



Workshops der
Wissenschaftlichen Konferenz
Kommunikation in Verteilten Systemen 2009
(WowKiVS 2009)

An emulation of VoD services using virtual network environments

Walter Fuertes and Jorge E. López de Vergara

14 pages

An emulation of VoD services using virtual network environments

Walter Fuertes¹ and Jorge E. López de Vergara²

¹ walter.fuertes@uam.es, <http://www.uam.es/>

² jorge.lopez_vergara@uam.es

Departamento de Ingeniería Informática, Escuela Politécnica Superior
Universidad Autónoma de Madrid, Madrid, Spain

Abstract: Virtualization platforms are a viable alternative for the implementation of IP network experimentation environments. These platforms facilitate the conducting of tests as if a real environment were used and therefore can reduce the risk of failure as well as investment and experimentation costs. This paper proposes to develop a method to improve the results obtained in virtual network environments, trying to resemble those obtained in a real environment. To carry this out, we have emulated a video-on-demand service over ADSL using Xen as a virtualization tool, just as it would have been through a real ADSL connection. Connectivity, IP addressing, switching, routing and video streaming were tested to check the functionality of virtual network environments. Then, the bandwidth, the delay, and the inter-arrival time of video streaming packets were measured both in real and virtual environments. Finally, these parameters were tuned in the virtual network environments obtaining a similar behavior in clients and servers of both cases.

Keywords: Emulation, performance measurement, traffic monitoring, video streaming, virtualization.

1 Introduction

Virtualization platforms are a potential technology to reproduce a real network topology using a virtual environment. They enable interconnected equipment to be emulated, which only requires the resources of a single physical computer. Virtualization can be used to maintain, execute and test multiple software environments [MHH⁺07], and to provide facilities for network dimensioning. In addition, virtualization platforms allow the evaluation of network service provisioning prior to production; therefore they can reduce the risk of network failures, as well as investment and experimental costs.

Currently there are at least two ways to achieve the networks dimensioning and comparative performance measurements. One alternative is to set up a mock-up network infrastructure in parallel, but this would require new equipment and hardware devices making it an expensive solution. The other alternative is to use simulation tools, such as NS2 [NS2], among others, which are used to evaluate the performance of networks. Nevertheless, these simulators are trying to replicate the performance of a real system (end-to-end delay, packet loss, etc.), using software, but are not able to closely reproduce the features and behavior of the hardware in a real system (emulation of new devices, configurations, architectures) [Rim07].

Facing these two alternatives, this work proposes to use virtualization platforms to assess the network service provision. According to the results obtained in [Mun08], virtualization tech-

nologies are approximately close to the ones derived from a real and simulated environment. However, virtualization technology generates a performance overhead caused by the virtualization layer [BDF⁺03]. This would mean that virtualization has not yet matured to ensure results close enough to the network service provision in real environments. Hence, which factors would be required to improve the results of such emulation? And, which operational conditions suggested for virtualization can be most effective?

In this context, as a contribution, this paper proposes to implement a method to improve the results obtained in virtual network environments, trying to resemble those obtained in real environments. In particular, our main goal is to obtain a result as close as possible to a real case. These improvements will provide strategies to interact with real time applications and to ensure their service delivery in similar conditions before putting them into production. In addition, these enhancements could improve user perception.

To carry out this work, we have emulated a real network service, namely video-on-demand (VoD) over ADSL by means of the Xen virtualization tool [BDF⁺03]. Firstly, we designed and implemented the required test-bed to make the comparisons between virtual and real environments. Secondly, we adjusted the network environment parameters to emulate the operational aspects of the real environment. With these results, we have found new guidelines to adjust the emulation parameters in virtual network environments.

The remainder of this paper has the following organization: Section 2 provides some background concerning virtualization networks, the Xen virtualization tool, video streaming and VideoLAN solution. Then, in section 3 the experimental setup is described which is used to emulate the VoD over ADSL service in the virtualization platforms. Section 4 shows a set of tests which were performed to improve our evaluation results. Section 5 discusses related work. Finally, the conclusions and future work are given in Section 6.

2 Background

2.1 The virtual network environment

Within the scope of this research, virtualization is in essence a technique to share hardware resources. It can be used to partition physical equipment to support multiple virtual machines, interconnect them, and to share hardware resources, such as CPU, memory and input/output devices. It provides an extra abstraction layer between the hardware and the operating system (OS). This technique enables, via hardware, to have several guest OSs of diverse types executing simultaneously [Jon06].

In this paper a virtual network environment can be defined as a set of virtual equipment (both end systems, routers and switches) connected collectively in a given topology deployed on one or multiple hosts, which emulates an equivalent system in which the environment is perceived as if it were real. The virtual network environment encapsulates a set of applications within a virtual network, enabling service configurations for a specific network in a realistic way.

2.2 Xen

Xen [BDF⁺03], is an open source virtualization tool, based on the para-virtualization technique. Xen provides more efficient processing and minor overhead which results in better performance than other virtualization platforms. Within the context of this paper, Xen provides the infrastructure to deploy and manage a virtual network environment which can be configured to emulate the provision of the VoD service. Finally, based on the above and considering that a lower consumption of CPU and memory of the host results in a better response of virtual machines, we have chosen Xen to be the tool used in our research. It provided the best results in our previous assessments published in [FV07] and confirmed in [BDF⁺03] and [WCCG08].

2.3 Video streaming

Video streaming is a method to transfer digital data in real-time. This process converts video and audio into a compressed digital format, and then distributes the data through the computer networks. The diverse ranges of video communication and streaming applications that do exist have different operating conditions and properties. Video communication applications may be used for point to point, for multicast, or broadcast communications, and video may be pre-stored, so-called VoD, or may be encoded in real-time, *e.g.*, video conferencing [ATW02]. However, systems that provide these services currently require large amounts of centralized resources and significant bandwidth to accommodate their subscribers. Considering the aforementioned reasons, and because it is a widely used service, we have chosen VoD here with the aim of trying to measure their network parameters and emulate this service using virtual network environments.

2.4 VideoLAN Solution

VideoLAN [Vid] is a software solution for video streaming designed to stream moving picture experts group (MPEG) videos or audio video interleave (AVI) files on high bandwidth networks. The VideoLAN VLC (initially VideoLAN client) can be used as a video server or as a client (as a video player) to receive, decode and display MPEG streams under multiple OSs.

In the context of this paper, we have selected the VideoLAN solution because it is free video transmission software for both for VoD and live video. In addition, the video format can be changed. VideoLAN is also a portable multimedia player that works both for Microsoft Windows and Linux and it can use diverse media storage and transmission.

In this article the VideoLAN solution basically has two main functions: *i)* as server to transmit video files of different format to the client; and, *ii)* as a client, it to control the communication between the client and the server, and implement the video play classic functions.

3 Emulation experiments

This section describes the method developed to improve the results obtained in virtual network environments. This method has been divided into five steps, as shown in the next flowchart (see Figure 1). Its design, implementation details, settings and assessment are the basis of our research and will be explained in the following paragraphs:

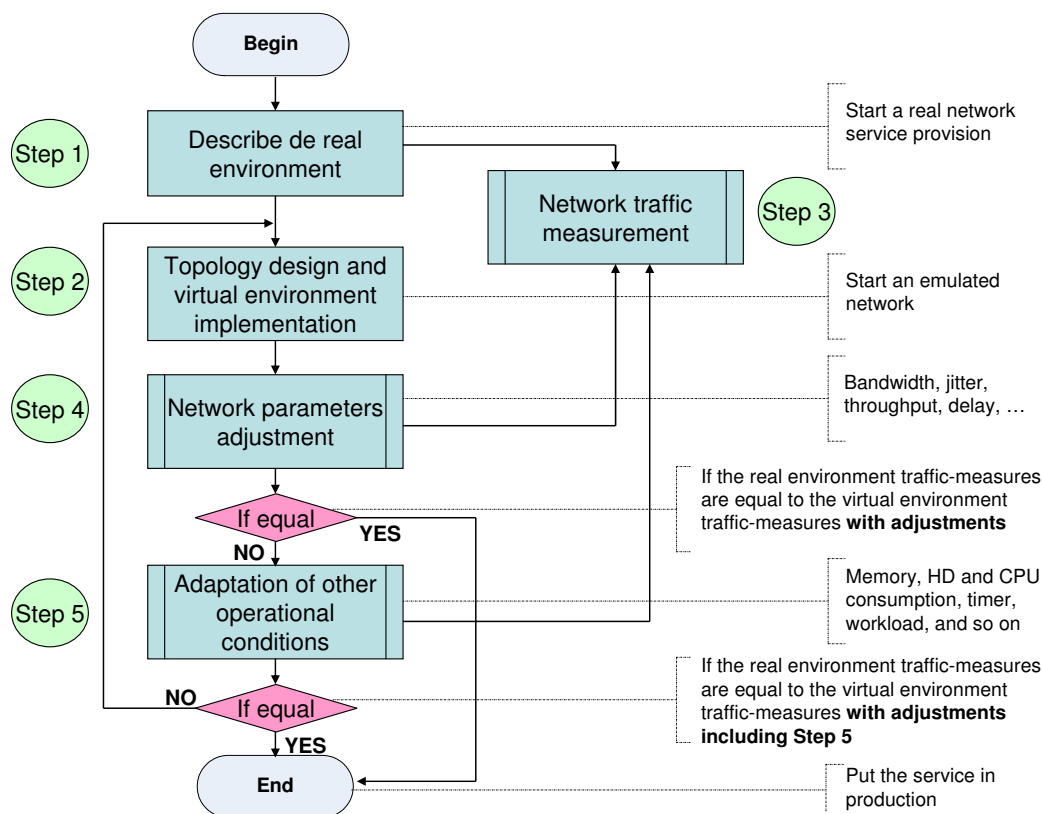


Figure 1: Flowchart to implement the method to improve the results obtained in virtual network environments.

3.1 Step 1: Description of real environment

As outlined in the flowchart, the first step was to describe the real environment and describe the experiment (see Figure 2). In this research we selected the VoD in an ADSL connection. ADSL was chosen because it was specifically designed to exploit the one-way nature of most multimedia communication. *i.e.*, the large amounts of information flow toward the user and only a small amount of interactive control information is returned. Next we tested the emulation of the service conditions for streaming VoD versus a classic 1.2 Mbps ADSL connection. Finally we accomplished the experiments described in Subsection 3.3.

Before proceed to Step 2, it is important to mention that in this research we did not consider the emulation of different transmission characteristics of ADSL connections concerning the last mile (*e.g.*, cable length, signal attenuation, or concurrent users). We focused on the bandwidth and delay, and we supposed that other properties would be masked by them. Given the obtained results (see section 4), this assumption can be considered acceptable.

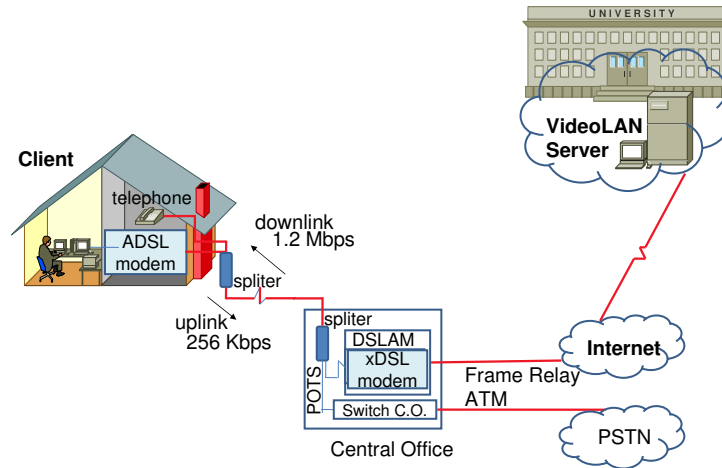


Figure 2: Testing environment of an ADSL connection.

3.2 Step 2: Design and implementation of the virtual network environment

As a second step in the method, to experiment the VoD service, we designed and implemented the virtual network environment shown in Figure 3. In this new environment, we installed and configured the VideoLAN server in a virtual machine located on one server in our laboratory at the University. On the other side we installed a VLC media player as a client on the same host, to display the video, especially given the need to handle a graphical environment and considering that Xen has limitations in such requirements. Then we fine tuned the link between Router 2 and Router 1 to make it similar to the case of an ADSL connection.

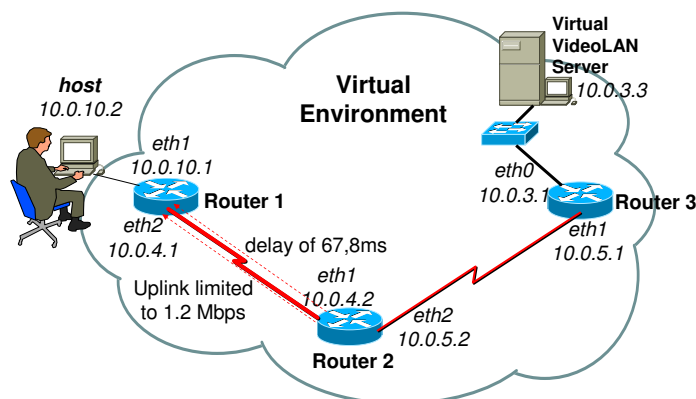


Figure 3: Virtual network environment used to emulate a VoD over an ADSL connection.

As shown in Figure 3, the elements of the virtual environment can be mapped to some elements of the real ADSL environment in Figure 2. (*i.e.*, the VideoLAN Server in each environment as well as the clients). Similarly, Router 1 maps to the home ADSL router, Router 2 maps to the ISP router, and Router 3 maps to the server router. Other routers could be also considered in the path between Router 2 and 3 if necessary.

The following procedure has been used to implement the proposed design in a virtual network environment: First of all we created the first virtual machine and installed the guest OS, Linux Debian. Then, we cloned the first virtual machine to the remaining virtual machines, in order to reduce the installation time. At this point, it should be noted that the routers depicted in Figure 3 are virtual machines to which the functionality of routing devices was assigned. After this, we added virtual interfaces, configured IP addresses, routing and started services. Then, we synchronized the clock of each virtual machine with a network time protocol (NTP) server within the real host. This allowed the virtual machines to synchronize their system time. This is very necessary because virtual machines (and routers in this case) work by time-sharing the host physical hardware, and a virtual machine cannot exactly duplicate the timing behavior of a physical machine. Lastly, we created and executed the respective programs that automatically constructed and started the environment. As a final point, we installed the software for traffic monitoring both in the virtual network environment and the host. It is also worth mentioning that all tests were done using open source software.

3.3 Step 3: Network Traffic Measurement

Once both environments were implemented, we took appropriate network traffic measurements according to step 3 of our method. We transmitted video files both in the real (Figure 2) and virtual (Figure 3) environments in the downstream direction from the VideoLAN server to the client. We conducted several tests to confirm that the traffic data capture was error-free.

To capture traffic for the two experiments described below, we used Tcpcdump [JLM] and disabled the promiscuous mode in the corresponding interfaces. Tcpcdump is a command-line tool whose main utility is to analyze the traffic going through the network. The logs obtained were visualized with Wireshark [Wir], which is a network protocol analyzer. Then, filters were applied to create flat files which were processed with a script to obtain the cumulative distribution function (CDF) of the video packet inter-arrival time. These probability distributions have been used to contrast the results, following the methodology presented in [GAH⁺07] to compare the performance of multimedia services on IP networks.

3.3.1 Real ADSL experiment

The first experiment consisted in the capture of video on real-time transport protocol (RTP) traffic in the real ADSL environment during the transfer of video (see Figure 2). This capture was made simultaneously both in the client at home and in the server located in the laboratory at the University, to obtain the CDF of the video packet inter-arrival time.

To obtain an appropriate value of the delay between the VideoLAN server and the ADSL client we applied the ping command with fixed-size packets of 1370 bytes. Ping measures the round-trip time (RTT) using the Internet control message protocol (ICMP) echo messages. With

this, based on the 4300 messages sent over 45 minutes we obtained a mean (μ) of 135.6 ms, with a correlation $r(0)$ of 40% and a standard deviation of 17ms for the RTT. It should be noted that the packet loss was negligible in the traffic measurement.

We used RTT instead of one way delay (OWD) that would be a better option for ADSL, given that OWD must be calculated between two synchronized nodes, which was not possible in the real environment where the experiment was carried out.

3.3.2 Virtualized environment experiment

The second experiment consisted in the capture of RTP video traffic in the virtual network environment (see Figure 3). All tests were carried out on a single host (Pentium D, 2.80 GHz, 1 GB RAM) with Linux Debian 4.0 and one Ext3 partition with 120 GB. In all virtual machines the same file systems and the same kernels (2.6.18-xen-686-GNU/Linux Debian) were installed.

3.4 Step 4: Adjustment of virtual network environment parameters

According to step 4 of our method, we adjusted the values of the parameters obtained in the previous experiment before transferring the video, to calculate the CDF of the video packet inter-arrival time in this environment and contrast the results. From these values the delay was configured to 67.8 ms (given the asymmetry of ADSL, we supposed as an approximation a symmetrical end-to-end delay and taking half of the RTT obtained for the real case). In addition, we limited the bandwidth on the link from Router 2 to Router 1 to 1.2 Mbps. It was also possible to emulate the ADSL uplink with a smaller bandwidth, but for the experiment this was not necessary because the video service is unidirectional (*i.e.*, from the VideoLAN server to clients).

To emulate the network parameters (bandwidth, end-to-end delay) at the eth1 interfaces of Router 2 and Router 1 of the virtual network environment we used the following Linux traffic control utilities: *i*) traffic control (tc) [Bro06], that encompasses the sets of mechanisms and operations by which packets are queued for transmission/reception on a network interface; *ii*) network emulator (NetEm) [Hem05], which is an enhancement of the traffic control facilities of Linux that allows adding delay, packet loss and other parameters [Kel06]. At this point we used this tool to emulate the end-to-end delay similar to what was obtained in an ADSL environment; and, *iii*) hierarchical token bucket (HTB), which is a useful queue discipline to limit the bandwidth rate. As shown in Figure 2, the ADSL connection last-hop has a maximum bandwidth rate of 1.2 Mbps, which must be limited in the virtual network environment. Once these adjustments were made, video traffic was captured in the virtual network environment, in order to make the comparisons described in section 4.

3.5 Step 5: Adaptation of other operational conditions

As a final step of the proposed method, the following conditions of the system operation where the emulation experiment was executed should be analyzed. This will identify mechanisms that can help to improve the method.

3.5.1 Dedicated server

Previous experiments were conducted in a dedicated server just for VoD service. When we added another service such as file transfer in the virtual network environment, performance degradation was increased. In summary, the degradation in response time can be explained because CPU resources are shared in the system. Among other solutions, this problem can be solved by adding the appropriate amount of resources [Fer06]. But the finest solution should be to balance the workloads in virtual machines.

3.5.2 Timer resolution

The OS uses these timers to provide a multitude of services. In the case of the virtual network environment, since there are several virtual machines running on a single platform, there is a variety of approaches to map the virtual clock into the physical platform clock, and any of these can cause clock skew. In addition, any changes in the clock behavior could lead to errors in computing the delivered performance. Due to clocking issues, utilization measurements from within virtual network environments are unreliable [Fer06]. In any case, to improve the measurements, the virtual network environment must be able to get access to high resolution timers. Higher resolution timers are needed to enable the system to process data at more accurate intervals. One manner to implement it is by adding patches to introduce a new subsystem, *e.g.*, high-resolution kernel timers (Linux hrtimers).

On the other hand, since the virtual machines must share the resources of the same CPU (provided by real host) and especially due to the complexities caused by emulating virtual hardware, such features are very time sensitive, leading to inaccuracy in system time of virtual machines. This fact has been well documented in [MRM06, Inc08]. Moreover, the installation of specific software for virtualization platforms, called Additions, is a better solution to enhance this synchronism [MRM06].

3.5.3 Other performance metrics

Besides of the traditional metrics that are already tested, the measuring of other performance metrics should also be considered such as number of users, number of virtual machines, workloads response time, CPU and memory utilization node, and so on. Moreover, as a final suggestion, the number of virtual CPUs must be equal to the number of physical core CPUs in the real environment, and; the amount of physical memory assigned to virtual machine must be equal to the physical memory in the real environment [CGS06]. This will create a fairer comparison.

4 Experimental results and discussion

4.1 Comparison between the ADSL server and virtual server

Figure 4 shows the results obtained by the CDF of video packet inter-arrival time for the real ADSL server and the virtual server. It is clear that the probability distributions are similar, although there is some disparity because of the overhead produced by the virtualization layer.

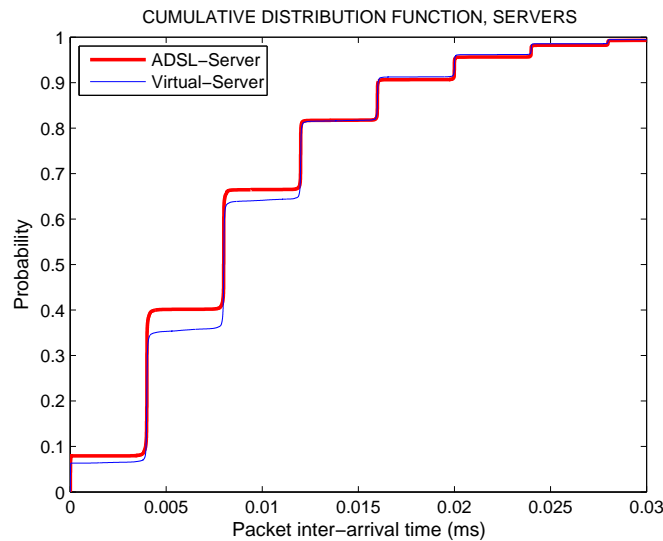


Figure 4: Cumulative probabilistic distribution of video packet inter-arrival time between the ADSL server and virtual server.

4.2 Comparison between real ADSL client and virtual environment client

Figure 5 shows the results obtained by the CDF of video packet inter-arrival time for the ADSL client, virtual client without adjustments, and virtual client with adjustments. This figure illustrates that the virtual client without adjustment is completely different from the experimental results of virtual network client with adjustments. Then again, the traces of the probability distribution of the two environments are also visually similar, but minor differences were found in the curves. This reveals the achievement of our method. Nevertheless, as noted in Figure 4, the CDFs are not exactly equal on the server side and hardly would be accurate on the client side. These results are related to adjustments made in step 4 (subsection 3.4) of our method.

4.3 Kullback-Leibler divergence

To compare the obtained results in our experiment we have calculated the Kullback-Leibler divergence [KL51], which measures the difference from a true probability distribution function (PDF) r to an arbitrary PDF v , as defined in equation (1):

$$D_{KL}(R||V) = \int_{-\infty}^{\infty} r(x) \log \frac{r(x)}{v(x)} dx. \quad (1)$$

Where $r(x)$ and $v(x)$ denote the PDF of R (real ADSL client) and V (virtual client) packet interarrival random variables.

The evaluation of equation (1) for the PDFs of the real ADSL client and the virtual client with adjustments gives a result of *divergence* = 0.0473. In addition, if the PDF of the real ADSL client is compared to the PDF of the virtual client without adjustments gives a result of *divergence* = 1.9972. As expected, these results show that the disparity is higher when the parameters are not adjusted in the virtual client.

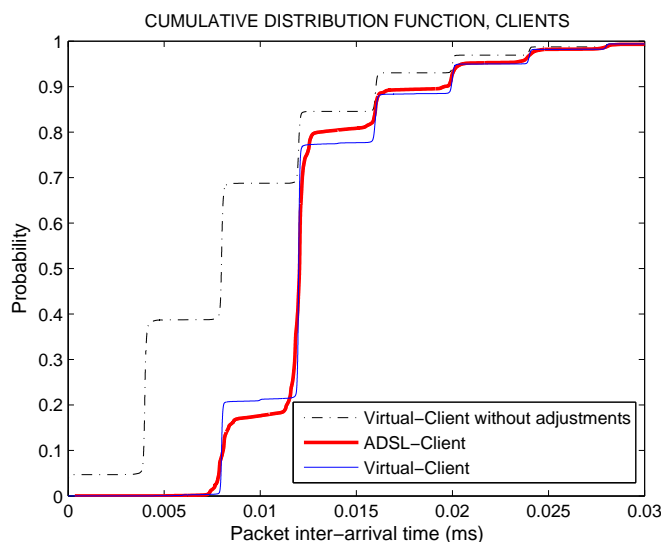


Figure 5: Cumulative probabilistic distribution of video packet inter-arrival time between the ADSL client, virtual client with adjustments, and virtual client without adjustments.

4.4 Discussion

As the experimental results have shown, our method has obtained relative similar results to emulate the conditions of a real environment using a virtual network environment. On the other hand this paper has demonstrated the viability of conducting these experiments. Moreover, given that a similar real infrastructure is not always available to compare the performance measurements, this research has shown that virtualized environments can be used to emulate a specific network service, whose results have been close to the real environment. Therefore, the results of the experiment have provided qualitative data related to how the services work, the perceptual quality of video service in terms of delay and limited bandwidth, the configurations required to transmit video in unicast or multicast, etc. Logically, these experiments can be done using real equipment. However, our proposal is intended to demonstrate that the virtualization platforms can be useful for the same task with lower hardware costs.

During the experiment we tried to control several factors. However, certain factors remain beyond our control (or they can not be controlled as yet). Some of them have been analyzed in this paper. Other factors such as the virtualization complexity and the virtual network hardware performance are out of the scope of this research. These factors generated some disparity in our results. In particular, because of: *i*) the overhead produced by the virtualization layer; *ii*) the conditions of network emulation (load, timers resolutions, and settings). The latter was caused because the parameters that NetEm could control were not sufficient to describe a complex network like ADSL and Internet, especially due to its dynamics and complexity [Hem05]; and finally, *iii*) the software and hardware components, in particular, do not provide high-precision timing guarantees [BGG⁺08].

Regarding scalability, it is possible to add more hops in the topology of the virtual environment, depending on the host real capabilities, where the virtual environment was deployed. Moreover, if the environment requires more resources, it is possible to distribute it among several

hosts, as proposed in [GFFM08].

With regard to virtualization as an approach to network emulation, the virtual network environment has fulfilled the aim to mimic the behavior of a VoD service, with the particularity that the functionality is not just theoretical or simulated, but rather a real time execution. Consequently, the advantage of this work, (despite the overhead imposed by the virtualization layer), is that these results give some certainty to experiment with other similar services without having to deploy them in a real environment.

Finally, the results obtained through the application of this method, lead us to research new guidelines to adjust the emulation parameters. In addition, it guides us to improve the conditions to provide the ideal environment for new experiments in virtualized environments.

5 Related work

There are very few papers that discuss the results of service emulation using virtualization environments. A first comparable research has been described by Fernando [Fer06]. In this paper the author explores the benefits of virtualization and discusses the difficulties in measuring such results in the real world. This work measures resource utilization from both the virtual machine monitor (VMM) kernel and the virtual machines. Our goal, however, is focused on finding guidelines to achieve better results in performance measurements in the virtual network environment.

A second comparable research has been described by Casazza *et al.* [CGS06]. Here, researchers explain a workload methodology to characterize the performance of server virtualization technologies to consolidate multiple physical servers. It presents a benchmark example using web server, e-mail server and database server. Our work, however, is focused on providing an infrastructure for conducting network dimensioning experiments. In addition, we have based ourselves on traditional performance metrics. Lastly, the comparison was made directly with data obtained between the two environments (real and virtual).

A third comparable work has been explained by Song *et al.* [SWL⁺07]. In this paper the authors study the interaction of VoD and other typical enterprise services in a computer consolidation environment context. They compare their results using benchmarks. Compared to this effort, our goal is to provide qualitative data associated to how the network services work using virtualization environments; how virtual network environments reproduce this behavior and how to mimic the results as much as possible to the real case.

Concerning service network emulation through virtualization environments, in [MHR07] network emulation test-beds provide a configurable virtual network environment for comparative performance measurements of real implementations. In the same context in [GGC⁺06] authors address the implementation through virtualization techniques of an IP multimedia subsystem (IMS) test-bed intended for the functional validation of services. Also, the research in [RSS07] presents the VMedia multimedia virtualization framework, for sharing media devices among multiple virtual machines. Compared with our work, these ones have the implementation of complex topologies as their main goal, so performance is a secondary goal. Unlike us, we were interested in finding a method to improve the results obtained in virtual network environments. Therefore, the performance was our primary goal.

Finally, another comparable work can be found in [BBKW03], where the performance of

virtual routers is evaluated for network research. However, our work has some differences. Apart from the use of Xen virtual machines, we have compared the results of the virtual environment with those obtained in a real environment, achieving similar values when applying our method to improve the results.

6 Conclusions

In this work, we have implemented a method to improve the results obtained in virtual network environments, contrasted with those obtained in real environments. We have performed a set of emulated experiments of a real VoD service of ADSL in a virtual network environment with Xen, using a single host computer. The experimental results have shown certain similarity in video packet inter-arrival time between both environments in the server side and client side. However, the probability distributions are not precisely the same, because the virtual network environment introduces a not quantified overhead and because the emulated ADSL delay is an approximation. In any case, the experiment results have provided qualitative data in relation to how the services work, the perceptual quality of video service, the required configurations, and so on. To conclude, we have presented a procedure to emulate network services in virtual network environments, emphasizing those factors that affect the experimental results.

As future work we will focus on how to quantify the overheads produced by the virtualization layer that affects the performance. We plan to study how the obtained results depend on the number of virtual machines and the number of connected devices that consume resources. Finally we will also explore how to implement and improve the delay emulation to increase the reliability of the measurements.

Acknowledgements: This work has been partially funded by Spanish Ministry of Industry, Tourism and Trade through Red.es under the PASITO project and by the Spanish Ministry of Science and Innovation under the DIOR project (TEC2006-03246).

The authors would like to thank the comments and good advice of José Luis García D., José Alberto Hernández and Bas Huiszoon, who helped to significantly improve this paper.

Bibliography

- [ATW02] J. Apostolopoulos, W. Tan, S. Wee. Video Streaming: Concepts, Algorithms, and Systems. Technical report, Mobile and Media Systems Laboratory. HP Lab. Palo Alto. Sep. 18th, 2002.
- [BBKW03] F. Baumgartner, T. Braun, E. Kurt, A. Weyl. Virtual routers: A tool for networking research and education. *ACM SIGCOMM Computer Communication Review. Vol. 33, New York, USA., Jul.*, pp. 127–135, 2003.
- [BDF⁺03] P. Barham, B. Dragovic, K. Fraser, S. H. T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield. Xen and the Art of Virtualization. In *Proc. of 19th ACM symposium on Operating systems principles*. Pp. 164–177. Bolton Landing, NY, USA, 2003.

- [BGG⁺08] N. Beheshti, Y. Ganjali, M. Ghobadi, N. McKeown, J. Naous, G. Salmon. Performing Time-Sensitive Network Experiments. In *Proc. of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*. San Jose, California. Pp. 127–128. 2008.
- [Bro06] M. A. Brown. Traffic Control HOWTO. *Version 1.0.2*. Oct., 2006.
- [CGS06] J. Casazza, M. Greenfield, K. Shi. Redefining Server Performance Characterization for Virtualization Benchmarking. *Intel Technology Journal*, 10 Aug., 2006.
<http://www.intel.com/technology/itj/2006/v10i3/>
- [Fer06] G. Fernando. To V or not to V: A practical guide to virtualization. *White paper, BMC Software, Inc.*, Jan., 2006.
- [FV07] W. M. Fuertes, J. L. de Vergara. A quantitative comparison of virtual network environments based on performance measurements. In *Proc. of the 14th HP Software University Association Workshop*. Munich, Germany, 8–11 July, 2007.
- [GAH⁺07] J. García-Dorado, J. Aracil, J. Hernández, S. López-Buedo, J. L. de Vergara, P. Reviriego, G. Huecas, S. Pavón, J. Quemada. A quality of service assessment technique for large-scale management of multimedia flows. In *Proc. of 10th IFIP/IEEE Int. Conf. on Management of Multimedia and Mobile Networks and Services (MMNS'2007)*. Pp. 173–176. San Jose, California, Nov. 2, 2007.
- [GFFM08] F. Galán, D. Fernández, M. Ferrer, F. J. Martín. Scenario-based Distributed Virtualization Management Architecture for Multi-host Environments. In *Proc. of System and Virtualization Management Workshop (SVM), CCIS 18. Munich (Germany)*, Oct. Pp. 49–60. 2008.
- [GGC⁺06] F. Galán, E. García, C. Chávarri, D. Fernández, M. Gómez. Design and Implementation of an IP Multimedia Subsystem (IMS) Emulator Using Virtualization Techniques. In *Proc. of the 13th HP OpenView University Association Workshop (HP-OVUA, France, May)*. Pp. 213–224. 2006.
- [Hem05] S. Hemminger. Network Emulation with NetEm. *Open Source Development Lab*. April, 2005.
- [Inc08] V. Inc. Timekeeping in VMware virtual machines. *White papers. Latest revision: 12 Aug.*, 2008.
http://www.vmware.com/pdf/vmware_timekeeping.pdf.
- [JLM] V. Jacobson, C. Leres, S. McCanne. *Tcpdump*.
[Available at: anonymous@ftp.ee.lbl.gov](mailto:anonymous@ftp.ee.lbl.gov)
- [Jon06] M. Jones. An overview of virtualization methods, architectures, and implementations. *Emulex Corp. Longmont, Colorado*, 29 Dec., 2006.
- [Kel06] A. Keller. *Tc Packet Filtering and NetEm Manual*, ETH Zurich, July 20, 2006.

- [KL51] S. Kullback, R. Leibler. On information and sufficiency. In *Annals of Mathematical Statistics*. Pp. 79–86. 1951.
- [MHH⁺07] J. N. Matthews, W. Hu, M. Hapuarachchi, T. Deshane, D. Dimatos, G. Hamilton, M. McCabe, J. Owens. Quantifying the Performance Isolation Properties of Virtualization Systems. In *Proc. of ExpCS07 San Diego CA 13 14 Jun. 2007*.
- [MHR07] S. Maier, D. Herrscher, K. Rothermel. Experiences with node virtualization for scalable network emulation. *Computer Communications*. Vol. 30, pp. 943–956, 2007.
- [MRM06] D. Marshall, W. Reynolds, D. McCrory. *Advanced Server Virtualization, VMware and Microsoft Platforms in the virtual Center*. In Averbach Publications. 2006.
- [Mun08] A. Munoz. Academic Research and Teaching with OPNET Software. Technical report, Univ. of Basque Country, Dep. of Electronics and Communications, 2008.
- [NS2] NS2. *The Network Simulator NS2*.
<http://www.isi.edu/nsnam/ns/>
- [Rim07] M. Rimondini. Emulation of Computer Networks with Netkit. Technical report, Dep. of Computer science and Automatization, University of Rome Italy, 2007.
- [RSS07] H. Raj, B. Seshasayee, K. Schwan. VMedia: Enhanced Multimedia Services in Virtualized Systems. Technical report, Published by Georgia Institute of Technology, No. GIT-CERCS-07-19, 2007.
- [SWL⁺07] Y. Song, H. Wang, Y. Li, Y. Sun, Y. Zeng. Can VoD streaming service co-exist with other services on a VM-based virtualized computing platform? In *Proc. of the Asian technology information China. HPC*. Pp. 95–103. 2007.
- [Vid] VideoLan. *VideoLAN Streaming Solution*.
<http://www.videolan.org/>
- [WCCG08] J. Walters, V. Chaudhary, M. Cha, S. G. J. S. Gallo. A Comparison of Virtualization Technologies for HPC. In *Proc. of 22nd International Conference on Advanced Information Networking and Applications (AINA)*. Pp. 861–868. Okinawa 25-28 Mar., 2008.
- [Wir] Wireshark.
<http://www.wireshark.org/>