

# A virtualized platform for analyzing LTE broadcast services

Carlos M. Lentisco<sup>†</sup>, Miguel Aguayo<sup>†</sup>, Luis Bellido<sup>†</sup>, Encarna Pastor<sup>†</sup>,

David De-Antonio-Monte\*, Alejandro García Bolívar\*

<sup>†</sup>Department of Telematics Engineering

Universidad Politécnica de Madrid

\*Alcatel-Lucent

Email: clentisco@dit.upm.es

**Abstract**—Nowadays, users are demanding access to video content anytime and anywhere. LTE broadcast services and video streaming services can reduce the resources used by telecommunication companies to give access to video content and open up new business opportunities. In this context, testbeds and simulation tools can help in the analysis of the different parameters that need to be considered in the deployment of video services based on LTE. In this paper, we describe the design and implementation of an open-source virtualized platform that supports both LTE broadcast services and video streaming services. The testbed is being used both over real LTE networks and LTE network emulations to analyze the performance of video streaming services and to validate new proposals.

## I. INTRODUCTION

The high demand of video streaming services anytime and anywhere poses some significant challenges. In particular, the distribution of video in Long Term Evolution (LTE) networks presents some issues such as bandwidth consumption or a radio channel with changing conditions. In this context, experimental platforms can be used to facilitate the analysis of a video streaming service.

The 3rd Generation Partnership Project (3GPP) defines the use of Dynamic Adaptive Streaming over HTTP (DASH) [1] as the standard for LTE multimedia broadcast and streaming services. DASH works on top of HTTP, allowing a client to adapt to changes in available bandwidth by switching between different representations of the same content encoded with different bitrates. Each representation is subdivided in several segments of the same duration, each identified by a different Uniform Resource Locator (URL). The definition of the different representations and segments is stored in a Media Presentation Description (MPD) file, that is downloaded by the client prior to the establishment of the DASH session. After parsing the MPD file, the client can start to request, sequentially, all the segments that compose the multimedia presentation.

However, this solution, based on TCP sessions, is not scalable when a mass audience is requesting simultaneously the same video. In order to efficiently deliver the same multimedia content to multiple receivers, 3GPP has defined Evolved Multimedia Broadcast and Multicast Service (eMBMS) [2], that uses a unidirectional common channel on which the data can be sent to multiple recipients. The 3GPP has also defined how to send the video content over eMBMS. The solution is based on the use of the File Delivery over Unidirectional

Transport (FLUTE) protocol [3], that works on top of UDP. The FLUTE protocol is used to send the DASH video segments to the recipients using IP multicast. Video segments that are lost can be retrieved directly using unicast connections.

Before the deployment of an LTE video service, an analysis of the different parameters that have an impact on the service can be crucial for its success. While this analysis can be done using simulation tools, such as [4], testbeds provide a useful tool that can bring the subsequent analysis closer to the real service. There are several works that describe testbeds used to emulate video streaming services in mobile networks. In [5], the authors describe a real-time testbed used to assess the end-to-end performance of multimedia services in an LTE network, but the analysis is based on video streaming over RTP instead of the current 3GPP proposal based on DASH. Other works [6] [7], follow an approach based on a simulation tool that emulates a video streaming service over a simulated LTE network. However, they do not simulate the FLUTE transmission of the video segments over a multicast channel. Finally, other works are based on different wireless networks such as Wi-Fi or 3G. For example in [8] the authors detail an experimental testbed that is used to evaluate streaming services in terms of video processing, audio and video coding, packetization, buffering, or transmission link.

In this paper, we present a virtualized platform for analyzing both unicast and multicast video streaming services over real or emulated LTE networks. The rest of the paper is organized as follows. In Section II, the main components that integrate the testbed are described. The deployment of two different experimentation scenarios over the testbed is detailed in Section III. A methodology that shows how to perform an analysis of a video streaming service, and how a proposal can be validated by using the testbed is presented in Section IV. Finally, in Section V, the conclusions are explained.

## II. A VIRTUALIZED PLATFORM TO ANALYZE MULTIMEDIA SERVICES IN LTE NETWORKS

This section describes the main components of a platform to support the analysis of multimedia services in LTE. By using open-source components and virtualization technologies, it has been possible to create a flexible platform that is easy to deploy and that can be configured and used in different ways to perform different kinds of analysis, both over an LTE network and over an LTE network emulation.

One of the main characteristics of our platform is that it is built over a virtualized environment based on Linux. Virtualization is managed using the open-source tool Virtual Networks over linux (VNX) [9], that gives high flexibility to our testbed, allowing us to easily deploy different virtual scenarios. The deployment of a scenario is performed by the use of a XML file, which defines the virtual machines that compose it, and how they are interconnected to each other. With VNX it is possible to create virtual scenarios with n-servers interconnected to m-clients. In the XML file, it is also possible to define commands that control the virtual machines, for example, it is possible to define a command that initiates a DASH session between an LTE terminal client and a content server. In order to facilitate the deployment of these scenarios over machines with different operating systems, another layer of virtualization can be used. We have encapsulated the whole testbed in a virtual machine created with VirtualBox. This provides portability to our testbed, since it is possible to take it easily to anywhere through a Open Virtual Appliance (OVA) file.

Below, we describe the different components that are included in the virtualized platform, which consists of three virtual machines: an LTE terminal client, a traffic shaper and a content server. For our analysis, we have combined these components to create two possible scenarios: (i) a fully virtualized scenario connecting the LTE terminal client and the content server through the traffic shaper; (ii) a virtualized client and a server, connected through a real network. The detail of how to deploy these scenarios is shown in the following section.

#### A. LTE terminal client

An LTE terminal client is composed of a DASH client, a FLUTE client and a intermediate HTTP cache. In the testbed, the LTE terminal client has been divided in two parts: a virtual machine executing the FLUTE client and an HTTP cache, and an external DASH client running outside of the virtual environment of VNX that requests the video content. This makes it possible to perform experiments by running a DASH client from different kinds of terminals (e.g., smartphones or tablets).

As a FLUTE client, we are using the MAD-FCL package, initially developed within the framework of the MAD project [10]. MAD-FCL is available for both Linux and Windows, and it supports unicast and multicast UDP transmissions. As an HTTP cache, we have included a standard Squid server. Finally, we have included two different DASH clients in the platform. The first one is DASH-JS, written in JavaScript and offered by the DASH Industry Forum (DASH-IF) [11]. The second is a VLC client running a DASH plugin [12].

#### B. Content server

The content server is responsible for storing and delivering the multimedia content to the users. Some videos have been pre-stored in the current version of the testbed in order to carry out various experiments, but it is possible to upload additional videos to the content server. The videos are encoded in H.264 using the x264 tool [13] at a resolution of 720p and several different bitrates. To create the video segments,

we have used MP4Box [14], that splits the video in video segments of a specific duration, and also generates the MPD of the multimedia presentation.

For the delivery of the DASH video segments over the multicast channel, the platform contains a FLUTE server, also provided by the same MAD-FCL package that is used for the FLUTE client. The content server includes also an Apache web server to provide access to the video segments using HTTP.

#### C. Traffic shaper

The traffic shaper provides the possibility of emulating a LTE network interconnecting content servers and LTE terminal clients. We have included scripts to modify the network conditions in terms of bandwidth, delay, packet lost and traffic type. The scripts are based on the use of a Class-based queuing (CBQ) discipline that classifies the incoming traffic, and the Linux Traffic Control (TC) tool together with the Network Emulation (NetEm) tool to control bandwidth, delay and losses. A simple text file can be used to specify the evolution of the network parameters throughout an experiment. Different parameters can be specified for TCP and UDP traffic. This is useful to specify the different bandwidth and delays that the multicast channel and the unicast channel would have in an LTE network.

### III. EXPERIMENTAL SCENARIOS SUPPORTED BY THE TESTBED

The different components of the platform have been combined in two different scenarios. A fully virtualized scenario connects an LTE terminal client and a content server through a simulated LTE network. The other scenario connects a virtualized client and a server, through a demonstration LTE network. Both scenarios, deployed with VNX, run in VirtualBox machines with a light version of the Ubuntu Linux distribution. The following is a description of the deployment and usage of each of these scenarios.

#### A. Client and server connected through an LTE network emulation

This scenario (Fig. 1) consists of a content server and an LTE terminal client interconnected by a traffic shaper that simulates the behavior of an LTE network. The definition and deployment of this scenario is based on a root filesystem, based on Linux Container (LXC), that includes all the software

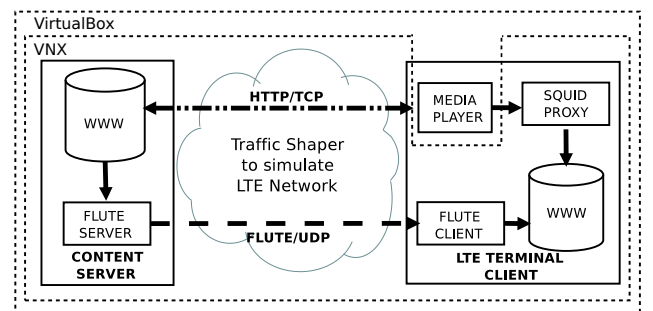


Fig. 1: Virtual LTE network scenario

components and the multimedia content. Using VNX, the same root filesystem can be used as an LTE terminal, a content server or a traffic shaper.

A single VNX command creates the scenario:

```
$ sudo vnx -f ltextreme_dash_multicast.xml -v --create
```

Once created, the user can perform various tasks on the testbed, such as establishing a DASH session, or starting the FLUTE server to send the DASH video segments to the multicast users. These tasks are also defined using VNX. A task can be also defined as a combination of other different tasks, which can be useful, for example, to establish a multicast session first and then booting a DASH player.

To sum up, we have defined several commands to control the behavior of the platform. Some of these commands are listed below:

- Start a DASH client (VLC or DASH-JS).
- Start a FLUTE session between the client and the server.
- Initiate the traffic shaping.
- Start a web browser.
- Clear the received multimedia content in the cache of the client.

#### B. Client and server connected through a demonstration LTE network

This scenario consists of a client and a server connected through a demonstration LTE network. This LTE network is composed of an Evolved Packet Core (EPC) and a radio access network. The EPC includes all the control and management components of a real LTE network and a gateway that can be used to access the Internet. The radio access network consists of two Evolved Node B (eNB)s, each of them with two antennas. The first eNB is used to transmit the data to an LTE terminal, and the second eNB is used to generate an interference signal. Combining the interfering eNB and a signal attenuator it is possible to simulate different conditions of the radio channel.

The LTE client and content server components of this scenario are deployed again using VNX on top of VirtualBox machines. In this case, however, because of the need of interconnecting the demonstration LTE network, the previous scenario was split in two, one to run the LTE terminal client, and the other to run the content server. Each scenario is defined in a specific OVA file to facilitate the deployment. Thus, it is possible to start the client on a PC, and the content server on

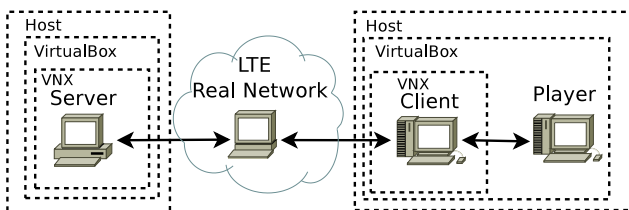


Fig. 2: Architecture for a client and a server connected through a real LTE network

another. In this case, the client is deployed on a Windows system and the content server on a Macintosh system. A representation of this scenario is shown in Fig. 2.

In this scenario, the server is connected to the Packet Data Network Gateway (P-GW) element in the EPC via Ethernet. The LTE terminal client is connected to the LTE network using a USB LTE modem. The use of virtualization makes it easy to use the USB LTE modem, for which only Windows drivers are available. However, in order to ensure the connectivity between the virtualized client and server, and due to the use of the double layer of virtualization, it is necessary to modify the network configuration of the virtual environment. A Network Address Translation (NAT) service in the client allows access to the connection of the USB LTE modem, and by adding port forwarding rules to redirect the traffic it is possible to send the FLUTE traffic directly to the client deployed with VNX.

While using two layers of virtualization facilitates in this case the deployment of complex scenarios, a possible issue can be a decrease in performance. However, in our experiments, this decrease in performance is not an issue. The tests show that the available bandwidth between the client and the server exceeds the requirements of a multimedia service for the video qualities that were used in the experiments.

#### IV. METHODOLOGY TO EVALUATE A MULTIMEDIA SERVICE OVER AN LTE NETWORK

This section provides a methodology to evaluate a multimedia service over an LTE network by using the testbed described previously. This methodology, depicted in Fig. 3, proposes to use the measurements and analysis of results from the platform interconnected through a real LTE network as

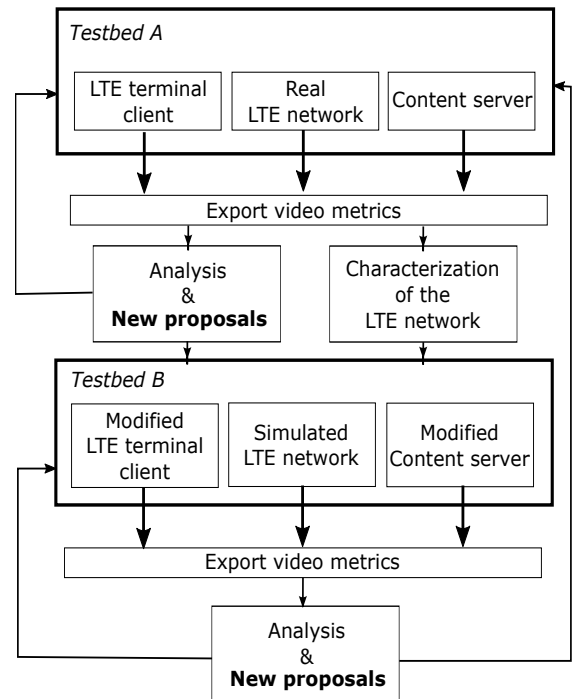


Fig. 3: Methodology to evaluate a multimedia service over a real LTE network by using the testbed

inputs to additional tests and analysis over the simulated LTE network. The simulated LTE network can then be used to work on new proposals regarding the distribution of multimedia content over LTE, which could again be validated over the real LTE network. To make this possible, it is necessary to monitor several video metrics, so as a part of the development of the platform, we have taken into account the different possibilities to extract these metrics. For example, the DASH-JS client has been modified to obtain a log of the buffer level or the video representation selected by the adaptation algorithm. In addition, we have also included tools to test the available bandwidth between the LTE terminal client and the content server, and tools to monitor the traffic.

A possible workflow would start by configuring the platform interconnecting the client and the server to a real LTE network (*Testbed A* in Fig. 3). The experiments can then be carried out and the results, (i.e., video metrics) can be exported from the platform to be analyzed. This first phase can already be useful to better understand a video streaming service over a LTE network. Moreover, the analysis can lead to new proposals, that can also be validated by using the testbed. To this end, further modifications to the platform in order to support the new features might be necessary (*Testbed B* in Fig. 3). Note that the validation of the modifications can be carried out initially over the simulated LTE network. Through the results previously extracted during the experiments, the network can be fully characterized. Both the bandwidth, delay and losses can be used by the traffic shaper to simulate the same real conditions. Once a new feature is tested using the traffic shaper based platform, the experiments can be carried out again, this time, over the real LTE network and a modified testbed with the added features.

As an example, we are going to show a part of an on-going work that uses this methodology. We have carried out several experiments by using the *Testbed A*, i.e., with a demonstration LTE network connecting the client and the server. The experiments consist of the establishment of DASH

unicast sessions of 5 minutes long with different radio channel conditions. Fig. 4 shows as an example the results obtained during one of the experiments. During this experiment, we have fixed some system parameters of the LTE network, such as the number of Physical Resource Block (PRB)s or the power emitted by the eNBs. The figure shows both the client buffer level (y-axis on the left), and the representation index requested by the DASH client (y-axis on the right). Indexes 1, 2, 3 and 4 correspond to video representations coded at bitrates of 250, 500, 1000, and 2000 kbps respectively. The figure shows how the DASH client is switching between different representations, due to the changing conditions of the radio channel, even if the network parameters are fixed. This can be caused by the DASH client using an adaptation algorithm that is not adapted to the characteristics of the LTE radio channel, in which packet losses are not caused by congestion but by transmission errors.

We are currently working on the analysis of application level and transport level alternatives taking into account the changing conditions that may affect the LTE radio channel and plan to use the methodology explained above for its validation.

## V. CONCLUSION

In this paper, we have presented an open-source virtualized platform providing a testbed to analyze LTE broadcast services and multimedia streaming services. By using open-source software components we have been able to create this new platform in accordance to the latest 3GPP proposals for LTE multimedia services.

We are using the platform on an on-going work to analyze the impact of the network conditions on the quality of a multimedia service over LTE. Our methodology combines experiments over a real LTE network and over LTE network emulations, and both scenarios are supported by the platform thanks to its flexibility and ease of deployment. We have already identified some issues such as the problems for current DASH clients to the changing conditions of the radio channel and we are currently addressing possible solutions at the application and transport level.

## ACKNOWLEDGMENT

This work was supported in part by the Spanish Ministry of Economy and Competitiveness, National Plan for Scientific Research, Development and Technological Innovation (IN-NPACTO subprogram), LTeXtreme project (IPT-2012-0525-430000).

## REFERENCES

- [1] ISO, "Information technology - Dynamic Adaptive Streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," 2014, ISO/IEC 23009-1:2014.
- [2] "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs", 3GPP TS 26.346 v11.6.0, Sep 2013.
- [3] T. Paila, R. Walsh, M. Luby, V. Roca, and R. Lehtonen, "Flute — file delivery over unidirectional transport," Internet RFC 6726, November 2012.
- [4] C. Lentisco, L. Bellido, A. Fuente, E. Pastor, R. Leal, and A. G. Armada, "A model to evaluate mbsfn and al-fec techniques in a multicast video streaming service," in *Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014 IEEE 10th International Conference, Oct 2014.

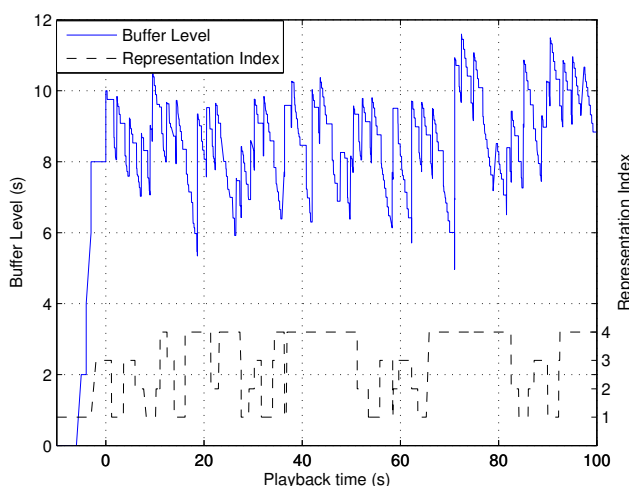


Fig. 4: Buffer level and representation index during the playback time

- [5] G. Gómez, J. Poncela-Gonzalez, M. Aguayo-Torres, and J. Entrambasaguas, "A real-time end-to-end testbed for evaluating the performance of multimedia services," in *Future Multimedia Networking*, ser. Lecture Notes in Computer Science, A. Mauthe, S. Zeadally, E. Cerqueira, and M. Curado, Eds., vol. 5630. Springer Berlin Heidelberg, 2009, pp. 212–217.
- [6] D. Munaretto, M. Zanforlin, and M. Zorzi, "Performance evaluation in ns-3 of a video delivery framework for next generation cellular networks," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, June 2013, pp. 642–646.
- [7] Q. Zheng, H. Du, J. Li, W. Zhang, and Q. Li, "Open-lte: An open lte simulator for mobile video streaming," in *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*, July 2014, pp. 1–2.
- [8] B. Lewcio and S. Moller, "A testbed for qoe-based multimedia streaming optimization in heterogeneous wireless networks," in *Signal Processing and Communication Systems (ICSPCS), 2011 5th International Conference on*, Dec 2011, pp. 1–9.
- [9] "Virtual Networks over linux (VNX)," <http://dit.upm.es/vnxwiki/>, [Online], 2015.
- [10] "MAD Project's Home Page," <http://mad.cs.tut.fi/>, [Online], 2015.
- [11] "DASH-IF forum website," <http://dashif.org/>, [Online], 2015.
- [12] "VideoLan organization website," <http://www.videolan.org/vlc/>, [Online], 2015.
- [13] L. Merritt and R. Vanam, "x264: A high performance h. 264/avc encoder," *online* [http://neuron2.net/library/avc/overview\\_x264\\_v8\\_5.pdf](http://neuron2.net/library/avc/overview_x264_v8_5.pdf), 2006.
- [14] "GPAC Multimedia Open Source Project website," <http://gpac.wpmines-telecom.fr/mp4box/>, [Online], 2015.