



Improving the User Experience by Web Technologies for Complex Multimedia Services

Tayeb Lemlouma

► **To cite this version:**

Tayeb Lemlouma. Improving the User Experience by Web Technologies for Complex Multimedia Services. 8th International Conference on Web Information Systems and Technologies, Apr 2012, Porto, Portugal. pp.444-451. hal-00827743

HAL Id: hal-00827743

<https://hal-univ-rennes1.archives-ouvertes.fr/hal-00827743>

Submitted on 29 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IMPROVING THE USER EXPERIENCE BY WEB TECHNOLOGIES FOR COMPLEX MULTIMEDIA SERVICES

Tayeb Lemlouma

IRISA Lab and University of Rennes I, Lannion, France

Tayeb.Lemlouma@irisa.fr

Keywords: Content accessibility - content adaptation - Web pagination - media segmentation - digital home - DLNA - device independence

Abstract: In this paper we address the accessibility to advanced networking functionalities and media rendering using a new framework. The objective of proposing this new framework is to make advanced services accessible for heterogeneous and limited devices. The context of this work concerns digital homes where several multimedia services exist and their use requires complex configurations and the ability to handle advanced networking operations. This required ability is not satisfied for handheld devices that are heterogeneous in terms of software and hardware capabilities. Indeed, even advanced mobile terminals do not support, at least natively, networking operations such as the multicast, services discovery and multimedia streaming with different protocols. In order to improve the user's experience in such environments, we focus on the Web accessibility and the adoption of the HTTP protocol that are nowadays supported by the majority of heterogeneous terminals such as mobile phones, TVs and game boxes. In order to demonstrate the feasibility and the validity of our framework we consider, as a digital home architecture, the *Digital Living Network Alliance* (DLNA) environment. We show that even if the existing architecture and multimedia/networking services are complexes, the proposed framework simplifies the architecture for the user and guarantees the delivery of an accessible and adapted response.

1 INTRODUCTION

With the increasing popularity and the emergence of smart powerful mobile devices, users become more demanding in terms of quality of experience. As mobile devices become widely used in a similar way like with desktop terminals, users require similar accessibility regarding network's content and services. Unfortunately, even with the clear and huge advance of mobile devices capabilities, many services still inaccessible natively. Without further extensions and advanced configurations, many services, content formats and delivery methods remain rich, complex and not adapted to "limited" environments compared to desktop platforms. In order to handle the heterogeneity of environments and terminals and provide a universal accessibility, it is necessary to determine a minimal set of compatible and standard functionalities that can be used in advanced and limited environments. In the context of content access, the HTTP protocol represents the most common and simple method used for delivering data. HTTP can be easily implemented in limited environments and can

deliver data from simple Web pages to media content. Our framework is based on this observation. It aims to improve the user's experience regarding the use and the access of complex multimedia services. The objective is to enable the use of complex services and content, using limited terminals, in a universal way. For limited devices, digital homes represent a challenging context because several multimedia services are available but their use requires complex configurations and the ability to handle advanced networking operations. This required ability is not satisfied for handheld devices that are heterogeneous in terms of software and hardware capabilities. Indeed, even advanced mobile terminals do not support, at least natively, networking operations such as the multicast, services discovery and multimedia streaming with different protocols. We believe that the adoption of the HTTP protocol coupled with the respect of Web accessibility guidelines improve the quality of experience of users regarding the access and use of multimedia and networking services even in complex architectures. We consider any network service as the union of the following three sets: (1) input argu-

ments, (2) advanced operations and (3) output results. Consequently, enabling a service in a limited environment becomes equivalent to perform the following: (1) delegate the execution of incompatible operations to an intelligent component, and (2) provide a standard and compatible interface that allows the user to interact with services and rich content and receives an adapted form of the service's output. The application of this approach will be validated in the *Digital Living Network Alliance* (DLNA) environment (DLNA, 2007). DLNA represents an ideal context that represents heterogeneity regarding the content, servers, services and terminals that can aim to access the home network services and content. The framework of *Universally*, that we present in this paper, is discussed regarding the provided Web interface and its compliance with the W3C recommendations and the content fragmentation (Web pagination and media segmentation).

2 DLNA

The objective of DLNA technology is to distribute digital content from a source to all the compatible devices (DLNA, 2007; DLNA, 2006). The key functions of DLNA are: connectivity and networking, device discovery and control, media management, media formats and media transport (Fig. 1). Connectivity and networking use the existing network platform at home. Device discovery and control is based on the UPnP Device Architecture (DA) 1.0 (UPNP FORUM, 2008b). Media management is based on the UPnP AV (UPNP FORUM, 2008a). Media formats are defined as a set of required and optional format profiles. The media transport in DLNA is done over HTTP. The RTP protocol is introduced recently and still optional (DLNA, 2007; DLNA, 2006).

Three entities are specified: devices, services and control points (CPs). Services are functions provided by a device, found and invoked by a CP. The main functions of a UPnP device are: IP addressing, discovery (periodic advertisement of services), description of services and capabilities, control in response of CPs requests, eventing to notify registered CPs and presentation in HTML to possibly monitor the device. This depends on the specific capabilities of the presentation page and device and if the device has a presentation's URL. SSDP messages are sent over HTTPMU (HTTP Multicast over UDP) in order to discover resources in the network (UPNP FORUM, 2008b). SOAP allows to specify available operations. In the audio video (AV) digital home, the three main entities are: the media server (MS) that offers me-

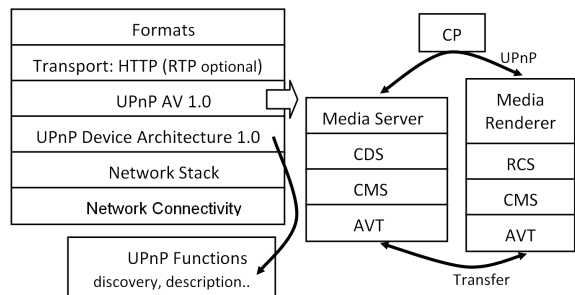


Figure 1: DLNA framework.

dia content, the media renderer that plays content and CPs that control what and how contents are played. MS provides three services: Content Directory Service (CDS), Connection Manager Service (CMS) and AV Transport Service (AVT) (Fig. 1). A media renderer provides the Rendering Control Service (RCS), the CMS and AVT services. CDS provides actions for CPs in order to list media items and relative metadata. RCS provides actions for a CP to control the rendering of a media resource. CMS handles connections with a device and AVT controls the player actions (play, seek, etc.). DLNA defines two main entities: Digital Media Server (DMS) and Digital Media Player (DMP). DMS is a UPnP/AV CDS device that provides media resources. DMP is a UPnP/AV CDS control point that can discover resources and render them.

In a DLNA/UPnP environment, the number, nature (UDP and multicast) and the periodicity of UPnP advertisements can be inconvenient. Also, in order to render a media resource, several messages and steps must be achieved. A mobile device, used in the digital home, has to support the multicast, UPnP Template Language parsing, SOAP protocol and the unreliable nature of UDP especially over a wireless access. Moreover, there is no guarantee that the original format and profile of the requested media resource will be supported. This situation is not adapted to heterogeneous environments where rendering capabilities, delivery and access technology, bandwidth, congestion probability and power consumption are not the same for all the existing devices.

3 UNIVERSALLY

The aim of our proposition is to enable and optimize the access and control of existing media resources in home for heterogeneous and mobile devices with limited access and rendering capabilities. To do so, we propose *Universally*: a new component that can

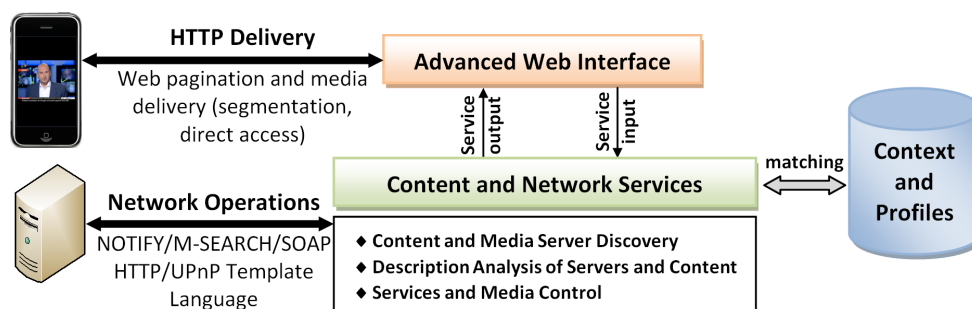


Figure 2: The *Universally* architecture.

be easily integrated into the digital home architecture and provide a universal access in a heterogeneous environment. The modular architecture of *Universally* is presented in Fig. 2. The key idea is to provide an easy media access only when it is needed by a user and requested by its device. Based on an Advanced Web Interface (AWI) module, devices can browse available DMSs and resources since it is done over HTTP. Once a media resource is found and displayed on the AWI with a hyperlink, the user can use it directly, or use an adapted version delivered in a real time. *Universally* delegates the UPnP main functions (discovery, control, etc.) to a specific module that ensures the access for limited devices such as those which do not support the multicast natively or are not compatible with DLNA. Since devices with advances capabilities can also use the proposed AWI, UPnP advertisements of media services are filtered and only responses to the *Universally*'s M-SEARCH requests are authorized. Unnecessary traffic is so avoided. The context module is used to describe the hardware and software capabilities of devices that aims to play a media resource (Lemlouma, 2012). If the original format of a media is not supported, it can be adapted then transferred to the device. Based on the device's profile, the Web interface is also dynamically adapted by performing a pagination of discovery results when the device explores media resources of available DMSs. Searching MSs are done only once when the user accesses the AWI interface. The AWI's home page is a simple page with an HTML refresh duration, after what the user is redirected to a new page where the result of services discovery is displayed. At any moment, the user can manually trigger a new services discovery. *Universally* has allowed to: avoid unnecessary traffic (NOTIFY and M-SEARCH), unify the eventual available presentations of MSs and devices, make the browsing of resources possible and optimal and avoid SOAP and XML UPnP handling and parsing for limited heterogeneous devices.

4 ACCESSIBILITY OF THE ADVANCED WEB INTERFACE

The W3C Web Content Accessibility Guidelines (WCAG) 2.0 (W3C-WCAG, 2008) propose recommendations for making Web content more accessible for users. The W3C Mobile Web Best Practices (MWBP) recommendation 1.0 (W3C-BP, 2008) addresses a series of recommendations designed to improve the user experience of the Web with mobile devices. The proposed recommendations refer to delivered content but are not intended to be the best practice for the processes of content creation and rendering on devices. The result of performing networks operations must lead to Web interfaces designed in a compliant way to the two previous W3C standards. As we focus on the accessibility of limited devices, we consider, in the design of our generated interfaces, the recommendations of the MWBP standard. We have evaluated the Advanced Web Interface according to the sixty statements of MWBP 1.0.

Table 1 summarizes the support of MWBP 1.0 by the advanced Web interface of *Universally*. The large compliance, regarding the MWBP statements, is due to the structure simplicity of our Web interface that hides the support of advanced networking functionalities. Some statements such as the user input are not applicable since the user's interaction is mainly based on exploring existing media servers and items.

An important number of statements are ensured thanks to: the use of content pagination with small size, navigation hyperlinks used in content exploration and pages navigation, the use of the same and simple template for different pages, the simple and short URI that identifies the hosting server of *Universally*. Page refreshing, used while a network operation (media servers searching) is triggered, is compliant with the statement #14 of the MWBP 1.0. At anytime, the user can stop it using the stop hyperlink (Fig. 3).

Table 1: The support of MWBP 1.0 by the Advanced Web Interface of *Universally*.

Best Practice Group of MWBP 1.0	# of MWBP 1.0 Statements	# of N/A Statements	# of Supported Statements
Overall Behavior	4	0	4
Navigation and Links	12	4	8
Page Layout and Content	12	2	10
Page Definition	25	4	21
User Input	7	6	1

5 EVALUATION OF THE DELIVERY SIZE

According to the statement #21 of the MWBP 1.0, a delivered page must be limited in size. In our approach, we propose to make a balance between the scrolling and pagination of the result presented in the Web interface. Scrolling long pages can affect the user experience and requires a long time load. On the other hand, an important pagination implies multiple requests to receive the relevant result. In order to optimize our process, we consider three input parameters: the screen size of the user's device, the size of delivered page and the network access technology used by the device. We distinguish four types of displayed pages: *searching progression*, *negative result*, *media servers list* and *items list* (Fig. 3). The two first pages are displayed in the case where *Universally* is looking for media servers or when no media servers are found. The size of these two pages is small (1,6 Kb) which is compliant with the MWBP recommendation. The *servers list* page includes hyperlinks to different found media servers. The *items list* page includes links to folders and media resources items that a media server hosts. The following is an example of links extracted from our experimentations.

1. `Me-PC: TVersity Media Server`
2. `All`
3. `Cool-T-Magic Key(2 Jays Remix)(video)
`

The first line represents the third found media server (named "*Me-PC: TVersity Media Server*") in the digital home network. The second one represents a folder item (named "*All*") of the media server. The third line represents a video item of the found media server (existing in the "*All*" folder).

The HTML code of each hyperlink has the following format: `<a accesskey="[access key of the hyperlink]"`

`href="[arguments]">[name of the item]
([type of media item])
`

The *access key* parameter is used to be compliant with statement #9 of the MWBP 1.0. It is set to the position of the link in the HTML page; it can vary from 1 to 20, so it is coded with two characters. The source of the hyperlink ([arguments]) is formatted for media servers and folders as follows: `ms?serv=[identifier of the server]&id=[identifier of the folder]` (set to "0" if the link represents a media server) `&inf=[starting index of items that will be explored if the link is hit]&total=[number of XML child items (folders and/or media items) used in pagination]`. For a media resource link (e.g. a video or audio file), the format is similar to media servers and folders links with the new attributes: *idmedia* and *action*. The *idmedia* is an identifier of the resource. It is used to avoid including the hard link provided by the media server that is in general very long. Using the *idmedia* attribute allows to adapt, in real time, the original media if it is not supported by the user's terminal (*action=adapt*). Otherwise, the user can directly access to the media without adaptation (*action=deliver*). The delivered page's size of the found servers and items list can be calculated as follows:

$$Size = sizeof(templateOfThePage) + n \cdot sizeof(itemLink) + sizeof(navigationImages) \quad (1)$$

The size of the template is fixed (3,83 Kb in our implementation). The size of a navigation icon (next or previous) is 1,69 Kb. Note that only one navigation icon is displayed in the first and the last page of the paginated result (Fig. 4). Also, even that the same navigation icon is displayed twice on the bottom and the top of the page (in respect to the MWBP statement #6), the icon size is considered only once in calculating the size of the delivered page. This is done because the browser requests the icon only one time, and thanks to its cache, the icon is displayed in the top and the bottom of the page without sending a new request.

The maximal length of a hyperlink's text (the [name of the item] value) is set to 30 characters (followed by the "." string if the original text is greater

than 30. According to the format discussed previously (`[name of the item]` (`[[type of media item]]
`), and the maximal encoding length of each attribute, we have evaluated the maximal length of a complete link to 121 bytes.

With the result of (1), the maximal overall size of a delivered page (including n links) is then:

$$\begin{aligned} \text{Size(Kbytes)} &= 3,83 + 0,118 \cdot n + 2 \cdot 1,69 \\ &= 7,21 + 0,118 \cdot n \end{aligned} \quad (2)$$

The advanced Web interface of *Universally* delivers a paginated result where the number of displayed links in each page depends to the profile of the user's terminal (the screen size) and is limited to a maximum of 20 links. This property, with the result of (2), guarantees the compliance with the MWBP statement #21 in respect to the *default delivery context* which was defined as being the minimum delivery context specification necessary for a reasonable experience of the Web (W3C-BP, 2008). Indeed, the considered metric here is the size of each delivered page. Whatever services complexity, in our case the result's size will never exceed 10 Kbytes which is suitable for any network access technology (cellular, wireless and wired) and required in a default delivery context. This satisfied property means that the size of each received result (after the network discovery of services) still very small. Hence, this will not affect negatively the user experience in discovering multimedia resources.

6 PAGINATION

The content pagination is applied to deliver fragments of the original content in a suitable way that is well balanced between the number of necessary requests and the size of each fragment. In a digital home environment, *Universally* applies pages pagination regarding the list of discovered media servers and the list of items hosted by a media server or stored in an explored folder. In other term, if the list of discovered media servers or items of a given folder is long and not adapted to the terminal's limitations, the list will be divided in small pages interlinked with next and previous hyperlinks (Fig. 4). In order to paginate the content of a root element (media server or folder), the number of child elements must be known in advance. To do so, *Universally* uses the invocation of the SOAP actions: *Browse* and *BrowseMetadata* sent to the *Content Directory* service of the media server. This service is identified by the URI: `urn:schemas-upnp-org:service:ContentDirectory:1#Browse`. The

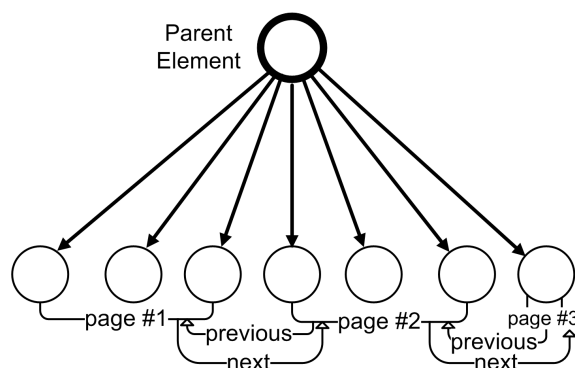


Figure 4: Pagination and interlinking of a parent element.

number of child elements is returned within the SOAP result in the XML attribute *childCount*. In the following, an example of the result of a SOAP *Browse* action received, in *Universally*, from a media server.

```
<DIDL-Lite xmlns="..">
  <container id="0" childCount="4" parentID
    ="-1" restricted="true">
    <dc:title>root</dc:title>
    <res protocolInfo="http-get:*:image/jpeg:
      DLNA.ORG_PN=JPEG_TN">
      http://192.168.1.10:5001/get/0/..
    </res>
    <upnp:class>
      object.container.storageFolder
    </upnp:class>
  </container>
</DIDL-Lite>
```

The *container* tag, specifies the identifier of the current element (here it is the root of the media server, $id=0$) and the number of child elements ($childCount=4$). As shown in Section 5, every browsing link, displayed within our Web interface, includes the *inf* and *total* arguments. The first argument is used to display child items (of the current parent element) starting from the *inf* position (Algorithm 1, line 2). The second argument has the value of the *childCount* attribute. It allows to check if there are more elements to display in a next page or not (Algorithm 1, line 3). If it is the case, a *next* link is included in the presentation page. As mentioned previously, the number of the child elements to display in a page (*itemsPerPage*) is set according to the terminal's profile. Handling terminal's profiles is based on the *Multimedia Universal Profiling* (MUP) schema that we have defined based on the W3C RDF language (the proposed RDF schema will not be discussed in this paper) (Lemlouma, 2012). MUP allows to describe the profiles that are stored in a profiles repository and specify the displaying and rendering capabilities of a

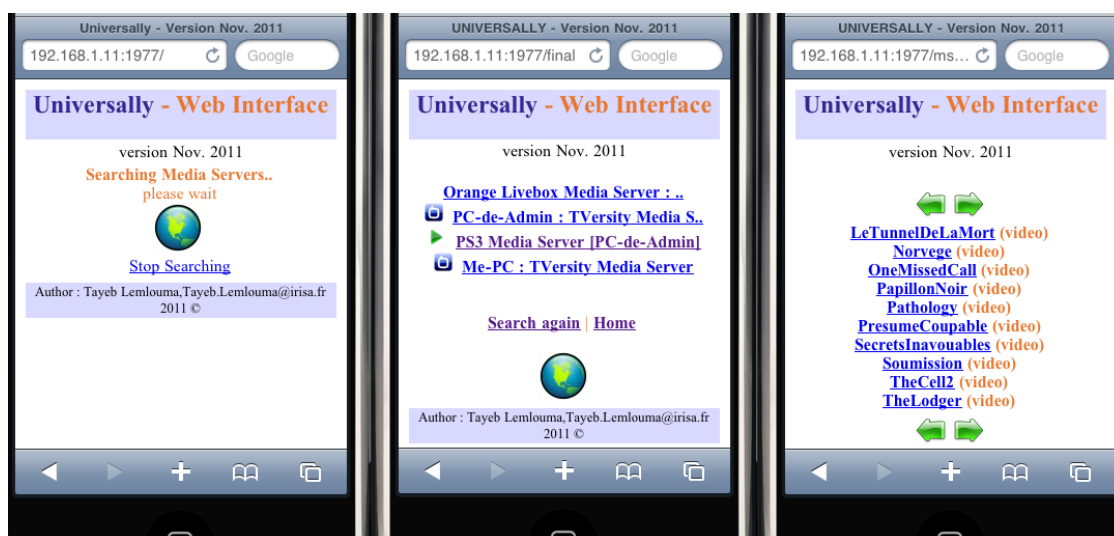


Figure 3: The advanced Web interface of *Universally*.

Algorithm 1 Result pagination and exploration

Require: inf , $itemsPerPage$ and $childCount$.

Ensure: Display items in a paginated way.

- 1: $nextHTML \leftarrow ""$; $previousHTML \leftarrow ""$;
- 2: **displays** $itemsPerPage$ items **from** the parent element starting from the item position: inf ;
- 3: **if** $(inf + itemsPerPage) < childCount$ **then**
- 4: $nextHTML \leftarrow$
 $'<a\ accesskey="N" \ href="/ms \ ?serv = ID$
 $\&id=objectID+\&inf=(\ inf+itemsPerPage)\&$
 $total=(\ childCount) ">'$;
- 5: **end if**
- 6: **if** $inf > 0$ **then**
- 7: $previousHTML \leftarrow$
 $'<a \ accesskey="P" \ href="/ms \ ?serv=ID$
 $\&id=objectID+\&inf=(\ inf-itemsPerPage)$
 $\&total=(\ childCount) ">'$;
- 8: **end if**
- 9: $displayingHTML \leftarrow '<..>'+ \ previousHTML+'$
 $<..>'+ \ nextHTML+'<..>'+ \ displayingHTML$
 $+ '<..>'+ \ previousHTML+'<..>'+ \ nextHTML;$

device. Near to 14000 different user agents are stored (Lemlouma, 2011).

Figure 5 shows an overview of the HTTP content-length variation over the access time. The measurements concern all the HTTP responses (HTML and images) of *Universally* to a mobile phone while exploring four media servers during about one minute. The results show the conformance of *Universally* re-

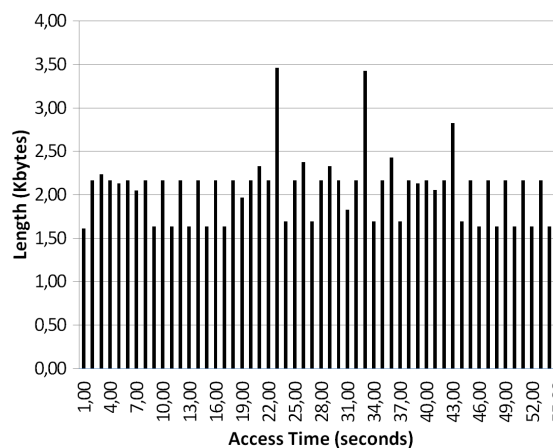


Figure 5: Content pagination and HTTP content-length over access time.

sponses regarding the size metric adopted in the *default delivery context* of MWBP as predicted and discussed in Section 5. Here, the maximal size is near to 3,5 Kb.

7 REAL TIME MEDIA ADAPTATION

Similar to content pagination, media delivery has to improve the user experience and be adapted to limited devices. In a digital home environment, a large amount of the stored content is rich media content (movies, music, pictures, etc.) having often a high quality and requires an important storage space. Sev-

eral media delivery protocols and methods were proposed for different environments including limited devices (Ma K. J.; Barto and R., 2011). In order to gain access to a large amount of heterogeneous devices and avoid complex network requirements, we adopt the HTTP as the delivery protocol. For heterogeneous environments, HTTP represents a common protocol either for downloading or streaming media and structural resources. Furthermore, the protocol is easily accepted by firewalls and different network configurations which is not the case of other methods like RTP over UDP with dynamic ports. *Universally* considers the media fragmentation with the HTTP protocol by implementing the HTTP Live Streaming (Pantos and May, 2011). In *Universally*, when an original video has to be adapted (based on the terminal's profile (Lemlouma, 2012)), our advanced Web interface delivers a hyperlink pointing to a playlist file (a m3u8 file, as shown bellow) that contains links to different video fragments.

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-MEDIA-SEQUENCE:4
#EXTINF:10,
http://.../0_3_976-5-00000004.ts
#EXTINF:10, ..
```

The playlist content is updated each time a new video fragment is created. In a video adaptation instance, the playlist includes links to the last five created segments. Our implementation of the media fragmentation is based on the VLC transcoding of the VideoLAN project (VIDEOLAN, 2011) with a video segment's duration of 10 seconds. Figure 6 shows an overview of the different segments length downloaded by the native player (*AppleCoreMedia*) of a mobile phone (an iPhone 4GS, iOS 5.0.1) during a real time video adaptation. The original video, of our experimentation, is a HD video with a size of 1,02 Gbytes, a duration near to 25 minutes, video codec of H264 MPEG4 AVC (part 10), a dimension of 1440 per 1080 and audio codec of A52 with a rate of 384 Kb/s. The adapted format is H.264 Base Profile, level 3 (ITU-T H.264, 2009) with a dimension of 320 per 240 resolution. The generated fragments allow the player to download and use segments buffering for a smooth play-out. Ideally, since each segment - generated in real time- represents 10 seconds of the original video, the player should buffer required segments prior to the current instant of video playing. For instance, at $t=50$ seconds (from the beginning of a video), the fifth segment must be previously generated and downloaded. The generation and download of a segment depend on the segmenter implementation, the original and targeted video transcoding, the

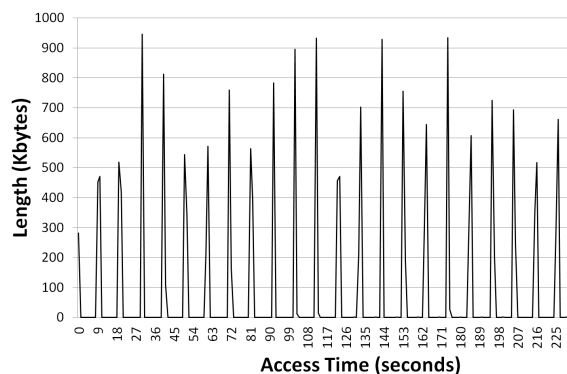


Figure 6: Media segmentation streaming.

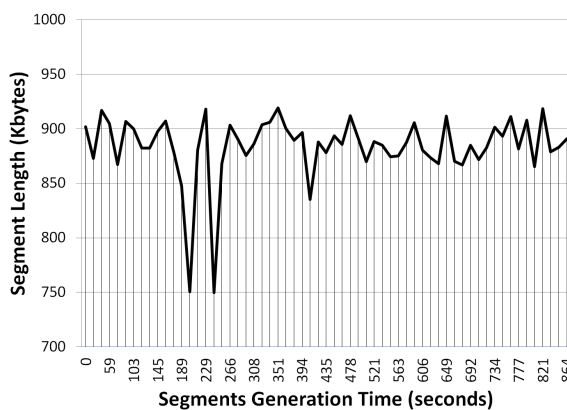


Figure 7: Media fragments generation.

segment size and the available link bitrate between *Universally* and the user's terminal. Figure 7, shows the generation times and lengths of sixty segments of 10 seconds of our HD test video. For the user experience, the observed video segmentation delays vary from 0 to 4 seconds. This real time media adaptation enables users to explore quickly existing media items and start to render them in an optimal way independently to the original size and format constraints.

8 CONCLUSIONS

In this paper, we addressed the accessibility to advanced networking functionalities and media rendering using a new framework. The objective of proposing this new framework is to make advanced services accessible for heterogeneous and limited devices that are natively unable to: handle networking protocols, interact with other components of the network and achieve complex networking and access actions. We have shown that based on a standard Web interface and the HTTP protocol, the user experience (in discovering, using and accessing complex networking and multimedia services) was improved. First, thanks

to the *Universally* approach, limited terminals are now able to discover and access complex services ; this was not initially possible for terminals that do not support advanced networking operations such as the *multicast*. Secondly, the approach guarantees a standard and accessible network response which is very limited in size: a maximum of 10 Kb for the Web interface and 950 Kb for media segments whatever the complexity of the existing digital home network. Whether with the content pagination or the media segmentation, the quality of experience of the user is improved whatever the access technology used by its device. The content fragmentation regarding Web pages and media content has permitted to optimize the content delivery for terminals with a minimal set of capabilities. Within the DLNA environment, and thanks to *Universally*, searching, access and rendering of media resources and services is enabled from anywhere to mobile phones and other limited devices that do not support natively required DLNA networking functionalities and complex media rendering.

REFERENCES

- Lemlouma, T. (2011). An overview of the MUP repository's identifiers, january 2012. http://people.irisa.fr/Tayeb.Lemlouma/mup/UNIVERSALLY_identifiers_repository_version_01_2012.txt .
- Lemlouma, T. (2012). Source of the multimedia universal profiling (MUP) schema, january 2012. <http://people.irisa.fr/Tayeb.Lemlouma/mup/schema> .
- Ma K. J.; Barto, R. B. S. and R., N. (2011). Mobile video delivery with HTTP. In *IEEE Communications Magazine 2011, Volume: 49, Issue: 4, pp: 166-175*.
- Pantos, R. and May, W. (2011). HTTP live streaming, september 2011. <http://tools.ietf.org/id/draft-pantos-http-live-streaming-07.txt> .
- DLNA (2006). DLNA Network Device Interoperability Guidelines, october 2006. <http://www.dlna.org/> .
- DLNA (2007). Digital Living Network Alliances: DLNA overview and vision whitepaper, 2007. <http://www.dlna.org/> .
- ITU-T H.264 (2009). ISO/IEC 14496-10 AVC. advanced video coding for generic audiovisual services [s]. <http://tools.ietf.org/id/draft-pantos-http-live-streaming-07.txt> .
- UPNP FORUM (2008a). UPNP AV Architecture:1, 30 september 2008. <http://www.upnp.org/> .
- UPNP FORUM (2008b). UPNP Device Architecture 1.0, 24 april 2008. <http://www.upnp.org/> .
- VIDEOLAN (2011). VideoLAN project. <http://www.videolan.org/> .
- W3C-BP (2008). Mobile web best practices 1.0, mobile web initiative (mwi). In *W3C Recommendation, 29 July 2008*. <http://www.w3.org/TR/mobile-bp/> .
- W3C-WCAG (2008). Web content accessibility guidelines (wcag) 2.0. In *W3C Recommendation 11, December 2008*. <http://www.w3.org/TR/WCAG20/> .