UDC 621

H.T. Tran, S.M. Tran, M. Preda

Advanced video-on-demand scenario with the deployment of MPEG-4/AVC

Статья посвящена исследованиям, касающимся проектирования и применения системы видео по запросу. Для эффективного кодирования и запоминания содержимого мультимедиа-информации применяется самая современная методика сжатия видеоизображений MPEG4/AVC. Созданы удобные интерфейсы, как для поддержки сервисов, так и для их использования. Предлагаемое комплексное решение для системы видео по запросу позволяет не только просматривать видеоизображения в интерактивном режиме, как это сделано в других системах. Также предлагается возможность простого и эффективного запоминания изображения, его изменения и восстановления из памяти.

The work describes a research issue on designing and implementing a Video-On-Demand system. The latest technique of video compression – the MPEG-4 Advanced Video Coding – is deployed to efficiently solve the storage problem of multimedia contents. User-friendly interface for both service maintenance as well as service usage at end-user is also addressed. The proposed complete solution of Video-On-Demand service therefore not only inherits the interactive feature on watching video / movie as other system in the same branch, but also proves to be an efficient, simplicity-oriented way for storing, manipulating and retrieving multimedia contents.

Introduction

Storing and accessing continuous media (e.g. audio, video, animation,…) are the fundamental technology for all multimedia applications. The explosion in quality as well as quantity of multimedia contents in the digital era nowadays exposes – among the others – two crucial challenges to researcher as followings:

1. Service providers should make a large number of multimedia contents available to large-scale of users, who may be equipped with terminals having limited resources for the reason of low cost.

2. Among the possibly huge range of materials, service users should easily access the appropriate contents, which fit the most their taste or demand.

The emergence of Video-On-Demand (VOD) can be considered as an answer for the above questions. The technology itself ensures the mechanism toward the solution of the main problem: watching the willing content whenever demanded. However, the complexity of the issue, including efficient storage, economic transmission, well-organized handling and using,… cannot be achieved successfully by VOD individually. Other technologies on compression, networking and ergonomics must be also involved.

Toward such complete VOD solution, this paper introduces an issue on design and implementation of a VOD service, which combines a sophisticated video compression technique, namely MPEG-4 Advanced Video Coding (AVC) together with the VOD technology. The paper is organized as follows. The next section points out some key technique of VOD in general. Section 0 overviews the video compression standards in the MPEG (Motion Picture Encoding Group) family. Then the integration of the two technologies is presented in Section 0. The last section denotes some experiment and conclusion on the proposed system.

Overview on the underlying techniques of VOD service

Different from the traditional way of watching television with a preordered sequence of events defined by service provider, VOD applications add the means of interactivity to TV viewers. Users of the system no longer play a passive role while watching the broadcasted programs. Now they can decide what to see and when to see them. Typically, the interactivity in VOD means the possibility of selecting video and controlling its playback. Based on the behavior of control, VOD can be further classified into two groups. To provide a True Video-On-Demand (T-VOD) service, where users can watch any movies at any time, the system must reserve a dedicated channel at the video server and distribution network for each user. In contrast, Near Video-On-Demand (N-VOD) makes use of broadcast or multicast technology to enable multiple users to share a single video channel to reduce system cost substantially. The tradeoffs are limited video selection, fixed playback schedule, limited or no interactive control. In both scenarios, VOD system is generally comprised of server and client software application. VOD client applications run on users’ terminal (*e.g.* digital set-top box, computer, *etc.*) as native application, or they can be downloaded on the fly when subscriber accesses the VOD application via service access point (for instance, a website of the VOD server). In the rest of the paper, we concentrate on the T-VOD application. Therefore the term “VOD” should be understood as T-VOD except for specified otherwise.

The VOD architecture can be simply drafted as in Fig. 1. To select the movie to be viewed, the client interacts with the head-end through a cable connection or a Web-server through an Internet Service Provider (ISP) connection. Once the movie is selected, a Real Time Streaming Protocol (RTSP) session is established against a file containing the transport stream packets for the selected video. The packets are transferred to the client over Real Time Protocol (RTP) or MPEG-2 transports.

Typically, a VOD service requires the coordination of a broadcast with an interactive service. A management server (*i.e.* program scheduler. configuration and billing manager) coordinates the broadcaster and the ISP. Viewers select the movie to view using an interactive guide application delivered through the interactive network, which is typically IP-based (*e.g.* using the HyperText Transfer Protocol HTTP and RTSP protocols), but could also be Digital Storage Media Command Control (DSM-CC) based. Once a movie is selected, the media for that movie is delivered from a video streaming server. Receivers interface with both the interactive and broadcast networks and use local buffering and disk store as needed.

The relationship between the broadcast service provider (BSP) and the ISP in a VOD application is depicted in Fig. 1. The client terminal has a unidirectional video-in interface and a bidirectional IP-based interface. The video-interface connects to a downstream network typically controlled by a BSP, while the bidirectional interaction interface connects to an IP-based network typically maintained by some ISPs and backbone infrastructure organizations. The VOD service provider is responsible for coordinating both the unidirectional and the interaction channel.

Our VOD solution in Section 4 purely based on IP transmission. Therefore, in the rest of this section we will introduce shortly the associated protocols for the sack of completion. They are RTSP and RTP, which specify the necessary client-server interaction to stream multimedia presentations. We refer readers to reference section for detailed documentations on these protocols as well as the MPEG-based broadcasting DSM-CC Server.

The RTSP protocol [2] is a client-server multimedia presentation control protocol designed for efficient multimedia streaming over IP networks. It is not at all concerned with the actual delivery of the streams; that is achieved by the RTP (see below). RTSP is designed to work with time-based media, such as streaming audio and video, as well as any application where time-based delivery is essential. It also supports multicast-unicast hybrid solutions for heterogeneous networks like the Internet. The protocol is intentionally similar in syntax and operation to HTTP/1.1 6[6] so that extension mechanisms to HTTP can, in most cases, also be added to RTSP. To correlate RTSP requests with a stream identified as an RTSP Uniform Resource Locator (URL), RTSP servers need to maintain session state whose transitions are driven by control commands. The following commands are central to the allocation and usage of stream resource on the server:

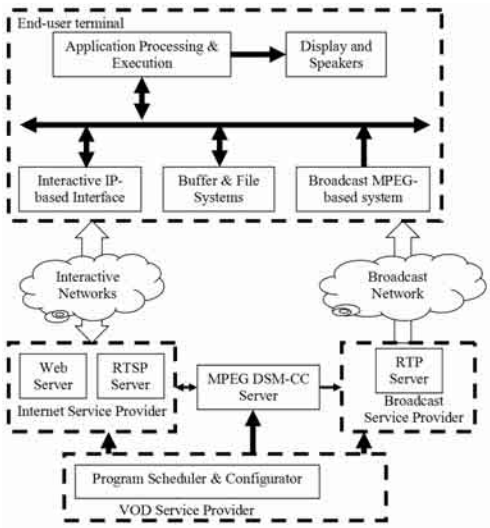

Fig.1. A typical VOD architecture; typically either RTP or DSM-CC is used, not both.

- *Setup*: Causes the server to allocate resources for a stream and start an RTSP session.
- *Play and Record*: Starts data transmission on a stream allocated via setup command.
- *Pause*: Temporarily halts a stream without freeing server resources.
- *Teardown*: Frees resources associated with the stream. The RTSP session ceases to exist on the server.

Note that the RTSP scheme requires that these commands are issued via a transport protocol that provides delivery guarantees, such as TCP 4. Though it may be implied from its name, RTSP does not provide real-time guarantee.

As opposed to RTSP, RTP [1] (being delivered on top of UDP [5] as oppose to TCP) does provide end-to-end network transport functions, which are suitable for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality of service for real-time applications.

RTP was originally designed with three application scenarios in mind:

– *Simple multicast audio conference*: Conference participant sends audio data in small millisecond chunks. Each chunk of audio data is preceded by an RTP header, and both are contained in a UDP packet, which will be fed into the Internet. The RTP header contains timing information and a sequence number that allow the receiver to reconstruct the timing produced by the source.

– *Audio and video conference*: When both audio and video media are used in a conference, they are transmitted as separate RTP sessions using two different UDP port pairs or multicast addresses. There is no direct coupling at the RTP level between the audio and video sessions. In the higher level (RTSP in case of VOD) must keep trace of the association.

– *Mixer and translator*: An RTP-level relay called mixer can make the adaptation between low- and high-speed networks so that narrow-bandwidth users can still receive reduced-quality media while broad-bandwidth users can enjoy the high-speed network access.

VOD clients need to be able to discover the available VOD programs on the server. It is done through the bidirectional interactive channel in Fig.1. However, the same mechanism used by IP multicast clients can also be applied. The Session Description Protocol (SDP) [3] provides the capability to discover multimedia sessions and supports session announcement, session invitation and other forms of multimedia session initiation. In general, SDP conveys sufficient information to be able to join a session (with the possible exception of encryption keys) and to announce the resources to be used to non-participants that may need to know.

Evolution of video compression in MPEG standard

As can be seen from the previous section, the stream controlled by RTSP may use RTP but not exclusively. That is the operation of RTSP does not depend on the transport mechanism used to carry continuous media. The encapsulation RTP in turn can be applied to various compressed types of data. In other words, deploying RTSP and RTP in VOD system, we can freely select the most efficient compression method to maximally reduce the redundancy occurring in the media data. In this section, we overview the international MPEG video compression techniques to select the best candidate for our VOD solution.

Two important standardization efforts for video compression were started in the late ’80s. One is the ITU-T (International Telecommunication Union Telecommunication Standardization Sector) standard for video-conferencing and video-telephony, known as H.261. The other one came under the name of MPEG (Moving Picture Experts Group) from ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) in order to define a video coding algorithm for application on digital storage media like CD-ROM (Compact Disc ─ Read Only Memory). In addition, audio coding is added into MPEG and the scope of the targeted applications extended to cover almost all applications, from multimedia system to VOD. MPEG’s first effort led to MPEG-1 standard that is used in various applications. CD-I (Compact Disc Interactive) and Video CD technology use MPEG-1 as the compression algorithm for audio and video. It was designed to support non-interlaced video coding up to 1.5Mbps with quality about the same as of the analog PAL (Phase Alternation by Line) or NTSC (National Television System Committee) system. MPEG’s second effort started in 1990. The main objective was to design a compression standard capable of different qualities depending on the bitrate, from TV (TeleVision) broadcast to studio quality. The work led to the MPEG-2 standard which is the expanded version of MPEG-1 with more sophisticated tools for audio video coding. The new standard is optimized for interlaced pictures, the popular format in TV technology. The MPEG-2 standard is capable of coding standard TV at about 4-9 Mbps. MPEG-4 is the result of another international effort in respect to the recent boom of multimedia applications delivered over the Internet. MPEG-4 enlarges the range of possible bitrate, starting from very low bitrate of 64 kbits/s or under but still ensuring the reasonable quality of the application. The standard is the first one, which addresses the composition as well as the interaction of a multimedia content. Also driven by the increasingly pervasive role of multimedia contents, MPEG started a new work item ─ MPEG-7 ─to provide a solution to high level understanding of media data, which is very useful for effective information retrieving and filtering from a huge database of media data. The latest effort of MPEG on standardization of multimedia framework, namely MPEG-21, aims at defining a normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain. Content-creators, producers, distributors and service providers will be provided with equal opportunities in the MPEG-21 enabled open market. From the viewpoint of content-consumers, they will benefit from the access to a large variety of content in an interoperable manner. Fig.2. summarizes the current members of MPEG family and their main objectives. In the scope of our VOD system, the MPEG-4 standard is the latest member that matches our requirement: a sophisticated compression method for video data.

Concerning the video compression issue, the standard provides two main methods for encoding video data: MPEG-4 video encoding part 2 and Part 10. The latter one – Advanced Video Coding (commonly referred as H.264/AVC) [7] – is the newest entry in the series of international video coding standards. It is currently the most powerful and state-of-the-art standard, and was developed by a joint video team consisting of experts from ITU-T's Video Coding Experts Group (VCEG) and MPEG. As has been the case with the past standards, its design provides the most current balance between the coding efficiency, implementation complexity, and cost – based on state of the integrated circuit design technology (CPU's, DSP's, ASIC's, FPGA's, *etc.*). In the process, a standard was created that improved coding efficiency by a factor of at least about two (on average) over MPEG-2 – the most widely used video coding standard today – while keeping the cost within an acceptable range.

H.264/AVC was developed over a period of about four years. The roots of this standard lie in the ITU-T's H.26L project initiated by the VCEG, which issued a Call for Proposals (CfP) in early 1998 and created a first draft design for its new standard in August of 1999. In 2001, when MPEG had finished development of its most recent video coding standard, known as MPEG-4 Part 2, it issued a similar CfP to invite new contributions to further improve the coding efficiency beyond what was achieved on that project. VCEG chose to provide its draft design in response to MPEG's CfP and proposed joining forces to complete the work. Several other proposals were also submitted and were tested by MPEG as well. As a result of those tests, MPEG made the following conclusions that affirmed the design choices made by VCEG for H.26L:

– The motion compensated Discrete Cosine Transform (DCT) structure was superior to others, implying there was no need, at least at that stage, to make fundamental structural changes for the next generation of coding standard.

– Some video coding tools that had been excluded in the past (for MPEG-2, H.263, or MPEG-4 Part 2) due to their complexity (hence implementation cost) could be re-examined for inclusion in the next standard.

– The integrated circuit technology had advanced significantly since the development of those standards and this had significantly reduced the implementation cost of those coding tools. (This was not a "blank check" for compression at all costs, as a number of compromises were still necessary for complexity reasons, but it was a recognition that some of the complexity constraints that governed past work could be re-examined).

– To allow maximum freedom of improving the coding efficiency, the syntax of the new coding standard could not be backward compatible with prior standards.

ITU-T's H.26L was a top-performing proposal, and most others that showed good performance in MPEG had also been based on H.26L (as it had become well-known as an advance in technology by that time). Therefore, to allow speedy progress, ITU-T and ISO/IEC agreed to join forces together to jointly develop the next generation of video coding standard and use H.26L as the starting point. A joint video team, consisting of experts from VCEG and MPEG, was formed in December, 2001, with the goal of completing the technical development of the standard by 2003. ITU-T planned to adopt the standard under the name of ITU-T H.264, and ISO/IEC planned to adopt the standard as MPEG-4 Part 10 Advanced Video Coding, in the MPEG-4 suite of standards formally designated as ISO/IEC 14496. As an unwanted byproduct, this standard gets referred to by at least six different names – H.264, H.26L, ISO/IEC 14496-10, JVT, MPEG-4 AVC and MPEG-4 Part 10.

Efficient VOD based on AVC

As seen in Section 0, deploying the RTSP and RTP protocols, we have the "routing traffic" and "carrying trucks" to facilitate the data-communication in a VOD application. To make the service efficient with minimum number of "traffic-jams" – leading to losing data – the data pumped into the carrying trucks must be the most compact as possible. That is, the data encapsulated in RTP packet must be compressed in the most cost-effective manner. The overview in Section 3 already name the most appropriate compression method – the AVC – for us to handle the most bit-consuming video data. Fig. 3. depicts the topology of our VOD scheme integrated with AVC compression technique.

The proposed VOD solution is based on IP network. Two servers for Web interface and streaming feature together with two types of users – super user and end-user – are all connected to the Internet. *Web Server* accepts commands from super user or *Administrators*, then redirect them to *Streaming Server* to keep the latter server up-to-date with the settings, maintenance of the *Administrator*. End-user can discover the available VOD program from the *Web Server*. Possessing the URL of the selected program, user then can connect to Streaming Server to obtain the content itself.

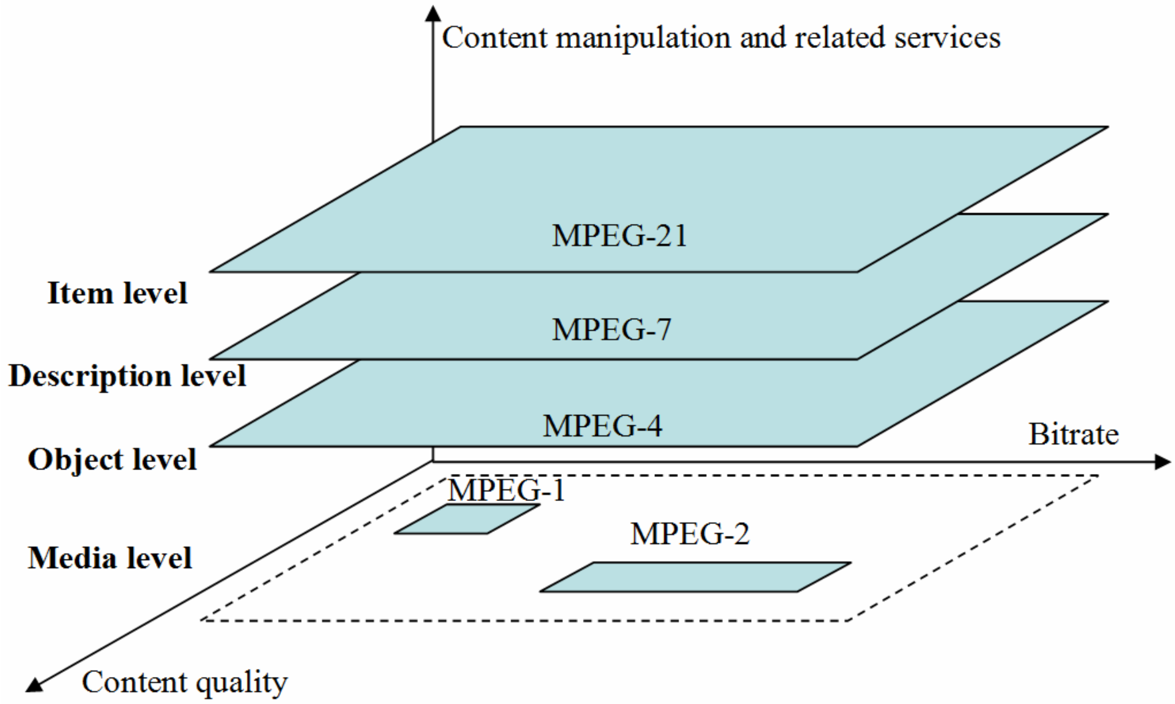

Fig. 2. Members of MPEG family

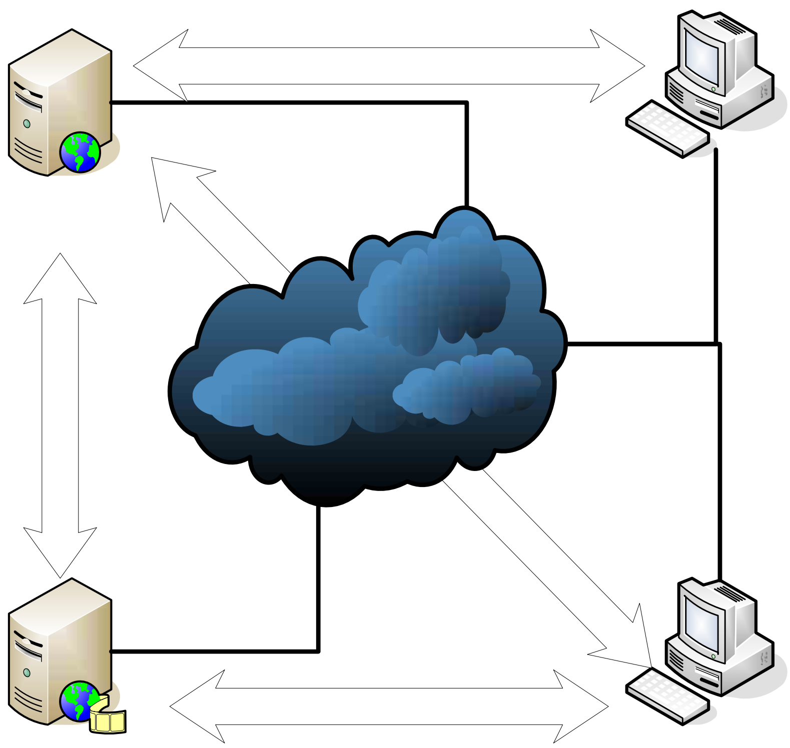

Fig. 3. The topology of the proposed VOD service.

Figure 4 exposes a deep look inside the operation of the VOD system. The dashed boxes can be considered as modules of the system, representing distinct functional entities on separate hosts. Note that, two hosts may not be necessarily installed in physically different computer machines. Three modules / components can be clearly distinguished in the VOD system; they are hosted by Web server, streaming server and end-user terminal.

Web server module: It consists of three interfaces (three Web pages) to aid users to interact with the system. The *Administrator interface*, which is guarded by login and password, is used by *Administrator* to activate as well as setting the attributes of a VOD program. From this main interface, Administrator can access the *Upload interface* to add a multimedia to the resource of the VOD system. The movies being uploaded are pre-encoded with AVC encoder also provided by Artemis. The current version of our AVC encoder supports avi file and MPEG-2 transport stream as input format; the output formats can be the byte syntax Annex B [10] or the pure MPEG-4 file [9] (no system information, having only one track atom for video data). After encoding phase, the AVC content can be uploaded to the Streaming server thanks to *Upload interface* and *File management*. The *Administrator interface* keeps trace of all uploaded contents as well as their settings. General users can check these information (with reading authority only) via *Browser interface*.

Streaming server module: It is a customized core of the VideoLan project [15], which is of type open source code. The framework for VOD objects in VLC (VideoLAN Client) media player of this project is exploited in our VOD system. In fact, in order to discovery the available VOD programs on the streaming server, we replaced the Session Announcement Protocol (SAP) of VLC with HTTP protocol (interactive or automatic access on the *Browser interface* in Figure 4) to ensure the accessibility of the service discovery. That is because the SAP adopts multicasting mode over the Internet, which may be disable on certain segment of the network to avoid flooding up the system.

End-user module: For the sack of ergonomics, we developed a C++ based, small client application with user-friendly interface. It assists user in browsing the available VOD programs together with their properties and access mode. After making the choice, the client application will launch VLC as an external player to connect to the targeted server and handle the received media stream.

Figure 5shows some typical interfaces of the processes in our VOD system and the relation between them. The arrows demonstrate a typical workflow of a VOD operation.

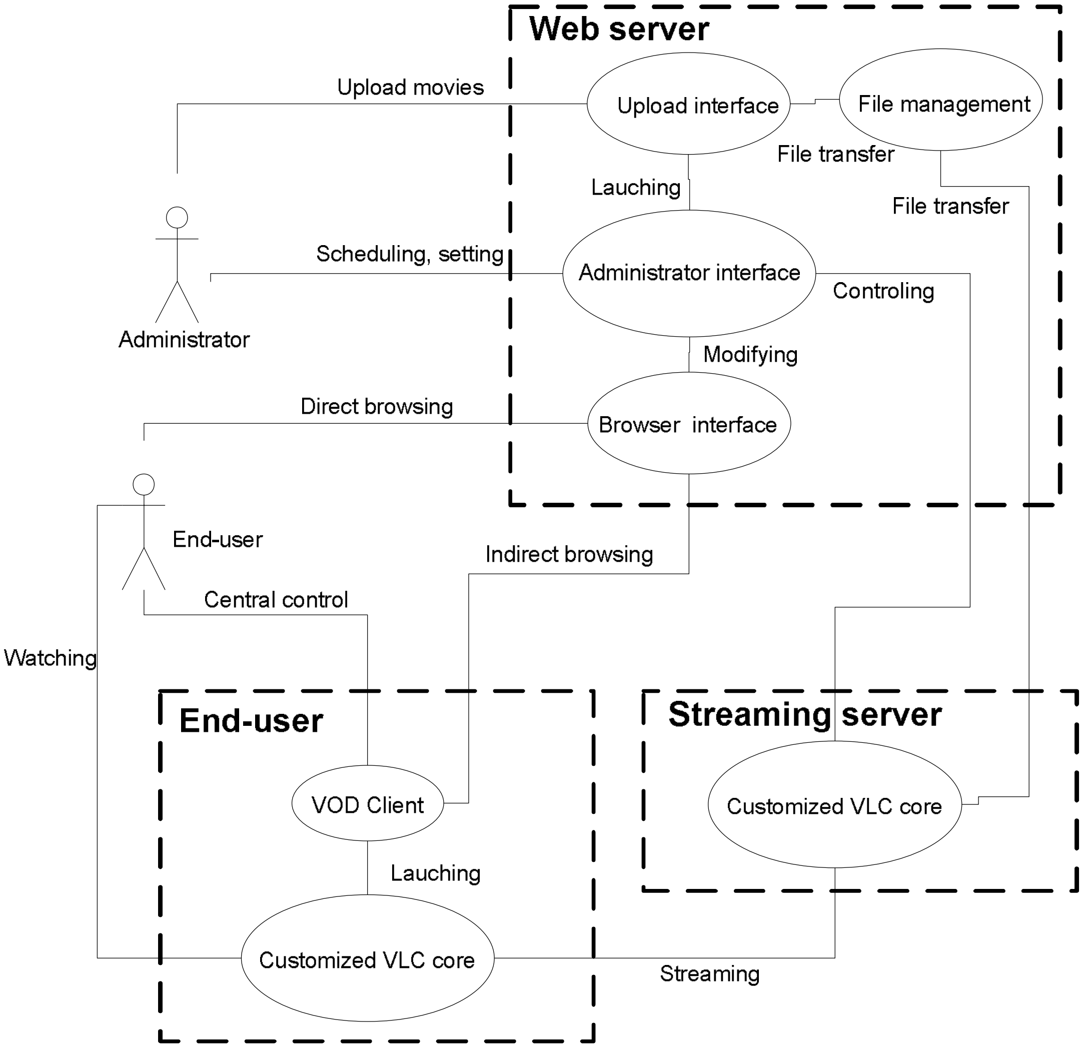

Fig. 4. Use case of the proposed VOD service

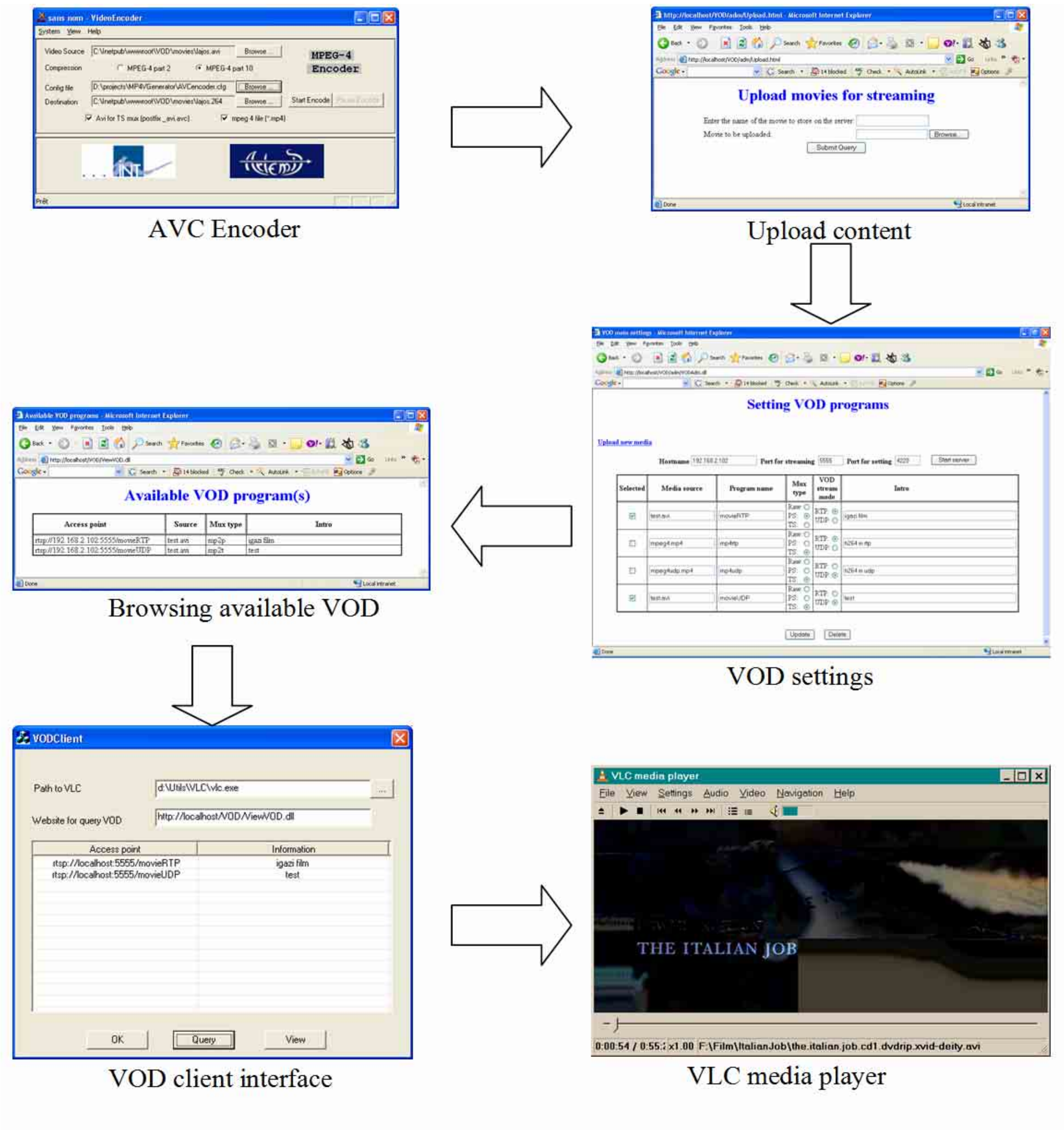

Fig. 5. Some snapshots of the interfaces of the VOD system

Conclusion

The paper introduces a design and implementation of a complete VOD service. It deploys the latest AVC encoding standard to efficiently transmit the bit-consuming video information. Besides implementing the mechanism of interactive streaming driven by users, we also put focuses on the convenience of the service: how conveniently service provider can deal with VOD programs, how simply user can access the service.

For larger capacity of VOD server with thousands or millions of VOD object in their resource, setting and maintaining the information of these objects should be registered into a database for safe and fast access. We are investigating on involving the database server into our VOD topology. To become a business product on the market, billing service is indispensable. Exporting audio from MPEG-2 transport stream (currently we only extract video out of this format) then transcoding / remuxing it back with the associated video in MPEG-4 format in the VOD scenario is also among our perspective target. Then we can make a translator for the already widely expanded MPEG-2 multimedia network to a more modest bandwidth network such as ISDN or mobile phone subscribers may have a great interest. Furthermore, in such narrow bandwidth channel, impact on quality is moderate thanks to sophisticate compression techniques like AVC; interactivity is added as in the VOD scenarios. These facts just imply a great demand on VOD in the near future.

References

1. *RTP:* A transport Protocol for Real-Time Applications, RFC 1889, Jan 96, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc1889.txt.
2. *Real-Time* Streaming Protocol (RTSP), RFC 2326. Apr 1998, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2326.txt.
3. *SDP*: Session Description Protocol, RFC 2327, Apr 1998, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2327.txt.
4. *Transmission* Control Protocol RFC 793, Sep 1981, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc768.txt.
5. *User* Datagram Protocol RFC 768, Aug 1980, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc768.txt.
6. *Hypertext* Transfer Protocol, HTTP/1.1, RFC 2616, June 1999, IETF, ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt.
7. *ISO/IEC* JTC1 / SC29 / WG11 N4350, Information technology — Coding of audio-visual objects — Part 2: Visual, July 2001.
8. *ISO/IEC* JTC1 / SC29 / WG11, Information technology — Coding of audio-visual objects — Part 6: Extensions for DSM-CC.
9. *WD* 3.0 ISO/IEC14496-1:2002, Information technology — coding of audio-visual objects: — Part 1: system, August 2002.
10. *Joint* Video Team of ITU-T and ISO/IEC JTC 1, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC),” document JVT-G050r1, May 2003; technical corrigendum 1 documents JVT-K050r1 (non-integrated form) and JVT-K051r1 (integrated form), March 2004; and Fidelity Range Extensions documents JVT-L047 (non-integrated form) and JVT-L050 (integrated form), July 2004.
11. *E. M. Schwalb*, iTV Handbook: Technologies and Standards, IMSC Press Multimedia Series, Prentice Hall PTR, 2004.
12. *L. A. Rowe*, J. S. Boreczky, D. A. Berger, D.W. Brubeck, J. E. Baldeschwieler, “A distributed hierarchical Video-On-Demand system”, http://bmrc.berkeley.edu/research/publications/ 1995/105/ICIP95.html.
13. *J. L.B. Lee*, “UVoD: An Unified Architecture for Video-On-Demand Services”, IEEE Communications Letters, V. 3, No 9, September 1999, p. 277-279.
14. *G. J. Sullivan*, P. Topiwala, A. Luthra, “The H. 264/AVC Advanced Video Coding standard: Overview and Introduction to the Fidelity Range Extension”, SPIE Conference on Applition of Digital Imgae Processing XXVII Special Session on Advances int he New Emerging Standard: H264/AVC, August, 2004.
15. *VideoLan* VLC Media player, http://www.videolan.org.