Multirate Video Multicast over the Internet: An Overview

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

Abstract

Multirate multicast is an effective method for video distribution to a set of heterogeneous receivers. In this article we present a comprehensive survey on multirate video multicast over the best effort Internet. We first review the key techniques in video encoding and network transport, and describe the representative approaches. We then study various trade-offs based on some important design issues and performance criteria, such as bandwidth economy, adaptation granularity, and coding complexity. Finally, we present some ongoing work and discuss possible avenues for future research.

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 is also an essential component of many current and emerging Internet applications, such as videoconferencing and distance learning, and thus has received a great deal of attention.

The Internet's intrinsic heterogeneity, however, makes video multicast a challenging problem. In traditional end-to-end adaptation schemes for unicast, the sender adjusts its transmission rate according to some feedback from its receiver. In a multicast environment, this solution tends to be suboptimal because there is no single target rate for a group of heterogeneous receivers. In other words, some receivers would be unfairly treated, and, at some branches of the multicast tree, the single-rate video stream would compete for bandwidth unfairly with other adaptive traffic, such as TCP flows. It is thus necessary to use multirate video multicast, in which receivers in a multicast session can receive video data at different rates according to their respective bandwidths or processing capabilities [1, 2].

Multirate video multicast over the best effort Internet remains a work-in-progress area. The goal of this article is to present a survey of the recent research in this area. We classify existing approaches according to the methods of generating the multirate video: stream replication, cumulative layering, and noncumulative layering. We then discuss representative techniques used in these approaches from both video coding and network transport perspectives. Since video and multicast are by themselves important research topics, we do not attempt to cover each in detail. Instead, we focus on their interactions that pose additional problems to designers; this specifically includes *the efficient transmission of multirate video streams to a large group of heterogeneous receivers using the Internet multicast infrastructure*. We also investigate the trade-offs of the approaches based on important design issues and performance criteria, including bandwidth economy, adaptation granularity, and coding complexity. Finally, some potential research issues are presented based on these investigations.

The rest of the article is organized as follows. We first present an overview of the existing solutions for video multicast. We then describe the representative multirate video multicast approaches, and study their trade-offs. Finally, we conclude the article and offer some potential research directions.

An Overview of Video Multicast Approaches

From the viewpoint of a video source, multirate video streams can be produced via two methods. The first is *information replication*; that is, the sender generates replicated streams for the same video content but at different rates. Each stream thus can serve a subset of receivers that have similar bandwidths.

The second is *information decomposition*. A commonly used decomposition scheme is *layering*, in which a raw video sequence is compressed into some nonoverlapping streams, or layers. The video quality is low if only one layer is decoded, but can be refined by decoding more layers. A receiver thus can selectively subscribe to a subset of layers according to its capacity or capability.

There are two kinds of layering schemes: *cumulative* and *noncumulative*. In cumulative layering, there is a layer with the highest importance, called 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 [3]. On the other hand, noncumulative layering supposes all layers have the same priority, and any subset of the layers can be used for video reconstruction [4]. Therefore, it yields higher flexibility than cumulative layering.

The above multirate adaptation approaches all rely on end-to-end services, where adaptation is performed on end nodes (the sender or receivers). The argument for *active services*, however, is that many applications can best be supported or enhanced 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 from its local region in a much easier manner. There are many trade-offs between end-to-end services and active services. In particular, the deployment of

0890-8044/03/$17.00 © 2003 IEEE

agents is mainly subject to network operators and service providers. In this article we focus only on multirate video multicast using end-to-end adaptation. Since no special assistance is required from intermediate nodes, the approaches discussed therein are readily applied in the current best effort Internet.

In Fig. 1, we give a preliminary taxonomy of the existing solutions for adaptive video multicast, where the circled approaches are our focus in this article. Their main objective is to tackle the heterogeneity problem in multicast, that is, to improve intra- and intersession fairness [5, 6] as well as TCP friendliness [7]. Meanwhile, the general requirements of scalability and stability for video multicast applications should be considered. In the following three sections we give more detailed descriptions of these approaches.

Stream-Replication-Based Multicast

Stream replication can be viewed as a trade-off between single-rate multicast and multiple point-to-point connections. Its feasibility is well justified in a typical multicast environment where the bandwidths of the receivers usually follow some clustered distribution. This is because they use standard access interfaces, for example, a 128 kb/s integrated services digital network (ISDN), 1.5 Mb/s asynchronous digital subscriber line (ADSL), or 10 Mb/s switched Ethernet, or might share some bottleneck links and hence experience the same bottleneck bandwidth. As a result, a limited number of streams can be used to match these clusters to achieve reasonably good fairness.

A representative stream replication protocol is the Destination Set Grouping (DSG) protocol [1]. In DSG, a source maintains a small number of video streams (say 3) for the same video content but with different rates. Each receiver subscribes to a stream that best matches its bandwidth. It can move among groups when its available bandwidth changes. It also monitors the video reception level and periodically reports this to the sender. A stream is then feedback-controlled within prescribed limits by its group of receivers. Specifically, if the percentile of the congested receivers is above a certain threshold, the bandwidth of the stream is reduced. If all the receivers experience no packet loss, its bandwidth is increased.

In practice, replicated video streams can be obtained through a set of independent encoders with different output rates at the source coding stage (e.g., through controlled quantization, pixel subsampling, or frame subsampling). This is the method used in the DSG experiments. Yet another method that can be applied to prestored video is *transcoding*. A transcoder converts an existing video stream into a new stream with a different format or rate. Since existing video coding standards, such as the MPEG and H.261/263 families, employ a similar sequential process, the *discrete cosine transform (DCT)-based motion-compensated hybrid coding*, fast transcoding can be achieved by directly manipulating data in the compressed domain (e.g., frequency filtering and quantization scale adjustment). In addition, the emerging MPEG-7 standard has defined *transcoding hints*, a set of meta-data in video streams that effectively help the transcoding procedure meet the speed and bandwidth requirements yet preserve high video quality.

Due to its simplicity, stream replication has been advocated in many commercial video streaming products, such as the *SureStream* mechanism provided by RelaNetworks' RealSystem G2. Nevertheless, the adaptation algorithms used in these products are relatively simple. Most of them use two or three pre-encoded streams to serve the receivers, while dynamic rate control on the sender's side and transcoding have seldom been considered.

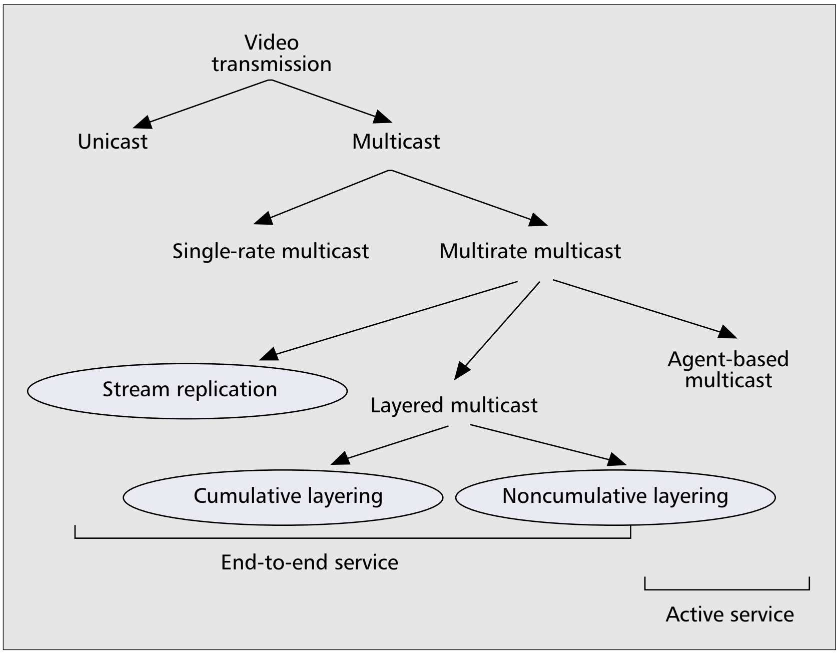
Cumulative Layered Multicast

The use of cumulatively layered or hierarchical information organization and prioritized transmission has long been an attractive idea for data delivery. The most important advantage of this approach, compared to stream replication, is that there is no significant information redundancy introduced by replication. However, cumulative layered video multicast imposes extra challenges in both networking and video coding areas. First, the source coder must have the ability to produce layered video streams. Second, to meet bandwidth constraints, some less important layers should be dropped prior to those important layers. This is nontrivial given that the best effort Internet does not provide differentiated services to the layers, and different branches could have different bandwidth constraints in a multicast tree.

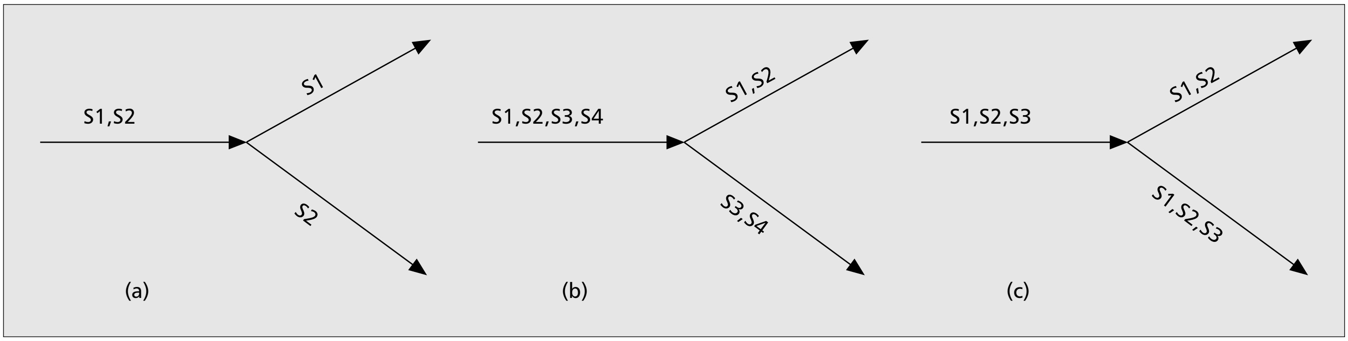
Cumulative Layered Multicast Protocols

McCanne, Jacobson, and Vetterli [2] proposed the first practical adaptation protocol for cumulative layered video multicast over the best-effort Internet. This protocol, known as Receiver-driven Layered Multicast (RLM), takes advantage of the dynamic group concept in the IP multicast model. A RLM sender transmits each video layer over a separate multicast group. The number of layers as well as their rates is predetermined. Adaptation is performed only at the receiver's end by a *probing-based* scheme. Basically, a receiver periodically joins a higher layer's group to explore the available bandwidth. If packet loss exceeds some threshold after the *join-experiment* (i.e., 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 induce packet losses experi-



■ Figure 1. *A taxonomy of video transmission approaches.*



■ Figure 2. *Illustration of the redundancy problem in upstream links;* S*i: a stream or layer: a) stream replication; b) noncumulative layering; c) cumulative layering (no redundancy).*

enced by other receivers sharing the same bottleneck link. These losses would occur frequently if all the receivers perform uncoordinated join-experiments. RLM incorporates a *shared learning* mechanism to solve this problem. With shared learning, the failure of a join-experiment conducted by a receiver is inferred by other receivers, thus avoiding the need for separate disruptive join-experiments. However, it reduces the scalability of RLM and significantly increases its convergence time. The difficulties associated with coordinating join and leave attempts motivated the design of the Receiver-Driven Layered Congestion Control (RLC) protocol [8]. RLC 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 is no packet loss during this experiment. Its convergence time can be much shorter than that of RLM since there is no coordination among the receivers.

In addition, it is well known that the original RLM does not ensure intersession fairness [6]; nor is it friendly to TCP traffic because its probing strategy is very aggressive [9]. 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 subsequent lower layer. This results in an exponential decrease of the bandwidth consumed in case of losses (like TCP). Nevertheless, the objective of TCP differs significantly from the objective of video transmission protocols. Although this solution interacts better with TCP, it could experience the same sawtooth behavior of TCP flows, resulting in unstable video quality. Therefore, rather than mimic the behavior of TCP, a more reasonable objective for video streaming would be to achieve a long-term fair share with TCP traffic. This is adopted by many model-based adaptation protocols, in which each receiver uses a model to estimate the equivalent bandwidth of a TCP connection running over the same path, and performs join and leave actions according to this estimated bandwidth [7, 10]. 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 the RTT between the sender and a receiver, require feedback packets, which may cause the well-known *feedback implosion* problem in large multicast sessions [11]. Some smart lightweight feedback loops have been developed to address this issue. For example, the Multicast Enhanced Loss-Delay-Based Adaptation (MLDA) protocol [7] employs an open-loop RTT estimation method as a complement to the closed-loop (feedback-based) method. It tracks the one-way trip time from the sender to the receiver and transforms it to an estimate of RTT. Link asymmetry can be compensated by low-frequency close-loop estimations.

Scalable Video Coding

In the video coding area, cumulative layered coding is often referred to as *scalable coding*, which can be achieved through scaling frame rate (*temporal scalability*), frame size (*spatial scalability*), or frame quality (*quality scalability*). Among them, temporal scalability is the most common scalability tool, and has been adopted in a diverse range of video compression standards, including H.263 and the MPEG family. In these standards, it is achieved by using intraframe coding (I frame), predictive coding (P frame), and bidirectional predictive coding (B frame), where a P frame depends on its previous I or P frame, and a B frame depends on a previous and a subsequent I or P frame. Layers can thus be mapped to different frame types; specifically, the base layer consists of the I frames. This has been used in experiments for many layered multicast protocols (e.g., [12]).

Noncumulative Layered Multicast

Noncumulative layering can be realized by the recent advances in multiple description (MD) video coding [4]. An MD coder generates multiple layers (referred to as *descriptions*) for the source signal. The descriptions are independent of each other and typically of roughly equal importance. For example, a simple two-description scheme can assign *odd* frames to one description and *even* frames to the other, and each description has its own prediction process and state information. As a result, low but acceptable video quality is achievable when *any* description is received, and can be refined when both descriptions are received and decoded together. There are also many other methods that have been proposed [4], such as the interleaving of subsampled lattice, MD scalar quantization, and MD transform. A common feature is that there are no critical layers needing special protection.

The basic idea of the adaptation algorithms for cumulative layered multicast, such as receiver-driven adaptation, can also be used in the noncumulative case [13]. Clearly, the resultant adaptation algorithm is more robust in the presence of packet loss. It also enables higher flexibility in layer subscription. For example, assume there are L layers; a layer bandwidth allocation $\{r_1, r_2, \ldots, r_L\} = \{2^0, 2^1, \ldots, 2^L\}$ can at most yield $2^L - 1$ distinct bandwidth levels for the receivers. Besides this exponential allocation, the work in [13] demonstrates that fine-grained adaptation can also be achieved by Fibonacci-like allocation schemes, and a fine-grained rate adjustment can be achieved by using at most three join or leave operations. This is very suitable for IP multicast in which join and leave are both costly operations.

Discussion and Comparison

In this section we try to comparatively discuss the above approaches; specifically, we focus on the following critical issues that are related to their performance and deployment: bandwidth economy, adaptation granularity, and coding complexity and efficiency.

Bandwidth Economy

A significant problem in stream replication is its high bandwidth redundancy. In particular, a link that is close to the sender may have to accommodate all the replicated streams, and its load thus can be very high, as illustrated in Fig. 2a. Although noncumulative layered coding does not produce replication among layers, redundancy could be introduced by the adaptation algorithm. Consider two receivers that share the same bottleneck link and select the subsets of the layers for subscription independently. If the two subsets are disjoint or have very little overlap, an upstream link has to accommodate their total bandwidth. An example is shown in Fig. 2b, where one receiver joins layers S1 and S2, and the other joins S3, and S4; consequently, the uplink has to deliver all four layers.

The redundancy problem is particularly severe when there are many shared links and the receivers are highly heterogeneous. Such situations are unfortunately common in a large-scale IP multicast network. The use of a Fibonacci-like layer rate allocation can alleviate the redundancy, although it cannot entirely eliminate it [13]. With the cumulative subscription policy, there is no such problem because at least one receiver will subscribe to all the layers delivered by an upstream link. Nevertheless, the adaptation granularity is sacrificed when imposing this constraint, as discussed below.

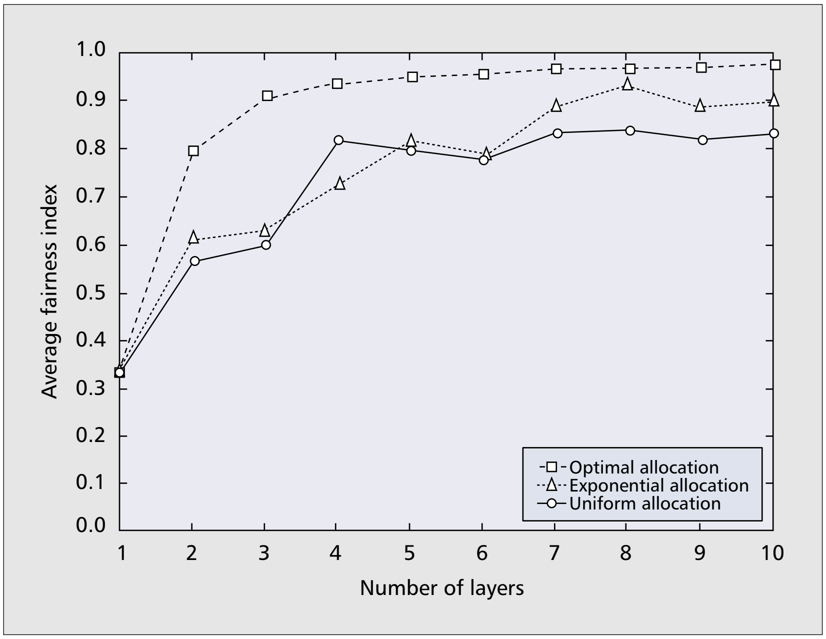


Figure 3. *Fairness as a function of layer number for different allocation schemes.*

Adaptation Granularity

The fairness of a multirate multicast scheme is closely related to its bandwidth adaptation granularity. If the granularity is very coarse, significant mismatches could occur between a receiver's expected bandwidth and the received video bandwidth. Consequently, the degree of intrasession fairness is degraded. In addition, a video stream cannot fairly compete for bandwidth with TCP flows if the expected bandwidth is estimated using the equivalent TCP throughput.

As discussed earlier, assume there are *N* streams or layers; the number of distinct bandwidth levels could be as high as $2^N - 1$ for noncumulative layering. For stream replication and cumulative layering, it is at most *N*. Hence, if *N* is small, a receiver's adaptability for bandwidth heterogeneity and network congestion is quite limited.

There are two possible methods to reduce the bandwidth mismatch. The first is to simply use more streams or layers. However, this will introduce more redundancy for stream replication and is often not supported by existing scalable video coders. For example, most temporary, spatial, or quality scalability algorithms have a fixed layering structure with only two or three layers. In addition, when the number of layers is increased, a receiver would have to perform a substantial number of join and leave operations for adaptation. The benefits from the improved granularity can be contradicted by such overheads, particularly in the current IP multicast model where join and leave remain costly operations.

The second method is to adaptively allocate the bandwidth to streams or layers to minimize the *expected* mismatch. This sender-based adaptation has been advocated in protocols like DSG [1] or MLDA [7]. Note that such a protocol needs a feedback loop for monitoring the receivers' states. To avoid feedback implosion, DSG and MLDA estimate states at coarse-grained levels: three in DSG (congested, loaded, and unloaded) and two in MLDA (the maximum and minimum receiver bandwidths). This can be done via some scalable feedback algorithms, but the resultant allocation is obviously suboptimal. An optimal algorithm specifically designed for a two-stream DSG protocol can be found in [5]. A more general example is the Hybrid Adaptation Layered Multicast (HALM) protocol [10]. The rationale in HALM is that to minimize the expected mismatch, the allocation algorithm needs only the bandwidth distribution of the receivers. Hence, sampling can be used to collect bandwidth reports from the receivers, and, in general, only limited reports are needed to make an allocation decision.

Another important problem in sender adaptation is how to control the rate of the video. For stream replication, it is relatively easy to fine-tune the rate for each single stream either at the source coding stage or by transcoding. For scalable coding, however, fine-grained rate control is often not supported in conventional algorithms. For example, the typical bit rates for MPEG-1 temporal scalability are 256 kb/s for the I stream, 900 kb/s for the I+P stream, and 1.5 Mb/s for the I+P+B stream, and they can be tuned only in a confined range. Fortunately, the recently developed MPEG-4 fine granularity scalability (FGS) standard has provided such support [3]. FGS uses embedded representations in compression, where layers can be generated by truncating the embedded stream at specified positions. As a result, it has a very flexible layering structure and can fine-tune the rate of each layer with fast responsiveness.

In Fig. 3, we show the degree of fairness vs. the number of layers for cumulative layered video multicast. There are three schemes in the comparison; one employs the optimal allocation at the sender's end, and the other two employ typical static allocation schemes: uniform allocation and exponential allocation. We calculate the fairness index based on the inter-receiver fairness function [5]. Some interesting observations can be made from this figure. First, for the optimal allocation, three to five layers offer satisfactory fairness. Any further improvement in using more than five layers is marginal. Second, the optimal-allocation-based scheme exhibits much better performance and often outperforms the two static schemes by 20 percent. A more interesting phenomenon due to the nonadaptability of the static schemes is that the expected fairness index does not monotonically increase with the number of layers. For example, with the uniform allocation, the fairness degree of using four layers is higher than that of five or

		Single rate	FGS	MDC
Forman, QCIF	Bit rate (kb/s)	79.3	102.3	114.6
PSNR = 33.3 dB	Overhead	–	29%	44%
Akiyo, QCIF	Bit rate (kb/s)	19.3	23.8	25.7
PSNR = 35.4 dB	Overhead	–	23%	33%

■ Table 1. *Comparison of coding efficiency.*

even six layers. On the other hand, we can prove that with the optimal allocation, increasing the number of layers always leads to a higher degree of fairness [10]. This also gives justification for the use of optimal allocation.

Coding Complexity and Efficiency

To date, the single-rate (or single-layer, single-description) video coding remains the most efficient and effective technique. Transcoding an existing single-rate stream to another stream usually sacrifices efficiency, but existing work shows that its bandwidth overhead is limited, usually less than 5 percent. The time taken to transcode a stream is also much less than that to encode it from scratch, because the DCT coefficients as well as motion vectors can be reused. In contrast, scalable coding usually has high computation complexity because of the iterative motion estimation and DCT transform for all the layers. The bandwidth redundancy is also accumulated in these iterations, which leads to significant quality degradation. Transporting the layers incurs bandwidth penalty as well (e.g., the extra bits for synchronizing layers). Multiple description coding is still in its infancy. To make each description provide acceptable visual quality, each description must carry sufficient information about the original video. This can reduce the compression efficiency even more than cumulative layering.

For illustration, Table 1 lists the bit rates for two standard test sequences, *Foreman* and *Akiyo*, using MPEG-4 single-rate coding, FGS coding [3], and a very recent MD algorithm [4]. It can be seen that the bandwidth penalty for layered coding could be higher than 20 percent. Given such a higher penalty, experiments show that layered multicast is not necessarily superior to stream replication for many network topologies [14].

Finally, on the receiver side, a scalable or MD video stream requires high computation power to assemble and decode multiple layers, and is usually not compatible with existing video decoding algorithms. On the contrary, replicated streams can serve receivers with much simpler and standard-compatible decoding algorithms. Since the streams are independent, they can even be of heterogeneous formats. Therefore, stream replication remains an effective technique to address user heterogeneity.

Summary and Research Issues

In Table 2 we summarize the video coding and bandwidth adaptation mechanisms for the discussed multirate multicast approaches, together with their relative performance using state-of-the-art techniques and under general conditions. We have also made some observations on future research on multirate video multicast. However, a wide range of open issues lie in this area that cannot be covered in this short conclusion. In the following, we list only those in which we are particularly interested:

Feedback loop: As discussed earlier, feedback is needed for sender adaptation as well as RTT estimation for TCP-friendly protocols. To avoid implosion, these protocols use probabilistic feedback algorithms or keep the feedback at low frequency. Some clustering algorithms have also been developed for receivers to share information with their neighbors [10]. Nevertheless, the accuracy and responsiveness of these mechanisms remain questionable in very large multicast sessions. Although we understand that feedback is crucial to these protocols, we do not believe an end-to-end feedback loop can simultaneously satisfy the requirements of scalability, accuracy, and responsiveness. A possible solution to this problem is the use of active service; for example, to deploy some feedback mergers in the network, as suggested in [12].

Bandwidth fairness: It has been shown that multirate multicast is "more fair" than single-rate multicast under the widely accepted max-min fairness notion [15]. However, a max-min fair allocation may not exist in the layering case where a

		Stream replication	Cumulative layering	Noncumulative layering
Video coding	Technique	Rate adaptive coding or transcoding	Scalable coding	Multiple description coding
	Efficiency	High	Low	Low
	Complexity	Low	High	High
	Compatibility	Compatible with existing decoders	Optional in standards, may not compatible with existing decoders	Usually not compatible with existing decoders
	Format	Can be heterogeneous in a session	Homogeneous in a session	Homogeneous in a session
Bandwidth adaptation	Sender algorithm*	Feedback-based rate allocation	Feedback-based rate allocation	–
	Receiver algorithm	Stream switching	Layer joining/leaving	Layer joining/leaving
	Redundancy	High	Low	High
	Granularity	Coarse	Coarse	Fine
* Sender adaptation is often used as a complement to receiver adaptation, and is available only in some of the protocols (e.g., [1, 7, 10])				

■ Table 2. *A summary of the multirate multicast approaches.*

receiver cannot subscribe to fractional layers [16]. Moreover, although the notions of intrasession fairness and TCP friendliness have been extensively used in existing protocols, the exact definition of fairness in the context of multicast is still a matter of debate. For example, it remains a question whether a multicast flow and a TCP flow should get exactly the same bandwidth share. In video multicast, such a problem is further complicated since a video stream generally needs a minimum bandwidth guarantee and its utility (perceptual quality) is often nonlinear with its bandwidth. In addition, improving fairness often conflicts with stability. For example, existing layered adaptation algorithms usually use a *greedy subscription* scheme (i.e., whenever spare capacity is discovered, a higher layer is joined). Additional mechanisms, such as a join-timer, are then used to mitigate bandwidth fluctuations. This implies that fairness is considered prior to stability, but such a policy is not necessary the best for video streaming, particularly when the video traffic and cross-traffic are nonstationary.

Stream or layer switching: Most existing adaptation protocols ignore the detailed operation for stream or layer switching. However, this is nontrivial given that a video stream generally has a complex syntax, and its content is highly dependent. For example, suppose in a stream-replication-based transmission a receiver plans to switch to another stream; here comes a basic problem: when should it switch? If it switches immediately, significant quality drift would occur if the next frame of the new stream is a P frame, which depends on its previous I or P frame. For low-bit-rate coding, there are often many P frames between two I frames, say 100 or more. This implies that drift would occur with high probability and could propagate to many following frames. On the other hand, if the receiver waits until the next I frame in the new stream comes, how does it access this information, especially when two streams could be asynchronous?

For layer switching, similar problem exists when an enhancement layer is to be added. FGS coding solves this problem by always using the base layer as the motion prediction reference for all enhancement layers. Nevertheless, this can noticeably reduce the efficiency of the prediction loop.

Note that, in the above example, we have not considered other practical limits, such as the join and leave latencies in the IP multicast model. How perceptual video quality is smoothed during this switch to avoid annoying sudden changes is also not easy. To conclude, there are many more practical issues that need to be addressed in a comprehensive system. These issues all offer interesting research topics, and their solutions become increasingly imperative with the increasing demands on video multicast applications.

References

[1] S. Cheung, M. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution," *Proc. IEEE INFOCOM '96*, Mar. 1996.

[2] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," *Proc. ACM SIGCOMM '96*, Aug. 1996.

[3] W. Li, "Overview of the Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 11, no. 3, Mar. 2001, pp. 301–17.

[4] Y. Wang and S. Lin, "Error-resilient Video Coding Using Multiple Description Motion Compression," *IEEE Trans. Circuits and Sys. for Video Tech.*, vol. 12, no. 6, June 2002.

[5] T. Jiang, E. Zegura, and M. Ammar, "Inter-receiver Fair Multicast Communication over the Internet," *Proc. NOSSDAV '99*, June 1999.

[6] R. Gopalakrishnan *et al.*, "Stability and Fairness Issues in Layered Multicast," *Proc. NOSSDAV '99*, June 1999.

[7] D. Sisalem and A. Wolisz, "MLDA: A TCP-Friendly Congestion Control Framework for Heterogeneous Multicast Environments," *Proc. IWQoS 2000*, June 2000.

[8] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like Congestion Control for Layered Multicast Data Transfer," *Proc. IEEE INFOCOM '98*, Apr. 1998.

[9] A. Legout and E. W. Biersack, "Pathological Behaviors for RLM and RLC," *Proc. NOSSDAV 2000*, June 2000.

[10] J. Liu, B. Li, and Y.-Q. Zhang, "A Hybrid Adaptation Protocol for TCP-Friendly Layered Multicast and Its Optimal Rate Allocation," *Proc. IEEE INFOCOM '02*, June 2002.

[11] J. Bolot, T. Turletti, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," *Comp. Commun. Rev.*, vol. 24, no. 4, Oct. 1994, pp. 58–67.

[12] B. Vickers, C. Albuquerque, and T. Suda, "Source Adaptive Multi-layered Multicast Algorithms for Real-time Video Distribution," *IEEE/ACM Trans. Net.*, vol. 8, no. 6, Dec. 2000, pp. 720–33.

[13] J. Byers, M. Luby, and M. Mitzenmacher, "Fine-Grained Layered Multicast," *Proc. IEEE INFOCOM '01*, Apr. 2001.

[14] T. Kim and M. Ammar, "A Comparison of Layering and Stream Replication Video Multicast Schemes," *Proc. NOSSDAV '01*, June 2001.

[15] D. Rubenstein, J. Kurose, and D. Towsley, "The Impact of Multicast Layering on Network Fairness," *IEEE/ACM Trans. Net.*, vol. 10, no. 2, Apr. 2002, pp. 169–82.

[16] S. Sarkar and L. Tassiulas, "Fair Allocation of Discrete Bandwidth Layers in Multicast Networks," *Proc. IEEE INFOCOM 2000*, Mar. 2000.

Biographies

BO LI (bli@cs.ust.hk) is an associate professor in the Computer Science Department, Hong Kong University of Science and Technology. He received his B.S. and M.S. from Tsinghua University (Beijing), and a Ph.D. from the University of Massachusetts at Amherst. He worked for IBM Networking Systems at RTP between 1994 and 1996. His recent research has been focused on video multicasting, content replication, and resource management in cellular networks. He has served as an editor or a guest editor for over 10 journals in IEEE and ACM.

JIANGCHUAN LIU (csljc@cs.ust.hk) received a B.S. degree (cum laude) in computer science from Tsinghua University, Beijing, P.R. China, in 1999. Currently, he is working towards a Ph.D. degree in computer science at Hong Kong University of Science and Technology. He is a recipient of a Microsoft research fellowship and had internships with Microsoft Research, Asia, in the summers of 2000, 2001, and 2002. His current research interests include video multicasting and service location in peer-to-peer and variable topology networks, on which he has published over 20 papers.