Real-Time Imaging 7, 221–235 (2001)
doi:10.1006/rtim.2001.0224, available online at http://www.idealibrary.com on IDEAL®

A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia

Though the integrated services model and resource reservation protocol (RSVP) provide support for quality of service, in the current Internet only best-effort traffic is widely supported. New high-speed technologies such as ATM (asynchronous transfer mode), gigabit Ethernet, fast Ethernet, and frame relay, have spurred higher user expectations. These technologies are expected to support real-time applications such as video-on-demand, Internet telephony, distance education and video-broadcasting. Towards this end, networking methods such as service classes and quality of service models are being developed. Today's Internet is a heterogeneous networking environment. In such an environment, resources available to multimedia applications vary. To adapt to the changes in network conditions, both networking techniques and application layer techniques have been proposed. In this paper, we focus on the application level techniques, including methods based on compression algorithm features, layered encoding, rate shaping, adaptive error control, and bandwidth smoothing. We also discuss operating system methods to support adaptive multimedia. Throughout the paper, we discuss how feedback from lower networking layers can be used by these application-level adaptation schemes to deliver the highest quality content.

Bobby Vandalore[1], Wu-chi Feng[2], Raj Jain*[,3] and Sonia Fahmy[4]

[1]*Amber Networks, Santa Clara, CA, U.S.A.*
E-mail: bobby@ambernetworks.com; vandalore.2@osu.edu
[2]*Dept. of CIS, Ohio State University, OH, U.S.A.*
E-mail: wuchi@cis.ohio-state.edu
[3]*Nayna Networks, Milpitas, CA, U.S.A. E-mail: jain@acm.org*
[4]*Purdue University, Lafayette, IL, U.S.A.*
E-mail: fahmy@cs.purdue.edu

Introduction

The Internet was designed for best-effort data traffic. With the development of high-speed technologies such as ATM (asynchronous transfer mode), gigabit Ethernet, and frame relay, user expectations have increased. Real-time multimedia applications including *video-on-demand*, *video-broadcast*, and *distance education* are expected to be supported by these high-speed networks. Organizations such as the IETF (Internet Engineering Task Force), ATM Forum, and ITU-T (International Telecommunications Union) are developing new protocols (e.g., real-time transport protocol), service models (e.g., integrated services and differentiated services), and service classes (e.g., ATM Forum service categories and ITU-T transfer capabilities) to satisfy the requirements of multimedia applications.

The Internet is a heterogeneous environment connecting various networking technologies. Even with

*Corresponding author: Dr. Raj Jain

1077-2014/01/060221 + 15 $35.00/0

networking support through service classes, the available network resources to a multimedia applications will change dynamically over time. For example, the network conditions may change due to difference in link speeds (ranging from 28.8 Kbps modem links to 622 Mbps OC-12 links) or variability in a wireless environment caused by interference and mobility. One way of achieving the desired quality of service in such situations is by massively over-provisioning resources for applications. This solution, however, leads to inefficiency. Without over-provisioning, network resources can be used efficiently if applications are capable of adapting to changing network conditions.

Adaptation of multimedia applications can be done at several layers of the network protocol stack. At the physical layer, adaptive power control techniques can be used to mitigate variations in a wireless environment. At the data link layer, error control and adaptive reservation techniques can be used to protect against variation in error and available rate. At the network layer, dynamic re-routing mechanisms can be used to avoid congestion and mitigate variations in a mobile environment. At the transport layer, dynamic re-negotiation of connection parameters can be used for adaptation. Applications can use protocols such as real-time streaming protocol (RTSP) [1] and real-time protocol (RTP) [2]. At the application layer, the application can adapt to changes in network conditions using several techniques including hierarchical encoding, efficient compression, bandwidth smoothing, rate shaping, error control, and adaptive synchronization.

This paper focuses mainly on the application layer techniques for adaptation. The rest of the paper is organized as follows. Challenges and issues that arise in supporting adaptive multimedia applications are discussed in the next section. The following section gives an overview of the compression methods and discusses techniques for adaptation based on these methods. The next section discusses the application level adaptive streaming techniques. These techniques include both reactive and passive methods of adaptation. Some example adaptive applications are then presented. Next, the enhancements of operating systems to support adaptive multimedia are discussed. The penultimate section presents related work to the issue of supporting multimedia applications and finally, the last section gives the summary of the paper. Throughout the paper, we also discuss how low-level network feedback (such as available capacity or error rate) can be/is used in adaptation methods.

Issues in Supporting Adaptive Multimedia

Before presenting the various application level adaptation techniques we first present the issues and problems related to supporting adaptive multimedia applications. Multimedia applications have different characteristics and requirements compared to traditional data applications. They are delay sensitive, to a certain extent loss tolerant, and maybe interactive unlike the data applications.

Challenges of supporting adaptive multimedia

The problem of supporting multimedia applications is a challenging problem because of the following network and system related phenomena:

Bandwidth: Multimedia applications have varying bandwidth requirement (e.g., amount of data varies from frame to frame in a video application). The available network capacity is also a varying quantity. Hence, matching the varying available bandwidth with dynamically changing bandwidth requirement poses a challenging problem.

Error Rate: Multimedia applications are sensitive to loss. For good quality they need certain minimum guarantees on error rate. Typical error requirements for audio and video are in the range of 10^{-4}–10^{-5} and 10^{-3}–10^{-4} respectively.

Delay and Jitter: 'Interactive' multimedia applications need minimum delay bounds to be guaranteed. Delay jitter[1] (variation in delay) degrades multimedia applications' performance since they need data in a periodic manner (e.g., MJPEG application with frame rate of 30 frames/s need complete frame data every 33 ms.)

Synchronization: Synchronization of different media arriving at different times is necessary at the user end.

Heterogeneity: Multimedia servers have to simultaneously handle receivers which have different characteristics and requirements.

User Interface: Application programming interfaces (API) have to be developed to control the adaptation process. Ideally, the user interface should be something

[1] For non-interactive multimedia applications, the jitter can be mitigated by buffering at the client side.

simple such as one knob for controlling each feature of the multimedia application.

Client/Server issues

The client/server model depicted in Figure 1 shows the various components at the server and client side of the system. The figure also shows the flow of feedback, which can be used at various layers of the network protocol and at the application layer to adapt to changing conditions.

Table 1 gives the list of techniques that can used in the various components of client/server model. These techniques are described in detail in the subsequent sections. The *C/S* column indicates whether the technique is a client side or server side issue. The *Dependency* column of the table explains how each technique relates to other components. The issue addressed and the advantage of the technique are also shown in the table.

Support for multimedia applications in the current Internet is at its initial stages. In corporate intranets where bandwidth is reasonably abundant, ranging from 380 Kbps to 100 Mbps, multimedia applications such as video conferencing can be easily supported. But in public networks such as the Internet, the quality of service achieved by multimedia applications is much smaller than in intranets. With IETF working to solve the QoS problem, we expect the support for multimedia applications to improve considerably in coming years. We believe (video based) interactive multimedia applications will be the next killer application after Voice-over IP (Internet Protocol). With the advent of optical networks which increases bandwidth dramatically, increasing processing speeds, and advances in compression technology, networked multimedia applications of the future may be able to support HDTV (high definition television) quality of video.

There are also several issues, which are not tackled both by the application level techniques discussed here and techniques used in other layers of protocol stack discussed elsewhere. A few of these are mentioned below:

Middleware: Designing and implementing a middleware to support adaptive multimedia applications is still unaddressed. Middleware should provide simple interfaces between the application, operating system and the network.

CPU to NIC bottleneck: Network interface cards (NIC) are designed to keep up with the growing link capacity. CPU speeds are also doubling every 18 months. But there is a bottleneck at the I/O bus, which connects the CPU to the NIC. Traditionally, NICs are treated as peripheral devices, but if the system has to keep up with the increasing speeds then they should be considered as peer devices to CPU.

Scalability: Wide scale deployment and usage of multimedia applications is still at its early stages. The problem of scalability will have to be addressed as the demand and capability of the multimedia applications grow.

Compression Level Methods

While the rest of the paper deals with multimedia in general, in this section we briefly examine video compression algorithms and their features which are

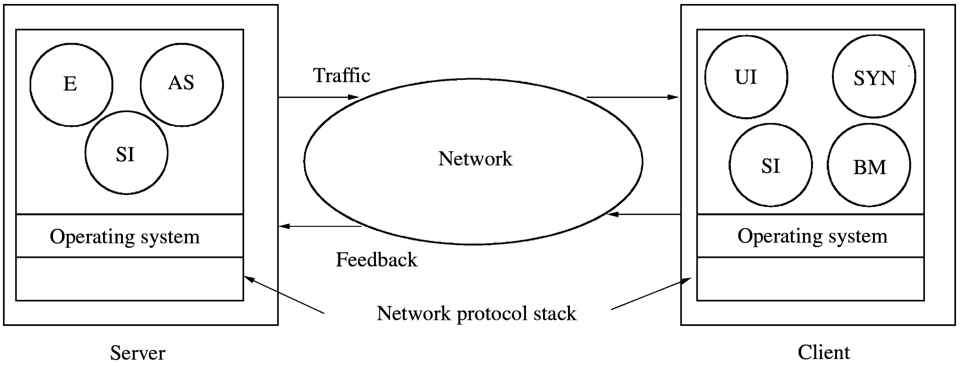

Figure 1. Client/Server model showing the various components of the server and client to support multimedia applications. E, encoding; AS, application streaming; SI, system interface; UI, user interface; SYN, synchronization; BM, buffer management.

Table 1. Adaptation techniques that can be used in the different components of the Client/Server (C/S) model

Component	C/S	Technique	Dependency	Issue tackled	Advantage
Encoding	S	Compression algorithms	Streaming scheme	Match source to available rate	Independent of network layer
Application streaming	S	Multiple Layer transmission	Encoding	Mitigate effect of congestion	Can support heterogeneous receivers
	S	Rate shaping	Encoding, network layer	Match source to available rate	Use network feedback
	S&C	Error control	Encoding, network layer	Mitigate effect of losses	Use network feedback
	C	Synchronization	Network layer	Synchronizing different media	Solves synchronization problem
	S&C	Smoothing	Network layer	Change transmission scheme based on a-priori knowledge	Better use of resources
Operating system	S&C	Process scheduling	Network layer	Treat multimedia application preferentially	Can satisfy real-time requirements

useful for adaptation. Two reasons for focusing on video are: (1) video requires larger bandwidth (100 Kbps to 15 Mbps) than audio (8 Kbps–128 Kbps), and (2) humans are more sensitive to loss of audio than video, because audio is important for comprehension. Therefore, audio is given higher priority and as a result only the video can be used for adaptation. Hence, we generally should bias towards adapting the video part of the multimedia application.

Transmitting raw video information is inefficient. Hence, video is invariably compressed before transmission. The three main compression techniques used for video are: (1) discrete cosine transformation (DCT) based, (2) wavelet transforms based, and (3) proprietary methods. Other methods of compressing video not discussed here include vector quantization [3,4] and content-based compression. Adapting to changing network conditions can be achieved by a number of techniques at the compression level (video encoder) including layered encoding, changing parameters of compression methods, and using efficient compression methods. In the event of bandwidth not being available to the video source it can reduce its encoding rate by temporal scaling (reducing frame rate) or spatial scaling (reducing resolution).

MPEG compression standard

DCT is the compression method used in the popular MPEG (Moving Picture Experts Group) set of standards [5]. MPEG standards are used for both audio and video signals. MPEG-2, MPEG-1 and JPEG (an earlier standard for still images) all use discrete cosine transformations. The transformed coefficients are quantized using scalar quantization and run length encoded before transmission. The transformed higher frequency coefficients of video are truncated since the human eye is insensitive to these coefficients. The compression relies on two basic methods: intra-frame DCT coding for reduction of spatial redundancy, and inter-frame motion compensation for reduction of temporal redundancy. MPEG-2 video has three kinds of frames: I, P, and B. I frames are independent frames compressed using only intra-frame compression. P frames are predictive, which carry the signal difference between the previous frame and motion vectors. B frames are interpolated, i.e., encoded based on the previous and the next frame. MPEG-2 video is transmitted in group of pictures (GoP) format which specifies the distribution of I, P, and B frames in the video stream.

There are several aspects of the MPEG compression methods which can be used for adaptation. First, the rate of the source can be changed by using different quantization levels and encoding rate [6,7]. Second, DCT coefficients can be partitioned and transmitted in two layers with different priorities. The base layer carries the important video information and an additional layer improves the quality. In the event of congestion, the lower priority layer can be dropped to reduce the rate [8–10].

Wavelet encoding

In wavelet compression, the image is divided into various sub-bands with increasing resolutions. Image data in each sub-band is transformed using a wavelet function to obtain transformed coefficients. The transformed coefficients are then quantized and run length encoded before transmission. In a sense, wavelet compression results in progressively encoded video. Two common approaches for wavelet compression are to use a motion-compensated two-dimensional (2D) wavelet function [11] or a 3D wavelet [12].

Wavelet compression overcomes the blocking effects of DCT based methods since the entire image is used in encoding instead of blocks. An important feature of wavelet transforms is the support of scalability for image and video compression. Wavelet transforms coupled with encoding techniques provide support for continuous rate scalability, where the video can be encoded at any desired rate within the scalable range [13]. A wavelet encoder can benefit from network feedback such as available capacity to achieve scalability [14].

Proprietary methods

Commercial applications such as Real Networks Inc.'s RealVideo and Intel's Indeo use proprietary methods for compression and adaptation. These proprietary schemes use both DCT based and wavelet based techniques for compression. A distinguishing feature of these methods is that they are optimized to work for particular bandwidths such as 28.8 Kbps and 56 Kbps. Some of the techniques used for adaptation by these applications are discussed later in the paper.

Application Streaming

Most multimedia applications transmit data in streams, and at the user end the data is used as soon as it is received. Application streaming is the technique used by such multimedia applications for sending multimedia data streams. At the application streaming level, adaptation techniques include layered encoding, adaptive error control, adaptive synchronization and smoothing. These methods can be classified into *reactive* and *passive* according to their approach towards adaptation. In *reactive* methods, the application modifies its traffic to suit the changes in the network. In *passive* methods, the application aims to optimize the usage of network resources. Rate shaping and smoothing of stored video are example applications of reactive and passive methods respectively.

Layered encoding

In layered encoding, a passive method, the video information is encoded into several layers. The base layer carries important video (lower order coefficients of DCT) and critical timing information. The higher layers improve the quality of video progressively. The receiver can get a reasonable quality with the base layer, and quality improves with reception of higher layers. The encoder assigns priorities to the encoded layers, with the base layer having the highest priority. When the network transmits layered video, it can drop lower priority (higher) layers in the event of congestion. A discussion of adaptive transmission of multi-layered video is given in [15]. Here the layered encoding method is made reactive by adding or dropping layers based on network feedback. The paper discusses both credit-based and rate-based approaches for providing feedback.

An optimal data partitioning method for MPEG-2 encoded video is discussed in [8]. In this method, the data in the MPEG-2 stream is partitioned into two layers (which can be done even after encoding). The two streams are transmitted over an ATM-based network, with ATM cells of the lower priority stream having their CLP (cell loss priority) bit set. The paper discusses an optimal algorithm to partition the MPEG-2 video stream. The problem is posed as an optimization problem and the Lagrangian optimization technique is used for finding the optimal partitioning for I, P, and B frames. Data partitioning methods can benefit from network feedback. For example, if the network indicates that more bandwidth is available, more data can be sent in the base layer, and conversely data in the base layer can be reduced when bandwidth is scarce.

Receiver driven Multicast

Receiver driven Layered Multicast (RLM), a reactive method, was the first published scheme which described how layered video can be transmitted and controlled [16]. In RLM, receivers dynamically subscribe to the different layers of the video streams. Receivers use "probing" experiments to decide when they can join a layer. Specifically, if a receiver detects congestion, the receiver quits the multicast group of the current highest layer (drops a layer), and when extra bandwidth is available, it joins the next layer (adds a layer). The network congestion is detected by packet losses. Extra capacity is detected by join experiments. In a join experiment, the receiver measures the packet loss after joining. The join experiment fails if the packet loss is above a certain threshold. Receiver join experiments are randomized to avoid synchronization. The overhead of join experiments in the presence of a large number of receivers is controlled by the receivers learning from the join experiments of others, instead of initiating their own. Currently, extra capacity is only estimated in RLM. The low-level network feedback can aid the receivers in measuring precisely the available capacity. Hence, this scheme will benefit if network layer feedback is used.

Layered video multicast with retransmission [17] is another method which uses layered video. The issue of inter-session fairness and scalable feedback control of layered video is discussed in [18].

Rate shaping

Rate shaping techniques are reactive and attempt to adjust the rate of traffic generated by the video encoder according to the current network conditions. Feedback mechanisms are used to detect changes in the network and control the rate of the video encoder.

Video has been traditionally transported over connections with constant bit rate (e.g., telephone or cable TV networks). The rate of the video sequence changes rapidly due to scene content and motion. The variable rate video is sent to a buffer which is drained at a constant rate. In such a situation, the video encoder can achieve constant rate by controlling its compression parameters based on feedback information such as the buffer occupancy level. A similar technique is used for adapting video and audio to network changes. In these cases, the feedback from the network is used instead of local buffer information. Control mechanisms for audio and video are presented in [7,19]. Rate shaping of the IVS video coder (which uses the H.261 standard) is discussed in [7]. The rate shaping can be obtained by changing one or more of the following:

- *Refresh rate*: Refresh rate (frame rate) is the rate of frames which are encoded by the video encoder. Decreasing the refresh rate can reduce the output rate of the encoder, but will reduce quality.
- *Quantizer*: This specifies the number of DCT coefficients that are encoded. Increasing the quantizer decreases the number of encoded coefficients and the image is coarser.
- *Movement detection threshold*: For inter-frame coding, the DCT is applied to signal differences. The movement detection threshold limits the number of blocks which are detected to be "sufficiently different" from the previous frames. Increasing this threshold decreases the output rate of the encoder. Again, this results in reduced video quality.

Two modes are used for controlling the rate of the encoder. In Privilege Quality mode (PQ mode), only the refresh rate is changed. In Privilege Rate mode (PR mode), only the quantizer and movement detection threshold are changed. PQ mode control results in higher frame rates, but with lower SNR (signal-to-noise ratio) than PR mode.

The packet loss information is used as feedback. The receiver sends periodically its current loss rate. The following simple control algorithm is used to dynamically control the rate of the video encoder:

```
If $median\_loss > tolerable\_loss$
    $max\_rate = \max(max\_rate/GAIN, min\_rate)$
else
    $max\_rate = \max(max\_rate + INC, min\_rate)$
```

In presence of congestion, indicated by loss, the rate is quickly decreased due to multiplicative factor $1/GAIN$. Otherwise, the rate is slowly increased by the additive factor INC. This multiplicative decrease, additive increase mechanism adapts well to network changes.

2D scaling changes both the frame rate and the bit rate based on the feedback [20]. Experimental results show that the system performs well in a rate constrained environment such as the Internet. A heuristic (success rate) is used to decide whether the rate can be increased. The low-level network feedback information, if available, can replace this heuristic. Rate shaping mechanisms use similar methods but differ in how the rate shaping is achieved. Other rate change approaches include block dropping [21] and frame dropping [22].

Error control

The error rate is variable in a wireless network due to interference, and the loss rate is variable in the Internet due to congestion. Multimedia applications need to adapt to changes in error and loss rates. Two approaches to mitigate errors and losses are Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). ARQ is a closed-loop and reactive mechanism in which the destination requests the source to retransmit the lost packets. FEC is an open-loop and passive method in which source sends redundant information, which can partly recover the original information in the event of packet loss. ARQ increases the end-to-end delay dramatically in networks such as the Internet. Hence, ARQ is not suitable for error control of multimedia applications in the Internet. It may be used in high-speed LANs where round trip latencies are small. Other error control methods include block erasure codes, convolutional codes, interleaving and multiple description codes.

Adaptive FEC for Internet audio. An adaptive FEC-based error control scheme (a reactive method) for interactive audio in the Internet is proposed in [23]. The FEC scheme used is the "signal processing" FEC mechanism [24]. Assume that the packets are numbered, $1, 2, \quad, n$. In this scheme, the *n*th packet includes, in addition to its encoded signal samples, information about previous packet $(n-1)$ which can be used to approximately reconstruct that packet if it is lost. The IETF recently standardized this scheme to be used in Internet telephony. The scheme works only for isolated packet losses, but can be generalized to tolerate consecutive packet losses by adding redundant versions of additional (previous) packets $(n-2$ and $n-3)$. The FEC-based scheme needs more bandwidth, so it should be coupled with a rate control scheme. The joint rate/FEC scheme can be used to adaptively control the rate and the amount of redundant information to be sent by the FEC method. The inventors of the scheme formulate the problem of choosing the FEC-method to use under the constraints of the rate control scheme as an optimization problem. A simple algorithm is used to find the optimal scheme. Actual measurements of the scheme for audio applications between France and London have shown that the scheme performs well and the perceptual quality of the audio is good.

Adaptive FEC for Internet video. An adaptive FEC-based scheme for Internet video is discussed in [25]. The packet can carry redundant FEC information for up to four packets, i.e., packet n carries redundant information about packets $n-1$, $n-2$ and $n-3$. Let $n-i$ indicate that packet n includes information about $n-i$. The different possible combinations of these methods are: (n), (n, n–1), (n, n–2), (n, n–1, n–2) and (n, n–1, n–2, n–3). These are numbered as combination-1 through to combination-5. Different combinations can be used to adapt to network changes. The network changes are detected through packet loss, and a loss threshold (high loss) is used in the algorithm for adaptation. The following simple adaptation algorithm was used:

```
If $loss \geq high\ loss$
    $Combinaton = \min(combination + 1, 4)$
else
    $Combinaton = \max(combination - 1, 0)$
```

This algorithm adds more error protection when there is more loss and less protection when the losses are low.

One way to use network feedback in this method is to couple the rate available and the FEC *Combination* used. For example, information about available rate and loss rate received as feedback from the network can be used to choose the FEC combination for error protection.

Adaptive synchronization

Synchronization is an important problem for multimedia applications. Synchronization problems arise due to clock frequency drift, network delay, and jitter. Adaptive synchronization can be used for multipoint multimedia teleconferencing systems [26, 27]. The adaptive synchronization technique proposed in [26] is immune to clock offset and/or clock frequency drift, does not need a global clock, and provides the optimal delay and buffering for the given QoS requirement. The adaptive synchronization technique can be used to solve both intramedia (in a single stream) synchronization and intermedia (among multiple streams) synchronization.

The main idea used in the synchronization algorithm is to divide the packets into *wait*, *no wait* and *discard* categories. Packets in the *wait* bucket are displayed after some time, *no wait* packets are displayed immediately, and *discard* category packets are discarded. The basic adaptive synchronization algorithm requires the user to specify the acceptable synchronization error, maximum jitter and maximum loss ratio. The sender is assumed to put a timestamp in the packets. At the receiver, the playback clock (PBC) and three counters for *no wait*, *wait* and *discard* packets are maintained. The algorithm specifies that when packets arrive early and enough wait packets have been received, the PBC is incremented. Similarly, when a threshold of *no wait* or *discard* packets are received, the PBC is decremented. This adaptive algorithm is shown to be immune to clock drift. Achieving intramedia synchronization is a straightforward application of the basic algorithm. For intermedia synchronization, a group PBC is used, which is incremented and decremented based on the slowest of the streams to be synchronized.

The network delay is only estimated in this adaptive algorithm. The adaptation can benefit if the low-level feedback provides accurate information on the delay experienced in the network.

Smoothing

One way to mitigate the rate variations of the multimedia application is to perform shaping or smoothing of the video information transmitted. Recent studies show that smoothing allows for greater statistical multiplexing gain.

For live (non-interactive) video, a sliding-window of buffers can be used, and the buffer can be drained at the desired rate. This method is used in SAVE (smoothed adaptive video over explicit rate networks) [6], where a small number of frames (30) is buffered in a window. The video is transmitted over the ATM ABR (available bit rate) service, where the feedback from the network is indicated explicitly. The SAVE algorithm (a reactive method) uses this feedback information to dynamically change the quantizer value of the MPEG-2 encoder. Note that this method already uses the low-level network feedback. Similar approaches have been proposed in [28,29]. For pre-recorded (stored) video, the a-priori video (frame) information can be utilized to smooth the video traffic at the source. Bandwidth smoothing (a passive method), can reduce the burstiness of compressed video traffic in video-on-demand applications. The main idea behind smoothing techniques is to send ahead large frames which need to be displayed later when there is enough buffer space at the client. There has been considerable research in this area resulting in several smoothing algorithms [30–34]. These differ in the optimality condition achieved, and whether they assume that the rate is constrained or the client buffer size is limited. A good comparison of bandwidth smoothing algorithm is given in [35]. In the next subsection, we discuss the idea of bandwidth smoothing in more detail.

Smoothing algorithms. A compressed video stream consists of n frames, where frame i requires f_i bytes of storage. To permit continuous playback, the server must always transmit video frames ahead to avoid buffer underflow at the client. This requirement can be expressed as:

$$F_{under}(k) = \sum_{i=0}^{k} f_i,$$

where $F_{under}(k)$ indicates the amount of data consumed at the client when it is displaying frame k $(k = 0, 1, \quad, n - 1)$. Similarly, the client should not receive more data than its buffer capacity. This requirement is represented as:

$$F_{over}(k) = b + \sum_{i=0}^{k} f_i,$$

where b is client buffer size. Consequently, any valid transmission plan should stay within the river outlined by these vertically equidistant functions. That is,

$$F_{under}(k) \leq \sum_{i=0}^{k} c_i \leq F_{over}(k),$$

where c_i is the transmission rate during frame slot i of the smoothed video stream.

Generating a bandwidth plan is done by finding m consecutive runs which use constant bandwidth $r_j, (j = 1, \quad, m)$. Within the jth run, the frames are transmitted at the constant rate r_j. The rate change occurs to avoid buffer overflow or buffer underflow. Mathematically, the runs of bandwidth plan must be such that the amount of frame data transfered forms a monotonically increasing, piecewise linear function. Different bandwidth smoothing algorithms result from choosing the rate changes among the bandwidth runs. Several optimal bandwidth allocation plan-generating algorithms are discussed in [30]. These algorithms achieve optimal criteria such as minimum number of bandwidth changes, minimum peak rate requirement, and largest minimum bandwidth requirement. We discuss below an online smoothing algorithm, a proactive buffering mechanism and two algorithms which combine smoothing and rate shaping techniques.

Online smoothing. Live video applications such as broadcasting of a lecture and news are delay tolerant, in the sense that the user does not mind if the video is delayed in the order of a few seconds (or even a minute). For these live video applications, smoothing techniques (passive methods) can significantly reduce the resource variability.

Several window based online smoothing algorithms (passive methods) are discussed in [36]. In the first approach, a hop-by-hop window smoothing algorithm is used. Here the server stores up to a window of W frames. The smoothing algorithm is performed over this window of frames taking into consideration the server and client buffer constraints. After the transmission of W frames, the smoothing algorithm is performed for the next set of W frames. This algorithm does not handle an inter-mixture of large I frames among P and B frames, since only in the first window the transmission of I frame is amortized. The consecutive windows can be aligned with an I frame at the end of each window.

While in the hop-by-hop algorithm, the server cannot prefetch data across window boundaries, the sliding-window method SLWIN(α) uses a sliding window of size W for smoothing. The smoothing algorithm is repetitively performed for every α frames time units over the next W frames. The sliding-window performs better but is more complex, since the smoothing algorithm is executed more times than in the hop-by-hop method.

Proactive buffering. Another passive method, rate constrained bandwidth smoothing for stored video, is given in [37]. Here, the rate is assumed to be constrained to a given value (for example, the minimum cell rate (MCR) in the ABR service). The algorithm proactively manages buffers and bandwidth. This method uses the rate constrained bandwidth smoothing algorithm (RCBS) [32] which minimizes the buffer utilization for a given rate constraint. In RCBS, the movie frames are examined in reverse order from the end of the movie. The large frames which require more than the constrained rate are prefetched. These prefetches fill the gaps of earlier smaller frames.

The proactive method identifies feasible regions of the movie. A feasible range is where the average rate requirement of the range is less than the constrained rate. The movie frames are examined in the reverse order and feasible regions are identified. The algorithm keeps track of the expected buffer occupancy at the client side and finds the maximal feasible regions. When the rate constraint is violated, frames are dropped. The dropped frames are placed apart to avoid consecutive frame drops. The proactive method results in maximizing the minimum frame rate for a given rate constraint. The low-level network feedback can be used in this method as follows: assume that the network can guarantee the constrained rate and inform the source through feedback if any extra bandwidth is available. This extra bandwidth and current buffer occupancy level can be used to decide if additional frames can be sent.

Bridging bandwidth smoothing and adaptation techniques. Passive adaptation techniques like bandwidth smoothing algorithms take advantage of a priori information to reduce the burden on the network. However, they do not actively alter the video stream to make them network sensitive. The reactive techniques usually do not take advantage of the a priori information and hence, may not provide the best possible quality video over best effort networks. Some recent work has focused on augmenting reactive techniques to take advantage of this a priori knowledge.

A priority-based technique (reactive method) is used to deliver prerecorded compressed video over best-effort networks in [38]. Multiple level priority queues are used in addition to a window at each level to help smooth the video frame rate while allowing it to change according to changing network conditions. The scheme uses frame dropping (adaptation technique) and a priori knowledge of frame sizes. The scheme tries to deliver the frame of highest priority level (base layer) before delivering the frames of enhancement layers. Adaptation is accomplished by dropping frame at the head of the queue if enough resources are not available.

Another algorithm which combines the smoothing and rate changing technique (frame dropping) is discussed in [39]. An efficient algorithm to find the optimal frame discards for transporting stored video over best-effort networks is given. The algorithm uses the selective frame discarding technique. The problem of finding the minimum number of frame discards for a sequence of frames is posed as an optimization problem. A dynamic programming based algorithm and several simpler heuristic algorithms are given to solve this problem.

Example Adaptive applications

In this section, we present two commercial adaptive applications: (1) real networks suite, and (2) Vosaic: video mosaic. These commercial applications are currently available and incorporate some of the adaptation techniques discussed in the previous section. They also incorporate additional optimization techniques. There are several other adaptive multimedia applications and architectures developed by academia such as Berkeley's vic [16], video conferencing system for the Internet (IVS) [7] developed by INRIA in France, Berkeley's continuous media tool kit (CMT) [40], OGI's adaptive MPEG streaming player [41], and MIT's View Station [42]. Most of these applications have contributed to the research results of adaptation methods discussed in earlier sections.

Real Networks' solutions

The Real Networks company provides commercial player (free) and server software for streaming applications. Their products include a number of features such as scalability (can support 500 to 1000 simultaneous streams using IP multicast), bandwidth negotiation, dynamic connection management, sophisticated error control, and buffered play. In this section, we review some of the streaming techniques used in Real Networks products for adaptively transmitting multimedia streams over the Internet.

RealVideo: Adaptive techniques. RealVideo [43] uses the RTSP streaming protocol and can run over both UDP and TCP protocols. RealVideo uses a robust version of UDP to reduce the impact of packet losses. It uses damage-resistant coding to minimize effects of packet loss in video. It also uses FEC-based methods when frame rates are low. The RealVideo supports two encoders: realvideo standard and realvideo fractal. The realvideo standard encoding can support a range of encoding from 10 Kbps to 500 Kbps. This encoder is specifically optimized to work over 28.8 Kbps and 56 Kbps modem lines.

SureStream: Multiple stream encoding. One approach to counter the changing network conditions at the server is to reduce the amount of data by dropping frames (stream-thinning). A limitation of this approach is that the resulting video is not of the same quality as the one optimized to the lower rate. The SureStream mechanism [44] overcomes this limitation by using two methods. First, it supports multiple streams encoded at different rates to be stored in a single file. Second, it provides a mechanism for servers and clients to detect changes in bandwidth and choose an appropriate stream. Changes in the bandwidth are detected by measurements of received frame rate.

SureStream uses the Adaptive Stream Management (ASM) functionalities available in the RealSystem API. ASM provides rules to describe the data streams. These rules provide facilities such as marking priorities and indicating average bandwidth for a group of frames. This information is used by the server for achieving adaptability. For example, the server might drop lower priority frames when the available rate decreases. A condition in the rule can specify different client capabilities. For example, it can indicate that the client will be able to receive at 5 to 15 Kbps and can tolerate a packet loss of 2.5%. If the network conditions change, the clients can subscribe to another appropriate rule.

The techniques used in RealVideo and SureStream can benefit from low-level network feedback. For example, instead of detecting bandwidth changes through measurements, the server can use the available bandwidth information from the lower network layer to the choose the appropriate stream to transmit.

Vosaic: Video mosaic

The design and implementation of *video mosaic* (*Vosaic*) is given in [45]. HTTP (hypertext transfer protocol) supports only reliable transmission and does not support streaming media. Vosaic uses the VDP real-time protocol which can be used to support streaming video in a WWW (World Wide Web) browser. Vosaic is built upon the NCSA Mosaic WWW browser. The URL (Uniform Resource Locator) links of Vosaic allows specification of media types such as MPEG audio and MPEG video. VDP uses two connections: an unreliable one for streaming data and a reliable one for control. In the control channel, the client application can issue VCR-like instructions such as play, stop, fast forward, and rewind. An adaptation algorithm is used to adjust the rate of the stream according to network conditions. The clients indicate two metrics: frame drop rate and packet drop rate as measured at the client, to the server as feedback using the control channel. The server initially transmits frames at the recorded rate and adjusts the frame rate based upon the feedback received from the client side. Experimental results show that the frame rate improves considerably when the VDP protocol and the adaptation algorithm are used (e.g., frame rate improved to 9 frames/s from 0.2 frames/s).

Vosaic can definitely benefit from low-level network feedback. Currently, the network condition is detected by measurements of the received frame rate at the client and sent to the server. Instead, the server can use network feedback such as available rate to dynamically adjust its frame rate.

Operating System Support for Adaptive Multimedia

Conventional operating systems are not designed for multimedia applications. For example, playback applications need to access CPU resources periodically during playout. This entails that the operating system provide ways for multimedia applications to access resources. To develop adaptive multimedia applications, there is a need for an operating system capability which can provide information about available resources. In this section, we discuss some techniques which are used in operating systems to support adaptive multimedia streaming.

Integrated CPU and network-I/O QoS management

Multimedia applications use multiple resources, and these resources such as CPU availability and bandwidth change dynamically. An integrated QoS management system to manage CPU, network and I/O resources is proposed in [46]. This cooperative model enables multimedia end-systems and OS to cooperate dynamically for adaptively sharing end-system resources. The thesis of this work is that end-system resources should be allocated and managed adaptively. The proposed OS architecture called AQUA (Adaptive Quality of service Architecture) aims to achieve this objective.

In AQUA, when an application starts, it specifies a partial QoS (for example, a video application can specify frame rate and may not specify bandwidth requirement). The OS allocates initial resources such as CPU time based on this QoS specification. As the application executes, the OS and the application cooperate to estimate the resource requirements and QoS received. Resource changes are detected by the measuring QoS. Then, the OS and the application renegotiate and adapt to provide predictable QoS with current resource constraints. To enable these functionalities, the AQUA framework includes a QoS manager, QoS negotiation library, and usage-estimation library.

The application specifies an adaptation function when the connection is setup. The QoS manager calls this function when it detects changes in QoS. Using this methodology, a CPU-resource manager and network-I/O manager has been implemented in AQUA. A composite QoS manager uses the services of both CPU and network-I/O managers. This facilitates an integrated way to manage resources in AQUA. The AQUA framework can use low-level network feedback to detect current availability of network resources. The QoS measuring function can be enhanced by using the network layer feedback.

Adaptive Rate-controlled scheduling

Multimedia applications need to periodically access resources such as CPU. The operating system needs to schedule multimedia applications appropriately to support such needs. The CPU requirement of a multimedia application might dynamically change due to the frame rate change caused by scene changes or network conditions. A framework called Adaptive Rate-controlled (ARC) scheduling is proposed to solve this problem in [47]. It consists of a rate-controlled online CPU scheduler, an admission control interface, a monitor, and a rate adaptation interface.

ARC operates in a operating system which supports threads (Solaris 2.3). The threads can be of three types: RT (real-time), SYS (system) or TS (timesharing). RT threads have the highest priority in accessing the CPU. The online CPU scheduler schedules the threads belonging to these classes based on their priorities.

Adaptive rate-controlled scheduling is achieved as follows: multimedia threads register with the monitor thread during connection setup. The monitor thread is executed periodically (every 2 s). It estimates for each registered thread whether the CPU usage *lags* (how fast the thread is running ahead of its rate) or *lax* (measures how much of the CPU is unused). This estimation is given as feedback to the multimedia application which increases or decreases its CPU access rate accordingly. This method can benefit by low-level network feedback. For example, the multimedia application can use the available bandwidth indicated in network layer feedback and change its encoding rate and also change its CPU access rate.

Related work

In this section, we present a summary of some related work. Most of these works are related to supporting multimedia, though not directly dealing with the problem of adapting multimedia streaming to changing network conditions. When appropriate, we identify if the work could be used for achieving adaptation of multimedia streaming application.

- *MPEG-2 error resilience experiments*: A study of the error resilience of MPEG-2 encoded video streams is given in [48]. Here, the error resilience MPEG-2 system layer is studied whereas the previous studies only concentrate on video layer. In the MPEG-2 system layer, several program streams might be combined to form a transport stream. The transport stream is multiplexed over the network, in this case the ATM network. The study shows that packet loss rates almost doubles when the impact of system layer is included in some error rates. This study will be useful in designing and studying adaptive error control scheme for streaming multimedia transmitting MPEG-2 encoded video.
- *Fast buffer fill up scheme*: A fast buffer fill up scheme for video-on-demand application running over ATM ABR service is presented in [49]. The MCR to be used for the application is based on a estimated value that depends on the GoP (group of pictures) used in the MPEG-2 encoding. To obtain a fast buffer fill-up during mode changes, such as fast forward and rewind, a high PCR (peak cell rate) is used. The PCR/MCR value is found from the expected value of the ACR (allowed cell rate).
- *Indeo video product, Intel*: The Intel's Indeo video [50] product provides support for progressively downloading video clips. The video file is stored in a hierarchical manner. The video is retrieved at lower frame rate using lower resolution quickly. If the user decides to continue, the frame rate and quality is increased as the download progresses. This supports users who have lower bandwidth to retrieve the video and download it based on what they see in the initial part of the video clip. The file encoding format provides the flexibility of specifying which frames of the video are key frames. The video producers can specify the scene-change frames as key frames.
- *Adaptive transport protocol for multimedia*: A new transport protocol HPF, that supports multiple interleaved reliable and unreliable streams is presented in [51]. Currently, the Internet does not support heterogeneous flows. Multimedia applications have to get the different media as different streams and then synchronize them. The HPF protocol supports interleaved flows which is required by multimedia applications. The congestion control and reliability mechanism (which are integrated in TCP) of the transport are decoupled. This allows support for interleaved reliable and unreliable streams. Priorities are used within substreams as indicated by the applications-defined hints. This enables the scheduler to drop low priority packets in the event of congestion. An adaptive multimedia application can use the HPF protocol.
- *Resilient multicast*: IP multicast is a popular delivery mechanism of streaming media of conference applications. Contrary to the belief that continuous media cannot do recovery of lost packets, one can show that there is a tradeoff between desired playback quality and degree of interactivity [52]. A new model of multicast called *resilient multicast* is proposed. In this scheme each client can decide its own tradeoff between reliability and real-time requirements. A resilient multicast protocol STORM (Structure-Oriented Resilient Multicast) is designed, in which groups self-organize themselves into distribution structures and use the structure information to recover lost packets from adjacent nodes. The distribution structure is dynamic and a lightweight adaptation algorithm is used to adapt to changing networking conditions. This scheme can be used to design and implement adaptive multicast streaming applications.
- *Scalable and adaptable video-on-demand*: The design and implementation of a scalable and adaptable video-on-demand server is discussed in [53]. In this system the movies are striped across multiple disk drives to boost I/O, reduce startup latency and prevent hot spots. Variable sized striping is used since MPEG encoding has variable-sized frames. The video stream is preprocessed and a RTP-like header which contains necessary information for the client is added. RTSP is used in the connection setup phase and RSVP is planned to be used for reserving resources along the path. An initial testbed to evaluate the system has been built and experiments demonstrate that the system reduces startup latency, buffering requirements and maximizes the number of concurrent users.
- *Efficient user-space protocols with QoS guarantees*: Multimedia applications running on traditional OSs have limitations due to overhead of data movement and inflexible CPU scheduling. A discussion of protocol implementation in user-space using real-time up-calls (RTU) which can provide QoS guarantees is given in [54]. The RTU mechanism has minimal overhead for concurrency control and context switching compared to thread-based methods. Efficient data movement techniques such as batching of I/O operations to reduce context switches, direct movement of data from applications

to network adapter, and header-splitting at receiver to maintain page alignment are augmented with the RTU mechanism. Experiments on the RTU implementation show that QoS can be guaranteed even in the presence of other competing best effort applications. This framework is amenable to support adaptive multimedia applications since RTU protocols are implemented in the user-space.

- *Rate-adaptive packet video*: A framework for sharing available bandwidth to transport rate-adaptive video is presented in [55]. The framework uses an ABR-like flow control and switch algorithm to give feedback to the rate-adaptive sources about the currently available bandwidth. The switch scheme used decouples the rate used in the flow control and the actual rate of the source. The source has the flexibility to adjust its rate only at certain time intervals which is controlled by a parameter. The work also discusses how to renegotiate the MCR parameter and use weight functions to reflect the changing requirements of video applications.
- *Video staging using proxies*: In LAN environments, large bandwidth is available due to high speed technologies (Gigabit Ethernet, ATM), but bandwidth is a scarce resource in a WAN environment. A proxy server can be used in LAN to overcome the resource restrictions when retrieving multimedia streams over WAN [56] Using a proxy server reduces the backbone bandwidth requirement. The paper also discuss how smoothing techniques can be used at the proxy server when clients have playout buffers.
- *Prefix proxy caching*: In this work the authors propose a prefix caching technique to mitigate effects of resource variability and startup latency [57]. In this technique a proxy server stores the initial frames of a long movie clip. When the client requests the stream in future, the proxy server starts sending the initial frames. In addition, the proxy server also contacts the video server and requests the transmission of later frames. The proxy also performs *workahead smoothing* into the clients playout buffer.

Summary

Only best-effort service is widely supported in the current Internet. The Internet is also a heterogeneous network and is expected to remain heterogeneous. Lots of efforts are being made by standardization bodies (IETF, ATM Forum, ITU-T) to support QoS in the Internet. Even with QoS support from the network the multimedia applications need to be adaptive.

Adaptation to changing network conditions can be achieved at several layers of the network protocol stack. In this paper, we surveyed several techniques for achieving adaptation at the application layer. A summary of these are as follows:

- Compression methods: An overview of DCT and wavelet compression methods was given. Aspects of these methods that can be used for adaptation were discussed.
- Application Streaming: The application methods are broadly classified into *reactive* and *passive* methods. Reactive methods are those where the application changes its behavior based on the environment. Passive methods aim to reduce the network burden by optimization techniques.
- Rate Shaping: Video encoder parameters (e.g., frame rate, quantization level) are changed to meet the changing network conditions.
- Error Control: These techniques use FEC-based methods to provide protection against changing error conditions.
- Adaptive Synchronization: An adaptive synchronization method for solving intermedia and intramedia synchronization problem was discussed.
- Smoothing: Smoothing techniques attempt to reduce the variability in the resource requirements of multimedia applications.
- Example Adaptive Applications: Techniques used for adaptation in real-world multimedia applications such as Real Networks' products (RealPlayer and RealServer) and Vosaic were presented.
- Operating System Support: Multimedia applications need periodic access of CPU and other system resources. Operating systems need to support such needs. Techniques to achieve this include real-time upcall, adaptive scheduling, and CPU management.

For each of these techniques, we discussed whether it could benefit from low-level network feedbacks. When appropriate, we discussed how the low-level feedback can be used for enhancing the adaptation technique.

References

1. Schulzrinne, H., Rao, A. & Lanphier, R. *Real Time Streaming Protocol (RTSP)*. RFC 2326, April 1998.

2. Schulzrinne, H., Casner, S., Frederick, R. & Jacobson, V. *RTP: A Transport Protocol for real-time applications. audio-video transport working group*. RFC 1889, Sept 1987.
3. Fowler, J.E. Adkins, K.C. Bibyk, S.B. & Ahalt, S.C. (1995) Real-Time Video Compression Using Differential Vector Quantization. *IEEE Transactions on Circuits and Systems for Video Technology* **5**: 14–24.
4. Fowler, J.E. & Ahalt, S.C. (1997) Adaptive vector quantization of image sequences using generalized threshold replenishment, In *Proc. of 1997 IEEE ICASSP*, pp. 3085–3088, April 1997.
5. ISO/IEC 13818-2. *Generic coding of moving pictures and associated audio information*. Technical report, MPEG (Moving Pictures Expert Group), International Organization for Standardization, 1994.
6. Duffield, N.G., Ramakrishnan, K.K. & Reibman, A.R. (1998) SAVE: An algorithm for smoothed adaptive video over explicit rate networks. In *Proc of IEEE INFOCOM*, April 1998.
7. Bolot, J.C. & Turletti, T. (1994) A rate control mechanism for packet video in the internet. In *Proc. of IEEE INFOCOM*, November 1994.
8. Eleftheriadis, A. & Anastassiou, D. (1994) Optimal Data Portioning of MPEG-2 Coded Video. In *First IEEE Int'l Conf. on Image Processing*, November 1994.
9. Pancha, P. & Zarki, M. (1992) Prioritized Transmission of Variable Bite Rate MPEG Video. *IEEE GLOBECOM*, pp. 1135–1138, December 1992.
10. Pancha, P. & Zarki, M. (1993). Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks. *IEEE Trans. on Circuits and Systems for Video Technology* **3**: 190–198.
11. Tham, J., Ranganath, S. & Kassim, A. (1996). Highly scalable wavelet-based video codec for very low bit-rate environment. *Journal of Selected Areas of Communications-Very Low Bit Rate Coding*.
12. Podilchuk, C.I., Jayant, N.S. & Farvardin, N. (1995). Three-dimensional subband coding of video. *IEEE Transactions on Image Processing* **4**: 125–139.
13. Taubman, D. & Zakhor, A. (1996) A Common framework for rate and distortion based scaling of highly scalable compressed video. *IEEE Transactions on Circuits and Systems for Video Technology* **6**: 329–354.
14. Cheng, P., Li, J. & Kuo, C.-C.J. (1997) Rate control for an embedded wavelet video coder. *IEEE Transactions on Circuits and Systems for Video Technology* **7**: 696–702.
15. Vickers, B., Albuquerque, C. & Suda, T. (1998) Adaptive Multicast of Multi-Layered Video: Rate-Based and Credit-Based Approaches. In *Proc. of IEEE INFOCOM, San Francisco, 1998*.
16. McCanne, S., Jacobson, V. & Vetterli, M. (1996) Receiver-driven layered multicast. In *ACM SIGCOMM, Stanford, CA, August 1996*.
17. Li, X., Paul, S., Pancha, P. & Ammar, M.H. (1997) Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery. In *Proc. NOSSDAV'97, May 1997*.
18. Li, X., Paul, S. & Ammar, M.H. (1999) Multi-Session Rate Control for Layered Video Multicast. In *Proc. IS&T/SPIE Multimedia Computing and Networking, Janurary 1999*.
19. Bolot, J.C. & Garcia, A.V. (1996) Control mechanisms for packet audio in the internet. In *Proc. of IEEE INFOCOM, November 1996*.
20. Nee, P., Jeffay, K. & Danneels, C. (1997) The performance of two-dimensional media scaling for Internet videoconferencing. *In Proc. of NOSSDAV, May 1997*.
21. Zeng, W. & Liu, B. (1996) Rate Shaping by block dropping for transmission of mpeg-precoded video over channels of dynamic bandwidth. In *ACM Multimedia*, 1996.
22. Ramanathan, S., Rangan, P.V., Vin, H.M. & Kumar, S.S. (1995). Enforcing application-level QoS by frame-induced packet discarding in video communications. *J. of Computer Communications* **18**: 742–754.
23. Bolot, J., Fosse-Parisis, S. & Towsley, D. (1999) Adaptive FEC-Based Error Control for Interactive Audio in the Internet. In *Proc. of IEEE INFOCOM, March 1999*.
24. Podolsky, M., Romer, C. & McCanne, S. (1998) Simulation of FEC-based error control for packet audio on the Internet. In *Proc. of IEEE INFOCOM, April 1998*.
25. Bolot, J.-C, & Turletti, T. (1998) Experience with rate control mechanisms for packet video in the Internet. *Computer Communications Review* **28**(1).
26. Liu, C., Xie, Y., Lee, M.J. & Saadawi, T.N. (1998) Multipoint multimedia teleconference system with adaptive synchronization. *IEEE JSAC* **14**: 1422–1435.
27. McManus, J.M. & Ross, K.W. (1996) Video-on-Demand over ATM: constant-rate transmission and transport. *IEEE JSAC* **1**: 1087–1098
28. Lakshman, T.V., Ortega, A. & Reibman, A.R. (1998) Variable bit rate (VBR) video: Tradeoffs and potentials. *Proceedings of the IEEE 36, May 1998*.
29. Lam, S.S., Chow, S. & Yau, D.K. (1994) An alogrithm for lossless smoothing of MPEG video. In *Proc. ACM SIGCOMM, September 1994*, pp. 281–293.
30. Feng, W., Jahanian, F. & Sechrest, S. (1997) An optimal bandwidth allocation strategy for the delivery of compressed prerecorded video. *ACM/Springer-Verlag Multimedia Systems Journal*.
31. Feng, W. & Sechrest, S. (1995) Critical bandwidth allocation for delivery of compressed video. *Computer Communications* **18**: 709–717.
32. Feng, W. (1997) Rate-constrained bandwidth smoothing for the delivery of stored video. In *Proc. of SPIE Multimedia Networking and Computing, November 1997*, pp. 316–327.
33. Feng, W. (1997) Time constrained bandwidth smoothing for interactive video-on-demand. In *Proc. of ICCC, November 1997*, 291–302.
34. Salehi, J.D., Zhang, Z.-L., Kurose, J.F. & Towsley, D. (1996) Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. In *ACM SIGMETRICS, May 1996*, pp. 221–231.
35. Feng, W. & Rexford, J. (1997) A comparison of bandwidth smoothing techniques for the transmission of prerecorded compressed video. In *Proc. of IEEE INFOCOM, April 1997*, pp. 58–66.
36. Rexford, J., Sen, S., Dey, J., Feng, W., Kurose, J., Stankovic, J. & Towsley, D. (1997) Online smoothing of live, variable-bit-rate video. In *Proc. of NOSSDAV. May 1997*, pp. 249–258.

37. Feng, W., Krishnaswami, B. & Prabhudev, A. (1998) Proactive buffer management for the delivery of stored video across best-effort networks. In *ACM Multimedia Conference*, *September 1998*.
38. Feng, W., Liu, M., Krishnaswami, B. & Prabhudev, A. (1999) A priority-based technique for the delivery of stored video across best-effort networks. In *Proc. IS&T/SPIE Multimedia Computing and Networking, January 1999*.
39. Zhang, Z., Nelakuditi, Z., Aggarwal, R. & Tsang, R.P. (1999) Efficient server selective frame discard algorithms for stored video delivery over resource constrained networks. In *Proc. of IEEE INFOCOM*, *March 1999*.
40. Patel, K. & Rowe, L. (1997) Design and performance of the berkeley continuous media toolkit. In *Multimedia Computing and Networking 1997*, *Proc. SPIE 3020*, pages 194–2–6.
41. Walpole, J., Koster, R., Cen, S., Cowan, C., Maier, D., McNamee, D., Pu, C. Steere, D. & Yu, L. (1997) A Player for Adaptive MPEG Video Streaming Over The Internet. In *Proc. 26th Applied Imagery Pattern Recognition Workshop AIPR-97*, *SPIE*, October 1997, pp. 249–258.
42. Tennenhouse, D. et al. (1995) The ViewStation: a software-intensive approach to media processing and distribution. *Multimedia Systems* **3**: 104–115.
43. *Real Networks*. Realvideo technical white paper. http://www.real.com/devzone/library/whitepapers/overview.html/.
44. *Real Networks*. Surestream technical white paper. http://www.real.com/devzone/library/whitepapers/surestrm.html/.
45. Chen, Z., Tan, S., Campbell, R. & Li, Y. (1995) *Real Time Video and Audio in the World Wide Web*. In WWW4 (Available at: http://www.vosaic.com/).
46. Lakshman, K., Yavatkar, R. & Finkel, R. (1998) Integrated CPU and network-I/O QoS management in an end system. *Computer Communications* **21**: 325–333.
47. Yau, D.K.Y. & Lam, S.S. (1997) Adaptive rate-controlled scheduling for multimedia applications. *IEEE/ACM Transactions on Networking* **5**: 475–487.
48. Frater, M.R., Arnold J.F. & Zhang, J. (1999) MPEG-2 video error resilience experiments: The importance of considering the impact of the systems layer. *Signal Processing: Image Communication* **14**: 269–275.
49. Zheng, B. & Atiquzzaman, M. (1998) Multimedia over ATM: progress, status and future. In *Proc. of ICC*, *1998*.
50. Intel. *Indeo Video Product*. http://developer.intel.com/ial/indeo/.
51. Dwyer, D., Ha, S., Li, J. & Bharghavan, V. (1998) An adaptive transport protocol for multimedia communication. In *IEEE Conference on Multimedia Computing Systems*, *1998*.
52. Xu, X.R., Myers, A.C., Zhang, H. & Yavatkar, R. (1997) Resilient multicast support for continuous-media applications. In *Proc. of NOSSDAV'97*, *1997*.
53. Berzsenyi, M., Vajk, I. & Zhang, H. (1998) Design and implementation of a video on-demand system. *Computer Networks and ISDN Systems* **30**: 1467–1473.
54. Gopalakrishnan, R. & Parulkar, G. (1998) Efficient user-space protocol implementations with qos guarantees using real-time upcalls. *IEEE/ACM Trans. on Networking* **6**: 374–388.
55. Hou, Y., Panwar, S., Zhang, Z., Tzneg, H. & Zhang, Y. (1998) On network bandwidth sharing for transporting rate-adaptive packet video using feedback. In *Proceedings of Globecom, November 1998*.
56. Wang, Y., Zhang, Z., Du, D. & Su, D. (1998) A network conscious approach to end-to-end video delivery over wide area networks using proxy servers. *IEEE/ACM Trans. no Networking* (submitted).
57. Sen, S., Rexford, J. & Towsley, D. (1999) Proxy prefix caching for multimedia streams. In *Proc. of IEEE INFOCOM*, *March 1999*.