Georg-August-Universität
Göttingen
Institut für Informatik

ISSN 1611–1044
Nummer IFI–TB–2005–06

Technischer Bericht

Architectural Thoughts and Requirements Considerations on Video Streaming over the Internet

Jun Lei, Ingo Juchem, Xiaoming Fu and Dieter Hogrefe

Technische Berichte
des Instituts für Informatik
an der Georg-August-Universität Göttingen

November 2005

Georg-August-Universität Göttingen
Institut für Informatik

Lotzestraße 16-18
37083 Göttingen
Germany

Tel.	+49 (5 51) 39-1 44 14
Fax	+49 (5 51) 39-1 44 15
Email	office@informatik.uni-goettingen.de
WWW	www.ifi.informatik.uni-goettingen.de

Architectural Thoughts and Requirements Considerations on Video Streaming over the Internet

Jun Lei, Ingo Juchem, Xiaoming Fu, Dieter Hogrefe
Institute for Informatics, Universität Göttingen
Lotzestr. 16-18, Göttingen 37083, Germany
Email: {lei,juchem,fu,hogrefe}@cs.uni-goettingen.de

November 17, 2005

Abstract

With increasing demands of multimedia information over the Internet, video streaming has been received explosive attentions. With respect to the real-time nature of video streaming, instable bandwidth, latency, noise, packet loss, retransmission and out of order packet delivery are all problems that can affect video streaming over the Internet. However, the traditional Internet traffic is not sensitive to these problems. Based on the general video streaming architecture, we give out some considerations on design and architectural mechanisms, namely, media server, media compression, media QoS control, media distribution services, media security mechanisms and protocol stacks for video streaming. For each of these areas, we present some existing methods and implementations. Then we propose architecture via overlay multicast integrated with proxy caching to achieve efficiency, flexibility and scalability. Finally, we conclude this issue and point out the research direction.

Keywords: video streaming, Internet, video compression, QoS control, media server, protocol

1 Background

Video streaming has been showing the potential to delivery a large amount of content, which can not be accomplished by broadcast,to broader users in a convenient way. Therefore, it could be used in lectures, sports, and entertainment or industry events. This technology enables simultaneous delivery and playback of a video. In general, video delivery consists of two species: video streaming and video delivery by download. Real-time video streaming has no possibility of prediction so that it is more difficult to accomplish. For On-demand video streaming, the general process is as follows: 1) partition the compressed video

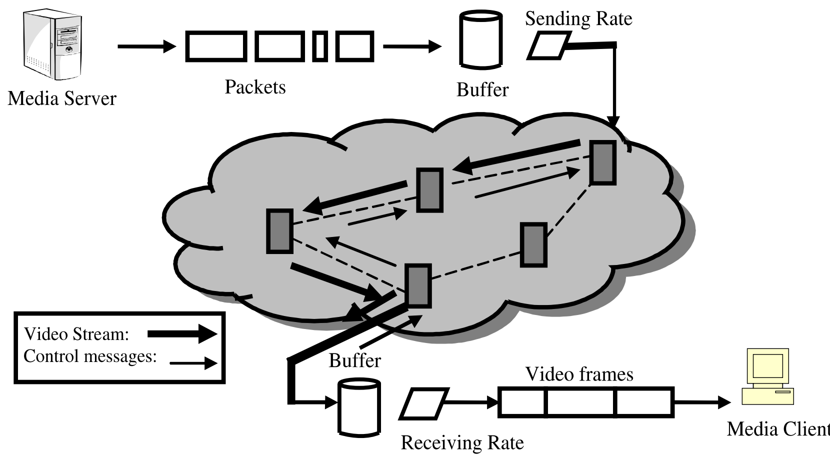

Figure 1: Basic architecture for video streaming over Internet

into packets; 2)start delivery of these packets to clients; 3) begin decoding and playback at the receiver when the other portions are still being delivered. Advances in computing and networking technology have made it feasible to deliver video streams across the Internet. To enable various use of video over the Internet, multicast must be added into the picture. The purpose of this document is to identify a selected set of issues that are introduced by video streaming over the Internet and prompts some architectural considerations towards effective and likely deployable video streaming architecture.

Firstly,we show a basic architecture for video streaming over Internet in Figure 1. Here, video data are firstly stored at a media server. Upon the client's request, the media server retrieves the video data. Then the transport protocols packetize the bit-streams and these packets may be dropped or delayed during transmission over Internet, e.g., due to network congestion. To improve the quality of video, a buffer is deployed before sending video data over the Internet, which is used to control the rate according to the current network status and QoS requirements. Before really starting to play at the side of the client, there is another buffer set to balance the receiving rate and playing rate. After that, the video is decoded and then replayed properly at the client. From this figure, several areas, e.g. media server, video operation, application-layer QoS control, are closely related and coherently constitute the video streaming architecture. In the next section, we will briefly describe some demands in each of above areas as requirements.

2 On requirements for video streaming

The Internet firstly was not designed for multimedia streaming. It is a shared medium and uses a *best effort* delivery mechanism, Internet Protocol

(IP) to deliver content. With respect to the real-time nature of video streaming, inconsistent bandwidth, latency, noise, packet loss, retransmission and out of order packet delivery are all problems that can affect video streaming over the Internet. However, the traditional Internet traffic is not sensitive to these problems. This section attempts to list some of these problems.

The important goal of video streaming is to perform the streaming in a manner that the sequence of constraints [1] should be met. Any data that is lost in transmission can not be used at the receiver, that is to say, the sequence of packets is vital to video streaming. Firstly, it needs to estimate the available bandwidth and adjust the transmitted video bit rate to the available bandwidth. Sometimes, there are various bandwidth requirements if a single sender streams data to multiple receivers. Secondly, the variation in end-to-end delay, i.e. delay jitter can affect the quality of streaming video as well. Thirdly, wired network may be affected by entire packet loss; wireless channels are afflicted by both bit errors and burst errors. In this way, the bandwidth constraint should care about not only the amount of bandwidth but also the consistency and quality of this bandwidth. Current streaming technology get around this problem by setting a buffer to contain a certain amount of content before really starting to play. So the part of file, generally, had to be completely downloaded before playing could commence. Otherwise, video streaming stops. Meanwhile, it may cause another problem that users have non-sequential access request [1]. For example, one client needs to start replaying from the middle of a video. Then it is difficult to retrieve this special part of a video.

Sometimes, it is not all about the quality of bandwidth; content creation, serving, usability and availability are also challenges that need to be overcome. Quality of service (QoS) mechanisms has been the focus because today's Internet lacks support for QoS assurance, which makes the transmission of video more challenging. Furthermore, the heterogeneity of the Internet's transmission resources and end-systems makes it to support different traffic characteristics among multiple receivers of the same video stream. When we refer to the security issues, there are still some problems for video streaming, including the authentication and authorization mechanisms, key management approaches for data confidentiality and flexibility and reliability in inter-domain communications. Finally, the access network technology may be another point of problems.

According to the above challenges in video streaming, we briefly describe some requirements in 4 areas as follows.

[1] Consider the time interval between displayed frames to be denoted by Δ, e.g. Δ is 33 ms for 30 frames/s video and 100 ms for 10 frames/s video. Each frame must be delivered and decoded by its playback time; therefore the sequence of frames has an associated sequence of deliver/decode/display deadlines:

1. Frame N must be delivered and decoded by time TN
2. Frame N+1 must be delivered and decoded by time TN $+\Delta$
3. Frame N+2 must be delivered and decoded by time TN $+2\Delta$
4. Frame ...

1. Media nature related requirements. It is the elementary demand for video streaming. To cope with the real-time feature and current Internet status, raw video should be processed before sending over the Internet. Then at the side of clients, these video might be processed again including decode, synchronization, error control. Steinmetz [2] and Little and Ghafoor [3] have presented several methods for the formal synchronization requirements in a multimedia environment.

2. Video streaming architecture requirements. The architecture plays a key role in the providing streaming services. To offer qualified multimedia services, video streaming architecture includes media server, network filtering, network monitor and client. Meanwhile, media server involves operating system,storage system and communication system. Using network filtering can maximize the video quality during network congestion. To cope with QoS mechanisms, network monitor is required to reflect the real-time network status, e.g. bandwidth utility, packet loss, delay.Moreover,we consider multicast delivery for media streams including IP multicast and overlay multicast methods.

3. Networking requirements. To satisfy heterogeneity demands of clients, different control mechanisms should be integrated into the basic architecture. Since Internet's best effort and unicast service model lacks both efficient multicast routing and QoS guarantees needed for high-quality video delivery.It also needs to concern about QoS control and different protocol stacks for video streaming. According to the functionalities, these protocols including network-layer protocols, transport-layer protocols and session control protocols are directly related to Internet streaming video.

4. Security requirements. Security issue has becoming more and more important in every area. However, there are still some weaknesses existing in multimedia streaming such as authentication and authorization mechanisms, access control and key management approaches for data confidentiality.

3 Architectural thoughts towards video streaming

In this section, we prove the architectural thoughts for video streaming corresponding to the above identified requirements. In each area, some existing problems and solutions are presented.

3.1 Media source processing

Since video transmission needs a large amount of bandwidth support, compression is usually required to achieve the transmission efficiency. Due to the special features of multimedia application, various media streams must be presented in a synchronized fashion. Otherwise, the displaying frames on the screen may not match with the sound heard by the client, which is disturbing. Some error control mechanisms are required as well, which will be shown in Section Error Control. Video will suffer from some errors caused by packet loss, packet delay or packet fragmentation.

3.2 Video compression

To achieve efficiency, raw video must be compressed before transmission. Video compression uses actually a coding/decoding system to standardize video types [4] [5]. Then at the side of clients, the encoded video data would be decoded and played in a proper way. Here, we only concern about MPEG type. The Motion Picture Experts Group (MPEG) has three open (ISO/IEC) standards which can be used for streaming.

MPEG-1 [6] is designed for a group of video and audio coding standards. It originally developed for Video Home System (VHS)[7] quality video on CD-ROM in 1988. MPEG-1 is mainly considered as a storage format and it does offer excellent streaming quality for the bit-rate it supports.

MPEG-2 [8] was published in 1994. It was typically used to encode video and audio for broadcast applications. It is also the coding format used by standard commercial DVD movies. MPEG-2 creates a video stream out of three types of frame data: intra-frames (I), forward predictive frames(P) and bidirectional predicted frames (B),that can be arranged in a specified order called the Group of Picture(GOP) structure.As an Internet streaming technology,it is probably not useful as it uses bit rates higher than those to which almost everyone has access.

MPEG-4 [9] [10] was introduced in 1999 and is a standard specifically developed for web streaming media and CD distribution, conversational services. It absorbs many of the features of MPEG-1, MPEG-2 standards and other related standards. It includes the features with the ability to represent audio, video, images, graphics and text as separate objects, and to multiplex and synchronize these objects into scenes. The MPEG-4 standards provide embedded error resilience capabilities to detect and localize errors recovering data, and to visually conceal the impact of errors.

The common video compression schemes like H.26x and MPEG-n standards take basis on a set of encoding principles. From the point view of a video source, video compression can be classified into two approaches: scalable and non-scalable video coding [11]. Today, the scalable video type is most commonly used. Scalable video encoder compresses a raw video into multiple substreams. One of them is the base substream and others are enhancement substreams. The base substream can be independently decoded and provide coarse visual quality; enhancement substream is decoded with base substream together to provide enhanced video quality. According to different network conditions, the receiver selects different quality levels of video. Using scalable video encoding provides a alternative solution to meet with heterogeneous demands of clients. More recently, the use of path diversity has been studied as an option to provide extra dimension of adaptability to video application. Apostolopoulos et al [12] investigated a method that relies on a simultaneously transmitting several substreams of the video over different paths, in which each substreams encodes a partial description of the video. The video can be decoded correctly, even if some of the substreams are missing.

3.3 Video synchronization mechanisms

Video synchronization is generally implemented at the side of clients. The clients should present video streams in the same way as they were originally generated by the media server. The major issues for media synchronization include how to specify and implement synchronization. Video synchronization falls into at least two types: intra-media, inter-media synchronization. These two types of synchronization have direct connection with three semantic layers of multimedia data: media layer, stream layer and object layer [13].

- *Intra-media synchronization*: It refers to the time relationship between presentation units of one media object. An example for this relation is the time between single frames of a video sequence. To ensure the video received with required jitter,throughput and latency, the real-time constraints must maintain across a single continuous media connection. Without this type of synchronization, the video may pause or stop.
- *Inter-media synchronization*: It is more complicated and concerned with the temporal relationships among different continuous media. A prominent example of inter-media synchronization is lip-synchronization scenario. Without inter-media synchronization, the movements of the lip of a speaker don't match the presented audio.

We need an integrated method with both intra-media synchronization and inter-media synchronization to adjust video, in order to accommodate audio under time constraints. As usual, three phases are involved in solving this problem: 1)normalized clock times which represents a relationship between the clock of the media server and the clocks of the destinations; 2)normalized relative time-stamps which should be preserved among the media data from the media server to the destinations; 3)detection of asynchrony which is based on policies to trigger the synchronization mechanism if the media streams are out of synchrony.

AT&T Labs presented a flexible framework for synchronization of multimedia streams. They utilize two collaborative modules, a transmitter-driven and a local inter-media synchronization module, to synchronize the incoming streams [14]. Whenever the first one is not enough to ensure reliable synchronization or cannot guarantee synchronization because the encoder does not know the exact timing of the decoder, the second one comes into play.

3.4 Media Server

Media server is the key role in providing streaming services. Normally, it needs to store pre-processed video data using compression algorithms and then save them into a selected storage device. For high-qualified video services, media servers are required to handle the multimedia data under time constraints and to provide interactive operations such like play/pause/stop, fast forward/fast backward. So, a media server typically consists of three sub-elements: a storage system, an operating system and a communication system. But in this section, we only discuss about the storage systems and operating systems. The communication systems are refer as the communicator involved with application layer QoS control and transport layer protocols implemented at the media server. So, detailed information will be shown at the "Media distribution services" and "Protocol stacks for video streaming" sections.

3.4.1 Storage systems

In this system, it firstly concerns about video content creation. The designer of the content will use various production tools to create the content. These tools convert video, or animation to video type format that the server can stream later. Production tools can epitomize the content for efficient delivery over the Internet, based on the nature of the material and the capabilities of the client computers. Local experience has shown that it is not usually sufficient to simply encode existing video content for streaming.Content producers need to be cognizant of the tremendous compression ratios that are common in this field.

Secondly, a storage system for processed video resources have certain features including high-throughput, large capacity and error-tolerance [15]. Although current technology can provide large disk capability for applications, frequent requests and high-throughput need reliable and flexible storage system. To support large-scale demands, hierarchical storage architectures are proposed. There is a "priority" exists in hierarchical storage architecture, where video files with high-priority, namely, frequently requested video streams are kept on disks or quick access devices; the remainder stays at the automated tape library. A storage area network (SAN) [16] [17] and network attached storage (NAS) [18]are two examples for large-scale streaming services. To prevent the disk errors, redundant information should be stored.On the other hand,it wastes resources. So, there is a trade-off between reliability and cost.

3.4.2 Operating systems

Operating systems are the bridges between hardware and applications. They are always used to satisfy more concrete interactive operational demands. Video streaming has extremely strict constraints on this issue. Existing systems can provide acceptable playing mechanisms; however, interactive operations like VCR have little achievements.

Interactive operations connect with server capabilities, such as file management, storage capabilities; network status like network congestion, packet loss; quality of video files and clients¡¯ capabilities with CPUs, memories and storage devices. Effective and flexible operating systems are required to fit into different implementation mechanisms.

3.5 Media QoS control

Due to the video features, application level QoS mechanisms are suggested to fit into inconsistent network and different requirements of users, such as rate control which is desirable for video streaming applications to employ congestion control to avoid network congestion.For video streaming, congestion control commonly relies on rate control by adapting the sending rate to the available bandwidth of networks. It is also one of required considerations for video streaming architecture.

3.5.1 Congestion control

Congestion control is used to prevent packet loss and to reduce delays. Current congestion control mainly refers to end-to-end algorithms executed by a transport protocol in order to regulate the rate of video stream to the available bandwidth. Existing rate control mechanisms can be classified into 3 categories: endpoint-based, hybrid rate control and path aggregates.

Under endpoint-based error control [19] [20], source and receiver are respectively responsible for adapting the video transmission rate. They all need feedback information about the network status. Based upon this information, the sender/receiver regulates the rate of video stream. On the other hand, this control schema should cooperate with scalable video for clients to choose appropriate quality level of video streams.

For hybrid rate control, both the sender and receivers regulate the rate of videos streams together. The example of hybrid rate control can be found in a layered multicast scheme [21]. Just reducing the bit-rate of encoded video to alleviate the congestion will also degrade the visual quality.Therefore, it needs to adjust the video bit rate to the available network bandwidth. This problem has been studied by several researchers, including studies by Lam et al. [22]; and Cuetos and Ross [24], which adopted TCP as the network transport.

Previous researches have shown that the use of path aggregates can overcome the bandwidth deficiency [25] [26]. The reduction in effective loss and delay is achieved by smart multiplexing and selecting the high latency paths to user¡¯s advantage.

3.5.2 Error control

Some conditions cause the diminished stream quality: network congestion; server or router failures; packets arriving out of order; packets arriving late; packet fragmentation. If one of the fragmentations is dropped, the original datagram must be fragmented again and retransmitted. The main reason for media fragmentation is that network can not support more than 1480 byte packets while generally, media encoder produced packets of 1000-4000bytes. So network has to break encoded packets into smaller fragments to fit for the network. When there is a fragment lost, entire packet is lost. Therefore, some researches using error control mechanisms to address these problems.

From view point of different compensation type, there are 4 methods used to mend the video error: 1) Link-layer error control; 2) retransmission-based; 3) error concealment; 4) error-resilient video coding.

Link-layer error control includes Forward Error Control (FEC) [27] and Automatic Repeat Request (ARQ)[28]. FEC is used to add redundant information so that the original message can be reconstructed in the event of bit errors; while ARQ is suitable for the varying wireless channel condition; the receiver notifies the source only when the packets are corrupted and needed to be retransmitted.

Associated with FEC and ARQ to deal the problems, type-II hybrid ARQ has constraint on the maximum number of retransmissions for a packet [29] [30]. N_r, the maximum numbers of retransmissions, is pre-defined and fixed. When the receiver detects the loss of packet N under the condition of $T_c+RTT+D_s < T_d(N)$, where T_c is the current time;D_s is a slack term; $T_d(N)$ is the time when packet N is scheduled for display, the receiver sends the request for retransmission of package N to the sender.

Retransmission-only error control is known as ineffective for real-time communication in multimedia applications due to its latency. However, *Buffer-controlled Retransmission-based Error Control (BREC)*[31] provides a conditional retransmission method with the control of buffer, which can be chosen for real-time media applications.

Error concealment comes into play by hiding errors from human perception. When it refers to one-way systems like video broadcasting, it is much more important because it is impossible to feedback and retransmit [32] [33] the media streams. The proposed scheme using the most possible error bits to conceal errors by flipping the "chosen" error bits pre-selectively.

The video error resilience techniques can be classified into the followings: 1)encoder based techniques;2)decoder based techniques;3)interactive based techniques; 4)proxy based techniques. An overview of video error resilience techniques is presented in [34]. Ge, Peng et. al. compare different error resilience algorithms for video multicasting on WLANs [35].

3.6 Media distribution services

Built on the top of IP protocol, media distribution services are used to achieve QoS and high-efficiency for streaming video over Internet. Streaming availability on the global Internet should ideally mean a server ready to stream content to any clients who have an interest in receiving it. Unfortunately the demand and availability of rich media content has lead to a breakdown of the traditional client server model. So video streaming is developed from one-to-many multicast video into many-to-many. Single server streaming content to diverse groups of clients across the Internet are ineffective in terms of both server load and network congestion. In our study, media server replication, inter-mediate server delivery and application level multicast are identified as most promising approaches. For multicast, current research on these domains has been mostly dedicated on application level multirate multicast, due to deployment issues of IP multicast [36].

3.6.1 Media content replication

Content replication is an effective method to improve the scalability of a media delivery system. It has some advantages, such as reduce the latency for clients, and reduce bandwidth occupation on network links. Caching and mirroring are usual ways to make copies of the original media files. But this method may not be feasible due to the large amount of video data and its low resource utility. Furthermore, a critical problem is replica placement, i.e. how many replicas should be placed and where to place them.

3.6.2 Inter-mediate server delivery

Different from content replication, inter-mediate server delivery techniques can take advantage of the resources to reduce the server burden and network resource occupation. The replication placement is based on the knowledge of long-term data access patterns, caching at inter-media server takes advantage of short-term locality of client requests. Gao et, al. [37] proposed a proxy-assisted catching which makes full use of sever channels for periodic broadcasting. The clients can join an on-going broadcast cycle at any time instead of waiting for the next broadcast cycle. They catch up with the current cycle by retrieving the missing initial frame from the local proxy server by a separate unicast channel.

Liu et, al. [38] find out that previous researches,combining proxy caching with video layering or transcoding, suffer from either coarse adaptation granularity or high computation overhead. So this model allows selected videos to partially cache at the proxies. Once a proxy intercepts the request from a client, it sends directly the prefix (cached portion) to the client and noncached portion, if necessary, will be fetched from the media server. Firstly, this method can fit for more complicated access networks such campus network. Secondly, it allows clients to terminate a video play permanently after request from the beginning of a video. Most importantly, this system supports for scalable adaptive videos.

Liu et al. [39] provide a cooperative system with both proxy caching and client caching. It combines the advantages of both proxy caching and peer-to-peer clients' communications. By using an efficient cache allocation algorithm, this system distributes video segments among the proxies and clients.

3.6.3 Application level multicast

IP multicast [23] was the first solution for multicast functionality in the Internet.In addition, multicast delivery is very flexible and enable a large number

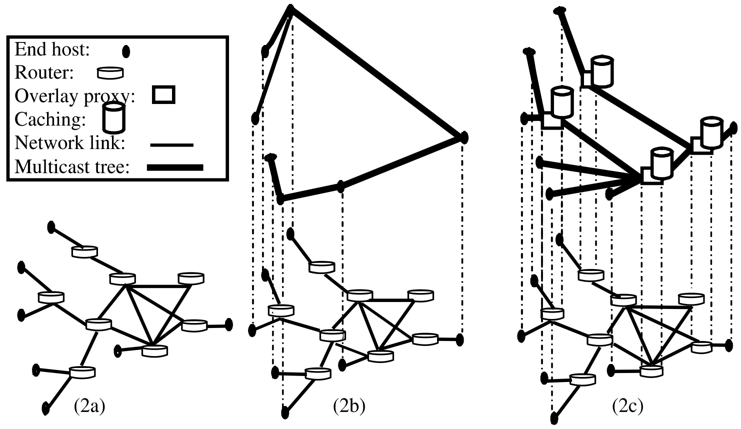

Figure 2: 2(a)IP multicast 2(b)Application Layer Multicast 2(c)Overlay Multicast

of senders to deliver content to any number of receivers.In past few year, it was most promising choice for content delivery over Internet, where a group address identifies a multicast group and any host can send messages to a group by simply sending to the group address. The router performs data delivery and message control. Unfortunately, native multicast has been proved extremely difficult to be efficient and scalable.Traditional multicast protocols such as DVMRP, PIM-DM and CBT, require the use of multicast-capable routers, making the entire implementation of multicast service difficult. By doing this, files are transmitted as one data stream over the backbone and only separated to the destinations by routers at the end of the path.

Since since the native IP multicast is still far from being widely deployed on the Internet due to many technical and marketing reasons, the focus has been transferred to application level multicast(ALM) which don't change network topology and can also be extended to a large scale group. Several researches have been studied at ALM protocols [40] [41]. Figure 2 shows the basic mechanisms for IP multicast, Application layer multicast and Overlay multicast. In this figure, we only show the dedicated source for one-to-many multicast scenario.

Today, more designs care about overlay multicast protocol because it has prominent potential [42] [43] [44]. First of all, it utilizes elaborately deployed intermediate proxies which construct "a backbone overlay" infrastructure. Under this system, multicast trees are established in terms of proxies and these proxies deliver the data to end host via unicast. As a result, overlay proxy placement, overlay link identification and bandwidth dimensioning are needed for overlay network establishment.In this way,multicast trees are used for data delivery and mesh is the key point of message control.

However, if we only use overlay multicast for video streaming, it would be insufficient, i.e. the overload of proxies, real-time playback requirements. It's better to set a cache with the proxy which used to store the prefix of video streams

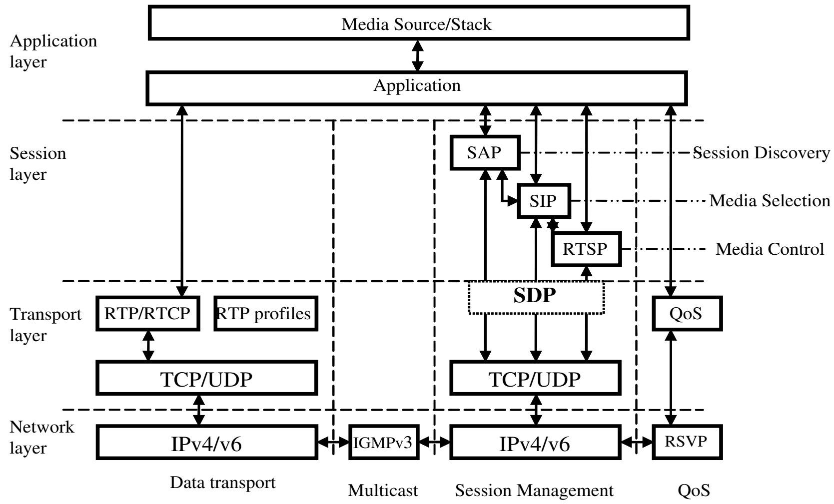

Figure 3: Protocol stacks for video streaming

or some special part of video streams.Therefore,network overload is alleviated and waiting time for clients is shortened. On the other hand, user-specified requirements e.g. many clients stop the video streams just after starting playback the beginning of the video, can be satisfied. If we deploy multirate video multicasts over networks, hierarchical requirements from users can be meet with, with respect to various capabilities of bandwidth. And we need protocols support for multicast including routing, resource reservation, reliability and congestion control.

3.7 Protocol stacks for video streaming

These protocols are designed for communication between clients and streaming servers. According to different functionality, we propose stacks in three layers: network layer, transport layer and session layer, shown in Figure 3. Network layer protocols mainly provide basic network service support. Transport layer protocols are the main communicator between endpoints. Session control protocols are defined to control the delivery of media streams within an established session.

3.7.1 Network Layer

In the network layer, IP is chosen as the main network layer protocol for video streaming. For multicast purpose, Internet Group Management Protocol(IGMP) is used between IP hosts and their inter-mediate multicast agents like routers to support the management of groups (creation of transient groups, periodic confirmation of group membership, addition or deletion of members of a group). Since traditional Internet lacks QoS support for multimedia transmission, a user can use Resource Reservation Protocol(RSVP) [45]to request a specific QoS from the network. RSVP carries the request through the network, visiting each node which the network uses to carry the stream. At each node, RSVP tries to make a resource reservation for the stream based on local admission control and policy control.

3.7.2 Transport layer

Considering the transport layer, Real Time Protocol (RTP) [46] is a transport mechanism for real-time data. It includes two main components: RTP and RTCP. RTCP mainly offers control-related function to media source such as congestion control,feeding back to application the quality of data distribution. Moreover, it needs to support media synchronization mechanism by providing *time-stamping, source identification, and participant identification and corresponding RTP timestamps* as well. To ensure playback successfully, RTP employs sequence numbering to place incoming packets in the correct order.

Furthermore, it is the responsibility of RTCP to provide QoS feedback to an application. Based on feedback, sender and receiver can adjust the transmission rate; determine the network status (i.e. local congestion); evaluate the network performance of media distribution.

3.7.3 Session control layer

Real-Time Streaming Protocol (RTSP) [47] provides control for media streaming. The main function is to support VCR-like operations, such as fast forward, rewind, play control. In the session layer, session Initiation Protocol (SIP) [52] is used to create, modify and terminate sessions. SIP mainly combines with Session Description Protocol (SDP) to transmit signaling messages. Session Announcement Protocol (SAP) [53] is created to assist multicast sessions, to communicate the relevant session setup information to prospective participants. It is always in conjunction with SIP and RTSP protocols. During multicast streaming of a video, SAP multicasts periodically packets containing a description of this session and remote participants in other session directories can use these description as a start tool to join the session. By using SAP, multicast services can disperse to broader area and attract more participants taking part in the media sessions.

3.8 Security issues

Several researches have been studied in video streaming, in the past few years; however, very little concern has been focused on security issues. For example, under overlay multicast video systems, little concern about security is taken seriously despite that it got many achievements in member management and data transmission. On the other hand, security and scalability are also requirements from current network applications as well.Totally, 3 phrases should be pointed out to reach the goal of transmitting a secured video stream over the Internet: authentication, authorization, video confidentiality.

3.8.1 Authentication

It is the fundamental request from services that the source identity, user identity, and the integrity of video streams during delivery can be confirmed. To exchange and validate the source and user identity, digital certificates or signatures are used. Under this condition, the server need to confirm the user is the exact valid receiver and the user need to ensure the server is the legitimate provider. It is also used to check the integrity of video streams. But current mechanism focus on one-way authentication using third party authority

3.8.2 Authorization

Although the users and server are mutually authenticated, there are a series of policies used to verify the exact services between the service provider and users who are allowed to access the video stream. The type of services, the duration time of services and the accounting type are basic content for authorization.

Cryptography is a usual way to provide for authorization mechanism, where the server and users exchange encrypted key under protected tunnel. In addition, some conditional access systems may employ watermarks embedded in a video stream to show the permission in the access models.

3.8.3 Video confidentiality

It is a necessary security measures for some applications because it is possible to replicate uncountable illegal video streams without this protection. And it is also a hard problem since many systems can only prevent "casual" copying via serial number or time stamps. For a media streaming applications, it involves the identifier of clients, the identifying protected video streams and defining some conditions for legal video copy. Some other protections focus on watermarks and confidentiality information embedded in the video stream headers. It is generally accepted to use cryptography, digital signatures, video watermarking and personal information binding for securing a digital video stream.

4 Some design and implementation considerations

4.1 Compressions proprietary

Despite the open standards of MPEG most people use one of the big three proprietary formats. These are *RealMedia, Quicktime*and *Windows Media*. All three have specific advantages which have allowed them to gain ground in the market - mainly because they are free, and support the Real Time Streaming Protocol (RTSP) [54].

4.1.1 RealMedia

This is a very popular player which is very widely distributed and available for all major OS platforms. RealPlayer is up to version 10 developed with Intel, coupled with their SureStream technology, will probably keep them in a dominant position [55]. RealSystem 10 supports over 40 media formats. Surestream is an automatic multi- bit rate technology that will adjust the streamed data rate to suit the client's connectivity. Also supported is Synchronized Multimedia Integration Language (SMIL) which allows mixed multimedia content to be delivered in a synchronized way.

4.1.2 QuickTime

Originally developed in 1991 version 4 now claims more than 100 million copies distributed world-wide. Quicktime's major advantages are its maturity and the large number of codec available for it [56]. It features an open plug-in feature to allow third party codec to be added. MPEG1 and MPEG4 codec are currently available.

The plug-in feature has allowed over 200 digital media formats to be supported by Quicktime 7 producing very impressive codec. As with RealPlayer, SMIL is available and now RTSP is also supported. Prior to version 4 only progressive streaming, not true real time streaming was available in Quicktime.

4.1.3 Windows Media Player

Windows Media Player (currently at version 10) is the posterity to the streaming world [57]. Because of that it can support fewer codec than previous two. Microsoft has put some work into their RTSP implementation and it is considered more efficient than others. SMIL is supported, but only at a basic level.

4.2 Streaming Servers

Real time streaming requires specific servers. RealNetworks, Microsoft and Apple all provide streaming servers [58]. These servers give you a greater level of control over your media delivery but can be more complicated to set up and administer than a standard HTTP server. Also, real time streaming uses special network protocols, such as RTSP or MMS (Microsoft Media Server).

Each of them provides tools for creating or converting content into a format that can be handled by their servers and epitomized for Internet Streaming. RealNetwork's RealProducer 10 will convert from a number of raw formats (AVI, MPEG-1, AU, AIFF etc) and is free for the basic version. Apple's Quicktime Player also provides content authoring and import/export facilities and Media On-Demand Producer is free from Microsoft.

The content creator can also create a Synchronized Multimedia Integration Language (SMIL) file to synchronize several clips within a presentation. A typical example of this is a lecture or presentation with associated slides where

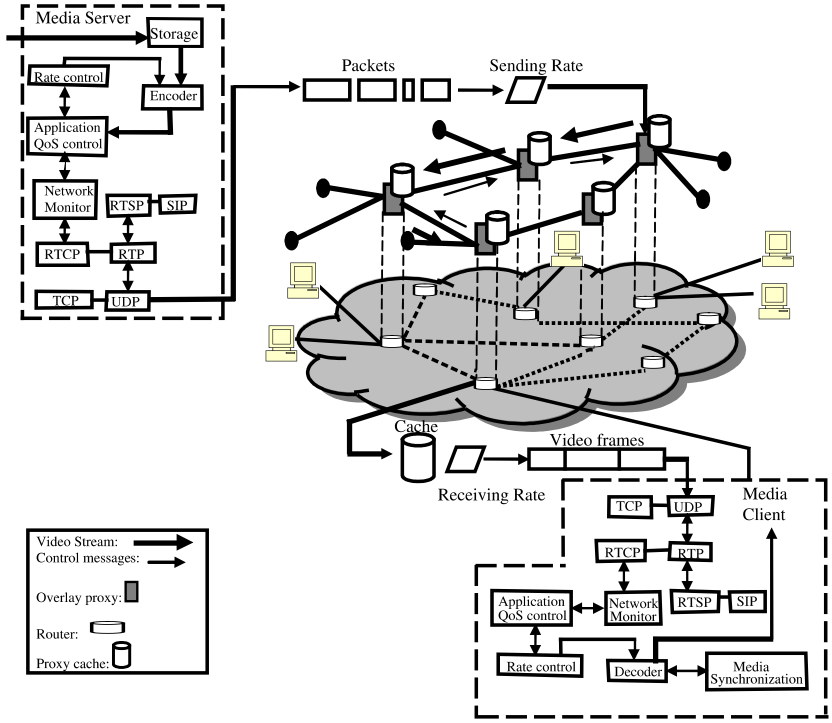

Figure 4: Efficient Overlay Multicast architecture with proxy caching support

the presentation of the slides can be synchronized with the audio content of the lecture. Creating content with SMIL [59] based on XML takes more time and effort but the results are worth it. RealNetworks can supply the Oratrix Development program GRiNS [60] for the creation of SMIL texts. There are many examples on the web showing how, for very little bandwidth, excellent media rich presentations can be compiled which are much more informative and interesting than the statically presented video.

4.3 Efficient Overlay multicast video architecture

Based on the aforementioned considerations on architecture, namely, media source processing, media server, media QoS control, media distribution services, some protocol stocks for video streaming and security issues, we suggest that the video streaming architecture could make use of overlay multicast with proxy caching support and application QoS control as well, see in Figure 4.

In this system, we combine caching with overlay proxies to store some video prefix or special part of video streams, by which clients can directly receive the required video without delay. Then noncached portions, if needed, are retrieved from the media server. For overlay multicast, we need to consider three phrases: construction of the overlay multicast tree, the maintenance of the overlay structure and management of endpoints. Meanwhile,four methods are provided for overlay multicast tree:1)tree-first construction approach [49]; 2)mesh-first construction approach [48]; 3)hierarchical construction [50]; 4)Mesh/tree hybrid construction [51]. To more efficient, we can deploy caches into the clients' side which will assist the proxy cache. Moreover, there are still some open issues such as scalability of overlay multicast, key management and access control of groups, QoS control-based overlay multicast issues.

4.4 Assessment of video quality

Streaming video quality is sometimes bad even when there are enough resources to handle the traffic load. The nature of the Internet and its use of IP mean that a video streaming is competing with other data transmissions, and in general there is no way of guaranteeing sufficient bandwidth to ensure an uninterrupted media streaming. So, we need a fast, accurate and comprehensive way to analyze the quality of video streams. In general, there are two ways to assess streaming video quality: objective analysis and subjective analysis. Objective assessment relies on some measurements such as packet loss, delay and throughput which are easy to collect and understand. However, the accuracy can not be assured and the exact type for measurement depends on system and certain streaming mechanisms. In contrast to objective analysis, subjective assessment provides the exact user thought of the stream. But only one outcome is not enough.

5 Conclusions and Outlook

In this document, we firstly identify some requirements for video streaming over Internet in the context of a basic architecture. Then we present the advanced architectural thoughts including 6 areas, based on recent researches. As a result, an efficient overlay multicast video streaming architecture is also proposed, incorporating overlay multicast with proxy caching mechanism. After that, we present several existing implementations and commercial supports for above mechanisms. If we use scalable video for adaptive application, different quality levels of video can be transmitted [61]. For instance, the higher leveled user can receive enhancement part of video with high Quality such as advanced text, information and playback with high real-time guarantee (1024*768). The coarsest resource supported users can only receive the basic video (640*480) and playback with poor guarantee.

It is of great importance to consider the loss pattern because the effect of the perceptual video quality with different loss patterns (i.e. the burst length of packet losses). When we look into the streaming over wireless network, it would be more complicated. The key issue in wireless environment is to detect correctly whether the network is in congestion or wireless loss. Accurately, we need to know the packet loss caused by wireless problem (multipath shading) or network congestion. Moreover, forward loss ratio need to be estimated including packet loss ratio and packet error ratio. Then, it is required to measure smoothed RTT for that one packet chosen as observation sample is unreliable.

Based on above mechanisms and approaches, we conclude this issue as follows.

- Media nature related requirements. For efficiency, high flexibility and low complexity, several video coding techniques are developed to control QoS fluctuations over networks. To achieve the synchronization of video and audio, adaptive and flexible synchronization is required, e.g. client-driven and network-driven cooperative method.

- Active research focus on how to build a scalable, efficient, cost-effective and incremental deployable infrastructure for media distribution. Therefore, it is required to provide reliable, efficient, scalable multimedia servers, i.e. hierarchical storage system with innovative operating system to support interactive functions.

- Internet can not provide sufficient solutions for all services, multicast is a desirable solution for multimedia streaming over the Internet. Furthermore, if we can solve scalability, efficiency and reliability problems, overlay multicast is the best choice for video streaming. Integrating MPLS backbone into multicast is a probable solution to support QoS since MPLS is an emerging technology that allows efficient transmission mechanism. However, path recovery, real-time constraints and security issues are still waiting to be solved in an appropriate way.

- Video streaming under wireless environment will be more complicated than wired situation. Under wireless networks, packet losses may be caused by network congestion and wireless networking. Furthermore, the security issues would be NP-hard problem.

Acknowledgment

This work is partly supported by T-Systems International GmbH under the EC Celtic Project VIDIOS - VIdeo DIstribution Over MPLS networks supporting heterogeneous format environments.

References

[1] Y. Cui, B.C. Li, K. Nahrstedt, "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks", *IEEE Journal on Selected Areas in Communications*, Vol.22, No.1, Jan. 2004.

[2] R. Steinmetz, "Synchronization Properties in Multimedia Systems", *IEEE Journal on Selected Areas in Communication*,Vol.8, No.3, pp. 401, Apr. 1990.

[3] T.D.C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects", *IEEE Journal on Selected Areas in Communication*,Vol.8, No.3, pp. 413-427, Apr. 1990.

[4] T. Sikora, "MPEG Digital Video Coding Standards", *Digital Electronics Consumer Handbook. McGraw Hill*, 1997.

[5] M. Ghanbari, "Video Coding-an introduction to standard codecs", *The Institution of Electronical Engineers*, 1999.

[6] MPEG-1, "Generic coding of moving pictures and associated audio information", *draft International Standard*,1994.

[7] Video Home System, *http://www.mediabit.de/lexikon/vhs.html*

[8] MPEG-2, "Coding of moving pictures and associated audio for digital storage media at up to 1.5 mbps", 1993.

[9] F. Pereiera and E. Touradj, "The MPEG-4 Book", *Prentice Hall*, 2002

[10] A.N. Netravali and B.G. Haskell, "Digital Pictures: Representation, Compression, and Standards", *Plenum Press*,Chapter 3.1995.

[11] D.P. Wu, Y.W.T. Hou, W.W. Zhu et al., "Streaming Video over the Internet: Approaches and Directions", *IEEE Trans. on Circus and Systems for Video Technology*, Vol.11, No.2, Mar. 2001.

[12] J.G. Apostolopoulos, T. Wong, W. Tan et al., "On multiple description streaming with content delivery networks", *in Proc. of IEEE INFOCOM*,June 2002.

[13] R. Steinmetz and K. Nahrstedt, "Multimedia: Computing, Communications and Applications", *Upper Saddle River, NJ: Prentice Hall*, 1995.

[14] ATT Labs research, http://public.research.att.com/viewPatent.cfm?Number=6177928

[15] J. Gemmell, H. M. Vin, D.D. Kandlur, et. al., "Multimedia storage services: a tutorial", *IEEE Computer Magzine*, vol.28, No.5, pp.40-49, May, 1995.

[16] A. Guha, "The evolution to network storage architectures for multimedia applications", *in Proc. IEEE international Conf. on Multimedia Computing and Systems*, pp.68-73, June 1999.

[17] D.H.C. Du and Y.-J. Lee, "Scalable server and storage architectures for video streaming" *in Proc. IEEE International Conf. on Multimedia Computing and Systems*,pp.62-67, June 1999.

[18] G.A. Gibson and R.V. Meter, "Network attached storage architecture", *COMMUNICATIONS OF THE ACM*, Vol.43, No.11, November 2000.

[19] T. Montgomery, "A Loss Tolerant Rate Controller for Reliable Multicast", *Technical Report, NASA-IVV-97-011*, 1997.

[20] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer", *in IEEE INFOCOM*, 1998.

[21] Q. Guo, Q. Zhang, W. Zhu et., "Send-adaptive and receiver-driven video multicasting", *IEEE International Symposium on Circuits and Systems*, Sydney, Australia, May 2001.

[22] L.S. Lam, J.Y.B. Lee, S.C. Liew and W. Wang, "A Transparent Rate Adaption Algorithm for Streaming Video over the Internet", *AINA¡¯04*, 2004.

[23] S.Deering, "Multicast routing in internetworks and extended LANs", *in Proc. ACM Sigmetrics*, June 1988.

[24] P.de Cuetos and K.W. Ross, "Adaptive Rate Control for Streaming Stored Fine-Grained Scalable Video", *Proc. NOSSDAV*, pp.3-12, May 2002.

[25] S. Varadarajan, H.Q. Ngo, and J. Srivastava, "An adaptive, perception-driven error spreading scheme in continuous media streaming", *Distributed Computing Systems for Video Technology*, pp 475-483, Sept. 2000.

[26] D. Wu, Y.T. Hou, W. Zhu et, al., "On end-to-end architecture for transporting MPEG-4 video over the Internet", *IEEE Trans. on Circus and Systems for Video Technology*, Vol.10, pp.923-941, Sept. 2000.

[27] M. Luby, L. Vicisano, J. Gemmell, et al., "Forward Error Correction (FEC) Building Block", *RFC 3452*, Dec. 2002.

[28] S. Lin, D.J.C. Jr., and M.J. Miller, "Automatic-Repeat reQuest error-control schemes", *IEEE Communications Magazine*, 22(12):5-17, Dec. 1984.

[29] Q. Zhang and S.A. Kassam, "Hybrid ARQ with selective combining for fading channels", *IEEE J. Select. Areas Commun.*, Vol.17, pp.867-880, May 1999.

[30] P. Bucciol, E. Masala and J.C.D. Martin. "Perceptual ARQ for H.264 video streaming over 3G wireless networks", *IEEE Communications Society*. 2004.

[31] C.H. Wang, R.I. Chang, J.M. Ho, et al. "Rate-sensitive ARQ for real-time video streaming",2001.

[32] A. Narula and J.S. Lim, "Error concealment techniques for an all-digital high-definition television system", *Proc. SPIE Conf. Visual Commu. And Image Proc.*, pp. 304-314. 1993.

[33] X.W. Ding, K. Roy, "A novel bitstream level joint channel error concealment scheme for realtime video over wireless networks", *IEEE INFOCOM*, 2004.

[34] Y. Wang and Q.F. Zhu, "Error control and Concealment for video communications: A Review", *Proc. of the IEEE*, Vol.86, No.5, May 1998.

[35] P. Ge, P.K. Mckinley, "Comparisons of Error Control Techniques for Wireless Video Multicasting", *IEEE 2002* ,2002.

[36] C. Diot, B. Levine, J. Lyles, et al., "Deployment issues for the IP multicast service and architecture", *IEEE Network*, Jan.2000.

[37] L.X. Gao, Z.L. Zhang, and D. Towsley, "Proxy-assisted Techniques for Delivering Continuous Multimedia Streams", *IEEE/ACM Trans. on Networking*, Vol.11,No.6, Dec. 2003.

[38] J.Liu, X.W Chu, and J.J. Xu. "Proxy Cache Management for Fine-Grained Scalable Video Streaming", *Proc. IEEE INFOCOM'04*, Hong Kong, March 2004.

[39] A.T.S. Ip, J.C. Liu, J.C.S. Lui, "COPACC:A Cooperative Proxy-Client Caching System for On-Demand Media Streaming", *In Proceedings 4th International IFIP-TC6 Networking Conference*, Waterloo, Canada, May, 2005.

[40] Y. Cui, B. C. Li, K. Nahrstedt. "oStream: Asynchronous Streaming Multicast in Application-Layer Overlay Networks", *IEEE Journal on Selected Areas in Communications, Special Issue on Recent Advances in Service Overlay Networks*, Vol. 22, No. 1, pp.91-106, January 2004.

[41] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", *in Proc. ACM SIGCOMM*, Aug.2002.

[42] Z. Li, P. Mohapatra. "HostCast: A New Overlay Multicast Protocol", *Proc. of IEEE ICC 2003*,2003.

[43] Y. Zhu, B.C. Li. "Multicast with Network Coding in Application-Layer Overlay Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 22, No.1, Jan. 2004.

[44] S. Banerjee, C. Kommareddy, K. Kar, et al. "OMNI: An Efficient Overlay Multicast Infrastructure for Real-time Applications", *Special Issue of Computer Networks on Overlay Distribution Structures and their Application*, 2005.

[45] R. Braden, L. Zhang, S. Berson, et al., "Resource Reservation Protocol (RSVP)Version 1 Functional Specification",Sep. 1997.

[46] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC 1889: RTP: A Transport Protocol for Real-time Applications", January 1996.

[47] H. Schulzrinne, A. Rao, R. Lanphier, "RFC 2326: Real Time Streaming Protocol (RTSP)",April 1998.

[48] Y.H. Chu, S.G. Rao and H. Zhang, "A case for End System Multicast", *SIGMETRICS*, 2000.

[49] P. Francis, "Extending the Internet Multicast Architecture", http://www.icir.org/yoid/docs/, Apr. 2000.

[50] S.Banerjee, B. Bhattacharjee, C. Kommareddy, et al., "Scalable Application Layer Multicast", *SIGCOMM*,2002.

[51] W.J. Wang, D. Helder, S. Jamin et al., "Overlay Optimizations for End-host Multicast", *NGC02,ACM*, pp.154-161,2002.

[52] J. Rosenherg, H. Schulzrinne, G. Camarillo, and et. al., "RFC 3261: SIP: Session Initiation Protocol",June 2002.

[53] M. Handley, C. Perkins, E. Whelan, "RFC 2974: Session Announcement Protocol",October 2000.

[54] MMUSIC WG, H. Schulzrinne, A. Rao, R. Lanphier, et al., "Real Time Streaming Protocol (RTSP)",draft-ieft-mmusic-rfc2326bis-06.txt, Feb. 2004.

[55] Realplayer, http://www.real.com/

[56] QuickTime, http://www.apple.com/quicktime/

[57] Windows Media Player, http://www.microsoft.com/windows/windowsmedia/default.aspx

[58] An Introduction to Streaming Video, http://www.cultivate-int.org/issue4/video/

[59] S. Bugaj, D. Bulterman, B. Butterfield, et al. "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification",June 1998.

[60] Oratrix development, http://www.oratrix.com/

[61] X.X. Lu, S. Tao, M.E. Zarki, et al., "Quality-based Adaptive Video Over the Internet", *Communication Networks and Distributed Systems Modeling and Simulation*, 2003.