

# BoxingExperience: Measuring QoS and QoE of Multimedia Streaming Using NS3, LXC and VLC

Javier Bustos-Jiménez  
NIC Chile Research Labs  
Email: jbustos@niclabs.cl

Rodrigo Alonso  
Universidad de Chile - DCC  
Email: ralonso@dcc.uchile.cl

Camila Faúndez  
NIC Chile Research Labs  
Email: camila@niclabs.cl

Hugo Méric  
Inria Chile  
Email: hugo.meric@inria.cl

**Abstract**—Quality of Experience (QoE) is defined as the overall acceptability of an application or service, as perceived subjectively by the end user including the complete end-to-end system effects. That means QoE may be influenced by the user expectations and context, adding a subjective component to measurements. Nevertheless, it has been studied that some metrics of Quality of Service (QoS), more related to the application and thus closer to the user's side, can be correlated with users evaluations for multimedia transmissions.

In this article, we build a modular framework to study the relation between QoS and QoE metrics for multimedia transmissions following the same separation of concerns as the Internet protocol suite. This framework, called *BoxingExperience*, relies on open-source softwares (NS3 and VLC) and Linux containers. We tested our framework in the context of video streaming from a network camera to several clients. Our simulations show that *BoxingExperience* can easily simulate, on a typical desktop computer, a scenario where multiple clients are connected to one streaming server.

## I. INTRODUCTION

Quality of experience is defined by ITU-T as “the overall acceptability of an application or service, as perceived subjectively by the end user”, which “includes the complete end-to-end system effects” and “may be influenced by user expectations and context” [1]. Nowadays, QoE has gained notoriety in both, an academic research and an industry perspective, given that as it is noted by the work of Schatz et al., “QoE is supposed to enable a broader, more holistic understanding of the qualitative performance of networked communication systems and thus to complement the traditional, more technology-centric Quality of Service (QoS) perspective” [2]. Then, QoS parameters such as packet loss, throughput, jitter, delay, bitrate are considered more technology-centric objective measurements of quality than those used for QoE which takes into account the subjective user component.

Nevertheless, some of those parameters are more related to the application itself (thus closer to the user) than the network operation, so they are more correlated with the users perception. For instance, former studies investigated the correlation between user's perception (via mean opinion score survey) and burst packet loss (that could be considered as a frame loss) [3], [4], jitter [4] and rebuffering [5]. Thus, a separation of concerns (such as in the Internet protocol suite) between network and QoE/QoS parameters is necessary for a better understanding of how network perturbations could affect

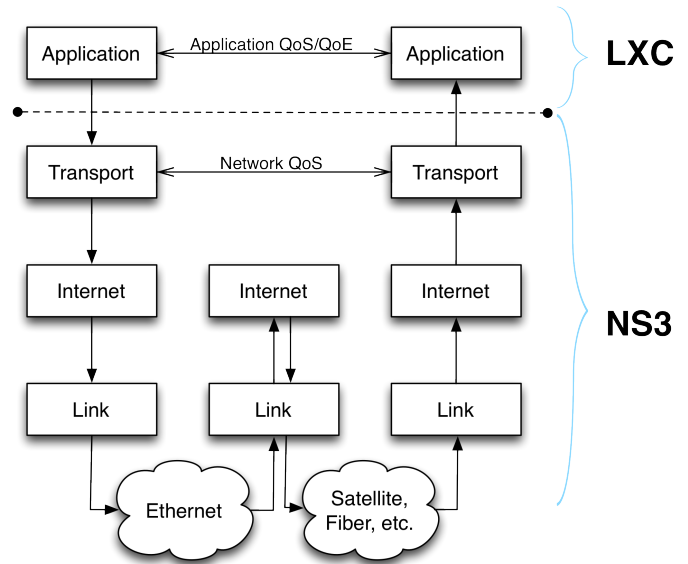


Fig. 1. BoxingExperience separation of concerns with NS3 and LXC

the perceived QoE, as it was suggested for QoS by Vittorio Ghini [6, Chapter 2].

In this paper, we introduce *BoxingExperience*: a modular testbed implemented over NS3 and Linux Containers (LXC) to study multimedia QoS/QoE metrics for different network parameters, maintaining the separation of concerns of the Internet protocol suite. The structure of *BoxingExperience* is illustrated in Fig. 1.

The article is organized as follows: Section II presents the related work. We introduce *BoxingExperience* in Section III followed by our use case: video streaming from network cameras using VLC in Section IV. Finally, conclusions and future work are presented in Section V.

## II. RELATED WORK

Several works already performed research both at the network/transport and application layers. In most cases, the targeted application is multimedia transmission. The goal is generally to obtain a video quality metric which depends on the network state (bitrate, delay, etc).

The first complete framework for evaluating the quality of video transmitted over network was EvalVid [7]. EvalVid

supports several video codecs (MPEG-4, H.264, etc) and the network part is either a real network or a simulated one. The software implements the computation of delay, jitter and loss; allowing to link the video quality in terms of Peak Signal-to-Noise Ratio (PSNR) to the QoS.

Several works combined the framework EvalVid with the network simulator NS2 [8], [9]. The first implementation combining both tools was proposed in [8]. The objective was to replace the simple network loss model used in EvalVid by a more realistic network. Indeed, the NS2 simulator enables simulations over many network scenarios: large topology, wireless network, etc. To interface EvalVid with NS2, the principle is to rely on trace files that enable both tools to communicate with each other. The previous work has been extended to enable simulations of rate adaptive video [9]. Rate adaptation consists in adjusting the encoding bit rate to the network situation in order to achieve better video quality and reduce congestion. Varying the quantization parameter is an efficient way to obtain rate adaptive video. The implementation proposed in [9] also relies on trace files.

More recent works propose to evaluate multimedia transmissions over network with NS3. For instance, an implementation of EvalVid combined with NS3 is available [10]. It offers to NS3 the same functionalities provided to NS2 by [8]. Also, a new QoE evaluation tool has been introduced in [11]. This software is directly implemented in NS3 and it does not use EvalVid for the video part. However, it enables the same kind of multimedia analysis as in previous studies, e.g., the computation of frame-by-frame PSNR in the received video.

Another tool that brings together a multimedia module and a network simulator is the Open Evaluation Framework for Multimedia Over Networks (OEFMON) [12]. The multimedia module is implemented using DirectShow, while the network simulator is implemented in QualNet. OEFMON is a promising tool that provides many features: various codecs, on the fly encoding adaptation to network information, no-reference video quality measurements, etc.

In all the previous works, the considered scenario involves one video server communicating with one video client through a network. However, each framework manages the competing flows in the network in a different way. For instance, the competing flows in [8] are entirely managed by the network simulator, while [9] enables some nodes to transmit rate adaptive video. An important feature of BoxingExperience is that it enables to run simulations with multiple servers and clients, allowing more realistic scenarios.

Finally, some works already proposed to integrate Linux containers. Connecting NS3 with LXC for simulating several applications running in mobile environments has been presented in the work of Zhang and Qin called *TapRouter* [13], and in the work of Skjegstad et al. [14]. Moreover, connecting several network simulators (NS3) that are running inside Linux containers is presented in the work of Calarco and Casoni called *NetBoxIT* [15]. However, the authors did not investigate the application of *NetBoxIT* for multimedia transmissions.

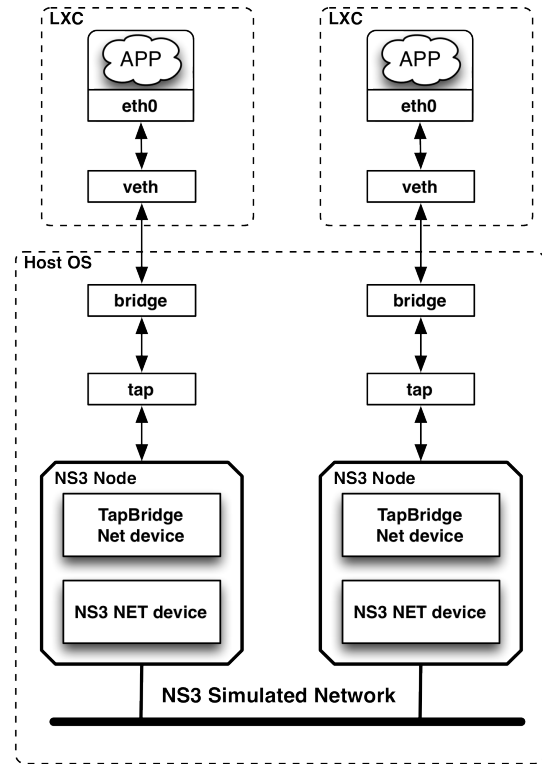


Fig. 2. Connecting NS3 with LXC via virtual ethernet (veth) and tap devices

### III. BOXINGEXPERIENCE: CONNECTING VLC WITHIN LXC'S WITH NS3

In this section, we present the main components of BoxingExperience and show how they are interconnected.

**NS3.** NS3 is an open-source discrete-event network simulator for Internet systems [16]. NS3 has a modular architecture and it relies on a set of libraries distributed from [17]. An API is also available. Libraries belong to different packages:

- **CORE:** generic functionality, such as callbacks, debugging objects, etc;
- **SIMULATOR:** schedules, events, etc;
- **COMMON:** independent objects as packets;
- **INTERNET:** models and protocols related to Internet such as TCP/UDP;
- **NODE:** consist of abstract classes.

NS3 provides realistic network models with the simulated physical/link layers that mimic the behavior of real equipments very strictly, this is why it is widely used in academic research. Also, it natively supports different emulation interfaces, *tap* devices, and real-time simulations where data are treated by the emulator with the same timing they would have when traversing a true network.

**LXC.** Linux container is a user space interface for the Linux kernel containment features [18]. It enables to run multiple isolated Linux systems on a single machine. LXC's rely on several kernel features to contain processes: kernel namespaces (ipc, uts, mount, pid, network and user), AppArmor and SELinux profiles, seccomp policies (a secure computing

mode), chroots (using `pivot_root` in order to change the root file system), kernel capabilities and control groups (cgroups). Among several virtualization techniques (VMware, VirtualBox and LXC), the authors in [15] noticed that LXCs achieve the best performance (best throughput and negligible computing overhead).

As depicted in Fig. 2, the typical infrastructure used to connect NS3 with a set of Linux containers is via NS3 native support of tap devices bridged with container's virtual ethernet interface [13], [14]. This is the basis of BoxingExperience.

**BoxingExperience.** BoxingExperience relies on LXCs connected through NS3. Moreover, as we are interested to study the relation between QoS and QoE for multimedia transmissions, each LXC will run the VLC media player. The VLC media player is a free and open-source software that can be used as a server (VLS) and as a client (VLC) to stream and receive/display multimedia streams [19].

Even if VLC may be directly linked to NS3, the usage of LXCs has two main advantages. First, we can easily change the application running inside the containers. Thus our proposal will also work with applications that cannot be connected directly to NS3. Secondly, LXCs enable to monitor the resource usage (for instance CPU or memory) independently for each container.

Our current application consists in multiple clients connected to a network camera as depicted in Fig. 3. More details concerning the streaming process are given in the next section. In our framework, there is exactly one LXC for each client and one for the network camera. The containers are connected via NS3. The clients run VLC, while the camera runs VLS. For BoxingExperience, we propose to modify the client (VLC) and server (VLS) sides in order to add QoS/QoE metrics and feedback, locating each component of the VLC framework inside a Linux container.

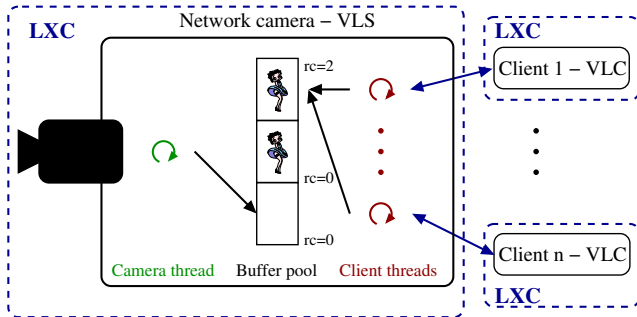


Fig. 3. BoxingExperience in the context of video streaming from a network camera to multiple clients. The arrows between the LXCs represent the connections via NS3.

#### IV. USE CASE: VIDEO STREAMING FROM A NETWORK CAMERA

This section first gives some details about the distribution of video frames from a single network camera. Then, we show a first evaluation of such a system using BoxingExperience. This work is in an early stage and still in progress; we expect to present more results in the near future.

##### A. Use case presentation

We study the distribution of video frames from a network camera to multiple clients. To deliver the frames, the solution considered here relies on a pool of buffers implemented at the server side (the camera) [20]. As shown in Fig. 3, the server involves two parts: a *camera thread* reads the frames from the hardware and writes them into the pool of buffers, while a *client thread* picks a new frame from the pool and sends it to the corresponding client. There is one client thread per connected client and the number of buffers is generally much smaller than the number of customers (due to limited hardware resources at the camera). Also, the camera thread can only write a frame in a buffer that is not currently being read by a client, otherwise it will produce an uncompleted and invalid frame at the client side. Thus, the camera and client threads require to be synchronized to avoid simultaneous access.

The main issue is to offer to the clients with a fast connection a high frame rate when many clients with a slow connection are connected to the camera. Indeed, while slow connection clients read the buffers, the camera cannot write new frames in these buffers and the performance of the fast connection clients decreases. In [20], several algorithms have been introduced in order to manage the buffer pool and offer good performance (in terms of frame rate) to fast connection clients even if the server is under heavy load.

The algorithms proposed in [20] were then implemented into the client and server sides of the VLC Media Player. The performance evaluation is presented in [21]. However, the previous work did not simulate a real network and the relation between QoS and QoE is not available. By using BoxingExperience, we tackle these two limitations.

##### B. Preliminary results

**Simulations setup.** All the simulations have been performed on a Dell OptiPlex 990 Quad-Core i5-2400 with a 3.10 GHz clock, 16 GB RAM running Ubuntu 12.04 LTS (32-bit) as operating system, `ns-3.19`, `lxc 0.7.5`, and `vlc 2.1.0-git` Rincewind.

Before starting simulations, preliminary steps include:

- By now X11 must be installed in LXC in order to be able to watch the video as a normal user;
- Each instance of LXC is managed as a virtual machine running in user space, thus each instance of VLC must be manually started in all LXCs and we suggest to automate connection management with a customized script;
- NS3 is not able to perform real-time simulation when it runs out of resources.

The configuration of BoxingExperience is as follows: there is one LXC for the VLC server streaming the Big Buck Bunny movie. Each transmitted frame has an average size of 400 Kb. The simulations involve  $n + 2$  VLC clients with one LXC associated to each client. All the clients are connected via a simulated network with a 1 Gbps full duplex link. The client with the fastest connection has a speed of 13,600 Kbps that allows a reception of 34 frames-per-second (FPS). The client

with the second fastest connection has a speed of 9,600 Kbps, ensuring a reception of 24 FPS. The  $i$ -th slow connection client ( $1 \leq i \leq n$ ) has a speed of  $(3.5 + 2.0 \times (i - 1)/n) \times 400$  Kbps allowing a reception of  $(3.5 + 2.0 \times (i - 1)/n)$  FPS.

Finally, we added some useful QoS/QoE metrics to VLC client and server, such as framerate and jitter.

**Results.** Fig. 4 illustrates the interface of BoxingExperience with 9 clients connected to one server. The terminal in the upper right corner corresponds to NS3, the terminal in the lower right corner to the server and all other terminals are the clients connected to the camera. For each client, the video is played inside the terminal.



Fig. 4. BoxingExperience interface: 9 clients connected to one server

Concerning the performance (in terms of FPS) of the fastest connection client, the behavior of our simulations was similar to the one obtained in previous works [20], [21].

Finally, we investigate the maximal number of slow connection clients that a simulation can support. We use NS3 with a real-time scheduler in order to lock the simulation clock with the hardware clock. However, NS3 may not keep up with real-time. In that case, two synchronization modes are available: BestEffort or HardLimit. We choose the HardLimit mode that enables a more realistic behaviour. In that mode, the simulation will abort if the simulation time consumes more time than the wall clock time with a certain tolerance threshold. This tolerance threshold corresponds to the HardLimit attribute in NS3 [22]. The maximal number of slow connection clients depends on that parameter as it is presented in Table I. In conclusion, the more realistic is the simulation (i.e., a small value of the HardLimit attribute), the less clients are involved.

TABLE I

BOXINGEXPERIENCE: NS3 SIMULATION LIMIT IN A DESKTOP COMPUTER

NS3 HardLimit attribute	Number of slow connection clients ( $n$ )
100 ms (default value)	42
80 ms	39
60 ms	29
40 ms	19
20 ms	8

## V. CONCLUSIONS AND FUTURE WORK

In this article, we introduced a modular framework to study the relation between QoS and QoE metrics for multimedia transmissions called BoxingExperience. Based on the same separation of concerns as the Internet protocol suite, BoxingExperience relies on Linux containers and open-source softwares: NS3 for network simulation and VLC (inside Linux containers) for streaming/displaying multimedia content. We tested our framework by evaluating the performance of a particular streaming system with BoxingExperience. We concluded that the behavior was similar to the one presented in previous works [20], [21].

As future work, we plan to extend the API of VLC in order to increase and improve QoS/QoE metrics and statistics. Also, we plan to investigate how BoxingExperience behaves in other simulated or emulated environments. For instance, if a massive content distribution is to be tested, it could be useful to measure QoS/QoE when the distribution is made through different networks (broadband, mobile, etc). Then, all simulations could be executed in a multicore high-level computer using NetBoxIT [15] (see Fig. 5(a)), or they can be distributed in a local network environment through network routers (see Fig. 5(b)). Finally, we would like to study if our framework could be executed in mobile devices, in order to test in a controlled environment the impact of network's variations in the perceived mobile QoE.

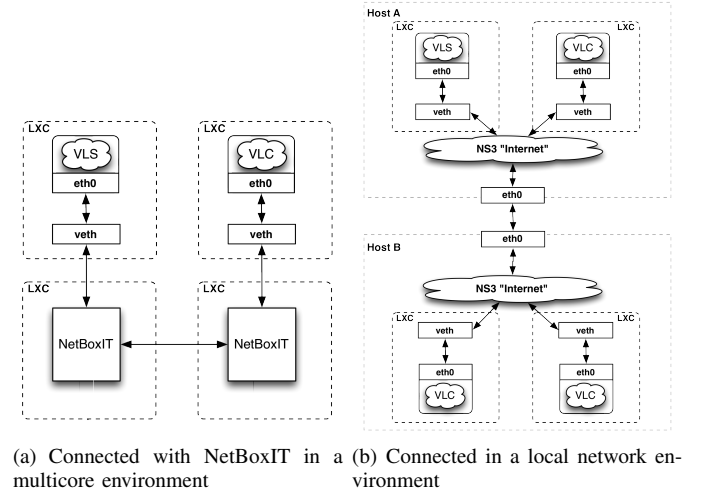


Fig. 5. Several ways to connect BoxingExperience

## REFERENCES

- [1] I. T. Union, "Vocabulary and effects of transmission parameters on customer opinion of transmission quality, amendment 2," ITU-T Recommendation P.10/G.100, Tech. Rep., 2007.
- [2] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From packets to people: quality of experience as a new measurement challenge," in *Data Traffic Monitoring and Analysis*. Springer, 2013, pp. 219–263.
- [3] K. Piamrat, C. Viho, J. Bonnin, and A. Ksentini, "Quality of experience measurements for video streaming over wireless networks," in *Proceedings of Sixth International Conference on Information Technology: New Generations*. IEEE, 2009, pp. 1184–1189.

- [4] J. Pokhrel, B. Wehbi, A. Morais, A. Cavalli, and E. Allilaire, "Estimation of QoE of video traffic using a fuzzy expert system," in *Proceedings of Consumer Communications and Networking Conference (CCNC)*. IEEE, 2013, pp. 224–229.
- [5] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2011, pp. 485–492.
- [6] V. Ghini, "QoS-adaptive middleware services," Ph.D. dissertation, University of Bologna, Italy, 2001.
- [7] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid - a framework for video transmission and quality evaluation," in *Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 2003, pp. 255–272.
- [8] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, "An evaluation framework for more realistic simulations of MPEG video transmission," *Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 425–440, 2008.
- [9] A. Lie and J. Klaue, "Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video," *Multimedia Systems*, vol. 14, no. 1, pp. 33–50, 2008.
- [10] GERCOM. [Online]. Available: <http://gercom.ufpa.br/>
- [11] D. Saladino, A. Paganelli, and M. Casoni, "A tool for multimedia quality assessment in NS3: QoE Monitor," *Simulation Modelling Practice and Theory*, vol. 32, no. 0, pp. 30–41, 2013.
- [12] C. Lee, M. Kim, S. Hyun, S. Lee, B. Lee, and K. Lee, "OEFMON: An open evaluation framework for multimedia over networks," *Communications Magazine, IEEE*, vol. 49, no. 9, pp. 153–161, 2011.
- [13] J. Zhang and Z. Qin, "Taprouter: an emulating framework to run real applications on simulated mobile ad hoc network," in *Proceedings of the 44th Annual Simulation Symposium*. International Society for Computer Simulation, 2011, pp. 39–46.
- [14] M. Skjegstad, F. T. Johnsen, and J. Nordmoen, "An emulated test framework for service discovery and manet research based on ns-3," in *Proceedings of the 5th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2012, pp. 1–5.
- [15] G. Calarco and M. Casoni, "On the effectiveness of Linux containers for network virtualization," *Simulation Modelling Practice and Theory*, vol. 31, pp. 169–185, 2013.
- [16] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, 2008.
- [17] NS3. [Online]. Available: <http://www.nsnam.org>
- [18] Linux Containers. [Online]. Available: <https://linuxcontainers.org/>
- [19] VLC. [Online]. Available: <http://www.videolan.org/vlc>
- [20] J. Bustos-Jiménez and J. M. Piquer, "Frame allocation algorithms for multi-threaded network cameras," in *Proceedings of the 16th International Euro-Par Conference on Parallel Processing: Part I*. Springer, 2010, pp. 560–571.
- [21] G. Muñoz, H. Méric, J. Piquer, and J. Bustos-Jiménez, "Performance evaluation of streaming algorithms for network cameras," in *Proceedings of IEEE INFOCOM Workshop on Communication and Networking for Video*. IEEE, 2014, pp. 1–6.
- [22] NS3 Manual. [Online]. Available: <http://www.nsnam.org/docs/release/3.11/manual/html/realtime.html>