp. 53-60.

[15] Stancil, D., Gist, N., and Jiang, Y., 2007, “REAL: The Remote Educational Antenna Laboratory,” *Proceedings of the IEEE International Symposium on Antennas and Propagation*, pp. 5399-5402.

[16] Enloe, C., Pakula, W., Finney, G., and Haaland, R., 1999, “Teleoperation in the Undergraduate Physics Laboratory-Teaching an Old Dog New Tricks,” IEEE Transactions on Education, 42(3), pp. 174-179

[17] Jodl, H., 2007, “RCL – Remotely Controlled Laboratories, http://rcl.physik.uni-kl.de

[18] Tzafestas, C., Palaiologou, N., and Alifragis, M., 2006, “Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation,” IEEE Transactions on Education, 49(3), pp. 360-369

[19] Ogot, M., Elliott, G., and Glumac, N., 2003, “An Assessment of In-Person and Remotely Operated Laboratories,” Journal of Engineering Education, 92(1), pp. 57-63

[20] Nickerson, J., Corter, J., Esche, S., and Chassapis, C., 2007, “A Model for Evaluating the Effectiveness of Remote Engineering Laboratories and Simulations in Education,” Journal of Computers and Education, 49(3) pp. 708-725

[21] Esche, S, 2005, “On the Integration of Remote Experimentation into Undergraduate Laboratories—Pedagogical Approach,” International Journal of Instructional Media, 32(4), pp. 397- 407.

[22] Lang, D., Mengelkamp, C., Jager, R., Geoffroy, D., Billaud, M., and Zimmer, T., 2007, “Pedagogical Evaluation of Remote Laboratories in eMerge Project,” European Journal of Engineering Education, 32(1), pp. 55-72.

[23] Esche, S., Chassapis, C., Nazalewicz, J., and Hromin, D., 2003, “An Architecture for Multi-User Remote Laboratories,” World Transactions on Engineering and Technology Education, 2(1), pp. 7-11.

[24] Troger, P., Rasche, A., Feinbube, F., and Wierschke, R., 2008, “SOA Meets Robots – A Service-Based Software Infrastructure for Remote Laboratories,” International Journal of Online Engineering, (4)2.

[25] Salzmann, C., and Gillet, D., 2007, “Challenges in Remote Laboratory Sustainability,” *Proceedings of the International Conference on Engineering Education*, Coimbra, Portugal.

[26] Lindsay, E., Long, P., and Imbrie, P., 2007, “Workshop – Remote Laboratories: Approaches for the Future,” *Proceedings of the 37th ASEE/IEEE Frontiers in Education Conference*, Milwaukee, WI.

[27] Wimba Software, http://www.wimba.com/

[28] Sakai Software, http://sakaiproject.org/portal

[29] Wikipedia, http://en.wikipedia.org