Multi-Method Virtualization
An Architectural Strategy for Service Tuning

Erik Moore
Co-Director, Center for Information Assurance Studies
Technical Lead, Academic Research Network
School for Computer & Information Sciences, Regis University
emoore@cyberetower.com

Daniel M Likarish
Director, Center for Information Assurance Studies
Degree Chair, Information Technology
School for Computer & Information Sciences, Regis University
dlikaris@regis.edu

Abstract
A wide variety of virtualization methods have recently come into common use. Using these methods as a suite of tools to tune service offerings has becoming a viable strategy for enhancing an Academic Research Network (ARNe) as it services a globally dispersed user population. This report will present a model for organizing and applying these virtualization methods to maximize service efficiency.

1. Introduction

1.1 Background

Expansion of the online teaching environment for computer, information and systems science at Regis University has resulted in a corresponding expansion of and reliance on an Academic Research Network, ARNe. ARNe provides course-enhancing services, resources, and environments to a large global student population, particularly in the area of technical graduate studies. The university has been providing virtual labs in support of technology courses for about ten years. The primary service at ARNe is to deliver research and virtual labs for a variety of computing courses. The designs of virtual laboratories (Vlabs) are modeled after traditional engineering or science laboratory facilities, with associated open lab hours laboratories where student perform individual research in a supported environment or prepare findings for classroom presentations based on experiential investigation of concepts through hands-on learning. As the complexity of the variety and type service has increased, so had the need to provide a method to prescriptively manage complex delivery of this environment in support of research projects and course materials.

The introduction of server virtualization, social networking, application hosting and system emulation in the academic computing environment has created opportunities to expand services to students and faculty. In concert with the opportunities presented by these rapid technological developments, the increase in services has resulted in a increasing requirement to tune delivery of materials to students and faculty that have come to expect Web 2.0 style services. Application hosting provides schools the ability to deliver complex applications to the student’s desktop while maintaining a centralized support and maintenance model. Thus, a variety of complex multi-threaded applications can be delivered as an integrated package in support of a variety of curricular requirements, research needs, learning styles, and assessment of learning objectives.

Advances in virtualization of operating systems (OS) provides students with Internet-based access to stable hosted virtual machines. Virtualization also provides an easier method of developing integrated environments in which to develop curricular-supporting resources. These designs are based on physical laboratory resources that are usually found in traditional campus-based facilities. For example, three hours of lecture can be supported by an out of class laboratory experience using virtual systems loaded with an application package tuned to the learning objectives and assessments of that course. Virtualization from simple to complex networks result from the appropriate combination of virtualized disparate systems, and network devices. A variety of emulation applications, such as GNS3, provide an extension of the individual OS into complex networks through the simulation of network devices.

The advent of social networking tools in support of courses added additional complexity to the delivery of course content to students. Regis University provides students in our graduate practicum program with
access to video/desktop conferencing and multi-user 3-
dimensional metaverse spaces like Second Life. The
ad hoc development of complex systems has resulted
in the need to develop a model based management
method to accurately tune delivery of course materials.

1.2 Context of Model Development

Following a general approach presented by various
network models, such as the OSI Model, virtualization
methods can be divided into layers that slice resources
between the user and service initiation point. This
report describes a model and strategy for applying
virtualization methods along a service pipe to
maximize the efficiency of the cost and bandwidth of
each segment of the pipe. This model was designed to
support an academic research network, a five node
regional VPN-based WAN that services about 3000
internationally distributed users whose locations are as
diverse as the South Pole, India, Iraq, and the South
Pacific, and includes a large population in the United
States. The primary users are students engaged in
laboratory experiments and exercises associated with
graduate level coursework. But there is also a
significant percentage of students, faculty, alumni, and
partners that participate in collaborative research, pilot
projects, and course support efforts.

2. Organizing Virtualization by Layers

The layers of the service virtualization model
divide virtualization methods into five distinct
categories. ARN team managers use virtualization
methods within each category to either reduce or
expand usage of resources in a particular component of
a service. The layer diagram represented in Table 2
came about as the result of a need for a collaboration
framework with a large graduate student population
who provide design and development work for a
complex set of services within ARNe. To use the
diagram, the graduate students who develop, design,
and support ARNe services analyze a particular service
line. Then they consider the virtualization methods
that might expand or reduce the use of components
within that service line in order to use it efficiently
while working to create a responsive and cost effective
service. The table options reflect the particular choices
that existed in our workspace and not necessarily the
full range of choices available currently. Much like the
Open Systems Interconnection (OSI) model layers in
Table 1, [1], the Virtualization Layer Model (VLM)
divisions in Table 2 represent virtualization of user
experience at the top layer and physical enablement at
the lower layers. While the individual technology
option in every layer of the VLM could be expanded to
map across the OSI layers if they were considered as a
technology, the options are placed based on the
primary user function that it virtualizes. Below is an
example of a technology component set mapped to the
OSI Model in a traditional way.

<table>
<thead>
<tr>
<th>OSI Layer</th>
<th>Example Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Telnet</td>
</tr>
<tr>
<td>Presentation</td>
<td>ASCII</td>
</tr>
<tr>
<td>Session</td>
<td>Secure Sockets</td>
</tr>
<tr>
<td>Transport</td>
<td>TCP</td>
</tr>
<tr>
<td>Network</td>
<td>IP</td>
</tr>
<tr>
<td>Data Link</td>
<td>Ethernet 802.3</td>
</tr>
<tr>
<td>Physical</td>
<td>10BASE-T</td>
</tr>
</tbody>
</table>

Also, like the OSI model, the Virtualization layer
Model is designed to create a modularity, so that
technologies can be defined, developed, implemented,
and maintained within a larger constellation of
enabling technologies. However, layered models are
insufficient to visualize end-to-end service offerings in
the case of this research facility, so additional models
will be introduced later that describe topological
progression of services.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Technology Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community/Social</td>
<td>SecondLife™, OpenSimulator™, Adobe Connect™, TeamSpeak™, SharePoint™ Portal, Moodle™ LMS, Angel™ LMS</td>
</tr>
<tr>
<td>Application</td>
<td>Citrix (XenApp)</td>
</tr>
<tr>
<td>System</td>
<td>VMware™, Virtual Box™, Xen™</td>
</tr>
<tr>
<td>Storage</td>
<td>HDSTM Thunder, HP EVA™, iSCSI software</td>
</tr>
<tr>
<td>Network</td>
<td>GNSSTM, Cisco™ VLANNing, VMware™ Virtual Networks</td>
</tr>
</tbody>
</table>

Unlike the OSI model, the VLM extends into the
virtualization of social environment. This is not
mappable to the OSI application layer, but is
comprised of technologies that virtualize traditional
environments in which technology is used, where inter-
personal discourse occurs, or how personal
environmental orientation occurs. These include
laboratory space, campus environments, meeting facilities, network operations centers, incident theaters of operations, etc. These virtualized environments are rendering non-technical cuing, scope-setting, and communications enabling conditions that would normally be experienced by the user in addition to the OSI application layer resources they perceive through the command line, system GUIs, or interactive topological diagrams of networked systems.

Also referenced in the development of the VLM was the Storage Network Industry Association™ (SNIA) Shared Storage Model v2.0 (SSSM2), seen in Table 3.[2] This model focuses on how storage blocks are aggregated and its layers divide the path between where the data is actually stored on a physical medium and where processing of data takes place in an application. Unlike the OSI and VLM models that place intermediate and enabling layers of processing “down the stack”, the SNIA places intermediate processing in the middle Block Virtualization/Aggregation layer between the Upper Application Layer and the bottom Storage Device Layer where data is physically stored. Below is a simplified version of the SSSM2 model that indicates minimally the sequence of layering.

<table>
<thead>
<tr>
<th>SNIA Layer</th>
<th>Example Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Email Server</td>
</tr>
<tr>
<td>File</td>
<td>File System</td>
</tr>
<tr>
<td>Block Virtualization / Extent Aggregation</td>
<td>SAN Switch or Storage Virtualization Device</td>
</tr>
<tr>
<td>Block Subsystem</td>
<td>HDS Storage Device</td>
</tr>
</tbody>
</table>

The VLM did not scale to a broad description of virtualization as the management team tried to apply it to the ARN network. But it did call to attention another requirement for model development, the need for “end to end” modeling of a virtualized service that mapped out the intermediary components of a service to consider where bottlenecks are occurring and whether service activity is matching resource capacity along the service path. This led to the development of the Service Pipe modeling presented shown first in Figure 1.

Because of the highly dispersed user population, the target bandwidth of minimum-capacity user connections has been limited to 56kbps. Text base community virtualization is used for broad client support (the options in italic in Table 2.) Primarily these are in the areas of collaborate workspace (SharePoint) and Classroom environment (Moodle and Angel Learning Management Systems (LMS)). Multi-media options complete a full service suite for the research team members who have access to higher bandwidth connections to the Internet or local access in a lab facility.

The current ARN Application Layer of the VLM contains our current “local desktop services” virtualization through, Citrix Metaframe™. This allows users to log on to a desktop portal and access the unique service they need, without the underlying processing requirements of offering them completely virtualized operating systems or direct access to virtualized storage offerings. The application layer offerings available in industry present two distinct technologies. The first is full local desktop environment that provides a local desktop that interacts much as the desktop on a PC, Linux, or Macintosh operating systems, but does not require full system virtualization. This first technology can be accessed both specialized thin hardware and PC-based software clients. Those we have tested in our facility include SunBlade hardware thin clients, Windows Terminal Services desktop emulation, and Citrix. The second technology is application terminal service that presents a specific application in a window, generally initiated from a web link (for example opening a word processor from a web link on the Citrix Metaframe Desktop.). This is different from a web application using HTLM to interact with programs, and relies on screen refreshes to interact specifically with an instance of an application. This second technology has proven to be the most efficient for many service requirements inside ARNe. Both technologies generally rely on reproducing user interface output of application or system components and transmitting screen refresh information.

Options in the system virtualization layer of the VLM are generally reserved in ARNe for clients who require extended capabilities beyond desktop and application virtualization. That is because our team have found it much is less efficient than application, in terms of processor and memory load required to present an offering. Access to the virtual machine guest operating system is still accessed through some type of desktop virtualization, whether through a VMware client, VNC, Citrix, Microsoft Terminal Services, or other remote desktop applications which create the virtual presence of the system locally for the client.

Instead, system virtualization has found its primary use in ARNe by creating efficiencies of management and processing resources internal to the network services data center. This type of data center configuration using virtualization has been widely considered, as by W. Vogels [3]. But this report will present uses where offering virtualized systems directly to the client has been used to significant
advantage. When this is done internal to ARNe, generally clients are provided a service from that guest machine instance as if it were a hardware-based system. This includes traditional services like web, email, and other services.

Storage Virtualization options have been explored in ARNe, but Storage Area Network service within ARNe are not generally offered to external clients at the block layer because this service has stricter latency requirements than most other services. However, it is included in this model because it is a significant service that can be used to analyze virtualized systems that fits effectively into the structure of the VLM. Block virtualization of SAN traffic to remote machines would generally also create unnecessary additional bandwidth requirements through the service pipe to the external user. Instead, if SAN block-level storage services are necessary, it is offered to systems that exist within the ARNe facility, and those machines are offered to external users through some desktop virtualization method. This practice of moving virtualization within the service pipe to tune bandwidth and latency is a significant way to enhance service performance.

Network virtualization is very important in ARN because we have a large number of students and faculty studying networks and network security. The primary technologies used to create virtual networks for ARNe clients include the network device emulator suite GNS3 and VLAN technologies such as 8021Q. The choice between them is primarily based on performance. If clients are interested in running a limited number of network devices and are not using significant network capacity, GNS3 emulation is sufficient. The GNS3 network is then offered to the client through virtualized desktop technology discussed earlier. If clients need significant well-defined network capacity, Cisco VLAN technology are used to segment off existing physical network resources and offer it as a virtual network that exists inside the ARNe facility. While it would be simple to provide access to this virtual network resource through virtual private network (VPN) technologies, generally ARNe has consolidated access through a single Citrix portal in order to provide a single point of security access. access control lists are created to provide desktop virtualization to a first node inside the VPN. This technique is also important in order to isolate client machines from activity on the network that would be harder to control with a VPN.

### 3. Mapping Virtualization With a Service Line Pipe Model

Services lines can be represented as a long pipe from the initial service to the client interface. In figure 1 there are two layers of virtual network. The internal VMWare client is run on a Citrix server, and the Citrix server offers that application out through the Citrix portal as an ICA client. While this initial configuration allowed for a consolidation of access to the Citrix portal, the two-step rendering of the desktop, first with VMware, and then again through Citrix, created a lag in response that was unacceptable for service provisioning. Note, “I” stands for “Internet”, and relates to ISP service options and tested characteristics in particular troubleshooting incidents. It will not be discussed here, but is generally taken as a constant in service provisioning from a particular location.

![Figure 1: Service line virtualization Citrix/VMware](image)

Reconfiguration of the service pipe required four actions. First, the VMware client is distributed to the end users. Second, a security portal must be created to allow for the authorized pass-through of VMware Desktop protocol Traffic. Third, the VMware servers must be made available through this security portal. Fourth, The client population should have a link to this portal present in their community virtualization layer. A security portal could be a product such as VMware vCenter Lab Manager.

![Figure 2: Authentication and VMware pass-through](image)

The change here above represents a troubleshooting scenario that occurred in ARNe and led to the
development of service pipe diagramming as a means to analyze the virtualization components of service offerings. This report presents below three cases where service pipe modeling is used to analyze service offerings.

### 3.1. Case 1: Distributed Oracle™ Admin

Regis University teaches many different database classes using Oracle. When the program was introduced, Oracle was included with the textbook. It was initially considered that the most cost effective way to have students perform lab exercises was to have them install Oracle locally on their systems and then provide lab guides for the students to follow in order to gain direct experience with the technology. One important component of the success of this approach was providing adequate support for students to successfully complete the lab. With an international student population, and minimal system requirements for the curricular program, students came to this class with a wide variety of operating systems that included a wide range of software installations and network configurations.

In the initial service pipe diagram (Figure 3), there is no virtualization of the Oracle Application. However, the student uses a web browser to access the LMS and potentially the technical help desk directly. These two components in concert comprise the typical support environment found in a scheduled laboratory session live on campus. While this service is not using a specialized application or protocol to virtualize the experience, the web-based resources create the context in which Oracle is used by the student. Support is available to the student in three primary forms. Peer support in forums, instructor assistance in forums, and the help desk.

![Figure 3: Oracle client install](image)

The lab designers hoped that distributing Oracle across a large number of client machines on a one-to-one basis would have the effect of distributing processing load in proportion to use across many individual client stations rather than having many instances of Oracle on one concentrated set of institutional servers in the research lab. The goal was to avoid having the costly requirement of acquiring more processing resources for ARNe. In addition, expected network traffic for Oracle would be nonexistent, so no additional network resources would be required.

However, the most limited resource in this case proved to be something other than the potential cost of centralized server processing. It was the help resource required for students to successfully complete the lab. While the processing was dispersed to a many-to-many model, the help desk became the only resource successful in troubleshooting the disparate problems. The help-desk support resources were moved to a one-to-many model that relied on live human response. Students were given a CD with their textbook and asked to load up Oracle on “Whatever system they were working on.” This created a heterogeneity problem that amplified the support requirements by the number of users having different systems, (virtually everyone.) Generally in online labs associated with classes there are three tiers of available support. 1) Students rely on peers to provide basic hints and suggestions for managing technologies associated with labs, though completing labs and answering questions is clearly described as individual responsibility. But because the Oracle installations were distributed across highly divergent systems, and Oracle is sensitive to those changes, the peer-oriented many-to-many support model held little value. 2) Instructors generally provided the next level of support for their students at a ration of about one to fifteen. Instructors are not compensated to do significant troubleshooting on fifteen different machines. The instructor’s primary responsibilities reside in explaining the content of the course and assisting with laboratory exercises once the Oracle software is up and running. So the faculty began to offload the Oracle technical support questions onto the help desk resources. 3) The help desk was operated by one individual at a time. Because of this one-to-many support requirement an immediate long cue of service requests was generated, creating untenable delays in the students’ ability to get Oracle working for lab assignment deadlines. Further, because of the heterogeneous nature of the client systems, the knowledge base used to facilitate support was generally inadequate and hard to navigate.

To tune this problem, ARNe managers redesigned the service line to offer students a virtual desktop so that students logged in through a Citrix server. The Citrix server created a homogeneous thin client that could be distributed to students in the Oracle class in addition to most of the other labs that ARNe provided support for. This thin client was not as sensitive to the changes in operating system configurations, and many
different clients were available to support the different operating systems. This meant that the help desk requirements were dramatically reduced using the configuration illustrated in the following service pipe. The Oracle Relational Database Management System (RDBMS) managed multiple instances of Oracle on a single server. Students logged into Citrix, and launched Putty, a communications utility to access their particular database on a separate server running the Oracle RDBMS. While Putty is a thin network client, the use of Citrix meant that there was no additional client-installed software in addition to what was needed for other laboratory courses. This means that no course-unique knowledge is necessary to support client connections and clients software installations in the database classes. It also means that the standard external-facing security available in Citrix could apply to the students in the database classes.

Figure 4: Oracle application virtualization

ARNe managers attempted a second phase of service tuning in the Oracle Service Line, as follows, because maintenance of the Oracle instances over time became problematic with graduate assistance turnover, requiring significant training to facilitate long-term maintenance of the RDBMS. The team wanted to use Virtualization to take snapshots of the RDBMS server at the beginning of each semester to reset courses. The ARNe management team researched similar installations to determine the viability of this strategy and while Oracle Corporation refused support, some users were reporting success with Oracle™ on VMware™.[5]

Figure 5: Oracle VMware attempt

When the ARN management team implemented Oracle™ as a VMware guest, several problems occurred. Oracle™ only accessed half of available memory, and a significant portion of memory was devoted to the virtualization process. Because the RDBMS was database and processor intensive, the service slowed below satisfactory performance. This attempt at virtualization for the convenience of maintenance failed. However, there are currently new offerings by Oracle that include a customized virtual machine built on the Xen™ hypervisor.[6] Currently the ARNe management team is beginning to implement this as the next generation of RDBMS platforms to support these classes.

3.2. Case 2: GNS3 and VLANs

Network virtualization meets a broad range of needs, but the functional requirements significantly determine which method of virtualization is appropriate, depending on available resources. This section compares virtualization options for two similar service lines that used VLANning and GNS3. The goal for network labs in general is to provide configurable network space for teaching complex network configuration, forensic analysis, and research project activities. But the virtualization options selected depend greatly on the specific activities that will be required of students as part of their laboratory experience.

The first service line supports a set of undergraduate traffic analysis labs. These labs need to present network equipment passing traffic at machine speeds and servers and clients that generate this traffic. Generally OpNet™ would be a tool of choice for simulating load and capacity planning. OpNet™ is a network simulation tool that uses a proprietary interface to create a broad array of machine, software, and network simulation for network designing, proof-of-concept modeling, and performance load planning. But OpNet™ is generally not part of the undergraduate program, and teaching the OpNet interface for students to complete one set of labs was not practical because it is complex and dissimilar to other technology they use. Since the actual hardware for this lab was available as a surplus within existing ARN network resources, the hardware was partitioned using 802.1q VLAN tagging. An access control list was developed to allow for desktop virtualization protocols to enter, and access a single server inside the network. From there, students could jump to other network nodes and configure applications from a software library resident on that server.
As mentioned earlier, VPN clients create problems in this scenario by exposing to attack the ports of the particular client that is logging in at the time, and by introducing another set of software that ARNe clients must install, for which ARNe clients must receive technical support. Because network resources are limited, in this scenario, the students need to schedule separate timeslots to work in the lab and save their configuration and performance data in separate directories.

In contrast to performance testing, the virtualization requirement of a lab for an “Introduction to Routing and Switching” class is quite different. Students do not need realistic speeds in regard to traffic patterns, but the requirement for introducing a broad range of devices is higher. In this case, GNS3 became the preferred option because it, and the underlying emulators, can create an environment with a broad range of Cisco™ router-like functionality, network connections firewalls, VLANable switches, and network clouds. In the initial configuration, the ARN management team deployed GNS3 on a stand-alone server because of concerns about underlying hardware requirements to achieve the necessary performance. The system was placed inside ARNe, and students used the ICA client to access the VNC™ client. The students then jump from the Citrix™ server to VNC, a desktop remote access tool, in order to access the desktop of the system where GNS3 is running.

This should be more efficient in terms of physical resources (processor time), because the dedicated system will become unnecessary. This configuration should also help reduce internal ARNe LAN traffic. Both of these things further suggest that system maintenance and monitoring for this service line will also be significantly reduced.

### 3.3. Case 3: Scenario Immersion for Information Assurance

For education, training, and research purposes in Information Assurance, the ARN management team has developed 3D multi-user metaverse environments in order to provide simulated environments for both training and research purposes. Information assurance has special requirements which are readily met by immersion experiences. These requirements come about because Information Assurance is a field where the social interaction and technological implementation act as a coherent system in which the information security officer must make comprehensive choices that affect both aspects of a scenario. Options for creating a service that creates this immersion experience include Second Life and OpenSim. Both of these environments allow multiple people to gather together in a virtual work and work as a team to overcome challenges that can be adapted on the fly by a team of event managers who are also in the virtual world. In addition CyberCIEGE is used to provide single-user scenario immersion in which students interact with a database of options in order to overcome challenges. This option is more like a CD Game where one is “playing against the software” in the hopes of making correct choices and other personalities in the world are for various network topologies and functional considerations.

The ARNe managers are moving this service line to a cleaner model of implementation that will eliminate the secondary desktop virtualization using VNC.[7]

The Citrix server offer it directly through Citrix desktop virtualization.
generated by software. Multi-user metaverse space like Second Life and OpenSim, as well as game interfaces like CyberCIEGE have special considerations when it comes to virtualization in a service line. This section compares attempts to implement these three components in a service line to provide information assurance training immersion environments to a distributed population.

The primary consideration of implementing service lines that use 3D rendering space is to make sure that world rendering takes place on the client. The standard implementation of CyberCIEGE is to download the application and install it on the client system as illustrated below.

![Figure 9: CyberCIEGE Download and Run](image)

This implementation proved successful, and as long as minimum hardware requirements were met, troubleshooting of this desktop application software was not significant for the student population engaged in the supported course. However, the ARNe management team did a pilot study to determine if it could reduce the technical support requirements involved in the installation of a new piece of software on student machines. A copy of the software was loaded onto the Citrix server and run locally.

![Figure 10: CyberCIEGE™ on Citrix](image)

This configuration proved to be fatal to the service, because the rendering of the interactive display image in CyberCIEGE apparently overloaded the rendering process and stream of the Citrix client/server application set. The refresh rate and level of detail that needed to be forwarded to the client was just too much.

Similar to CyberCIEGE, Second Life and OpenSimulator require a rendering browser that is located on the end-user workstation. However, a Simulation Server, or Grid of Servers to create a large connected space, is required to be a common node of connection of all clients so that events in the simulation can occur relatively simultaneously and interactively among all participants. In the case of Second Life, geographically dispersed students do not connect to the ARNe at all, but instead connect directly to the Second Life Grid™ presented by Linden Labs. Students and faculty present within the ARNe facility join students in the metaverse by also connecting to the Second Life Grid as illustrated Below. The Second Life Browser (SLB) is a client for a stream of information that updates objects within the metaverse simulation server. But 3D rendering results are only traveling within processes in memory and the graphics card on the client machines.

![Figure 11: Second Life](image)

In addition, video streams, audio streams, and web pages can also be experienced inside the Second Life world. But this traffic is handled as a separate input stream to the browser on the client that connects directly to the media source and does not pass through the Second Life Grid. In this sense, significant bandwidth planning and service analysis is necessary to provide the media. Client systems must be capable of handling both the 3D rendering and live streaming media sources. This load further limits the potential clients to a subset of the ARNe client population and makes these services necessarily either an optional luxury or the basis for a new set of hardware and networking requirements for the information assurance program. The Second Life implementation with media stream path is illustrated below, showing how metaverse updates and streaming media are integrated in by the browser.
While this configuration can provide a rich immersion experience, many clients such as partnering agencies requiring first-responder training cannot operate in a space that is open to the public or where security parameters are outside the controls of ARNe. Therefore, OpenSimulator is the option that the ARNe management team is piloting to meet the need of this group. OpenSimulator is an open source simulation server that operates under the BSD license. It allows ARN to house the simulation server, and provide Secure Socket Layer (SSL) Tunneling of the communications links if necessary.

OpenSimulator may eventually be the preferred environment for all coursework as the average bandwidth of client ISP connections expand and media-rich interactive environments are developed within this type of immersive space. Currently however, it is the constellation of virtualization methods that must be managed for different sets of users in order to provide the most effective services to the clients.

4. Modifications of Service Pipe Models

Network Architects intending to use the Service Pipe model to render the virtualization of service lines could vary the modeling conventions presented here to significant advantage. First, it is possible to illustrate the embedding of services by nesting service visually, as in Figure 14. This can be done to reflect the perception of “nested” services such as the media streams inside second life, or technological nesting, such as the SSL wrapping that might occur to enhance security on an OpenSIM connection. Another model variation used in the ARNe management team discussions on the whiteboard is to vary the height of each resource along the service pipe to reflect the cost of that service. Then as the service becomes allocated, its use of each resource along the pipe is rendered in proportion to average usage. This was not illustrated in this report because of space constraints and because the methods of doing this currently have inconsistencies.

Other variations of this modeling strategy exist within the ARN management team. The authors encourage readers of this article who apply these techniques to get in provide feedback and extensions to the modeling system.

5. Conclusion

The work in ARNe at Regis University has demonstrated how the use of a new system of virtualization modeling can significantly improve services by facilitating a more virtualization-savvy service design. While the initial virtualization layer model alone proved insufficient to provide this structure and awareness, this layered model in conjunction with the service pipe modeling method facilitated clear improvements in services. Often tuning services was not a matter of enhancing hardware infrastructure or purchasing new software, but the benefits were realized through more virtualization-aware designs within the same infrastructure. This model seems to be generalized enough to apply to a broad range of service enterprises with highly dispersed end users.

6. References

IEEE Transactions on Communications, Vol 28, no. 4, April 1980 pp 425 - 432


