Scalable e-Learning Multimedia Adaptation Architecture

Mazen Almaoui and Konstantinos N. Plataniotis

Dept. of Electrical and Computer Engineering, University of Toronto
{mazen, kostas}@dsp.utoronto.ca

Abstract. A neglected challenge in existing e-Learning (eL) systems is providing access to multimedia to all users regardless of environmental conditions such as diverse device capabilities, the heterogeneity of the underlying IP network, and user modality preference. This paper proposes a novel two-tier transcoding framework capable of adapting eL multimedia to meet the end-user environmental challenges. This two-tier architecture consists of 1)an application layer transcoder that adapts the presentation format of the eL content as viewed in a browser to meet device capabilities and user modality preference, 2) a bitstream transcoder that transforms multimedia streams to conform to the device's processing capabilities and to adapt the encoding rate to meet the network's fluctuating bandwidth. Results demonstrate the eL multimedia transcoding for mobile devices and its low overhead delays.

1 Introduction

The maturity of the Internet has given rise to effective collaborative e-Learning (eL) webcasting. Such applications allow various people from different physical locations to communicate and interact together over the Internet. Fig. 1 shows a screen shot of the web interface for the ePresence eL webcasting system that has the capability of streaming video, audio, text, and slides to the end-user [1].

The increasing multimedia processing capability of mobile devices such as Personal Digital Assistants (PDA), Pocket PCs, and Smart Phones as well as advances in wired and wireless networks in terms of multimedia delivery have added a new dimension to how people collaborate. Conventional eL applications such as ePresence cannot support multimedia delivery that adapts to unique device and network conditions. In light of this fact, there is a growing aspiration for a Universal Multimedia Access (UMA) Framework [2] that provides seamless access of multimedia to anyone, at anytime, by adapting eL content to unique device capabilities, unreliable IP networks, and user personal preferences.

End-user devices differ in a multitude of processing capabilities such as framerate, resolution, and number of audio channels supported. This diversity in device capabilities requires media content to be adapted, and media streams that cannot be processed by the device to be dropped. This must be performed in real-time to avoid delays in delivering eL content and to provide the user with the experiences that an individual attending the actual live event would have [3].

M. Kamel and A. Campilho (Eds.): ICIAR 2005, LNCS 3656, pp. 191–198, 2005.

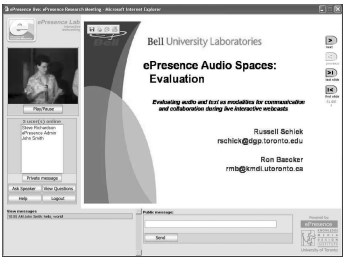


Fig. 1. ePresence: interactive e-Learning application

The second challenge to consider relates to the underlying network's bandwidth capacity and fluctuations determined by factors such as packet loss, delay, and jitter [4]. This diversity in bandwidth characteristics gives rise to a need for multimedia adaptation to provide variable bit-rate multimedia encoding.

Lastly, we address the challenge of delivering a given quality of experience [2] to the end user. This involves negotiation of a Quality of Service (QoS) based on personal preferences and preferred modality. This must be negotiated, not granted because not all devices are capable of processing all types of media.

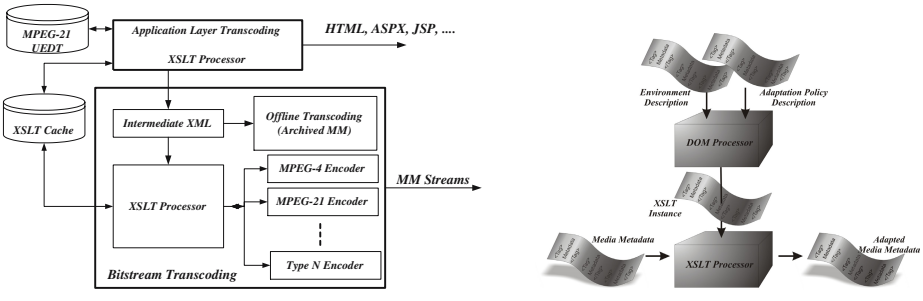
In order to resolve the three key challenges, multimedia transcoding [5] is utilized to adapt eL content. Here, two types of transcoding are required; application layer transcoding and multimedia bitstream transcoding. Application layer transcoding adapts the presentation format of the eL content [6] to meet device processing capabilities and user preferences (Fig. 1). Bitstream transcoding [5] is performed on the the actual media to parse, transform, and truncate the underlying multimedia streams to adapt the encoding rate to the network's fluctuating bandwidth capacity. Both transcoding approaches require descriptions of varying device capabilities, network conditions, and user preferences. This can be done using metadata, a tool used for describing, indexing, and searching the properties of the user's environmental conditions, and properties of the actual media. Thus, metadata-driven transcoding [6] is needed to achieve application layer transcoding and to provide adaptation parameters to achieve bitstream transcoding. This paper proposes a novel scalable two-tier architecture for implementing metadata-driven transcoding. Section 2 gives a brief overview of existing eL systems architectures. Section 3 explains the proposed two-tier transcoding architecture. Section 4 visually demonstrates the importance of transcoding for an eL system by applying our approach to the ePresence architecture and provides the delay implications of real-time bitstream transcoding.

2 Related Work

Web-based eL systems such as the ones discussed in [7,3] include a combination of video, audio, slides, chat session, and whiteboard functionality. A four-tier electronic educational system (EES) model was proposed in [8] that provides a degree of data personalization to the end-user. The top layer (instructional layer)

allows educators to specify which media to include in the system (video, slides, etc.). However, the end-user, does not have the flexibility to specify which eL material they desire and no consideration is put forth concerning what modalities individual user devices can process. The lower layers take the instructions from this layer to create the final presentation format as seen through a browser. Similarly, a metadata-based approach to delivering personalized course material for a specific user learning needs was proposed in [9]. In particular, IEEE LTSC Learning Object Metadata (LOM) is used to allow professors to deliver personalized material. For example, the user can choose the language or difficulty level of the eL material. Programmable models such as Netscript [1], ANTS[2], and SmartPackets[3] provide solutions for handling network traffic, however do not address adaption to meet device and user preference needs.

These eL solutions do not adapting multimedia to meet each user's unique device capabilities, modality preference, and adapting the media encoding rate to take into account fluctuating network conditions. Transcoding must be incorporated into existing eL applications all of the above mentioned user needs.



(a) Proposed two-tier metadata-driven transcoding architecture

(b) Metadata Transformation Engine

Fig. 2. System overview

3 Proposed System

The objective of the proposed transcoding system is to adapt eL multimedia to match a user's environment characteristics (device capability, network conditions, modality preference). The proposed architecture is shown in Fig 2(a).

Section 1 presented a motivation for metadata-driven transcoding. Metadata syntax is represented using Extensible Markup Language (XML). Metadata description can be done through the MPEG-21 standard [10]. The tools offered by the standard include the Usage Environment Description Tool (UEDT) [2] to describe device capabilities, network conditions, and user characteristics as

[1] http://www1.cs.columbia.edu/dcc/netscript/
[2] http://www.cs.washington.edu/homes/djw/papers/00755004.pdf
[3] http://www.net-tech.bbn.com/smtpkts/smtpkts-index.html

well as the natural environment characteristics. The processing and transformations of this metadata can be accomplished using Extensible Stylesheet Language Transformation (XSLT) [5]. Application layer transcoding requires transformation rules to process user metadata in order to determine the presentation preferences. Bitstream transcoding requires these rules to process intermediate XML and physically adapt multimedia to match the device's processing capabilities and network available bandwidth. Due to the short length of this paper, details of XSLT sheets will not be explained, refer to [10] for more detail.

A generic metadata transformation system is proposed in [11] as shown in Fig. 2. This solution consist of a Document Object Model (DOM) processor that creates XSLT sheets and an XSLT processor that transforms XML with the created XSLT sheet. The problem with DOM processing is its computational cost making it an unviable solution for a real-time eL application. Our method performs DOM processing a priori and caches XSLT sheets for transcoding. Here, an XSLT rule for providing multimedia adaptation is chosen from a set of previously cached sheets to provide best-effort service by replacing the DOM processor with a selection phase that is more efficient for real-time transcoding.

The XSLT sheet selection processes is comprised of three steps. First, the cached sheets are pre-filtered to determine a small set of possible sheets matching the user request. The pre-filtering is done based on parameters that remain *unchanged* throughout a session assuming the same device is used from the beginning of the eL webcast until the session is terminated. Then, one of the sheets from this set is chosen to be used during transcoding. Finally, the XSLT sheets chosen are passed to transcoding modules to transform XML metadata to produce the appropriate presentation template and to transcode the multimedia streams. This will be explained in more detail in subsequent sections. The process of determining which sheet to select is up to the system designer.

Application layer transcoding begins by obtaining the user's UEDT metadata. The main idea behind the application layer transcoding module is to choose the best matching XSLT rule from the XSLT cache that will adapt the eL content layout (which is represented by an XML file) to meet user environment characteristics. The sheet selected depends on the user's device capabilities and the desired modalities. This module produces and outputs the proper template that will display the desired modalities in a browser (Fig. 1). The template produced depends on the underlying eL system interface (e.g. HTML, JSP, etc.). The application layer transcoding module then passes control to the bitstream transcoding module to adapt multimedia streams as shown in Fig. 2(a).

The next step in the adaptation process is bitstream transcoding. The bitstream transcoding module utilizes adaptation parameters from the application layer module to transform the underlying streams. Depending on the implementation of the eL system, some of the components shown in Fig. 2(a) can be excluded. This will depend on whether transcoding will be conducted offline (i.e. on-demand) or in real-time (i.e. live and on-demand). As shown in Fig.2(a), the bitstream transcoding engine consists of the intermediate XML, offline and real-time (MPEG-4) encoders, and the XSLT processor.

As a residual of application layer transcoding, intermediate XML metadata is passed to the bitstream transcoding module. This intermediate XML contains the metadata describing the modality and resolutions reserved for each as supported by the browser template produced by the application layer transcoder. Additional information included are name and address of the multimedia that the browser embedded protocols will attempt to access eL multimedia and other optional parameters as required by the system.

Offline bitstream transcoding is in essence scalable coding [2]. Multiple scaled copies of the same media are encoded and stored on the content server. The multimedia copy that best matches the end-user's environment parameters is delivered. Scalable coding can only be used for on-demand eL systems and is not a viable solution for live webcasting. In contrast, real-time bitstream transcoding requires only one version of the multimedia to be stored on the content server. Multimedia is adapted in real-time to meet the end users environmental needs. There are various bitstream encoding solutions [5] capable of providing real-time transcoding. The XSLT processor in Fig. 2(a) can be used to transform the intermediate XML metadata to any format required by the underlying eL system's encoder in order to perform metadata-driven transcoding. For example, if the MPEG-21 Framework is used, the intermediate XML could be the environment description and the the bitstream description (BSD) [5] of the multimedia streams. The XSLT processor would process the user's environment description to determine which XSLT sheet to retrieve from the cache in order to adapt the requested multimedia streams. The output of the XSLT processor can be the bitstream adaptation rules required by the MPEG-21 encoder to transcode the multimedia streams to meet the end-user environmental needs.

The MPEG-4 Standard scalable tools are used to provide temporal, spatial, and SNR scalability, hence can serve as the bitstream transcoder. The set of rules obtained from the XSLT cache together with the intermediate XML serve as input into the XSLT processor. The intermediate XML contains parameters for temporal (e.g 15 f/s for PDA, 25 f/s for PC) and spatial (qcif for PDA, 4cif for PC) scaling of video and slides and audio scaling parameters. The XSLT processor outputs a script file that take into account the user's environment characteristics. This script is utilized by the MPEG-4 encoder to transcode the multimedia streams. Real-time transcoding solutions are not limited to the proposed MPEG-21 and MPEG-4 encoding solutions. The important point of the XSLT processor (in combination with the intermediate XML) is to provide an extensible architecture to incorporate any bitstream encoding solution that is a viable or desired solution to virtually any eL system.

4 Results

Experimental results of our system for achieving application layer and bitstream transcoding will be presented in this section. This will demonstrate why application layer transcoding is crucial to delivering personalized eL material to meet the user's device, network, and personal preference to in order to affectively utilize ePresence interface functionality. Results of the proposed bitstream transcoding

approach will also be evaluated in terms of how delay will affect real-time multimedia adaptation and delivery of ePresence material. Testing is conducted using an FFMPEG encoder [4], Darwin Streaming Server [5], and a QuickTime Player to provide an end-to-end MPEG-4 delivery framework. Note that multicasting is not used because (a) the ePresence server resides over a restricted unicast network gateway and (b) eL material needs to be delivered to each user separately to match their unique environmental needs. However, a multicast solution would be an important research topic to investigate for ePresence.

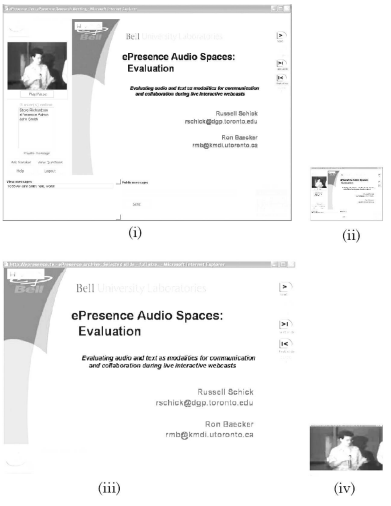
Application Layer Transcoding: Fig. 1 shows the current browser template used to represent eL content. This template is intended for PCs with suitable multimedia capabilities with a large enough real-estate to display and process all available modalities shown. Our application layer transcoding decision process produces the same template if it is determined that the end-users device is capable of processing all available modalities. Hence this guarantees our solution meets the interface requirements of the current ePresence system.

Application layer transcoding becomes crucial for mobile devices accessing ePresence that have minimal multimedia processing capabilities as it determines which XSLT rule is needed to match the device's capabilities in processing the requested modality. Fig. 3(a) clearly displays this problem. Fig 3a(i) and 3a(ii) show how the display size affects the clarity of the delivered media for a PDA and Smart Phone (SP) on the current ePresence system. Hence, it would not be logical to stream all the available media to a PDA or Smart Phone. Fig 3a(iii) and 3a(iv) show how the application layer transcodor takes all these factors into account and logically adapts the available media and presentation format using XSLT rules. Note that other possible adaptations (i.e. choice of modalities) can be chosen for a PDA and SP. This ensures the proper delivery of material to meet the device needs while trying to meet the end users modality request.

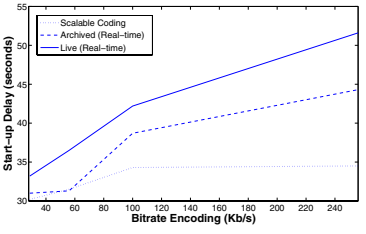
Bitstream Transcoding: Real-time video delivery applications must stream content with negligible delay in order to produce a live lecture experience. The Real Time Streaming Protocol (RTSP) [4] can deliver a continues flow of video (e.g. 28Kb/s for dial-up to a PDA, 100-300Kb/s for high speed to a PC) with negligible streaming delays . This invovles pre-buffering a portion of video at the user device to ensure that the buffer can maintain a steady flow of media to playback while the rest is streamed over the webcast session. Although RTSP can ensure a steady flow of media in the user device buffer, the effect of transcoding delay must also be addressed and how it effect this steady flow. Transcoding inevitably will cause delay due to the fact that video must be converting between formats (e.g raw video to MPEG-4 format) or scaled (e.g. change frame-rate).

One of the key challenges that designers of transcoding servers face is making sure that the streaming server always has a continues flow of buffered media to deliver. Fig. 3b shows an approximation of start-up delays at the beginning of an eL webcasting session as a result of transcoding. These results are from real

[4] sourceforge.net/projects/ffmpeg/
[5] http://developer.apple.com/darwin/projects/streaming/



(a) (i) PDA (no transcoding), (ii) SP (no transcoding), (iii) PDA (transcoded), (iv) SP (transcoded)



(b) Start-up delays incurred during scalable coding, archived and live real-time transcoding

Fig. 3. Results: (a) Application layer transcoding, (b) Bitstream transcoding

experiments using the above mentioned streaming framework to a user that is 50Km away from the eL system server. It is expected that scalable coding will producing the lowest overhead due to the fact that no transcoding is needed. The start-up overhead observed is due to pre-buffering at the user device. As mentioned in Section 3, scalable coding can only be used for on-demand eL systems and cannot be used for live sessions. eL sessions are archived using MPEG-4 encoding. This archived multimedia is transcoded by scaling the video and audio to match the end users environmental characteristics. As Fig. 3b shows, there is a noticeable delay increase in comparison to scalable coding (which skips the transcoding step). The amount of delay increases proportional to the encoding bit-rate. This is due to the computational intensity and memory usage that the transcoder experiences as the target encoding bit-rate (e.g. 100Kb/s) increases. For live eL sessions, the encoder requires the aid of a capturing device to transfer and synchronize live video and audio from devices (e.g. video camera). As shown in Fig. 3b, transcoding live session incurs the highest amount of delays. This additional overhead is due to two factors: 1) delay from the capturing device, 2) high computational intensity of transcoding raw video and audio to MPEG-4.

The encouraging conclusion that can be drawn from Fig. 3b is that start-up delay overhead is going to be roughly under one minute. Assuming that an eL session will be on average one hour in duration (e.g. course lecture), this is an acceptable delay for the end-user to cope with. The benefits of adding transcoding to the current ePresence architecture will be worthy of the tradeoff

of a one minute start-up delay that the system will incur. Note that this delay will probably increase as the user's distance from the server increases (e.g overseas).

5 Conclusions

This paper has addressed the problem of Universal Multimedia Access in the context of an e-Learning applications. More specifically, seamless delivery of multimedia content to diverse end-users in unreliable network conditions has been considered. The proposed solution is a real-time application level and bitstream transcoding solution that can be be accomplished with low delay overhead to deliver eL content to any user, despite there environmental restrictions.

References

1. Baecker, R., Moore, G., Zijdemans, A.: Reinventing the lecture: Webcasting made interactive. In: Proc. of HCI Int'l. Volume 1. (2003) 896–900
2. Bormans, J., Gelissen, J., Perkis, A.: MPEG-21: The 21st century multimedia framework. IEEE Signal Processing Magazine **20** (2003) 53–62
3. Deshpande, S., Hwang, J.: A real-time interactive virtual classroom multimedia distance learning system. IEEE Trans. on Multimedia **3** (2001) 432–444
4. Schulzrinne, H., Rao, A., Lanphier, R.: Rfc 2326: Real time streaming protocol (rtsp). Technical report (2004)
5. Timmerer, C.: Resource Adaptation using XML with the MPEG-21 Multimedia Framework. PhD thesis, Institut fur Informaionstechnologie, Universitat Klagenfurt, Germany (2003)
6. van Beek, P.: Metadata-driven multimedia access. IEEE Signal Processing Magazine **20** (2003) 40–52
7. Brotherton, J., Bhalodia, J., Abowd, G.: Automated capture, integration, and visualization of multiple media streams. In: Proc. of the IEEE Int'l Conf. on Multimedia Computing and Systems. (1998) 54–63
8. Cloete, E.: Electronic education system model. Computers and Education **36** (2001) 171–182
9. Paris, A., Simos, R., et al.: Developing an architecture for the software subsystem of a learning technology system-an engineering approach. In: Proc. of the IEEE Int'l Conf. Advanced Learning Technologies. (2001) 17–20
10. Vetro, A., Trimmerer, C.: Information technology-multimedia framework MPEG-21-part7: Digital item adaptation. Technical Report 21000-1:2007, ISO/IEC (2003)
11. Kinno, A., Yonemoto, Y., et. al.: Environment adaptive XML transformation and its application to content delivery. In: Symposium of Applications and the Internet. (2003) 31–36