# Adaptive Video Multicast over the Internet

**Jiangchuan Liu and Bo Li**
*Hong Kong University of Science and Technology*

**Ya-Qin Zhang**
*Microsoft Research, Asia*

**This article surveys existing solutions to adaptive video multicast, providing a taxonomy of solutions according to several distinct features. We also review the principal techniques from both video coding and network transport perspectives. Finally, we discuss the evaluation methodologies and metrics for adaptive video multicast systems.**

Multicast extends the traditional unicast communications with efficient multipoint communications in which data can be sent to a set of destinations simultaneously. This extension to the Internet, often called *IP multicast,* has been supported in many commercial routers and is undergoing incremental deployment in the Internet. Given the rapid development and deployment of multimedia applications and the multireceiver nature of video programs, real-time video distribution has emerged as one of the most important IP multicast applications. It's also an essential component of many current and emerging Internet applications, such as videoconferencing and distance learning.

Because network conditions are dynamic and there's no quality of service (QoS) guarantee in the current best-effort Internet, bandwidth adaptability becomes an essential requirement for video distribution over the Internet. Generally, a nonadaptive stream suffers two deficiencies. First, it can either underuse the available bandwidth or cause congestion collapse, which eventually degrades the video playback quality. Second, it can potentially lead to unfair bandwidth allocation, as the dominant traffic in the Internet (such as transmission-control protocol, or TCP, traffic) is adaptive. To solve such problems, one possible solution is to reengineer the network to provide necessary QoS support via resource reservation or service differentiation. Nevertheless, the current state of QoS deployment is only in an experimental stage and not yet ready for mainstream use. To be compatible with the near-term architecture and capability of the Internet, real-time video multicast applications have to adapt to the dynamic network conditions but still offer reasonable playback quality to the receivers.

There are, however, many challenges for adaptive video multicast over the Internet. First, the size of a multicast session varies from tens of receivers for videoconferencing to thousands of receivers for Web TV. The IP multicast model is reasonably scalable in terms of control overheads, but it provides only a basic underlying framework for group communications. Still, we must tackle the problem of scalability in designing high-level adaptation mechanisms. Second, it's important to consider that the Internet is a heterogeneous network. For example, receivers could have different capacities and processing capabilities even in the same session. An adaptation algorithm should thus maximize the total system throughput and fairly distribute video data to each receiver commensurate with its individual demand.

A lot of solutions to adaptive video multicast have been proposed in recent years,[1–19] addressing various issues and challenges. In this article, we present a comprehensive survey on this active research, first classifying the representative approaches according to the video rates delivered to the receivers in a session, as well as the places for bandwidth adaptation. We then provide detailed discussions on each approach. Our investigation covers both video coding and network transport techniques, with an emphasis on their interactions. Finally, we consider the evaluation methodologies and metrics for adaptive video multicast systems.

## Adaptation approaches for video multicast

We use two distinct properties to classify existing video multicast approaches:

- The video rates delivered to the receivers in a session. Existing approaches generally fall into two categories—single rate and multirate.
- The place where adaptation is performed. It's either at end systems (*end-to-end service*) or at intermediate network nodes (*active service*).

According to these properties, we classify the existing approaches as *single-rate adaptation* (sin-



1070-986X/03/$17.00 © 2003 IEEE

Published by the IEEE Computer Society

gle-rate, end-to-end), *simulcast* (multirate, end-to-end), *layered adaptation* (multirate, end-to-end), and *agent-based adaptation* (multirate, active). We'll give more detailed descriptions for each in the next few sections.

### Single-rate adaptation

Single-rate adaptation is a direct extension to the traditional feedback-based unicast adaptation. In this approach, the sender maintains one video stream and adjusts the rate based on the receivers' states. To achieve high scalability, generally the receivers detect the states. A receiver can also use metrics that suit its local network technologies, as the heterogeneity of today's Internet precludes any attempt to use a single metric to determine the state.

To reflect the dynamic network conditions, the state information should be sent back to the sender on time. With unicast, the time scale of feedback delay is close to one or several round-trip times (RTTs). With multicast, since the source needs to know the global state of all the receivers, the convergence time could be much longer if the receivers aren't synchronized. On the other hand, if they're synchronized for feedback, the root link in the multicast tree will become congested and the sender will be overwhelmed with the responses, resulting in a feedback implosion problem.[1] Therefore, we need a scalable feedback mechanism for such a system.

Early work on scalable feedback mainly focuses on its use for packet retransmission in reliable data multicast protocols. However, this isn't the principal objective for video multicast. The INRIA Videoconference System (ivs),[1] one of the pioneer video multicast systems, uses a probing mechanism to solicit feedback information in a scalable manner. To aid the design of a general algorithm, three generic notions for network states are in this system—unloaded, loaded, and congested.

With these notions, a receiver can decide its states according to the packet loss rate—for example, a persistent 5 percent or more loss rate indicates congested. When the source wishes to solicit state information from its receivers, it sends out a randomly generated 16-bit key and a number indicating how many digits of the key are significant. Each receiver also generates a key and responds to the solicitation only if the keys and the states match. The number of significant digits starts from 16 and decreases after each epoch. For a typical worst-case RTT of 500 ms, it requires at most 16 seconds to finish a cycle of probing. The source can then estimate the state distribution based on an analysis of the matching probability. Specifically, a logarithmic relationship exists between the number of replies and the number of receivers. Hence, the algorithm can scale to very large sessions.

**Although incapable of solving the problem of heterogeneity, single-rate multicast can serve as a building block for advanced video multicast protocols, such as the simulcast protocols.**

The real-time and real-time control protocols (RTP/RTCP)[20] aim to support real-time media distribution of groups of any size; therefore, they also provide a scalable state reporting method. When the size of a session grows, each receiver reduces the frequency of its RTCP report accordingly, so that the total bandwidth for the reports remains constant (let's say 5 percent of the session bandwidth). However, RTP/RTCP doesn't address the conversion of different reports into a global measure for rate control. In ivs, we can do this by comparing the fraction of congested and unloaded receivers (say 30 percent of the receivers being congested implies the sender should reduce the video rate). Nevertheless, in ivs and many other single-rate multicast systems, the choice of such parameters is ad hoc because no perfect target rate exists in the context of heterogeneous multicast. As a result, the receivers in a session aren't fairly treated. Low-capacity receivers suffer congestion while high-capacity receivers have their bandwidths underutilized. Although incapable of solving the problem of heterogeneity, single-rate multicast can serve as a building block for advanced video multicast protocols, such as the simulcast protocols.

### Simulcast

Single-rate multicast is an extreme toward multiuser support for video distribution. The

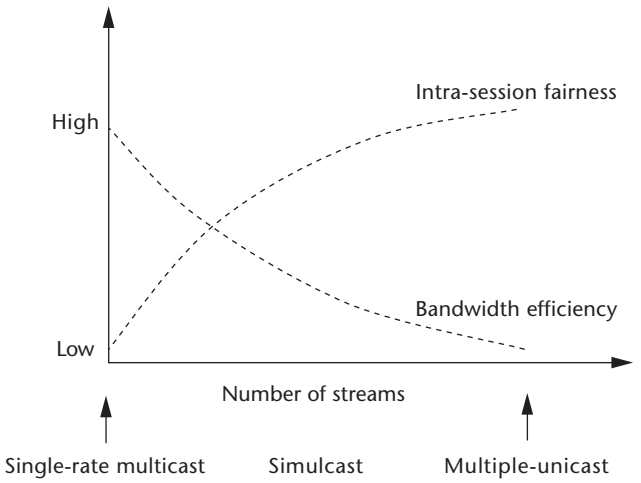



***Figure 1. A comparison of single-rate multicast, simulcast, and multiple unicast.***

opposite is the traditional unicast delivery where each user has a connection to the sender. Their differences lie on the bandwidth efficiency and control granularity. If a video program is distributed using individual feedback-controlled connections, the bandwidth efficiency is low but we can achieve ideal fairness as each receiver receives a customized copy. In contrast, using a single multicast stream has good bandwidth economy, but very low granularity in terms of rate control.

Simulcast is indeed a trade-off between these two extremes, as Figure 1 shows. Evidently this is well justified in a typical multicast environment, where the receivers' bandwidths in a session usually follow some clustered distribution. Note that they either use standard access interfaces—for example, a 128-Kbps Integrated Services Digital Network (ISDN), a 1.5-Mbps asymmetric digital subscriber line (ADSL), or a 10-Mbps switched Ethernet—or they might share some bottleneck links and hence experience the same bottleneck bandwidth. Therefore, a simulcast protocol can use a limited number of streams to match these clusters without introducing high redundancy.

A representative simulcast protocol is the destination set grouping (DSG) protocol.[2,21] In DSG, the source maintains three streams carrying low-, medium-, and high-quality versions of the original video. A receiver subscribes to a stream that best matches its bandwidth. Then DSG uses an intrastream protocol for the sender to adjust the stream rate within a prescribed range. This protocol is a variation of the one used in ivs. In addition to this sender-based adaptation, DSG also supports receiver-based adaptation through an interstream protocol. It lets a receiver change to another stream when the current one can't satisfy its requirement. For example, suppose the current stream has operated at its low end. If the receiver is still congested, it can move to a lower-rate stream.

Experimental studies on simulcast with real video codecs by Lameillieure and Pallavicini[22] demonstrate the feasibility of real-time video simulcast. Actually, simulcast has already been advocated in many commercial video-streaming systems. For example, RealNetworks' RealSystem G2 supports simulcast under the name of SureStream, which generates a fixed number of streams at prescribed rates, and a receiver can dynamically choose a stream commensurate with its bandwidth.

### Layered adaptation

Bandwidth redundancy caused by replication remains a drawback of simulcast. Li and Ammar[3] propose some solutions to this problem. The idea is to infer the receivers' capacities together with their locations through an end-to-end probing algorithm. We can then use heuristics to limit the total bandwidth of all the streams generated by the sender as well as the aggregate bandwidth in a local area.

A more attractive solution involves information decomposition.[5] A commonly used decomposition scheme is *cumulative layering*, in which a raw video sequence is compressed into some nonoverlapped streams, or layers. There's a *base layer*, which contains the data representing the most important features of the video. Additional layers, called *enhancement layers*, contain data that progressively refine the reconstructed video quality.[5] Consequently, different subsets of layers can serve the receiver with heterogeneous capacities or capabilities.

For layered adaptation, one straightforward solution is network-driven, or so-called *prioritized transmission*.[8,13] In this scheme, we assign packets from the base layer's highest priority, whereas we assign packets from enhancement layers progressively lower priorities. When there isn't enough bandwidth, routers discard the lowest-priority packets, thereby preventing loss of the base layer or high-priority enhancement layers. Nevertheless, it requires routers to implement some priority-based packet scheduling policy, which is considerably more complex than existing first in, first out (FIFO) drop-tail policies, and this hasn't been widely supported by the current Internet.

McCanne et al.[5] proposed the first practical

adaptation protocol for layered video multicast over the best-effort Internet. This protocol, known as receiver-driven layered multicast (RLM), is a pure end-to-end adaptation protocol and requires a FIFO drop-tail router only. A RLM sender transmits each video layer over a separate multicast group. It predetermines the number of layers as well as their rates. It performs adaptation only at the receiver's end by a *probing-based scheme*, where a receiver periodically joins a higher layer's group to explore the available bandwidth. If packet loss exceeds some threshold after the *join experiment*—that is, when congestion occurs—the receiver should leave the group. Otherwise it will stay at the new subscription level.

One drawback of this probing-based scheme is that one receiver's join experiments can introduce packet losses experienced by other receivers that share the same bottleneck link, and such losses would occur frequently if all the receivers perform join experiments independently. The RLM incorporates a shared learning mechanism to solve this problem. With shared learning, other receivers infer the failure of a join experiment conducted by a receiver, thus avoiding separate disruptive join experiments. However, it reduces the scalability of RLM and significantly increases the convergence time.

The difficulties associated with coordinating join and leave attempts motivated Vicisano, Rizzo, and Crowcorft to propose their receiver-driven layered congestion control (RLC) protocol.[7] Their scheme calls for synchronized join experiments, where the sender temporarily increases the sending rate on a layer, and a receiver will join a higher layer only if there's no packet loss during this experiment. Because no coordination exists among the receivers, its convergence time can be much shorter than that of RLM.

It's well known that the original RLM doesn't ensure intersession fairness,[10] but it also isn't friendly to TCP traffic because its probing strategy is very aggressive.[11] The RLC uses receiver-driven join/leave actions to mimic the behavior of TCP congestion control by a careful choice of the join timer and the rate of each layer—say, twice as much as the previous layer. Experimental results demonstrate that, under idealized conditions, RLC achieves good intersession fairness and TCP friendliness.

Nevertheless, the objective of TCP differs significantly from that of video transmission protocols. Whether a video flow should fairly share the bandwidth with TCP traffic on both short and long-time scales is still in debate. For example, although RLC interacts better with TCP, it could experience the same sawtooth behavior of TCP flows. Such short-term bandwidth oscillations are undesirable for video transmission.

**Rather than mimic the behavior of TCP, a more reasonable objective for video streaming would be achieving a long-term fair share with TCP traffic.**

In addition, it's impossible for a layered video stream to be totally fair to TCP flows because its adaptation granularity on the receiver's side is at a layer level.[16] Therefore, rather than mimic the behavior of TCP, a more reasonable objective for video streaming would be achieving a long-term fair share with TCP traffic. This is a relatively loose notion for TCP-friendliness and has been widely cited in existing protocols for streaming applications.[12] In practice, we can achieve this by model-based adaptation, where each receiver uses a model to estimate the equivalent bandwidth for a TCP connection running over the same path and performs joining and leaving actions according to this estimated bandwidth. There has been significant research on modeling TCP throughput. A general conclusion is that such a model relies on the packet size, loss event rate, and round-trip time (RTT). The estimations of some parameters, such as RTT between the sender and a receiver, require feedback packets. Therefore, in a heterogeneous multicast environment, how to efficiently estimate these parameters for a large number of sender–receiver pairs becomes a challenging issue. Researchers have developed some smart lightweight feedback loops to solve this problem.[12]

### Agent-based adaptation

The multirate adaptation approaches, however, all rely on the end nodes (the sender or receivers) playing active roles, whereas the nodes inside the network do nothing but packet scheduling and forwarding in the network layer. We

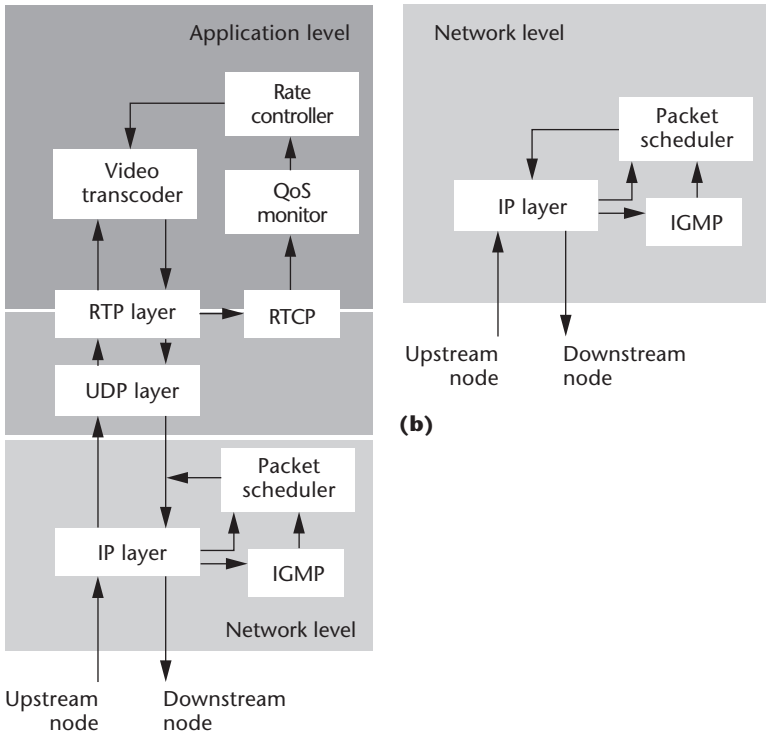



***Figure 2. Two types of network nodes: (a) a node with application-level active services, and (b) a node with network-level services only.***

base this on the *end-to-end argument*,[23] which is a set of architectural principles that guide the placement of functions within a network. Such principles are often construed to preclude the implementation of any kind of higher-level function within a network. However, the argument for active services[19] is that we can best support or enhance many applications using information or intelligent services only available inside a network. For example, we can deploy several agents in a large-scale network; the agents partition the network into several confined regions, and each agent can thus handle the requirements more easily from its local region.[19]

In Figure 2a, we present a generic diagram of a network node that provides active services. Compared to a traditional network node, shown in Figure 2b, an active node (such as a video gateway or agent) offers a richer set of services for video transmission at the application level. For example, the gateway or agent can manipulate the data and control information of the video streams and decompose an original stream into multiple streams with different bandwidth or format requirements by transcoding. Such service has also been embodied in the design of RTP by providing an RTP-level mixer and translator.[20] A mixer is capable of mixing streams and producing new timing information. A translator can translate formats and semantics across transport protocols. For example, it can convert a multicast packet to a number of unicast packets for easy transferrence across a network with no multicast support. A translator may also perform transcoding, but unlike the mixer, it doesn't affect any synchronization information associated with the stream.

As we discuss in the next section, several methods exist for transcoding a video stream. Nevertheless, most of them involve complex computations. Current studies have shown that, for a small number of streams, transcoding doesn't significantly increase the end-to-end delay because we can perform it within the interframe display time.[24] However, for a high-speed link, such as a 155-Mbps link, a node can spend only 26 µs to receive, process, and forward a packet (assuming an average size of 500 bytes). This is clearly not long enough for any state-of-the-art transcoding operation if we need to filter a certain amount of traffic.

Besides transcoding, we must address a number of management issues in an active-service architecture. A key issue is the agents' topological placement. Numerous agent placement algorithms have been proposed in different contexts, such as Web proxying, data replicating in content delivery networks, and loss repairing for data multicast. Given a network topology, these algorithms basically place a set of agents to minimize or maximize some client-perceived measures, such as average bandwidth or latency.[25] We can use many of them for placing multicast video agents as well. In contrast to this static configuration approach, self-organized approaches[18] form groups out of colocated receivers with similar reception levels. A group representative is responsible for dynamically locating a suitable agent or receiver willing to provide customized transcoding for the group. Despite its high complexity, this dynamic scheme offers a much more flexible framework for agent deployment than the static configuration.

## Support for video coding techniques

Here we'd like to review the video coding support for the adaptation algorithms. In particular, let's focus on the techniques related to the major compression standards, including the MPEG and H.261/263 families.[26] From the considerations of compatibility and interoperatibility, these standards play the most important roles in building practical video transmission systems.

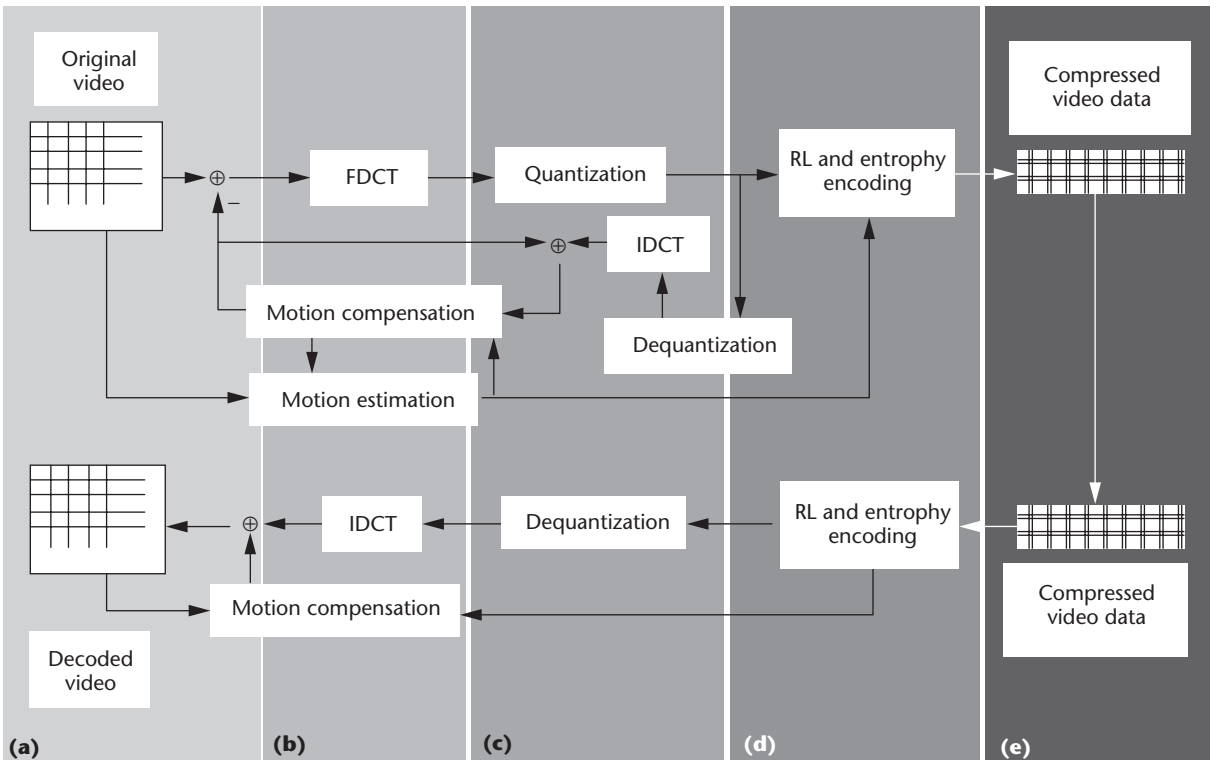



*Figure 3. A generic diagram for discrete cosine transform-based, motion-compensated coding.*

All these coding standards employ a similar sequential process, the DCT-based, motion-compensated hybrid coding.[26] It removes spatial correlation by block-based discrete cosine transform (DCT) and temporal correlation by motion-based predictive coding. As Figure 3 shows, we first divide a frame of a raw video into $8 \times 8$ blocks. A block serves as the minimum coding unit and four blocks constitute a macroblock. In intraframe coding (I-frame), transforms are applied to the original blocks, as Figure 3b shows. In interframe coding (P-frame), for each macroblock, we used a motion estimator to find a best-matching macroblock in a reference frame (a previous I or P frame). The two macroblocks are subtracted and their difference is then transformly coded based on $8 \times 8$ blocks. After this forward DCT stage, the DCT coefficients are quantized and encoded using a run-length encoder. Finally, we use an entropy encoder (or Huffman coder) to further compress the data and produce the bitstream with a specific syntax. On the decoder side, we can perform inverse operations to reconstruct the video frames and display them.

### Rate control for source coding

In the standards discussed here, the video stream generated by a source coder can be either variable bit rate (VBR) or constant bit rate (CBR). VBR video has some promising advantages, such as better quality, shorter delay, and lower average bandwidth. However, existing studies show that receiver-driven layered multicast exhibits significant and persistent instability with a VBR source.[10] This results in frequent layer shifts (such as one shift every 45 seconds), which is generally intolerable by end users. Sometimes this instability is severe enough to cause the relative allocation of sessions to flip-flop. In other words, the system never reaches a steady state. Simulation results also show that when the system size increases, the frequency of layer shifting doesn't reduce much, implying that multiplexing isn't an effective solution to this problem.[10]

Therefore, to achieve a stabilized transmission, a CBR source is often more desirable. In addition, because of its predictable traffic patterns, it makes resource allocation and network management easier. In existing video coding systems, two methods are usually used to achieve a desired yet constant bit rate. As Figure 4 shows (next page), a buffer first smoothes out the bit rate variations. Because of the delay constraint, the buffer size is limited and a buffer malfunc-

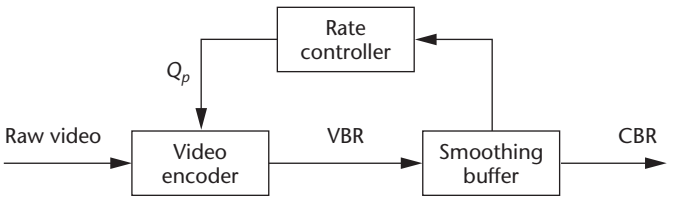



*Figure 4. Rate control for source coding. ($Q_p$ is the quantization scale.)*

tion, such as overflow or underflow, may occur if the buffer doesn't maintain the original video rate within a specified range. To ensure a normal operation point, we also control the expected bit rate at the compression stage. Meanwhile, we optimize the perceptual video quality. We can generally use the quantization scale as the control parameter to trade off these requirements.

### Transcoding

In practice, a set of independent encoders can replicate video streams for simulcast using different parameters for rate control. We can also obtain them through media scaling mechanisms at the postencoding stage—specifically, through transcoding.[19] A transcoder converts an existing video stream into a new stream with a different format or rate. It's also widely used in the active-service architecture.[17]

A straightforward approach for transcoding is to decompress the video stream, process it, and then recompress it. This strategy is called *spatial domain processing* because it's performed on the original pixels (see Figure 3a). We can apply a rich set of operations in this domain—for example, pixel downsampling and color reduction. However, the efficiency of spatial domain processing is relatively low because it involves computationally intensive procedures for fully decoding and encoding. Since state-of-the-art compression standards have similar processes, we can achieve fast transcoding by directly manipulating the compressed data in the frequency domain. In Figure 3c, the data size is the same as that of the raw video, but operations at this stage can avoid the computation for forward-DCT and inverse-DCT transforms. Examples include frequency filtering and quantization scale adjustment. After the run-length coding, the many zeros produced by the forward-DCT have been removed and the data are considerably smaller. Operations that are feasible here include color to monochrome conversion, data partitioning, frequency filtering, and simple codec conversions. For resolution downsampling, we can reuse the motion vectors in the original stream by interpolation. Thus, we can simplify the most time-consuming process in video coding (motion estimation) by transcoding. Operations on the fully compressed data (Figure 3e) are standard-specific and relatively simple, such as intelligent frame dropping.

### Scalable coding

Cumulative layered coding has also received great attention in the video coding community, where they frequently use the term *scalable coding*.[27] Generally, the perceptual quality of a frame-based video sequence depends on the frame rate, frame size (resolution), and frame quality. By scaling any of them, we can obtain temporal, spatial, and quality scalability, respectively.

Temporal scalability is the most common scalability tool adopted in a diverse range of video compression standards, including H.263 and the MPEG family.[26] In these standards, we achieve this scalability by using intraframe coding (I-frame), predictive coding (P-frame), and bidirectional predictive coding (B-frame). As Figure 5a shows, a P-frame depends on the previous I- or P-frame, and a B-frame depends on a previous and a subsequent I- or P-frame. Layers thus can be mapped to different frame types. Temporal scalability has also been used in many experimental studies for video multicast (such as Cheung et al.[2] and Vickers et al.[13]).

We can achieve spatial scalability by pixel subsampling, which scales the frame size for different layers (Figure 6b). Note that in an adaptive

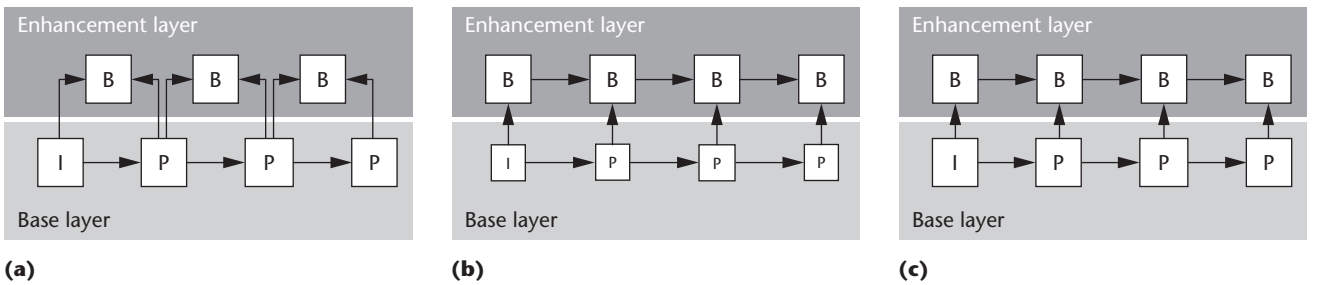



*Figure 5. Examples of frame-based scalabilities. (a) Temporal scalability, (b) spatial scalability, and (c) signal-to-noise ratio scalability.*

transmission, switching of frame size during a video playback may result in an extremely poor user experience. Therefore, even if a receiver obtains only a subsampled video at a certain time, it has to interpolate this video to the full terminal resolution for displaying. In this sense, we can view spatial scalability as a special case of quality scalability.

We can also refer to quality scalability as signal-to-noise ratio (SNR) scalability, since SNR often measures video quality.[26] Figure 5c shows a block diagram of a two-layer SNR scalable encoder. First, we code the input video at a low bit rate (using, for example, a coarse quantization scale) to generate the base layer bitstream. A second encoder then codes the difference between the original video and the base layer to generate the enhancement layer's bitstream.

The MPEG-2, MPEG-4, and H.263+ standards also support a combination of individual scalabilities to form hybrid scalability for certain applications. Spatial–temporal scalability is perhaps the most commonly used scheme. This hybrid scheme provides a 2D space for scaling. However, for a specific enhancement layer, we have only one choice, either to enhance the spatial resolution or to enhance the temporal resolution. The choice is obviously application-dependent. For example, in a soccer game broadcast, we'd like to first enhance the frame speed to render actions smoothly. On the other hand, to clearly display the quasistatic slides in remote learning, we prefer higher spatial resolution. Unfortunately, because the structure of a codec is generally fixed, a system can choose only one design policy, preventing it from satisfying conflicting user desires.

The fixed structures of these scalability tools also limit their adaptability for bandwidth heterogeneity and network congestion. For example, the typical bit rates for MPEG-1 temporal scalability are 256 Kbps for the I stream, 900 Kbps for the (I + P) stream, and 1.5 Mbps for the (I + P + B) stream, and we can only tune them in a limited range. To achieve flexible and fine-grained rate adaptation, the MPEG-4 group called for proposals in 1998 under the name of fine granularity scalability (FGS).[27] After several core experiments, the MPEG-4 group chose the bitplane coding of DCT residues because of its comparable efficiency and implementation simplicity. Bitplane coding uses embedded representations in compression. For illustration, there are 64 ($8 \times 8$) DCT coefficients for each video block. The most significant bits from the 64 DCT coefficients form bitplane 0, the second most significant bits form bitplane 1, and so forth. In the output stream, the bitplanes—not the coefficients—are placed sequentially. We thus generate layers through an assembling/packetization procedure, which can truncate the embedded stream at any specified position. This differs from the traditional scalable coders that use a fixed layering structure and perform rate control at the source coding stage. As a result, it can fine-tune the rate of the enhancement layer quickly. The original FGS only uses the based layer as the reference for motion compensation, which is inefficient when its bit rate is relatively low. An improvement on FGS, called Progressive FGS (PFGS),[28] addresses this issue by using a smart reference scheme. It not only uses the base layer but also some enhancement layers for motion compensation. It thus achieves better quality while retaining the flexibility of the original FGS.

MPEG-4 also introduces a content-based visual data representation,[26] which we consider an effective vehicle for user interactions for a variety of future multimedia applications. In MPEG-4 video,


*Figure 6. An example of the video object concept and content scalability in MPEG-4. (a) The scene consists of five objects, including two announcers, the caption, ballet stage, and background. (b) One object shown separately. (c) A user indicates that the background and ballet dancers are less interesting, and consequently some packets of these two objects are dropped to meet the bandwidth constraints. Without content scalability, packets are uniformly dropped across the whole rectangular frame, resulting in (d) poorer subjective quality.*

we view a scene as a composition of video objects with intrinsic properties such as shape, motion, and texture. We perform the encoding and decoding process on the instances (images) of video objects at each given time, which we call video object planes (VOPs). We can achieve object-based temporal scalability and quality with video object layers (VOLs), which represent either the base layer or enhancement layers of a VOP. Moreover, object-based representation yields another type of scalability, namely content or object scalability (see Figure 6). This information decomposition scheme differs from the traditional frame-based layering schemes because objects are independent or partially independent, and yet differs from stream replication because objects are nonoverlapping. Hence, it provides a more flexible and efficient framework for adaptive video dissemination. Nevertheless, to our knowledge few multicast systems have exploited the potential of such scalability.

## Evaluation procedures and metrics

Conducting objective evaluations of the merits of various adaptive video multicast techniques is a challenging task. There has been a significant amount of research on monitoring multicast sessions, such as Mtrace (see ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/) and Mantra.[29] However, experiments for adaptive video multicast over the Internet are still preliminary, usually with limited active members (less than 200), and relatively simple video coding schemes. On the other hand, analytical models are difficult to build for such complex systems. Therefore, simulation remains the most common tool for studying their performance. There are two fundamental issues we must address in simulation:

- the system's offered load—that is, the nature of the packets it transmits over the network; and
- its performance characteristics—that is, how the playback quality depends on the network services.

We characterize the former by source traffic models, and evaluate the latter by performance metrics for individual receivers as well as the overall system.

### Source traffic model

Researchers have developed numerous traffic models for the single-layer VBR video.[30] However, for evaluating multicast systems, a layered source model is of particular interest. The original simulation of RLM uses a set of CBR streams with fixed packet sizes to represent the layered traffic. Unfortunately, this simple model fails to reflect the nature of real video traffic—specifically, that the instantaneous traffic in each layer varies over time, and that there's a high correlation between the instantaneous traffic in each layer. Gopalakrishnan et al.[10] propose an abstract layered source model to capture these characteristics. The model relies on two parameters, $A$ and $P$, where $A$ is the average number of packets generated per interval, and $P$ characterizes the packet distribution in the interval. When $P = 1$, the model produces CBR-traffic. As $P$ increases, traffic becomes more bursty (VBR-like). Researchers have also used real video traces in simulations, such as Vickers et al.[13] Nevertheless, to our knowledge, complex source traffic models (like the Markovian model[30]) or video traces for advanced video codes (such as the MPEG-4 FGS coder[27]) have seldom been used in existing studies.

### Local metrics

We can use local metrics to characterize the service experienced by an individual receiver, and the local measures of some representative receivers can partially represent the performance of a multicast system. In principle, we can use any traditional metric for unicast applications as a local metric. From a networking perspective, typical metrics include bandwidth, loss rate, delay, and delay jitter. Clearly, it would be desirable to translate these low-level parameters into quality measures that can be more readily appreciated by end users. This has been addressed using pure mathematical measurements, such as peak-rignal-to-noise ratio (PSNR), or using evaluation tools for the human visual systems (HVS).[26] However, an accurate evaluation must consider all relevant factors, including the image resolution and frame rate, human spatial–temporal contrast sensitivity, and the impact of fluctuation. This remains a complex undertaking and requires further examination and integration.

### Global metrics

For a video multicast system with heterogeneous receivers, its performance not only depends on an aggregation of local metrics like the overall loss rate, but also on the stability of the system and the time scales over which events occur. Unfortunately, no standardized metrics have been

*Table 1. A summary of the adaptive video multicast approaches.*

| Approach | Adaptation Mechanism | Network Requirement | Coding Requirement |
|---|---|---|---|
| Single-rate adaptation[1] | Scalable feedback control | — | Rate control |
| Simulcast[2,3] | Scalable feedback control | — | Rate control<br>Transcoding |
| Layered multicast | | | |
| Network-driven*[13] | Priority dropping | Priority identification | Scalable coding |
| Receiver-driven*[5–7,12,13] | Joining/leaving groups | — | Scalable coding |
| Active service[17,19] | Transcoding in agents | Active service node | Transcoding |

* Sender-based adaptation is used in Sisalem and Wolisz,[12] Vickers et al.,[13] and Liu et al.[14] as well.

adopted in this area. Here we describe two global metrics commonly used in existing studies.

**Convergence time.** We can generally assume that a video multicast system will eventually reach an optimal operating point and maintain its state except for infrequent and brief excursions due to transient congestions. This optimal operating point relies on intrinsic system characteristics, such as the rate-control algorithm, session size, distribution of receivers' capacities, and distribution of receivers' locations. Hence, the convergence time is a justifiable metric to evaluate the stability and scalability of such a system. Examples using this metric include the performance evaluations of the RLM[5] and RLC[7] protocols.

**Fairness index.** A crucial and extensively studied analytical problem is how a multirate multicast impacts network fairness. Analytical results[9] suggest that, in this case, the max–min allocation is fairer than if we restrict the sessions to being single rate. Nevertheless, if a receiver can't partially subscribe to a layer, the rate it could receive is restricted in a discrete set. In this case, the max–min fair rate allocation might not exist. See Rubenstein et al.[9] and Sarkar et al.[16] for studies on this dilemma. A possible solution is to use an active service infrastructure that offers finer rate control through the application-level transcoding.

Quantitative evaluations of fairness rely on the normalized bandwidth, or *fairness index*. (Note that aggregate bandwidth isn't a well-defined metric for optimizing fairness for heterogeneous receivers. For example, algorithms using such a metric generally favor a receiver with broad bandwidth and meanwhile might sacrifice a number of receivers with relatively narrow bandwidth.) A representative intrasession fairness index is the interreceiver fairness function (IRF).[4] The fairness function of an individual receiver normalizes its actual received video rate by its expected bandwidth, and the global index of a session is the weighted average of the individual measures of all the receivers. Although originally defined for simulcast, we can extend IRF to layered multicast to access the performance loss incurred by a mismatch between the discrete set of possible subscription rates and the receivers' expected bandwidths.

## Conclusions

For simplicity, we summarize the existing adaptive video multicast techniques we've reviewed in Table 1. Note that we sorted them using the following features:

- adaptation mechanism,
- network requirement, and
- coding requirement.

Because of its simplicity, simulcast remains a promising approach to address user heterogeneity. It's supported in many commercial streaming systems. Meanwhile, layered multicast has seldom been used. Nevertheless, with the efforts on scalable coding algorithms and standards that are underway, undoubtedly more efficient layered multicast systems can be built in the near future.

Regarding the choice between the end-to-end service and the active service, as indicated in Saltzer et al.,[23] many trade-offs exist. For video applications, the scalability of transcoding agents remains a question. Moreover, the process of decompression and lossy recompression inherently accumulates quantization noise each time the cycle is invoked. This effect is most notice-

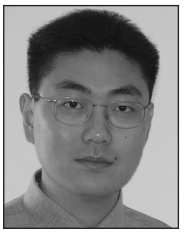
able when there's a concatenation of transcoding agents. To overcome these obstacles, the emerging MPEG-7 standard has defined *transcoding hints,* a set of metadata in video streams that effectively help a transcoding algorithm to meet speed and bandwidth requirements yet preserve high video quality.

Finally and most importantly, it's important to note that agent deployment is mainly subject to the management policies or market strategies of network operators and service providers. We've seen many commercial agent products available; for example, Cisco's Enterprise Content Delivery Network (Cisco E-CDN).[31] By deploying agents worldwide, a CDN enables multicast delivery of content-rich media from a corporate headquarters to low-bandwidth branch offices, so that it isn't necessary to rely on the multicast capability of the underlying network. Since Internet protocol multicast hasn't been widely deployed, the CDN provides a very attractive alternative that's readily available for the current Internet. We believe that such a solution remains dominant in recent years, although the more efficient Internet protocol multicast-based systems would eventually replace them. **MM**

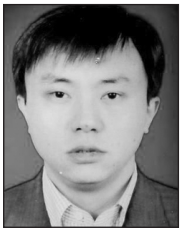
## References


1. J. Bolot, 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.
2. S. Cheung, 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.
3. X. Li and M. Ammar, "Bandwidth Control for Replicated-Stream Multicast Video," *Proc. High Performance Distributed Computing* (HPDC 96), IEEE CS Press, 1996.
4. T. Jiang, E. Zegura, and M. Ammar, "Inter-receiver Fair Multicast Communication over the Internet," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 99), 1999, pp. 103-114, http://www.nossdav.org.
5. S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-Driven Layered Multicast," *Proc. ACM Sigcomm Conf.*, ACM Press, 1996, pp. 117-130.
6. L. Wu, R. Sharma, and B. Smith, "Thinstreams: An Architecture for Multicast Layered Video," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 97), 1997, pp. 173-182, http://www.nossdav.org.
7. L. Vicisano, 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.
8. S. Bajaj, L. Breslau, and S. Shenker, "Uniform Versus Priority Dropping for Layered Video," *Proc. ACM Sigcomm Conf.*, ACM Press, Sept. 1998, pp. 131-143.
9. D. Rubenstein, J. Kurose, and D. Towsley, "The Impact of Multicast Layering on Network Fairness," *Proc. ACM Sigcomm Conf.*, ACM Press, 1999, pp. 27-38.
10. R. Gopalakrishnan et al., "Stability and Fairness Issues in Layered Multicast," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 99), 1999, http://www.nossdav.org.
11. A. Legout and E.W. Biersack, "Pathological Behaviors for RLM and RLC," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 00), 2000, http://www.nossdav.org.
12. D. Sisalem and A. Wolisz, "MLDA: A TCP-Friendly Congestion Control Framework for Heterogeneous Multicast Environments," *Proc. 8th Int'l Workshop on Quality of Service (IWQoS)*, IEEE Press, 2000, pp. 65-74.
13. B. Vickers, 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.
14. J. Liu, B. Li, and Y.-Q. Zhang, "A Hybrid Adaptation Protocol for TCP-Friendly Layered Multicast and Its Optimal Rate Allocation," *Proc. Ann. Joint Conf. IEEE Comptuer and Comm. Soc.* (Infocom 02), IEEE Press, 2002, pp. 1520-1530.
15. T. Kim and M. H. Ammar, "A Comparison of Layering and Stream Replication Video Multicast Schemes," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 01), 2001, pp. 63-72, http://www.nossdav.org.
16. S. Sarkar 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.
17. N. Yeadon et al., "Filters: QoS Support Mechanisms for Multipeer Communications, " *IEEE J. Selected Areas in Comm.*, vol. 14, no. 7, 1996, pp. 1245-1262.
18. I. Kouvelas, V. Hardman, and J. Crowcroft, "Network Adaptive Continuous-Media Applications Through Self Organised Transcoding," *Proc. Network and Operating System Support for Digital Audio and Video* (NOSSDAV 98), 1998, http://www.nossdav.org.

19. E. Amir, 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.
20. H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," *Request for Comments (RFC) 1889*, Internet Eng. Task Force (IETF), Jan. 1996.
21. X. Li, M. Ammar, and S. Paul, "Video Multicast over the Internet," *IEEE Network Magazine*, vol. 13, no. 2, Apr. 1999, pp. 46-60.
22. J.D. Lameillieure and S. Pallavicini, *A Comparative Study of Simulcast and Hierarchical Coding*, tech. report, European RACE project, subgroup WG2, Feb. 1996.
23. H. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Trans. Computing Systems*, vol. 2, no. 4, Nov. 1984, pp. 277-288.
24. J. Youn, J. Xin, and M.-T. Sun, "Fast Video Transcoding Architectures for Networked Multimedia Applications," *Proc. IEEE Int'l Symp. Circuits and Systems* (ISCAS 00), IEEE CS Press, 2000.
25. B. Li et al., "On the Optimal Placement of Web Proxies in the Internet," *Proc. Ann. Joint Conf. IEEE Comptuer and Comm. Soc.* (Infocom 99), IEEE Press, 1999, pp. 1282-1290.
26. Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.
27. W. Li, "Overview of the Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, Mar. 2001, pp. 301-317.
28. S. Li, F. Wu, and Y.-Q. Zhang, "Experimental Results with Progressive Fine Granularity Scalable (PFGS) Coding," *ISO/IEC JTC1/SC29/WG11, MPEG99/m5742,* Int'l Organization for Standardization, Mar. 2000.
29. P. Rajvaidya, K. Almeroth and K. Claffy, *A Scalable Architecture for Monitoring and Visualizating Multicast Statistics*, tech. report, Univ. of Calif., Santa Barbara June 2000.
30. M. R. Izquierdo and D. S. Reeves, "A Survey of Statistical Source Models for Variable-Bit-Rate Compressed Video," *Multimedia Systems*, vol. 7, no. 3, July 1999, pp. 199-213.
31. Cisco Systems, "Integrating the Cisco IP/TV Broadcast Server with the Cisco ECDN Solution," tech. report, http://www.cisco.com/warp/public/cc/pd/mxsv/iptv3400/prodlit/wlmmy_wp.htm.



**Jiangchuan Liu** is working toward his PhD degree in computer science at the Hong Kong University of Science and Technology and is a recipient of a Microsoft Research fellowship. His current research interests include video transmission over wired and wireless networks. He received his BS (cum laude) in computer science from Tsinghua University, Beijing, China.



**Bo Li** is an associate professor in the Computer Science Department at the Hong Kong University of Science and Technology. His research focuses on optical networks, capacity and resource management in wireless cellular networks, Web proxy replication, and video multicasting. He received his BS and MS in computer science from Tsinghua University, Beijing, and his PhD in computer engineering from the University of Massachusetts at Amherst.

**Ya-Qin Zhang** is the Managing Director of Microsoft Research, Asia. His recent interests focus on the research and commercialization of multimedia information technologies. He's been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia. He received his PhD in electrical engineering from George Washington University and had executive business training at Harvard University. He's been granted more than 40 US patents in digital video, Internet, multimedia, and wireless and satellite communications. He's a fellow of the IEEE and has received numerous awards, including the Circuits and Systems Society's Jubilee Golden Medal.

Readers may contact Ya-Qin Zhang at yzhang@microsoft.com.