2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies

Supporting handover between LTE video broadcasting and unicast streaming

Luis Bellido, Carlos M. Lentisco, Miguel Aguayo, Encarna Pastor
Department of Telematics Engineering
Universidad Politécnica de Madrid
Email: lbt@dit.upm.es

***Abstract*—The heterogeneity of the network poses new challenges on the paradigm of multimedia content distribution. Nowadays, different access technologies coexist in user terminals, so it is possible to improve the performance or avoid coverage problems by switching from one to another. This process is known as handover. In this work we have focused on vertical handover, which is the process of switching between different access technologies without losing connectivity. If a handover happens during a video streaming session, the session will need to adapt to the different channel conditions. This paper discusses the proposal of a new mechanism at the application level to support a handover between a unicast video streaming session and a video session based on LTE broadcasting.**

I. Introduction

Nowadays, the demand of services through different wireless networks is increasing. Due to their mobility, users may experience service interruptions. This issue is specially relevant in the case of video streaming applications, in which a continuous traffic needs to be transmitted. Both Video on Demand (VoD) and Live streaming services are affected, but the latter can be more sensitive to service disruptions because of possibly more strict delay requirements.

A handover is a way to address service interruptions. A horizontal handover is the process of switching between different cells using the same access technology. A vertical handover is the process by which a terminal will change the access network from one technology to a different one [1], for example from a Long Term Evolution (LTE) access using the cellular network to a WiFi access point. In this work, we are focusing on how to take advantage of a handover to avoid service interruptions in the case of a video streaming service.

Even if there are different streaming technologies, this paper is considering a service based on Dynamic Adaptive Streaming over HTTP (DASH) [2], which has been selected by the 3rd Generation Partnership Project (3GPP) as the standard for multimedia streaming. DASH is a set of technologies that are put together with the initial idea of streaming video over the HTTP, allowing a video streaming client to adapt to changing network conditions by switching between different representations of the same video content that have been encoded with different bitrates. DASH identifies different segments of a video representation by their Uniform Resource Locator (URL), and the definition of the different representations and segments is stored in a Media Presentation Description (MPD) file that is downloaded by the video streaming client in order to learn how the multimedia content has been prepared and how it can be accessed. Using the information from the MPD file, the streaming client will apply an adaptation algorithm to play the video stream choosing the video segments with the more adequate bitrate, taking into account an estimation of the available bandwidth.

While DASH as defined above is useful when a user is accessing streaming services over-the-top of an LTE cellular network, or a video-on-demand service provided by the telecom operator, the 3GPP also defines how DASH can be used to provide a video broadcasting service over LTE. This service is based on the standard data service for multicast/broadcast defined by 3GPP, the Evolved Multimedia Broadcast and Multicast Service (eMBMS) [3]. This data service uses a common multicast channel to send the same data to multiple receivers, thereby improving the efficiency in the use of network resources. Thus, in the video broadcasting service over LTE, DASH video segments are sent from the network to the terminals using multicast, so in this case all the video streaming clients receive the same content. Different representations of the same content at different bitrates are not needed because the sender just needs to send a single representation over a multicast channel that is pre-allocated for this communication.

The question that arises is how this kind of services, provided by the telecom operator, can continue to be accessed when there is a vertical handover. The discussion of how or when a terminal switches to a different access technology is out of the scope of this work, but it is common, for example, that terminals switch their data access from LTE to a WiFi access point when the WiFi access point is detected, with the goal of obtaining a service with less cost or to save battery. Video session continuity is studied in [4], but the aim in that case is to use session rate prediction to avoid video freeze. However, taking the decision of performing a handover might mean to lose (or gain) access to the video stream broadcast through eMBMS, and to the best of our knowledge there is not a mechanism to allow a video streaming application to handle a vertical handover moving between a multicast session and an adaptive unicast session in LTE networks.

In this paper, we are proposing a mechanism at the application level to support a seamless handover from/to a video stream that is broadcast over an LTE cellular network to/from a Wi-Fi/Femto access point from which the terminal retrieves the video stream using unicast. The rest of the paper is organized as follows. In Section II, the scheme used to send DASH segments over a multicast service is detailed. The LTE architecture for multimedia content distribution is then explained in Section III. The mechanism proposed to

978-1-4799-8660-6/15 $31.00 © 2015 IEEE
DOI 10.1109/NGMAST.2015.59

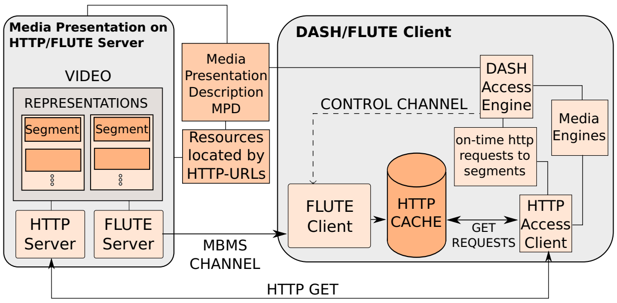


Fig. 1: Hybrid FLUTE/DASH architecture

support vertical handovers in video streaming over LTE at the application level is described in Section IV. Section V explains the experimental results validating the proposal. Finally, our conclusions are summarized in Section VI.

II. Hybrid FLUTE/DASH architecture

The 3GPP defines how DASH can be used to provide a video broadcasting service over LTE using eMBMS, replacing the unicast transmission of video segments over HTTP for a multicast transmission. Since the data is sent using multicast, and the eMBMS channel has a non-negligible error rate, Application Layer - Forward Error Correction (AL-FEC) techniques are used to recover from lost packets. File Delivery over Unidirectional Transport (FLUTE) [5], a protocol initially defined for a unidirectional delivery of files over the Internet, is the protocol proposed by the 3GPP to encapsulate both video segments, encoded following the DASH standard, and the AL-FEC redundancy.

FLUTE uses an eXtensible Markup Language (XML) metadata file describing the properties of the objects transmitted and the properties of the session. This file is called File Delivery Table (FDT) and is transmitted as an object of the FLUTE session. New updates of the FDT instances are sent to the receiver, so it can be aware of objects that are created dynamically. This characteristic makes it appropriate not only for video on demand, in which video segments are pre-encoded, but also for live streaming services, in which video segments are created on-the-fly.

Despite of the use of Forward Error Correction (FEC) techniques, a video segment sent with FLUTE over eMBMS might not be recovered by a client receiving the video. For example, when using Raptor codes [6], the video segment and the redundancy is packetized. If the error rate during the transmission of the video segment is high it might happen that not enough packets are received to recover the video segment. To provide a solution to this, the 3GPP proposes to use a hybrid FLUTE/DASH architecture (Fig. 1), in which video segments that are lost can be retrieved using an HTTP unicast access. The solution also facilitates the integration of standard DASH Access Engines that work using HTTP.

Fig. 1 shows that video segments are sent using FLUTE over a multicast session from the server to the client. The FLUTE client on the terminal receives the packets, decodes the video segment using the AL-FEC, and writes the video segment to an HTTP cache. The DASH Access Engine, through the use of the HTTP Access Client, can obtain the video segment from the HTTP Cache when the video segment has been received correctly via the FLUTE multicast session or from previous unicast transmissions. If the video segment is not in the cache, then it is retrieved using HTTP and stored in the cache. Therefore, this hybrid FLUTE/DASH architecture provides a solution in which the use of FLUTE and multicast is transparent to the DASH Access Engine, and in which the recovery of a lost video segment using HTTP is automatically done by the HTTP Access Client when the video segment needed by the DASH Access Engine is not found in the HTTP cache.

III. LTE architecture for multimedia content distribution

In an LTE video broadcast, a single video representation is sent to all the terminals using eMBMS. This video representation needs to be coded at a bitrate that depends on the bandwidth of the allocated multicast channel and on the chosen AL-FEC redundancy, among other parameters [7]. The video segments and the corresponding MPD file will be sent through eMBMS using FLUTE as explained previously, but they also need to be stored on a web server to provide error recovery through HTTP.

Fig. 2 shows the architecture for multimedia content distribution over LTE networks. Firstly, it shows that clients can connect to a multimedia content server that stores the DASH multimedia presentation. Additionally, if a Content Delivery Network (CDN) is deployed, clients can connect to one of its replicas directly, where a replica is a server that stores a duplicate of the video content already prepared. These replicas can be strategically located in the network to reduce the latency in the access to the video content.

In order to send a video broadcast session over LTE, the multimedia content is first delivered to the Broadcast Multicast Service Centre (BM-SC), the entity in LTE that distributes the content towards the Multimedia Broadcast Multicast Services Gateway (MBMS-GW). The BM-SC also provides protection against errors to the content sent over eMBMS and can act as a server for file error repair, providing an HTTP access interface to the terminals [8]. The MBMS-GWs forwards the content to the Evolved Node B (eNB)s belonging to a MBSFN area and then the eNBs forward the content to the terminals over the multicast channel. When the AL-FEC fails to recover from an error, the lost video segment is recovered by HTTP. Since the use of HTTP requires establishing a unicast channel, these video segments traverse a different path. A lost video segment enters the LTE network through the Packet Data Network Gateway (P-GW). Then, it is sent via the Serving Gateway (S-GW) to the nearest eNB to the user, which in turn, sends the lost video segment over a unicast radio channel.

Fig. 2 shows also other possible access points. The first one is a Femtocell access point, also known as Home eNodeB (HeNB). In LTE-Advanced (LTE-A), it is possible to connect directly to the Internet through the Local Gateway (L-GW) [9], so the terminal can establish a connection to another server in the CDN without going through the core of the mobile network. A terminal could also connect to a WiFi access point.

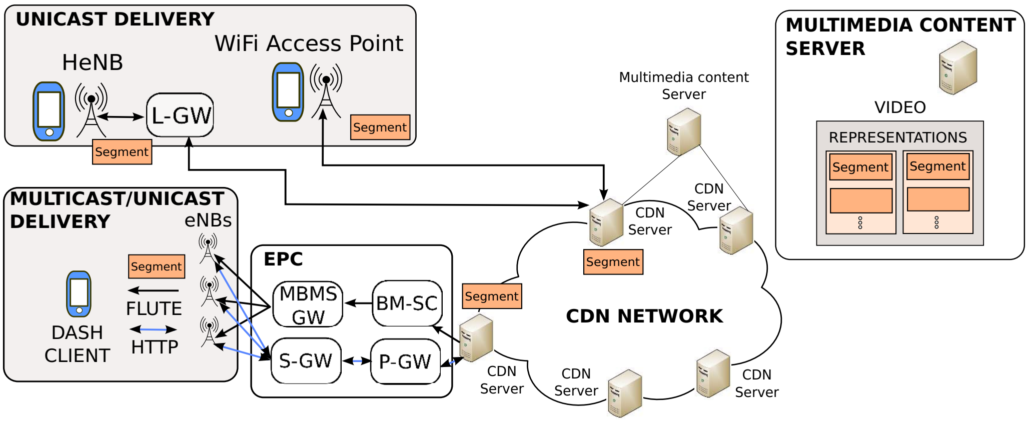


Fig. 2: Architecture for content distribution in LTE networks

Different CDN servers might be used depending on the access point, so a handover, from example from LTE to WiFi/Femto, might also involve an access to a different CDN server in order to obtain the lowest possible latency.

Fig. 3 shows possible vertical handover scenarios in LTE networks. We have illustrated two kinds of users: (i) users that are moving to/from a Wi-Fi/Femto access point and that would therefore switch between multicast sessions over eMBMS and pure unicast connections over WiFi/Femto; (ii) users that would switch between multicast sessions and LTE unicast sessions.

IV. Application level support for multicast and unicast video streaming services

In a multicast streaming service over LTE, a terminal receives a single representation of a video coded at a bitrate that depends on the multicast channel bandwidth and the AL-FEC redundancy. However, in a unicast streaming service several representations of the video, each with a different bitrate, are preferred. This allows each terminal to use a dynamic adaptation algorithm to retrieve one representation or another depending on the available bandwidth. To support video session continuity during a handover, it is necessary to address the possibility of switching between a multicast streaming and a unicast streaming. Besides, a mobile operator giving access to a video content not only through a broadcast session, but also through unicast over different access technologies, can increase the quality of the service and at the same time offload the core of the mobile network.

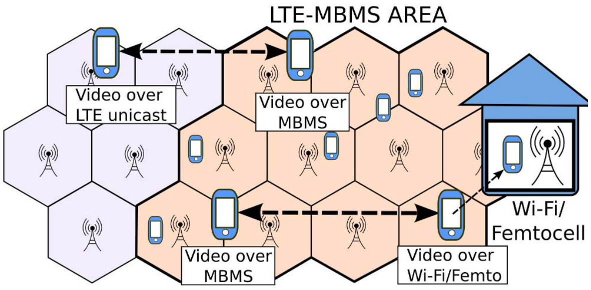


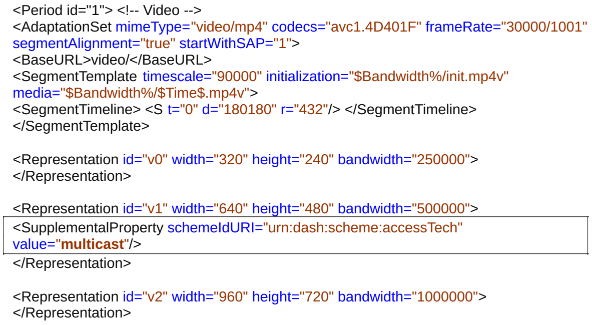
Fig. 3: Handover scenarios between eMBMS and unicast

At the application level, one issue is the access to the MPDs including the information about the video representations needed in each case. It would be possible to have a different MPD file for multicast and unicast. The multicast MPD file would contain just a single video representation and the unicast MPD file would contain the different available representations. The video player would choose one or the other depending on the availability of the multicast channel. From the point of view of the video player, a handover would simply mean using a different MPD for the video. However, in order to minimise the impact of a handover, the video player would need to load both MPD files at the beginning of the session, and keep both updated if they change. At the same time, there would be some information about the media that would be duplicated, i.e., it would be present in both MPD files.

Our proposal is to use just one MPD file containing all the video representations, adding new information to mark the representation that is used for multicast. In this case, the video player needs to be able to parse and process this new piece of information. Then, when accessing the unicast streaming, the video player can use the regular adaptation algorithm to retrieve representations according to the available bandwidth. But when accessing the multicast streaming, the video player needs to stop using the adaptation algorithm and start retrieving the representation that is marked as multicast. The following subsections detail how the information about a multicast representation can be included in the MPD file and how an experimental platform including an adapted video player has been developed to assess the validity of the proposal.

A. MPD multicast video representation

Previous versions of DASH did not support the addition of new information elements, so adding the information about what video representation is used for multicast would have meant to define new attributes for the MPD out of the standard. However, with the new revision of DASH [2], it is possible to use the new element *SupplementalProperty* that is specifically



```
<Period id="1"> <!-- Video -->
<AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F" frameRate="30000/1001"
segmentAlignment="true" startWithSAP="1">
<BaseURL>video/</BaseURL>
<SegmentTemplate timescale="90000" initialization="$Bandwidth%/init.mp4v"
media="$Bandwidth%/$Time$.mp4v">
<SegmentTimeline> <S t="0" d="180180" r="432"/> </SegmentTimeline>
</SegmentTemplate>

<Representation id="v0" width="320" height="240" bandwidth="250000">
</Representation>

<Representation id="v1" width="640" height="480" bandwidth="500000">
<SupplementalProperty schemeIdURI="urn:dash:scheme:accessTech"
value="multicast"/>
</Representation>

<Representation id="v2" width="960" height="720" bandwidth="1000000">
</Representation>
```

Fig. 4: Example of the *accessTech* tag in an MPD file

defined to support new functionality. *SupplementalProperty* specifies additional information about the containing element. This element is defined by a mandatory attribute that identifies the scheme (@schemeIdUri) and an optional attribute that specifies its value (@value). We propose to define a new scheme called *accessTech* that can be used to specify what video representation is delivered over eMBMS.

Fig. 4 shows a code snippet of an MPD file for a video that is broadcast over eMBMS, on which communication resources are pre-allocated. In the example the operator has chosen a bitrate of 500 kbps for the multicast video representation. In this case, the MPD shows how the 500 kbps representation needs to include the *SupplementalProperty* with scheme *accessTech* and value multicast. But the operator also makes the video content available for terminals that have no access to the multicast channel. For those cases, the operator offers two additional video representations, one encoded at 250 kbps and the other at 1 Mbps, both included also in the MPD. A terminal accessing the multicast transmission will just retrieve the multicast representation. But a terminal using a unicast streaming will run the adaptation algorithm and retrieve the video content taking into account the three different representations.

B. A prototype for hybrid multicast/unicast video distribution

Fig. 5 shows the components of an experimental platform used to assess the validity of the proposal [10]. By using virtualization it is possible to create an experimental platform that can be run on a single machine. For this reason, we have used a general purpose open-source virtualization tool called Virtual Networks over linuX (VNX) [11] to deploy a network scenario consisting of three virtual machines: a content server, a shaper and a terminal. The content server (VM#1) includes the multimedia content used for the experiments, a FLUTE server to send the video segments encapsulated as FLUTE objects and an Apache web server used to respond to unicast requests. The shaper (VM#2) is used to simulate the network. It includes scripts that are used to control the bandwidth, latency or losses of the simulated network, in this case, an LTE network. Finally, the terminal (VM#3) includes a FLUTE client, a DASH player, a proxy server and the HTTP cache.

In a multicast video streaming session the FLUTE server is sending video segments to the terminal. Once the terminal joins the multicast channel, the FLUTE client starts to receive the video segments transmitted over the FLUTE session. The video segments that are delivered correctly (error free) are copied to the HTTP cache. The DASH player is requesting the video segments via HTTP, so in case a video segment is received via FLUTE correctly, it should be retrieved from the HTTP cache, otherwise, it should be retrieved from the Apache web server. In order to model this behaviour using available software components, a web proxy server has been configured to check the availability of video segments in the cache. The DASH player sends the video segment requests through the proxy and the proxy in this case returns a redirect message with a new URL to the client. If the video segment has been delivered correctly over the multicast session, the URL redirects the DASH player to retrieve the video segment locally from the HTTP cache. If the video segment is not available in the HTTP cache, the URL redirects the DASH player to the Apache server, from which the video segment is retrieved using a unicast connection. Therefore, in this platform the decision to use the recovery mechanism to retrieve video segments affected by errors is implemented using the proxy web server.

In our experimental platform, the content server includes a video sequence coded in four different representations, all of them with a video segment duration of 2 seconds, but each with a different maximum bitrate: 250 Kbps, 500 Kbps, 1000 Kbps and 2000 Kbps. Representation 2, coded with a maximum bitrate of 500 Kbps, is the one used for video broadcasting, and this is signaled by using the tag *accessTech* in the MPD file.

The DASH player used in the platform is the one offered by the DASH Industry Forum (DASH-IF) [12], written in javascript. In order to use the multicast video representation, we have modified the DASH player to use the new tag *accessTech*, including a new multicast mode that avoids using the adaptation algorithm and sticks instead to the video representation labelled as multicast. To test the solution we have also modified the DASH player to manually switch between the new multicast mode and the regular mode using unicast and also to simulate a handover event at a specific video playing time.

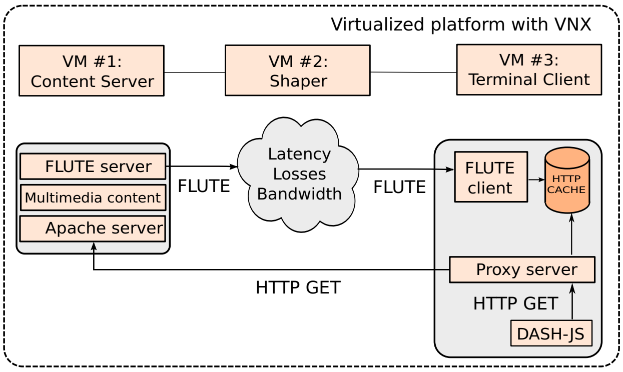


Fig. 5: Experimental hybrid FLUTE/DASH platform

V. Experimental results

By manually switching between the multicast mode and the unicast mode we have tested that the proposal works. But to better understand its impact on the implementation of a video player, a deeper analysis has been carried out by simulating handover events at specific video playing times and processing the information collected in the log files of the different software components of the platform.

Fig. 6 shows the video segments that are requested by the client during the playback of the video from playing time 50 secs. to playing time 60 secs. when a handover event at playing time 50 secs. is simulated. The handover event switches the video playing mode from multicast to unicast. While [1] points out that handover durations are typically no longer than 2 seconds, we have used handover duration values of up to 5 seconds to see more clearly the effect of the interruption of the network access. Fig. 6 shows how after the handover is completed, the video player starts requesting video segments of higher quality representations, i.e. it moves from representation 2 used for multicast to representations 3 and 4, at 1000 Kbps and 2000 Kbps respectively. This is a result of enabling the adaptation algorithm when the playing mode is switched to unicast after the handover.

Fig. 7 shows the level of the video player buffer, in seconds, during the handover. Since the video player always maintains a buffer around 8-10 seconds, even the longest handover is not exhausting the buffer completely, so video session continuity is achieved. While the above results are the ones expected from the point of view of the modifications made to the video player, they have made us realize of one possible flaw of the implemented solution. The result of a vertical handover is a connection to a new access network, with an unknown available bandwidth to access the server. Therefore, after the vertical handover, the client should start a new reset adaptation algorithm, requesting the video representation with the lowest bandwidth requirements, the same way it is done when the video playing starts from the beginning. The inclusion of this functionality in our prototype is currently an on-going work.

VI. Conclusion

With terminals that can use different access technologies, users expect to have more flexibility to access different services. In video streaming services, mobile network providers can use DASH over LTE, saving network resources by distributing video content over the eMBMS multicast channels. The hybrid FLUTE/DASH architecture proposed by the 3GPP for DASH over eMBMS is meant to work with DASH Access Engines with minimal modifications. Ideally, the hybrid multicast/unicast video retrieval process could be completely transparent to the DASH Access Engine. But in this paper we have shown that this might not be practical.

By including specific information in the DASH MPD file to signal the video representation used over the multicast channel it is possible to have a single MPD file that can be used for both the multicast video representation and other representations that can be delivered via unicast. While this proposal can simplify the management of a video content that is accessible through different access networks, it introduces some changes in the way video players interpret the content of the MPD file.

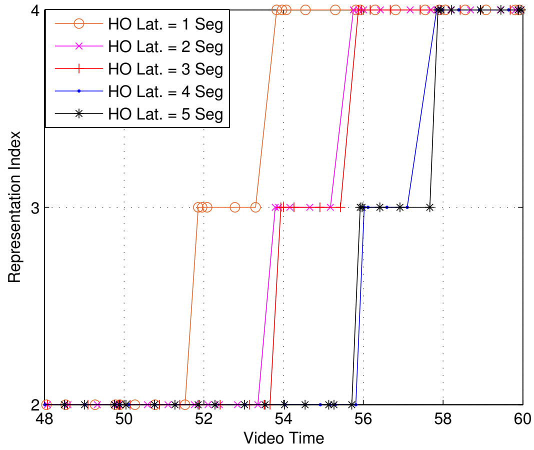


Fig. 6: Video segments requested during a handover

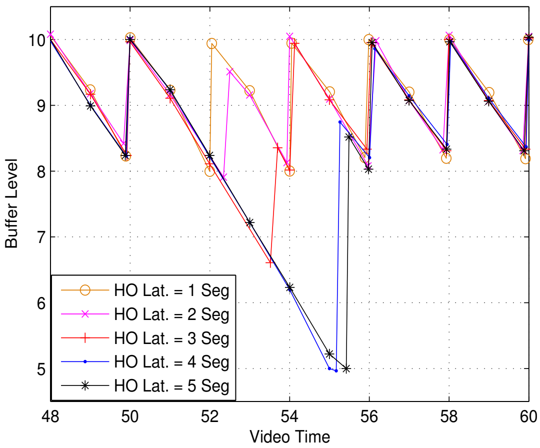


Fig. 7: Video player buffer during a handover

Our results of implementing the proposal above have also lead us to realize of additional changes needed for the adaptation algorithms used by the DASH Access Engine, like resetting the algorithm when there is a vertical handover. We are currently working on implementing and validating this and other future proposals.

Acknowledgment

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

References

[1] A. Ahmed, L. Boulahia, and D. Gaiti, “Enabling vertical handover decisions in heterogeneous wireless networks: A state-of-the-art and a

classification," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 2, pp. 776–811, Second Quarter 2014.

[2] ISO, "Information technology - Dynamic Adaptive Streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," 2014, ISO/IEC 23009-1:2014.

[3] *"Introduction of the multimedia broadcast/multicast service (MBMS) in the radio access networks (RAN)-Stage 2"*, 3GPP TS 25.346 v11.0.0, Sep 2012.

[4] V. Amarsinh and D. Satish, "Improved session handover in mobile wireless networks for video streaming," in *Advance Computing Conference (IACC), 2014 IEEE International*, Feb 2014, pp. 411–415.

[5] T. Paila, R. Walsh, M. Luby, V. Roca, and R. Lehtonen, "Flute — file delivery over unidirectional transport," Internet RFC 6726, November 2012.

[6] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, *Raptor Forward Error Correction Scheme for Object Delivery*, RFC 5053, IETF, Oct 2007.

[7] 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.

[8] T. Lohmar, M. Slssingar, V. Kenehan, and S. Puustinen, "Delivering content with LTE Broadcast," *Ericsson Review 2013*, 2013.

[9] C. Sankaran, "Data offloading techniques in 3gpp rel-10 networks: A tutorial," *Communications Magazine, IEEE*, vol. 50, no. 6, pp. 46–53, June 2012.

[10] C. Lentisco, M. Aguayo, L. Bellido, E. Pastor, D. De-Antonio-Monte, and A. García Bolívar, "A virtualized platform for analyzing LTE broadcast services," in *European Conference on Networks and Communications (EuCNC), 2015 IEEE 24th International Conference*, June 29/July 2 2015.

[11] "Virtual Networks over linuX (VNX)," http://dit.upm.es/vnxwiki, [Online], 2015.

[12] "DASH-IF forum website," http://dashif.org/, [Online], 2015.