SETIT 2005
3rd International Conference: **S**ciences of **E**lectronic,
Technologies of **I**nformation and **T**elecommunications
March 27-31, 2005 – TUNISIA

Tutorial on Multicast Video Streaming Techniques

Hatem BETTAHAR

Heudiasyc, Université de Technologie de Compiègne, France

Hatem.bettahar@hds.utc.fr

Abstract: This tutorial gives a survey of actual used techniques for multicast streaming video. We first dscuss important issues in video streaming namely video compression techniques and video compression standards. We then present the challenges in multicast video streaming and we give a detailed description of recent proposals for multicast video streaming. We classify these solutions into four approaches depending on the techniques used to adapt the reception video quality and on the source/receiver/network role .

Key words: QoS adaptation, multicast, multimedia applications, video streaming

1. Introduction

Multicast communication is an efficient solution for group applications in the Internet. Multicast conserves the network bandwidth by constructing a spanning tree between sources and receivers in the session. A single copy of the date is send to all the receivers through the multicast tree. Most multiparty applications over the internet are multimedia oriented. Applications such as videoconferencing, distant learning or network games use video, audio and data traffic. This traffic is bandwidth consumer. In addition, the current best effort service and the uncontrolled dynamic state in the internet require multimedia adaptation techniques to overcome these problems. For these reasons, combining multicast and video rate adaptation technique seams to be a suitable solution for video streaming over the internet. However, a set of open issues, such as receiver heterogeneity and feedback implosion, should be treated in order to have an efficient and viable solution for multicast video streaming.

This tutorial is organized as follows: in section 2 we give an overview of video streaming, and we stress the requirements for an efficient and streaming architecture over IP networks. In the section 3 we present some important issues in video compression. This is essential, because compression techniques are the base of many multicast streaming proposals. In section 4 we study the challenges in multicast video streaming: rate control, playout buffer and error control. Finally in section 5 we give a detailed description of recent proposals for multicast video streaming and we classify these solutions into four approaches depending on the techniques used to adapt the reception video quality and on the role of the source, the receivers and the network.

2. Video streaming

2.1. Video streaming architecture

A typical multicast streaming scheme is represented in Figure 1. A multicast source implements a streaming server which is responsible for retrieving, sending and adapting the video stream. Depending on the application, the video may be encoded on-line for a real-time transmission or pre-encoded and stored for an on demand transmission. Applications such as video conferencing, live broadcast or interactive group games require real-time encoding. However, applications such as video on-demand require pre-encoded video. Whence the multicast session is initialized, the streaming server retrieves the compressed video and begins the streaming with the adequate bit-stream rate.

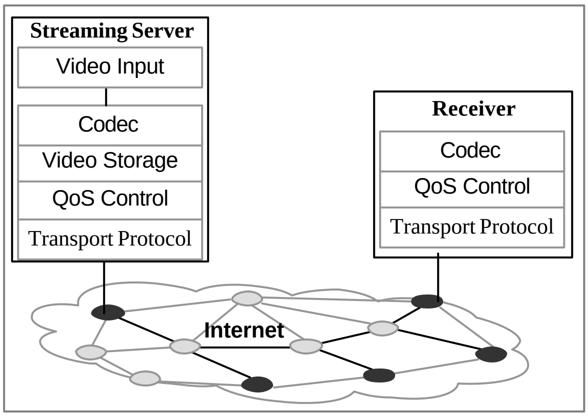

Figure 1. *Video Streaming Architecture*

2.2. Streaming protocols and standards

In this section we give a brief description of the network protocols for video streaming over the Internet. Figure 2 resumes the set of protocols described below.

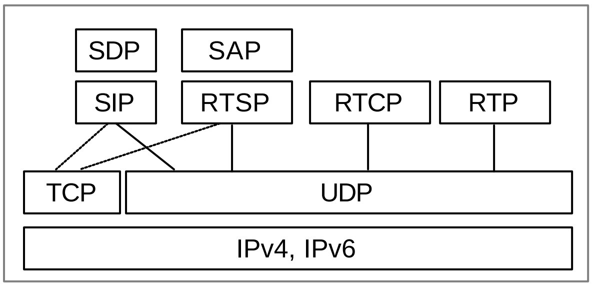

Figure 2. *Streaming Protocols Stack*

Real-Time transmission control: RTP/RTCP

The Real-time Transport Protocol (RTP) and Real-time Control Protocol (RTCP) (Schulzrinne &al. 1996) are IETF protocols designed to *support streaming media*. RTP is designed for data transfer and RTCP for control messages. Note that these protocols do *not* enable real-time services, only the underlying network can do this, however they provide functionalities that support real-time services. RTP does not guarantee QoS or reliable delivery, but provides support for applications with time constraints by providing a standardized framework for common functionalities such as time stamps, sequence numbering, and payload specification. RTP enables detection of lost packets. RTCP provides feedback on quality of data delivery in terms of number of lost packets, inter-arrival jitter, delay, etc. RTCP specifies periodic feedback packets, where the feedback uses no more than 5 % of the total session bandwidth and where there is at least one feedback message every 5 seconds. The sender can use the feedback to adjust its operation, e.g. adapt its bit rate. The conventional approach for media streaming is to use RTP/UDP for the media data and RTCP/TCP or RTCP/UDP for the control. Often, RTCP is supplemented by another feedback mechanism that is explicitly designed to provide the desired feedback information for the specific media streaming application. Other useful functionalities facilitated by RTCP include inter-stream synchronization and round-trip time measurement.

Session control: SIP/RTSP

The Session Initiation Protocol (SIP) and the Real-Time Streaming Protocol (RTSP) are used to control media streaming sessions. SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions such as Internet telephony calls (VOIP). SIP can also invite participants to already existing sessions, such as multicast conferences. RTSP establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It acts as a "network remote control" for multimedia servers. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. RTSP is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms bases upon RTP

Session description: SDP

The Session Description Protocol (SDP) describes multimedia sessions, for example whether it is video or audio, the specific codec, bit rate, duration, etc, for the purpose of session announcement, session invitation and other forms of multimedia session monitoring. Session directories assist the advertisement of conference sessions and communicate the relevant conference setup information to prospective participants. SDP is designed to convey such information to recipients. SDP is purely a format for session description - it does not incorporate a transport protocol, and is intended to use different transport protocols as appropriate including the Session Announcement Protocol (SAP).

Session Announcement: SAP

Session Announcement Protocol (SAP) is an announcement protocol that is used to assist the advertisement of multicast multimedia conferences and other multicast sessions, and to communicate the relevant session setup information to prospective participants. A SAP announcer periodically multicasts an announcement packet to a well-known multicast address and port. A SAP listener learns of the multicast scopes it is within (for example, using the Multicast-Scope Zone Announcement Protocol) and listens on the well known SAP address and port for those scopes. In this manner, it will eventually learn of all the sessions being announced, allowing those sessions to be joined.

3. Video compression techniques

In this section we describe briefly important issues in video compression. This is useful to understand some multicast streaming solutions described in the fifth section.

3.1. Video Compression overview

Video compression is the process in which the amount of data used to represent video is reduced to meet a bit rate requirement for a given application. This is achieved by exploiting temporal redundancy between consecutive frames in a video sequence as well as spatial redundancy within a single frame. In addition only video features that are perceptually important for human are coded. In current video compression standards, such as MPEG and H261 families, temporal redundancy is exploited by applying Motion Compensation techniques and the special redundancy is exploited by applying the DCT transormation (Discrete Cosine Transform). Using DCT, picture frames are first separated into square blocks of size 8x8. Each block is DCT transformed, resulting in another 8x8 block in frequency domain, whose coefficients are then quantized and coded. Most of the quantized high-frequency DCT coefficients do not need to be sent because they have nearly zero values, resulting in high compression. The remaining DCT coefficients are quantized and coded by exploiting their statistical redundancy, e.g., using variable length codes (like Huffman codes). This increases the average compression by assigning smaller code words to frequently encountered events. Applying inverse DCT on the quantized DCT coefficients recovers an approximate version of the original block. The 8x8 reconstructed blocks are pieced together to form the coded image. For motion compensation, the encoder typically estimates the motion (displacement of moving objects from frame to frame) assuming that a 16x16 block of pixels (macroblock) has been translated by a motion vector, which is found using block matching. The motion vector is then encoded and sent to the decoder. The difference between the predicted and actual frame is then compressed like a still-image as described above using DCT coding. Using this motion compensation technique, frames are separated in three types: intra-coded frames, or *I-frames*, where the frames are coded independently of all other frames, predictively coded frames, or *P-frames*, where the frame is coded based on a previously coded frame, and bi-directionally predicted frames, or *B-frames*, where the frames are coded using both previous and future coded frames. The selection of prediction dependencies between frames can have a significant effect on video streaming performance, e.g. in terms of compression efficiency and error resilience.

3.2. Video Compression Standards

We present in this paragraph the two principal standards currently used for video compression in streaming applications: the H.261/H.263 family (ITU 93,97), standardized by the International Telecommunications Union-Telecommunications (ITU-T) committee for compressing video streams for video conferencing applications and the MPEG family standardized by the International Organization for Standardization (ISO).

H.261/H.263 family

The first video compression standard to gain widespread acceptance was the ITU H.261 [11], which was designed for videoconferencing over the integrated services digital network (ISDN). H.261 was adopted as a standard in 1990. It was designed to operate at p = 1,2, ..., 30 multiples of the baseline ISDN data rate, or p x 64 kb/s. In 1993, the ITU-T initiated a standardization effort with the primary goal of videotelephony over the public switched telephone network (PSTN), where the total available data rate is only about 33.6 kb/s. The video compression portion of the standard is H.263 and its first phase was adopted in 1996 [12]. An enhanced H.263, H.263 Version 2 (V2), was finalized in 1997, and a completely new algorithm, originally referred to as H.26L, is currently being finalized as H.264/AVC.

MPEG family

The Moving Pictures Expert Group (MPEG) was established by the ISO in 1988 to develop a standard for compressing moving pictures (video) and associated audio on digital storage media (CD-ROM) (ISO/IEC 1993, 1996, 1999). The resulting standard, commonly known as MPEG-1, was finalized in 1991 and achieves approximately VHS quality video and audio at about 1.5 Mb/s [13]. A second phase of their work, commonly known as MPEG-2, was an extension of MPEG-1 developed for application toward digital television and for higher bit rates [14]. Currently, the video portion of digital television (DTV) and high definition television (HDTV) standards for large portions of North America, Europe, and Asia is based on MPEG-2. A third phase of work, known as MPEG-4, was designed to provide improved compression efficiency and error resilience features, as well as increased functionalities including object-based processing, integration of both natural and synthetic (computer generated) content and content-based interactivity.

Table 1 presents an overview of different versions within these two standard families.

Standards	Applications	Bit rate
H.261	Video teleconferencing over ISDN	px64kbs
MPEG-1	Video on digital storage media (CD-ROM)	1.5 Mbs
MPEG-2	Digital TV	2-20 Mbs
H.263	Video telephony over PSTN	>33.6 kbs
MPEG-4	Mulitmedida over internet, object based coding	variable
H.264/MPEG-4	Improved video compression	10's-100's kbs

Table 1: *overview of video compression standards*

4. Challenges in Multicast video streaming

In this section we begin by presenting basic video streaming challenges and then we discuss challenges introduced by the use of multicast communication in video streaming.

4.1. Challenges related video streaming

The best effort service offered by the Internet leads to an uncontrolled quality of transmission which affects the stream reception quality for receivers. Three basic problems are to be treated to keep an acceptable QoS during video streaming: Bandwidth management, packet loss and delay jitter.

Bandwidth management

Regarding the important number of flows' natures transmitted through the internet, the bandwidth management is still one of the critical problems to be solved. Actually, it's hard to have an instantaneous estimation of the available bandwidth within a path in order to adapt the stream bit rate. It is also hard to achieve an acceptable fairness in bandwidth allocation between concurrent flows with different natures and requirements (video UDP streams and TCP flows). There are many techniques to adapt the stream bit rate to the available bandwidth; we can use multi-coded stream, transcoding or layered compression.

Multi-coded stream: One way for a source to adapt its stream to the available bandwidth is to preview a number of pre-encoded streams for the original video. These streams should be coded at a few well chosen rates targeted for common network access speeds (for example 128 Kbps for an ISDN access, 1.5 Mbps for an ADSL access, and 10-Mbps for an Ethernet access). Here it is the receivers' role to choose the adequate stream depending on its own access speed to the network.

Transcoding: To adapt the bit rate we can just recompress the stream to the desired bit rate. The problem with recompression is that decoding and re-encoding leads generally to a lower quality than if the video was coded directly from the original source to the same bit rate. In addition recompression requires an extensive computation and may affect sensitive application. The solution is to use transcoding to recompress the stream. With transcoding techniques (Wee & al.1999) we selectively re-use compression decisions already made in the compressed media to reduce computation (work on the compressed data directly without decoding). This is possible because most codecs use similar compression techniques.

Layered compression: A more elegant way to adapt the bit rate is to code the stream using a layered compression technique. It consists on mapping different video frames to a set of layers during the compression process; in general a basic layer and a set of enhancement layers. The way frames are mapped into layers depends on the used scalability parameter: temporal, spatial or quality scalability. For example, using the temporal scalability (used in H.263 and MPEG) layers are mapped to different frame types, for instance I and P frames compose the basic layer, and B frames compose the enhancement layer. In order to adapt the reception rate, the receiver should cumulate a set of enhancement layers in addition to the base layer.

Packet loss

Even if video streaming applications are less sensible to packet loss and transmission errors than other reliable applications, losses and errors can have a very destructive effect on the reconstructed video quality. To avoid such problem video streaming system are designed with error control techniques. These techniques can be classified into four classes: forward error correction (FEC), retransmissions, error concealment, and error-resilient video coding. The main idea of FEC codes, such as Tornado codes, is to take K packets and generate N packets: as long as any K of N packet received, the original video can be reconstructed. One disadvantage of this technique is the additional bandwidth overhead. In addition we need to adapt redundancy ratio N/K to the loss rate which is difficult to estimate. Error control techniques based on retransmission need a feedback mechanism to inform the source about data to retransmit. Retransmissions may be harmful for multimedia delay constrained applications. Most video streaming application, avoid error and loss degradation by just using error concealing combined with error resilient. Error concealing (Wang & al. 1998) consists on using the spatial and/or temporal correlation to estimate the lost information (by spatial interpolation, temporal extrapolation or motion-compensated interpolation). Error resilient coding avoids error propagation which may cause more and more damage. This may be done by isolating the error by placing resynchronization marker within fixed interval in the stream (used in MPEG-4) or at the beginning of each picture (used in MPEG-1/2 and in H261/3).

Delay jitter

Delay jitter measures the end-to-end delay variation caused by the unstable network state. Delay jitter can cause damage (jerks) during video decoding at the receiver side because video frames need to be received at constant rate. This problem is solved by including a playout buffer at the receiver side. This buffer will be used to attenuate the jitter and to give more flexible way to apply error recovery and error resilience techniques on the video stream.

4.2. Challenges related to multicast sessions

In addition to the basic challenges in video streaming listed above, using multicast adds a number of problems to be solved. First of all, there is the problem of receivers' *heterogeneity*. In a multicast session different receivers may have different network access speed and so different QoS and bandwidth constraints. Also we can have different classes of receivers regarding the chosen charging level. This heterogeneity complicates the video adaptation technique to be used. This problem is detailed in the following section. In addition source rate adaptation techniques are based on feedback information sent by receivers, in the case of a multicast session this will lead to a *feedback implosion* problem. A third problem related to the use of multicast in video streaming is that error control techniques based on retransmission are hardly applicable in a multicast session.

5. Multicast video Streaming

We give, in this section, a detailed description of recent proposals for multicast video streaming. We classify these proposals depending on the used rate adaptation technique as well as on the source/receiver/network roles. We will focus essentially on layered adaptation technique which we believe to be the most promoting approach for the future. Readers can find exhaustive surveys about multicast video streaming (Matrawy & al. 2003) (Li & al. 2003) (Liu & al. 2003). Table 2 gives a summary of these approaches.

5.1. Single stream with source rate adaptation

With this approach the multicast source uses only one copy for the video stream and adjusts the bit-rate based on the receivers/network feedback (Cheung & al. 1996) (Li & al. 1999). This feedback can be based on RTP/RTCP quality reports. In order to adjust the stream rate the source can use transcoding. This simple approach has many drawbacks. First, it needs a scalable feedback mechanism (Bolot 1994). Second, receivers' feedback should be synchronized, which is very hard to achieve. Finally, the adjusted rate should satisfy the lowest receiver resource, which penalizes all other receivers.

5.2. Multiple streams with receiver switching

In order to avoid feedback problems, in this approach the source uses a set of pre-encoded streams of the original video (Cheung 1996). These streams are coded at a few well chosen rates targeted for common network access speeds (for example 128 Kbps for an ISDN access, 1.5 Mbps for an ADSL access, and 10-Mbps for an Ethernet access). The source sends each stream to a separate group address. The rate adaptation role is relegated to each receiver. After estimating the reception quality (loss rate), a receiver can adapt the bit-rate by switching between streams and joining/leaving the corresponding multicast group. To improve the efficiency of this approach, it can be combined with the first one. Actually, the source can use an intra-stream protocol to adapt the bit-rate for each stream using a less frequently feedback reports. Note that this is completely different from feedback reports in the first approach since no synchronization is needed between receivers. The source has to get, from time to time, some feedback to adapt the rate of each stream. Note that this technique is implemented in many commercial solutions such as SureStream from Real.

5.3. Layered Multicast Streaming

Layered video coding was proposed as an attractive and efficient solution for the problem of video adaptation in heterogeneous multicast sessions. In layered approaches, the raw video is encoded into a set of cumulative (or non-cumulative) layers (Amir & al.1996) (Vickers &al. 2000) (Bajaj & al. 1998). A basic layer contains the essential video information with a basic low quality and a set of enhancement layers are used to improve the quality of the received video. In the case of MPEG-4 (Pereira & al. 2002) coding, layers can be obtained by applying Temporal scaling, Spacial scaling of Fine Granularity scaling (FGS). The set of layers are sent over separate multicast groups. A receiver joins, first, the basic layer to obtain the basic video quality and then adapts the rate depending on its capabilities by joining/leaving enhancement layers. Depending on the set of joined layers, heterogeneous receivers will obtain different video quality. In the cumulative layered approach layers should be joined in a cumulative manner in the order of their relevance. However in non-cumulative approaches, layers are not ordered and receivers can join any subset of layers. Layered coding is also used in router (or agent) assisted techniques where rate adaptation is achieved by on-tree routers. In this approach, the quality adaptation is delegated to intermediate routers which accept or drop layers by filtering the video stream according to the congestion state of its interfaces. In the following subparagraphs we present a brief description of the state-of-art of layered multicast proposals.

Cumulative layered multicast protocols

The first adaptive layered multicast protocol was the Receiver-driven Layered Multicast (RLM) proposed by McCanne, Jacobson and Vetterli (McCanne & al. 1996). Using RLM, the sender streams each video layer on a separate multicast group. In the receiver side, rate adaptation is performed by periodic *Join-experiments* to higher layers. The receiver decides to keep or not the joined layered based on the packet loss experienced during this attempt. In order to coordinate join-experiments of receivers sharing the same tree branches, RLM uses a *Shared learning* mechanism where each join-experiment triggered by a receiver is inferred by other receivers. RLM has a number of drawbacks and cannot be applied as it is in real multicast streaming sessions. In fact, the way RLM manages join-experiments leads to a high convergence time and reduces the scalability of the protocol (Legout & al. 2000). In addition RLM lacks inter-session fairness and it is not TCP-friendly (Gopalakrishnan & al. 1999) (Sarkar & al. 2000).

To overcome the drawbacks of RLM, authors in (Vicisano & al. 1998) proposed the Receiver driven Layered Congestion control protocol (RLC). RLC solves the coordination problem between receivers by the active role of the source. The source periodically, and for a short time, increases the sending rate on a layer. If no packet loss is experienced during this period, the receiver may decide to join a higher layer. This way no coordination is needed among receivers and the convergence and scalability are significantly improved compared to RLM. In addition RLC ensures an acceptable fairness with TCP flows by using adequate join timers and a careful choice of the rate for each layer in a way that mimics the behavior of the TCP congestion control.

Non-Cumulative layered multicast protocols

Authors in (Byers & al. 2001) propose a non-cumulative approach based on their fast encoding technique called the *digital fountain* (Byers & al. 1998). This encoding scheme enables the source to send the video stream using independent fine grained layers. In the receiver side an RLM-like adaptation scheme may be used. The non-cumulative nature of layers makes this approach more robust to packet loss and enables a better adaptation flexibility.

Router/agent assisted layered multicast protocols

In this approach, layers adaptation is delegated to on-tree routers or to agents placed on the network. The Layered Video Multicasting with Retransmission protocol (LVMR) (Li & al. 1997) is an example of agent assisted protocols. LVMR uses an RLM-like adaptation scheme; however the rate adaptation control is delegated to a hierarchy of agents placed in the network. These agents help receivers to decide to join or leave layers. The main advantage of LVMR is that it reduces the volume of control information compared to RLM and it introduces the concept of recovery using retransmission from designated local receivers to reduce the recovery time. However LVMR introduces a considerable overhead and rises the problem of agents deployment and placement.

The Receiver-Selectable Loss Proprieties protocol (RSLP) (Gopalakrishnan & al. 2000) is an example of router assisted protocol. RSLP implements a RLM-like adaptation scheme combined with a two levels priority dropping mechanism at routers. Priorities, assigned to layers by the receivers, indicate to routers which layers to drop first in case of congestion. As routers should keep track of priorities assigned to layers, this may create scalability problems. In addition this protocol needs an efficient signalling technique between receivers and routers.

The Network-supported Layered Multicast protocol (NLM) (Nakauchi & al. 2001) is a fully router-assisted protocol. All of the congestion control and layers adaptation are delegated to intermediate routers. Each on-tree router filters, stops, or adds layers according to the congestion state of its interfaces. Even though NLM may give good fairness and stability performances, it needs a considerable complexity and processing overhead at the router level and hence it is not suitable for an efficient deployment on the internet.

In (Zhang & al. 2002), authors proposed a Router Assisted layered Multicast protocol (RALM). With this protocol, each on-tree router adapts the stream rate by a series of suspend/retry operations. If congestion is detected on one interface, a router will temporally suspend the less important layer based on priorities set by the sender. As soon as the congestion is away, the router will try to re-activate the suspended layer. If a layer is suspended on all the interfaces, the router leaves definitively the associate group and waits for an explicit join triggered by one of the receivers to re-join this layer. Unlike NLM, RALM is not a fully router-assisted protocol since the sender and the receivers play an active role in the adaptation process. However RALM still suffer from an important processing and state maintenance overhead at routers. One advantage of RALM is that it can be implemented incrementally since not all the routers need to implement RALM within a session.

5.4. Single stream with Active network adaptation

In contrast with other approach, in this approach the adaptation process is relegated to the intermediate routers. Some active routers may implement agents able to manipulate the video stream and to adapt its bit-rate or to decompose the original stream into multiple streams with different rates or formats using transcoding techniques (Amir & al.1998). This is useful for multicast sessions where agents can be placed in well chosen nodes on the multicast tree so that we can have more accurate QoS measurement and the adaptation will affect only the receivers within a given region. It is also useful to overpass non-multicast links in the network; the agent can convert the single multicast stream into a number of unicast streams. However this solution suffers from agent placement problem: should we use dynamic active agents or should we use a static agent placement. Note that this technique is used in Content Delivery Network such as Cisco Enterprise CDN.

Approaches	Source role	Receiver role	challenges
Single Stream	Rate adaptation	Reception quality feedback	Scalable feedback
Multiple streams	Multi-stream transcoding	Streams switching	Stable receivers switching
layered stream	Cumulative or non-Cumulative Layered coding	Layers Join/Leave	Inter-layer synchronization Inter-receiver synchronisation
Active network	Transcoding by agents within Active Routers		Active service nodes

Table 2: *Multicast video streaming approaches*

6. Conclusion

In this tutorial we have discussed first important issues in video streaming namely video compression techniques and standards. Then we presented the challenges in multicast video streaming. We then classified the different approaches proposed to overcome these challenges. Note that in all the approaches presented in this tutorial, there is a number of open issues that need to be treated. For instance, feedback implosion and receivers' synchronization in multicast sessions are still up-to-date challenges. In addition any proposed solution for multicast video streaming should take into consideration the problem of intersession fairness and should be TCP friendly.

We believe that using multicast communication in video streaming is a promising solution for the future especially for wide content distribution applications. But we also believe that a lot of effort should be done in this domain to propose efficient and viable solutions.

References

Amir A., S. McCanne, and R. Katz, "An Active Service Framework and Its Application to Real-Time Multimedia Transcoding," *Proc. ACM Sigcomm Conf.*, ACM Press, 1998, pp. 178-179.

Amir E., S. McCanne, M. Vetterli, "A Layered DCT coder for Internet video", *IEEE international conference on image processing*, Lausanne, September 1996

Bajaj S., L. Breslau, and S. Shenker, "Uniform Versus Priority Dropping for Layered Video," *Proc. ACM Sigcomm Conf.*, ACM Press, Sept. 1998, pp. 131-143.

Bolot J., T. Turletti, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," *Computer Comm. Review*, vol. 24, no. 4, Oct. 1994, pp. 58-67.

Byers J. W., Luby M., Mitzenmacher M., and Rege A., "A digital fountain approach to reliable distribution of bulk data," *Proc. of SIGCOMM*, 1998.

Byers J. W., Luby M., and Mitzenmacher M., "Fine-grained layered multicast," *Proc. of INFOCOM*, 2001.

Cheung S., M. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution," *Proc. Ann. Joint Conf IEEE Comptuer and Comm. Soc.* (Infocom 96), IEEE Press, 1996, pp. 553-560.

Gopalakrishnan R et al., "Stability and Fairness Issues in Layered Multicast," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 99), 1999

Gopalakrishnan R., Griffioen J., Hjsson G., Sreenan C., and Wen S., "A simple loss differentiation approach to layered multicast," *Proc. of IEEE INFOCOM*, 2000.

H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," *Request for Comments (RFC) 1889*, Internet Eng. Task Force (IETF), Jan. 1996.

ISO/IEC 11172, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s." International Organization for Standardization (ISO), 1993.

ISO/IEC 13818. "Generic coding of moving pictures and associated audio information." International Organization for Standardization (ISO), 1996.

ISO/IEC 14496. "Coding of audio-visual objects." International Organization for Standardization (ISO), 1999.

ITU "Video codec for audiovisual services at px64 kbits/s", ITU-T Recommendation H.261, Inter. Telecommunication Union, 1993.

IUT "Video coding for low bit rate communication", ITU-T Rec. H.263, Inter. Telecommunication Union, version 1, 1996; version 2, 1997.

Legout A. and E.W. Biersack, "Pathological Behaviors for RLM and RLC," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 00), 2000

Li B. and Liu J., "Multirate video multicast over the internet: An overview," *IEEE Network*, vol. 17, no. 1, pp. 24–29, January 2003.

Li X., M. Ammar, and S. Paul, "Video Multicast over the Internet," *IEEE Network Magazine*, vol. 13, no. 2, Apr. 1999, pp. 46-60.

Li X., Paul S., Pancha P., and Ammar M., "Layered video multicast with retransmission (lvmr): Evaluation of error recovery schemes," *Proc. Of NOSSDAV*, 1997.

Liu J., Li B., and Zhang Y.-Q., "Adaptive video multicast over the internet," *IEEE MultiMedia*, vol. 10, no. 1, pp. 22–33, 2003.

Matrawy A. and Lambadaris I., "A survey of congestion control schemes for multicast video applications," *IEEE Communications Surveys and Tutorials*, pp. 22–31, Volume 5, N. 2 2003.

McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. ACM Sigcomm Conf.*, ACM Press, 1996, pp. 117-130.

Nakauchi H. M. K. and Aoyama T., "A network-supported approach to layered multicast," *Proc. of IEEE ICC*, 2001.

Pereira F. and Ebrahimi T., "*The MPEG-4 Book*". Prentice Hall PTR, 2002.

Sarkar S. and L. Tassiulas, "Fair Allocation of Discrete Bandwidth Layers in Multicast Networks," *Proc. Ann. Joint Conf. IEEE Comptuer and Comm. Soc.* (Infocom 00), IEEE Press, 2000.

Vicisano L., L. Rizzo, and J. Crowcroft, "TCP-Like Congestion Control for Layered Multicast Data Transfer," *Proc. Ann. Joint Conf. IEEE Comptuer and Comm. Soc.* (Infocom 98), IEEE Press, 1998, pp. 996-1003.

Vickers B., C. Albuquerque, and T. Suda, "Source Adaptive Multi-Layered Multicast Algorithms for Real-Time Video Distribution," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, 2000, pp. 720-733.

Wang Y. and Q. Zhu, "Error control and concealment for video communications: A review," *Proceedings of the IEEE*, May 1998.

Wee S., J. Apostolopoulos and N. Feamster, "Field-to-Frame Transcoding with Temporal and Spatial Downsampling," *IEEE International Conference on Image Processing*, October 1999.

Zhang Z. and Li V. O. K, "Router-assisted layered multicast," *Proc. of IEEE ICC*, 2002.