

Sender-Adaptive and Receiver-Driven Layered Multicast for Scalable Video Over the Internet

Qian Zhang, *Senior Member, IEEE*, Quji Guo, Qiang Ni, Wenwu Zhu, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

Abstract—In this paper, we propose and analyze a new system architecture for video multicast over Internet, namely, the sender-adaptive and receiver-driven layered multicast (SARLM). In SARLM, the sender of a video source splits the video data coded by a scalable codec and a channel codec into multiple data streams, each of which corresponds to a separate multicast group. The sender can adjust the way in which the video sequence is split dynamically based on the receivers' network parameters collected through feedback. Meanwhile, a receiver can estimate available bandwidth based on a modified packet-pair technique and choose to reassemble and playback the video sequence for a given quality level by dynamically subscribing a given part or all of the data streams according to its network conditions. To optimize the sender's adaptation strategy, we introduce a quality-space (Q-Space) model to describe and analyze the mathematical relationship between the sending rate of different SARLM layers and the video quality received by a given receiver identified by its network characteristics including available bandwidth and packet loss ratio. Our simulation results demonstrate that, under the same network topology and condition, the SARLM architecture can achieve higher network throughput and better video qualities on the receiver side than the existing approaches.

Index Terms—Automatic repeat request (ARQ), bandwidth estimation, forward error correction (FEC), feedback implosion, layered multicast, receiver driven, scalable video, sender adaptive, streaming.

I. INTRODUCTION

MULTICAST is a promising technique for cost effectively delivering multimedia data to a large number of receivers simultaneously over the Internet. It is especially suitable for delivering popular content with a lot of concurrent requests, as it saves network resources by sharing the data streams across receivers with shared links. However, using one data stream with fixed bit rate and channel coding scheme can not best serve receivers that have heterogeneous and time-varying bandwidth and packet loss ratio. A lot of research has been conducted to

Manuscript received May 29, 2003; revised October 28, 2003. Part of this work has been presented at IEEE ISCAS2001 and IEEE ICME2001. This paper was recommended by Associate Editor M. Strintzis.

Q. Zhang and Y.-Q. Zhang are with Microsoft Research Asia, Beijing 100080, China (e-mail: qianz@microsoft.com; wwzhu@microsoft.com; yzhang@microsoft.com).

Q. Guo is with Microsoft Research Asia, Beijing 100080, China. He is also with Stanford University, Stanford CA 94305 USA (e-mail: qujiguo@hotmail.com).

Q. Ni was with Microsoft Research Asia, Beijing 100080, China. He is now with Planete Group, INRIA Sophia Antipolis, France, 06902 Sophia Antipolis, France (e-mail: Qiang.Ni@Sophia.inria.fr).

W. Zhu is with Intel Communication Technology Laboratory, Beijing 100020, China (e-mail: wenwu.zhu@intel.com).

Digital Object Identifier 10.1109/TCSVT.2005.844454

address this challenge by using layered multicast, which can be largely categorized into three categories: receiver-driven approach, sender-driven approach, and hybrid sender-driven and receiver-driven approach.

Receiver-driven multicast proposed first by McCanne *et al.* [1] uses a layered video codec to generate multiple data streams from a single video sequence, and sends each data stream as a separate multicast group. The receivers reassemble and playback the video with different qualities by subscribing different combinations of the multicast groups dynamically following a predefined joining/leaving policy. In the original proposal, due to lack of bandwidth knowledge on the receiver side, a receiver tried to subscribe more groups from time to time to get more data. Congestions occur when such attempts happen without enough bandwidth available, and consequently affect other receivers sharing the same bottleneck link. L. Vicisano *et al.* adopted coordinated join tests and sender-initiated probes to reduce the chances of failure of such subscribing tests [2]. Later on, Legout introduced a Packet-pair receiver-driven layered multicast (PLM) protocol [3] that estimates the available bandwidth using packet-pair for receivers to make subscription decisions. In [31], Tan *et al.* proposed the use of layered forward error correction (FEC) as an error control mechanism in a layered multicast framework. By organizing FEC into multiple layers, receivers can obtain different levels of protection commensurate with their respective channel conditions. However, in all these receiver-driven approaches the sender's strategy is predefined, and it may lead to joining/leaving oscillations at receivers when the network conditions are changing. Note that if the sender were able to dynamically adjust its parameters, such as the number of layers it splits the video, the bit rate in each layer, the protection level for each layer, the receivers may obtain the video with better quality.

Dealing with packet loss effectively is another important issue in video streaming over Internet. Automatic repeat request (ARQ) has been widely used to address packet loss, however, ARQ-based error control in multicast may lead to implosion of repeated requests to the sender. To avoid feedback implosion, most reliable multicast used explicit hierarchical retransmitters scattering throughout the network [24]. FEC with packet interleaving is also widely used for packet loss recovery of continuous media transmission [29], [30], but it introduces constant transmission overhead regardless of network conditions. Using hybrid FEC/ARQ, Nonenmacher *et al.* [28] proposed a scheme answering multiple negative acknowledges (NACKs) from different receivers with a single parity packet. Independently, Chou *et al.* studied how to use

FEC and pseudo-ARQ for receiver-driven layered multicast [13]. In their proposal, redundant packets are delayed and sent through different multicast channels. The receivers that have missed some of the original packets can dynamically subscribe some of these channels to recover from the packet loss. It is also a receiver-driven approach where the senders' behavior is independent of the receivers and the networks.

On the receiver side, the major challenges for layered multicasting schemes are how to make the joining/leaving decision and how to avoid feedback implosion. In multicast, the implosion problem is caused by a large number of feedback packets sent back to the sender simultaneously. To avoid feedback implosion, several solutions have been proposed, which are hierarchical [5], parameterized [10], and randomly delayed timers [11] based schemes, respectively. Hierarchical acknowledgment prevent feedback implosion by designating selected destinations as special feedback consolidating nodes. Since such methods need special functionality on the routers, they cannot be implemented everywhere. In parameterized schemes, multicast responses from one receiver suppress those from other receivers that have higher throughputs. The drawback is that the statistical properties of all receivers cannot be completely preserved. The random-delayed-timer based methods use a truncated exponential distribution to generate the timer to achieve efficient feedback suppression. In such schemes, a multicast or multi-cast-emulated feedback channel is needed for every receiver.

Another large category of approaches are sender-driven multicast, in which the receivers can actively make decision on their own. In such methods, the sender changes the bit rates dynamically according to the feedback. Bolot *et al.* proposed a scheme where the parameters of the source codec are adjusted according to the feedback [4]. The scheme also uses a probing mechanism to solicit feedback in a scalable manner and to estimate the number of receivers. The major limitation of this method is that it cannot serve heterogeneous receivers very well, because only a single layer is used. Vickers *et al.* introduced the source adaptive multilayered multicast (SAMM) algorithm [5], where the sender uses congestion feedback to adjust the number of layers and the bit rate of each layer. However, feedback mergers are required in the network, which are difficult to be implemented in practice.

Combining both receiver-driven and sender-driven approaches, Cheung *et al.* proposed a destination set grouping (DSG) algorithm [6], and Leannec *et al.* proposed a hybrid sender and receiver driven rate control scheme for layered video multicast [27]. In [6], a sender encodes video into independent streams with different rates, which are adjusted according to feedbacks, and each stream is targeted at a subset of receivers. This scheme does not efficiently utilize network bandwidth because it uses independent streams rather than layered streams. In [27], a mechanism for rate allocation in each layer is proposed for providing optimal bandwidth usage for all the receivers. However, feedback mergers need to be deployed throughout the network, just like SAMM scheme in [5].

To address the issues mentioned above, in this paper, we introduce a new architecture called sender-adaptive and receiver-driven layered multicast (SARLM) for video multicast

over the Internet. SARLM scheme uses a layered multicast approach with FEC and pseudo-ARQ to support heterogeneous receivers with different bandwidths and packet loss. It also specifically deals with dynamic characteristics of a video multicast session, where a receiver dynamically joins or leaves based on its network conditions including time-varying available bandwidth and packet loss ratio. Meanwhile, the sender also adapts its sending strategy to these time-varying factors. On the other hand, we acknowledge the limitation of sender-adaptation in several aspects, and propose a combined sender-adaptive and receiver-driven approach where such limitations are alleviated by the receivers' autonomous behaviors. The SARLM scheme is an application layer approach, and thus does not require any changes in network routers. For gaining more insights and deeper understandings of different layered multicast schemes, we also introduce a mathematical model, namely quality-space (Q-Space) model, to describe and analyze these schemes in a generalized fashion.

In SARLM, we suggest that each receiver's subscribing/leaving strategy to/from the video layers explicitly depends on estimated available bandwidth rather than blind joining tests. We propose a modified consecutive receiver based packet-pair (RBPP) algorithm to estimate the available bandwidth of each receiver, while other alternative algorithms can be plugged in without changing the framework. A gamma-distributed random timer is further proposed for generating scalable feedback sent back to the sender, which can largely reduce the number of required feedback.

The rest of this paper is organized as follows. In Section II, we present our architecture of SARLM for video multicasting. In Section III, we describe multicast group generation and parameters optimization on the sender side. Section IV discusses joining/leaving behavior management on the receiver side. In Section V, we propose a mechanism for generating scalable feedbacks. Experiment results are presented and analyzed in Section VI, and Section VII concludes this paper.

II. SARLM SYSTEM ARCHITECTURE FOR VIDEO MULTICASTING

Fig. 1 depicts our SARLM system, which consists of a video multicast server (the sender), a feedback analyzer that may or may not physically reside in the same location as the sender, and a number of receivers that dynamically join and leave the video session based on their users' conditions. The sender's role is to dynamically generate multiple data streams and in turn generate multiple multicast groups from the video source using a scalable video source codec and FEC and pseudo-ARQ. Each receiver monitors its network condition, and sends sparse feedback packets containing statistical information about its network conditions back to the analyzer. The receivers also make their subscribing decisions. As an important system component, the feedback analyzer collects the feedback packets, and then determines how the sender splits the video into multiple layers by providing the sender important parameters such as source rate and protection rate based on the feedbacks and the Rate-Distortion function of the given video codec. These three components form a loosely closed loop. Notice that a SARLM system does

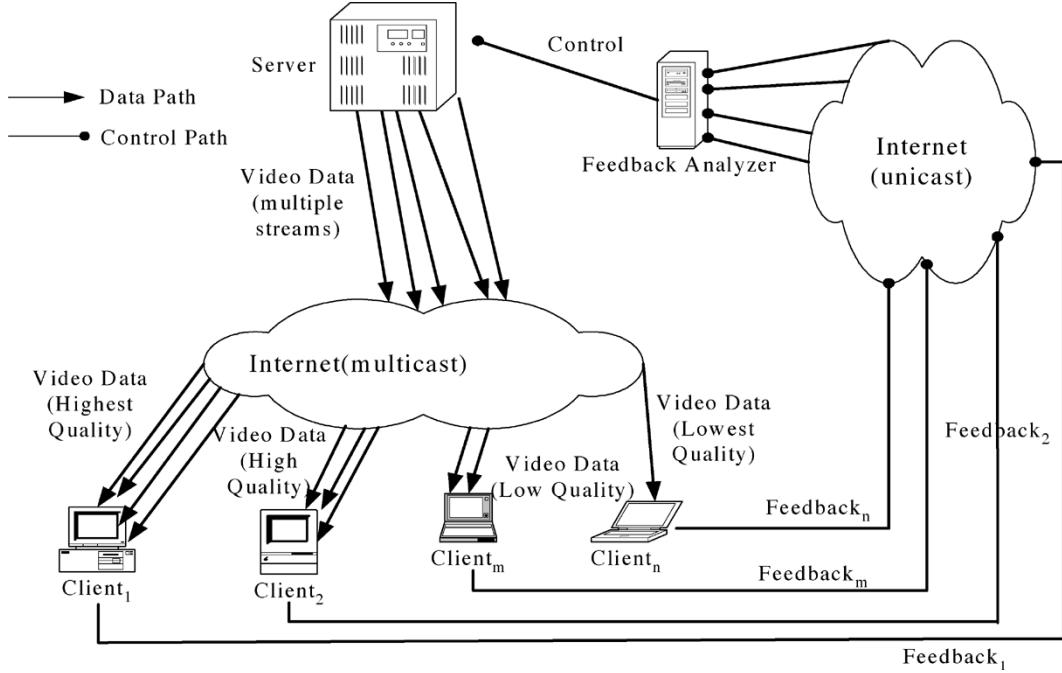


Fig. 1. Basic framework of a SARLM system.

not require any special distributed components across the network beyond standard multicast-capable routers.

In SARLM, the sender first classifies the receivers into groups dynamically based on their reported network parameters, and then adjust the layered multicast scheme in such a way that each group of receivers are expected (but not required) to subscribe certain group. This jointly provides the best video quality for the receivers in each group under their given network conditions. However, unlike sender-driven approaches, in SARLM the sender leaves the freedom of choosing which multicast groups to join to the receivers, therefore this approach is yet receiver-driven at this point.

There are several reasons for us to keep the architecture receiver-driven. First, the feedbacks to the sender are always delayed. Second, the classification may be sub-optimal due to computational constraints, which prevent the sender adaptation from being further delayed. Third, the network parameter estimation may not be precise all the time. In short, the sender can hardly have perfect knowledge about all the receivers' network conditions, and thus individual receiver should be in a better position to make its own subscription decision.

On the sender side, the multicast groups are formed as follows.

Source Coding: A quality scalable codec, such as MPEG-4 PFGS video coder [14], is used for source coding so that the sender is able to adjust the bit rate in each layer arbitrarily. In the rest of this paper, we use H to represent the number of layers, and $V_h (h = 1, \dots, H)$ to represent the data stream of a given layer of h . Due to the nature of layered/scalable coding, there is an explicit data dependency between V_k and any $V_j (1 \leq j < k)$, i.e., if a receiver subscribes V_k , it has to subscribe all $V_j (1 \leq j < k)$ as well. We call $V_h (h = 1, \dots, H)$ primary streams.

Channel Coding: Considering the characteristic of the scalable video codec, we apply unequal error protection (UEP) to different layers according to different rate-distortion relation as

well as receivers' statistics of each layer. In a given layer h , the sender applies the channel coding scheme such that a packet block of total K_h packets in the video source stream (primary stream) is interleaved, and is channel coded into a packet block of $N_h (N_h > K_h)$ packets. Whenever a receiver has received any K_h packets within the $N_h (N_h > K_h)$ packets, the original K_h source packets can be reconstructed. For computational convenience, the channel coding scheme usually can be designed in such a way that the first K_h packets in the corresponding N_h packets are actually the same as the original source packets. For example, the systematic Reed-Solomon (RS) channel coding scheme can be used to achieve our goals.

Multicast Group Formation: As described above, for a given packet block of K_h source packets in a given layer h , there are $N_h - K_h$ corresponding protection packets. The sender divides them into $F_h (F_h \leq N_h - K_h)$ data streams, namely *protection streams*. If $N_h - K_h$ is small, we can simply let $F_h = N_h - K_h$; otherwise an additional mechanism is needed to determine the value of F_h . And then the sender sends out each primary stream and each protection stream as an independent multicasting group. In the rest of the paper, we use $P_{i,j}$ to represent the j th protection stream in layer i .

Fig. 2 illustrates the formation of all data streams, each of which corresponds to a multicast group.

Protection Delay (Pseudo-ARQ): We adopt the idea of pseudo-ARQ [13] in our system. In each layer, protection streams are delayed compared to its primary stream. Then the second protection stream is delayed compared to the first stream, and so on. A receiver that has missed some packets in the primary stream can dynamically subscribe some of the delayed protection streams to recover from the packet loss. As soon as the packet loss is recovered, the receiver can drop the protection streams to free some network bandwidth.

The selection of delay parameters should consider the system latency (including the packet transmission delay from the sender

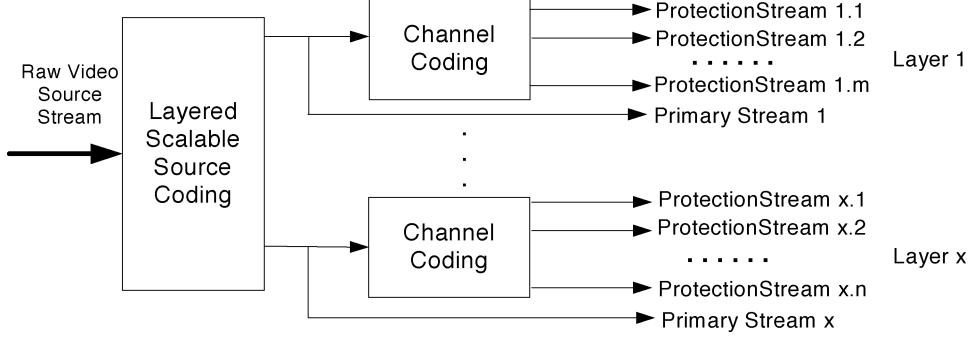


Fig. 2. Generation process of multiple streams.

to the receivers and the multicast group joining latency) and the size of video buffer possibly used on the receiver side. On the one hand, it is favorable to have large delay value so that the receivers can have enough time to assert that the source packets are most likely lost, and to subscribe the protection streams dynamically. However, on the other hand, if the delay value is too large, compared with the video buffer time on the receiver side, the protection packets become useless when it comes to the receiver.

On the receiver side, two different strategies are used for making decisions for joining/leaving multicast groups. For subscribing/leaving primary streams, a receiver applies a decision rule based on its dynamically estimated available bandwidth. A smoothing filter is used for getting rid of noise in bandwidth estimation. For subscribing/leaving protection streams, when the receiver asserts that a certain number of packets in a packet block have been lost, it subscribes the corresponding number of delayed protection streams for recovering from the loss. These two different strategies imply that subscribing/leaving protection streams may happen more frequently than subscribing/leaving primary streams. An intuitive explanation to such a difference is that experiencing packet loss without an observed available bandwidth change suggests a burst congestion, from which the receiver may recover with the pseudo-ARQ mechanism; while an observed available bandwidth change is likely to have a longer term impact on the communication channel between the sender and receiver due to forming factors such as routing changes, long term background traffics introduced, etc. More details about subscribing/leaving management on the receiver side will be discussed in Section IV.

The receivers also feed their observed/estimated network parameters, for example, available bandwidth, packet loss ratio, and network latency, back to the feedback analyzer/sender through a UDP unicast channel to avoid additional requirement in the network layer. To avoid feedback implosion, we propose that each receiver uses a gamma-distributed random timer to adjust its feedback sending rate, following control information given by the sender based on the total feedback density. We will discuss our feedback implosion suppression scheme in detail in Section V.

III. FEEDBACK ANALYSIS AND PARAMETER OPTIMIZATION AT THE SENDER SIDE

In SARLM, the feedback analyzer collects all the feedbacks from the receivers, and applies an optimization scheme based

on these data to determine how the sender should split the video source into multiple streams. In this section, the underlying model of layered multicast and its optimization solution are discussed.

A. Mathematical Model for Layered Multicast

In all layered multicast schemes, whether adaptive or not, the major idea is to serve heterogeneous receivers without losing scalability or introducing network inefficiency. In a simplified view, the heterogeneity of the receivers on the Internet is mainly captured by two parameters: the bandwidth and the packet loss ratio; while other parameters, such as network latency, jitter, etc., are usually as secondary considerations. To gain a deeper understanding of these layered multicast schemes, and also to introduce the theoretical foundation for the feedback analyzer in our SARLM system, we introduce a Q-Space model as follows.

Considering only the bandwidth and the packet loss ratio, we can describe a receiver as a vector

$$\vec{U} = (B_u, L_u) \quad (1)$$

where B_u is its available bandwidth, and L_u is its packet loss ratio. Note that, in some cases B_u can be a desired data rate instead if other data rate constraints other than bandwidth availability need to be applied. For instance, one such a constraint could possibly be TCP friendliness.

To further simplify the mathematical model, we take the following reasonable assumptions: 1) the layered multicast scheme uses a fixed number of layers, H ; 2) the packet losses are independent of each other; and 3) the rate-distortion (R-D) function of the scalable video codec in each layer is known.

Based on previous literature [22], [26], if we use a layered scalable codec, which has explicit data dependency across layers, for the given receiver \vec{U} , we can then obtain an optimal (minimized expected distortion) rate distribution that is described by a vector $(R_1, \dots, R_H, P_1, \dots, P_H)$, where R_h is the video source bit rate in layer h , and P_h is the bit rate of protection data. For a given $R_u = \sum_{h=1}^H R_h$ and $P_u = \sum_{h=1}^H P_h$, the vector $(R_1, \dots, R_H, P_1, \dots, P_H)$ can be calculated to achieve minimum expected distortion. Therefore, we can actually reduce the dimensions of the vector, and represent it as

$$\vec{M} = (R_u, P_u). \quad (2)$$

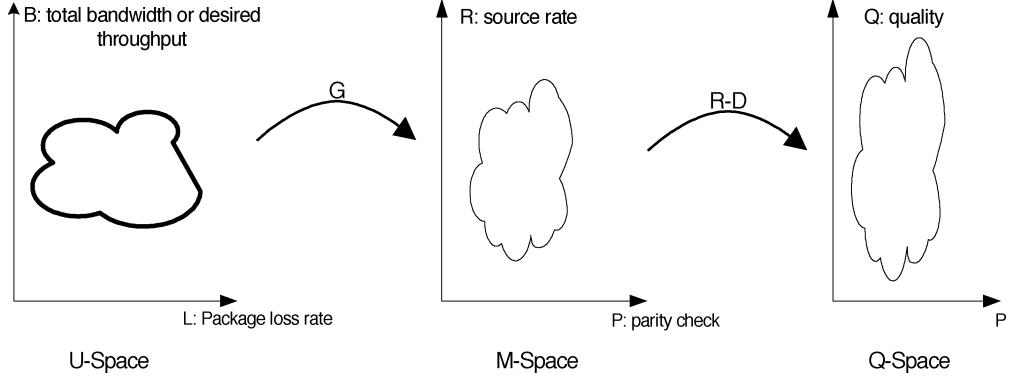


Fig. 3. Illustrations of the mapping from U-Space to Q-Space.

We have built a one-to-one mapping between \vec{U} and \vec{M} , which means each point in the M-Space corresponds to a given receiver \vec{U} through mapping function G , i.e.,

$$\vec{M} = G(\vec{U}). \quad (3)$$

With the known rate-quality relationship Φ , we have

$$Q = \Phi(R) \quad (4)$$

where Q is the video quality and R is the video source bit rate. That is to say, we can map each value of R_h to a value of video quality Q_h . In turn, we can also map each \vec{M} to $\vec{Q} = (Q_h, P_h)$. We call the range of \vec{U} , \vec{M} , and \vec{Q} U-Space, M-Space, and Q-Space, respectively.

In the real world, for each receiver the bandwidth and packet loss ratio is time-varying. Therefore, the description of a receiver illustrated as one point in U-Space is naturally extended to a random vector with a probability density function (PDF), $f(\vec{U})$, which indicates the probability density of having bandwidth B_u and packet loss ratio L_u at any time. In multicasting case, from the sender's point of view, the combined effect of all receivers, $f_{\text{all}}(\vec{U})$, corresponds to a PDF in U-space as follows:

$$f_{\text{all}}(\vec{U}) = \frac{1}{N} \sum_{n=1}^N f_n(\vec{U}), \quad (5)$$

where $f_n(\vec{U})$ is the PDF for a receiver n , and N is the total number of receivers.

In case that users have different priorities, additional weights can also be put into $f_{\text{all}}(\vec{U})$. By using function G , we can get its projection in M-Space: $f_M(\vec{M}) = f_{\text{all}}(G^{-1}(\vec{M}))$. Note that here G^{-1} exists, since mapping between \vec{U} and \vec{M} is one-to-one. Fig. 3 illustrates the mapping from U-Space to Q-Space. Since M-Space and Q-Space share the same P axis, a distribution in M-Space can be transformed into the corresponding distribution in Q-Space by a nonlinear scaling on the Q-dimension.

On the other hand, each point in the M-Space and in turn each point in the Q-Space corresponds to a rate distribution vector. And then we can describe a layered multicast scheme as a discrete set of rate distribution vectors, each of which targets at receivers with a fixed bandwidth and packet loss ratio. We call

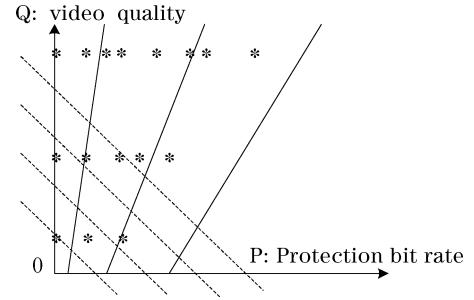


Fig. 4. Example of Q-space Coverage of multicast schemes.

such a set of points the *Q-Space Coverage* of a given layered multicast scheme.

Fig. 4 gives an example of Q-Space Coverage for several layered multicast schemes, each of which contains three layers. Each point is denoted as a star in the figure, and there are many stars in three horizontal lines, each of which corresponds to one layer. For instance, if we treat the three points on the Q axis as a set, it represents a receiver-driven layered multicast scheme proposed by McCanne *et al.* in [1]. As shown in the figure, this multicast scheme does not address packet loss issue. If we shift these three points horizontally to the right side of the Q axis, they then represent a receiver-driven layered multicast with a fixed channel coding scheme across layers. If we regard all the points as a set, it represents a receiver-driven layered multicast with FEC and pseudo-ARQ proposed by Chou *et al.* in [13]. In this example, there are three primary streams, and each one corresponds to a different number of protection streams. In general, such Q-Space Coverage figures provide a very clear view visually on how well a given multicast scheme serves heterogeneous receivers.

Changes on source bit rates and protection bit rates affect the locations of the stars, i.e., the *M-Space Coverage* of the multicast scheme. If we imagine that three horizontal lines that contain all stars actually move vertically along with time, all the moving stars represent the parameters of our SARLM scheme, where the source rate in each layer adaptively changes overtime.

In Fig. 4, each dashed line represents a constant bandwidth, which corresponds to a horizontal line in U-Space. Solid lines correspond to constant packet loss ratios. If the R-D function is linear and is the same among all the layers, they are straight lines as illustrated; otherwise, they are complex curves. In practice,

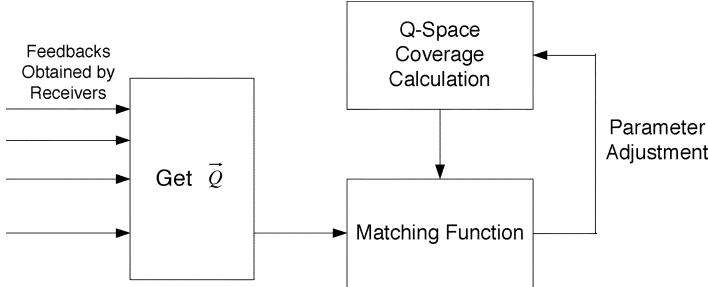


Fig. 5. Q-Space match and parameter optimization.

the protection bit rate is usually much smaller than the source data. Hence, all stars in each horizontal line have a lower bandwidth than those in a higher horizontal line.

Recall that each point in the Q-Space can also represent a receiver with a fixed bandwidth and packet loss ratio. Then a group of receivers also form a set of points in the Q-Space. Therefore, for achieving better quality on the receivers, the sender will adjust the corresponding parameters so that getting one whose coverage points in the Q-Space can better match receiver points. Theoretically, we can use an evaluation function to evaluate the matching degree, and then use the obtained result as a driving force to adjust the parameters. The whole process is illustrated in Fig. 5. This framework can be used in both the cases where we treat a user as a fixed point or a probability distribution in the Q-Space. On the receiver side in our SARLM system, it always tries to find the closest star within its current bandwidth and packet-loss-ratio constraint according to its current projection in the Q-space.

Notice that in the analysis above, we only considered bandwidth and packet loss ratio in this model. If taking the network latency into account, we can extend this model by adding another dimension.

B. Solution of the Optimization by Classification

We have proposed a theoretical framework for using matching to adjust the sender's parameters. In practice, even though the matching function is not difficult to design and compute, it is very difficult to adjust the sender's parameters so that a better matching can always be obtained. In this section we use dynamic programming and a classification approach to solve the parameter optimization problem. Specifically, we first determine the rate for each source group according to the classification of available bandwidth. Then the packet loss protection rate for each group is determined.

1) Determine the Data Rate for Each Primary Group: In the case that each receiver represented as a fixed point in the Q-Space, we project these points onto the Q axis by simply getting the Q values in the points, and get a discrete set of Q values, $Q_d(i)(i = 1, \dots, N)$, where N is the total number of receivers. In the case that each receiver is treated as a distribution, we have to take an additional step to make the projection discrete, and get a much larger set of Q values. In each case, we get a finite set of Q values $Q_d(i)(i = 1, \dots, N)$. Our optimization problem now becomes to find a set of positive value $Q_h(h = 1, \dots, H)$ such

that for any h $Q_h < Q_{h+1}$ and for all receivers the sum of video quality received is maximized, i.e.,

$$(Q_1, Q_2, \dots, Q_H) = \arg \max_{Q_1, Q_2, \dots, Q_H} \sum_{i=1}^N \tilde{Q}_d(i) \quad (6)$$

where $\tilde{Q}_d(i) = Q_h$, if $Q_h \leq Q_d(i) < Q_{h+1}$, and $\tilde{Q}_d(i) = Q_H$, if $Q_H \leq Q_d(i)$.

Conceptually $\tilde{Q}_d(i)$ stands for the video quality obtained by subscribing up to the highest layer applicable to the given receiver. And the sum of $Q_d(i)$ is the combined video quality received by all receivers given by

$$Q_{\text{all}} = \sum_{i=1}^N \tilde{Q}_d(i). \quad (7)$$

Using function (4), the optimal source bit rate in layer h equals

$$r(h) = \Phi^{-1}(Q_h) - \Phi^{-1}(Q_{h-1}). \quad (8)$$

It is easy to show that for any h , Q_h must be equal to one of $Q_d(i)(i = 1, \dots, N)$ to satisfy function (6). Thus, for all h , there are C_N^H possible values for (Q_1, Q_2, \dots, Q_H) , and the computational complexity is $O(N^H)$ assuming N is much larger than H .

An approximate solution to this optimization problem is first to classify all $Q_d(i)$ into H groups by some computational inexpensive classification methods, such as K-means, and choose the smallest $Q_d(i)$ value in each group as the corresponding Q_h . A byproduct of using K-means classification is that we can actually leave the total number of layers (H) undecided in the beginning, and place a variance constraint in each group. And then the classification scheme will come up with the value of H automatically. However, in practice we observe that dynamically changing the number of layers introduces additional complexity on the sender side and on the synchronization between the sender and the receivers. Therefore, we still recommend a scheme with fixed number of layers although it is theoretically applicable to adjust it as well.

Using the smallest value in each group coming from K-means may degrade the overall system performance due to the “outlier effect”, where a single point with a far smaller value of Q causes the whole group to pick an undesirably small value of Q_h . In this case, local adjustment can be applied. We first order $\{Q_d(i), i = 1, \dots, N\}$ from small to large. Starting from Q_1 we then try to move it to the next value in $\{\tilde{Q}_d(i), i = 1, \dots, N\}$, which is larger than the current Q_1 , and see if Q_{all} is increased. We stop moving Q_1 when such a move cannot increase Q_{all} or when Q_1 become Q_2 with the move. And then we iterate this local adjustment for all Q_h . We must acknowledge that although each step in the local adjustment always leads to increase of Q_{all} , it cannot lead Q_{all} to its maximum possible value in most cases. Therefore such local adjustment is just an attempt to avoid the outlier effect.

2) Determine the Parameters for Each Protection Group: The intuitive solution is to find the highest protection level for the receivers in each user category based on the classification. In Q-Space, this task is equivalent to finding

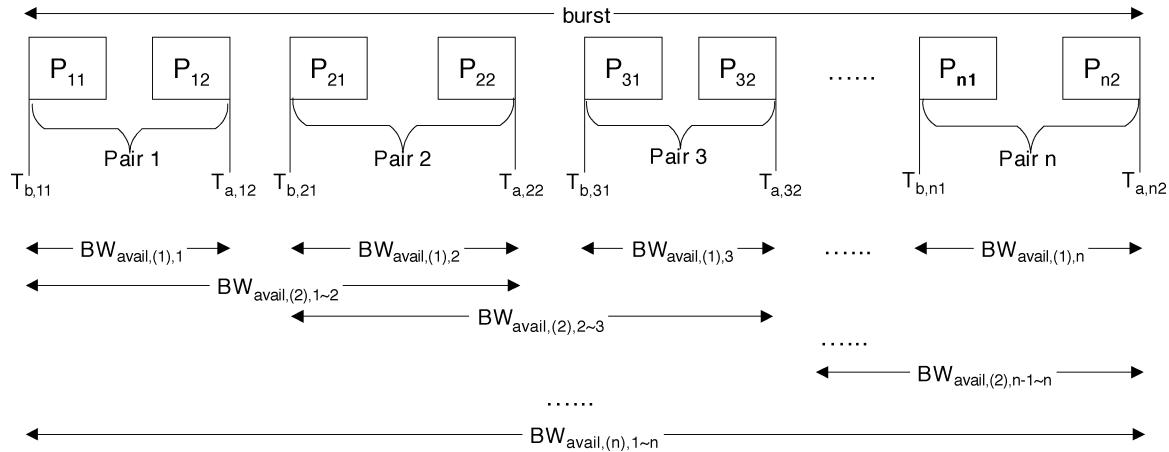


Fig. 6. Probing packets in consecutive RBPP scheme.

the maximum P in all $\overrightarrow{Q} = (Q, P)$ that belong to the same receiver category based on the classification. And then, the corresponding N_h and K_h values can be determined for the channel codec. Due to the features of FEC and pseudo-ARQ, this “overprotection” will not reduce the efficiency of network bandwidth usage. This is because as we discussed before, the protection data is divided into several independent groups so that only the receivers who need the protection packets can subscribe the corresponding protection streams dynamically.

IV. JOIN/LEAVE BEHAVIOR MANAGEMENT BASED ON BANDWIDTH ESTIMATION

In this section, we describe the policy for controlling receivers’ join/leave behaviors. To avoid the side-effect caused by blind join/leave scheme, in SARLM the receiver will make its decision based on its estimated network condition.

In unicast environment, a TCP throughput equation [7] is normally used to estimate the available bandwidth and thus support TCP-friendliness. However, such TCP throughput equation cannot estimate the available bandwidth without inducing losses. Moreover, it assumes that the network condition is stable during a large time period. On the other hand, the packet pair [20] scheme was proposed by Keshav to estimate both bottleneck bandwidth and available bandwidth. The main advantage of the packet pair scheme over TCP equation is that the packet pair does not require packet losses. To estimate available bandwidth in packet pair, the router is required to support fair scheduler [3]. In the following, we assume that every router is multicast capable and implements a fair scheduler. This assumption is practically feasible because it is known that RBPP [20] scheme can measure the available bandwidth by arrival times of packets, which is suited for multicast scenario. Packet-pair relies on the fact that if two packets are queued next to each other at the bottleneck link, one is t second apart from the other, t can be calculated as

$$t = \frac{\text{PacketSize}}{\text{BW}_{\text{calculated}}} \quad (9)$$

where PacketSize is the same size for all the packets, and $\text{BW}_{\text{calculated}}$ is the calculated bandwidth.

Considering the multicast application, we introduce a modified RBPP algorithm, called consecutive RBPP, to measure the available bandwidth. In our approach, we send n probing packet-pairs back-to-back, which are denoted as a burst, for all the multicast groups in every certain period of time (see Fig. 6). In our work, n is set to 8. Taking the channel efficiency into account, we do not use the particular control packets for probing. Instead, only data packets are used. All the multicast groups of the same session are considered as a single flow. We set a flag in the header of the first packet of a pair so as to make it easy for a receiver to identify the beginning of a packet-pair burst.

For each probing burst, we calculate several available bandwidth instances according to the following steps.

- 1) Upon receiving the i th probing pair, we record the time that starts to receive the first packet $T_{b,i1}$ and the time that the second packet arrives $T_{a,i2}$.
- 2) The available bandwidth instance that is obtained based on a single packet-pair can be calculated as

$$\text{BW}_{\text{avail},(1),i} = \frac{\text{PSize}_{P_{i1}} + \text{PSize}_{P_{i2}}}{T_{a,i2} - T_{b,i1}} \quad (10)$$

where $\text{PSize}_{P_{i1}}$ and $\text{PSize}_{P_{i2}}$ are the sizes of the first and the second packet of the i th packet-pair, respectively.

- 3) The available bandwidth instance that is obtained based on two consecutive packet-pairs can be calculated as

$$\text{BW}_{\text{avail},(2),i-1 \sim i} = \frac{\text{PSize}_{P_{(i-1)1}} + \text{PSize}_{P_{(i-1)2}} + \text{PSize}_{P_{i1}} + \text{PSize}_{P_{i2}}}{T_{a,i2} - T_{b,(i-1)1}}. \quad (11)$$

- 4) Similarly, we can obtain the available bandwidth instances based on different (up to n) consecutive packet-pairs, where

$$\text{BW}_{\text{avail},(n),1 \sim n} = \frac{\sum_{i=1}^n (\text{PSize}_{P_{i1}} + \text{PSize}_{P_{i2}})}{T_{a,i2} - T_{b,i1}}. \quad (12)$$

Once obtaining the instances of the available bandwidth, the main challenge becomes how to filter out the noise caused by time compressed and extended packets. In our consecutive RBPP scheme, we use the kernel density estimator to overcome the time compressed and extended problems. In order to use it, we first define a kernel function $K(t)$ with the property

$$\int_{-\infty}^{+\infty} K(x)dx = 1. \quad (13)$$

Here, the kernel function we use is

$$K(x) = \begin{cases} 1+x, & -1 < x \leq 0 \\ 1-x, & 0 < x < 1 \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Then the density at any point x is defined as

$$\frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - x_i}{h}\right) \quad (15)$$

where h is the kernel width, which controls the smoothness of the density function, n is the number of points within h of x , and x_i is the i th point, which is the individual estimated available bandwidth value. This density function has the desirable property that gives larger weight to samples closer to the point at which we want to estimate density. It is simple and fast to compute. The final available bandwidth calculated for each receiver $BW_{\text{available}}$ is the one with the largest density.

As stated above, in our SARLM scheme, each receiver controls its join/leave behaviors based on the corresponding estimated bandwidth. Let $BW_{\text{available}}$ be the available bandwidth estimated by the packet-pair method and BW_n be the current bandwidth obtained from n cumulative layers, i.e.,

$$BW_n = \sum_{i=1}^n \text{Rate}_i \quad (16)$$

where Rate_i is the sending rate of the i th layer. The join and leave policy on the receiver side can be described as follows.

- *Leave Policy*
Each receiver drops a layer instantly when its estimated bandwidth is lower than the cumulative bandwidth of the current subscribed layer (i.e., $BW_{\text{available}} < BW_n$).
- *Join Policy*
The receiver joins a new group if the minimum estimated bandwidth is greater than the cumulative bandwidth of the current subscribed group for the last certain period C , which equals 1 second in our work (i.e., $\min(BW_{\text{available}}) > BW_n$).

In summary, we drop a layer each time once our estimated bandwidth $BW_{\text{available}}$ lower than the current layer subscription BW_n , but we add layers according to the minimum $BW_{\text{available}}$ value received during the period C .

As described in [11], synchronization among all the receivers in the same downstream bottleneck link

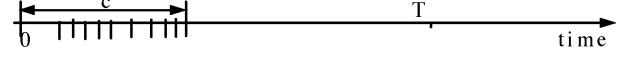


Fig. 7. Gamma-distributed timer setting.

is important for a multicast pruning mechanism. Our scheme gets both join and leave synchronization in the following:

- all the multicast receivers behind the same bottleneck will roughly receive the same probing packet pair at the same time because of the queue build-up. Then they drop layers at the same time when the estimated bandwidth is lower than the current subscribed group;
- if the receivers do not join the session at the same time, a late joiner will be resynchronized when its first drop occurs ($BW_{\text{available}} < BW_n$). By this way, the clock drift is also avoided.

V. SCALABLE FEEDBACK GENERATION MECHANISM

As is known, soliciting information from receivers in a multicast group of indeterminate size may create a so-called feedback implosion problem. In this section, we describe a mechanism for eliciting feedback information from the receivers in a multicast group to avoid feedback implosion.

In order to achieve higher suppression ratio and low feedback latency, we propose a probabilistic feedback approach for multicast based on gamma-distributed timers. The motivation to use two-parameter gamma-distributed timer instead of traditional exponentially distributed timer [11] is based on the fact that gamma-distributed timer has faster convergence speed suppression ratio. In our approach, sender uses probing packet to solicit feedback packets within its expected number in a scalable manner, and estimates the number of receivers based on the receiver's feedback. The receiver determines whether to send out the feedback message (FBM) based on its generated gamma-distributed timer and the delay between the sender and the receiver. In this way, feedback suppression can be achieved. Specifically, the density of our truncated gamma distributed timer is defined as

$$f_{Z_i}(z_i) = \begin{cases} \frac{1}{(e^\lambda - 1)} \times \alpha \times \frac{\lambda}{T^\alpha} \times z_i^{\alpha-1} \times e^{\frac{\lambda}{T^\alpha} z_i^\alpha} & 0 \leq z_i \leq T \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

where T is a fixed interval of our feedback, and λ and α are factors related to the number of receivers. The sender updates receivers' distribution after a certain interval T .

Our proposed scalable feedback scheme can be divided into the following four steps

- 1) The sender multicasts a feedback-request message with parameter set (I, λ, α, T) to all the receivers, where I is the identification number of the feedback interval.
- 2) Receiver i generates a gamma-distributed random timer $z_i \sim [0, T]$ upon receiving the request message (I, λ, α, T) (See Fig. 7). To suppress the other receivers' feedbacks, only the receivers which get the timer between $(0, 0 + c)$ can send feedback, where c is the delay between the receiver and the sender. When the timer z_i expires, receiver i sends feedback

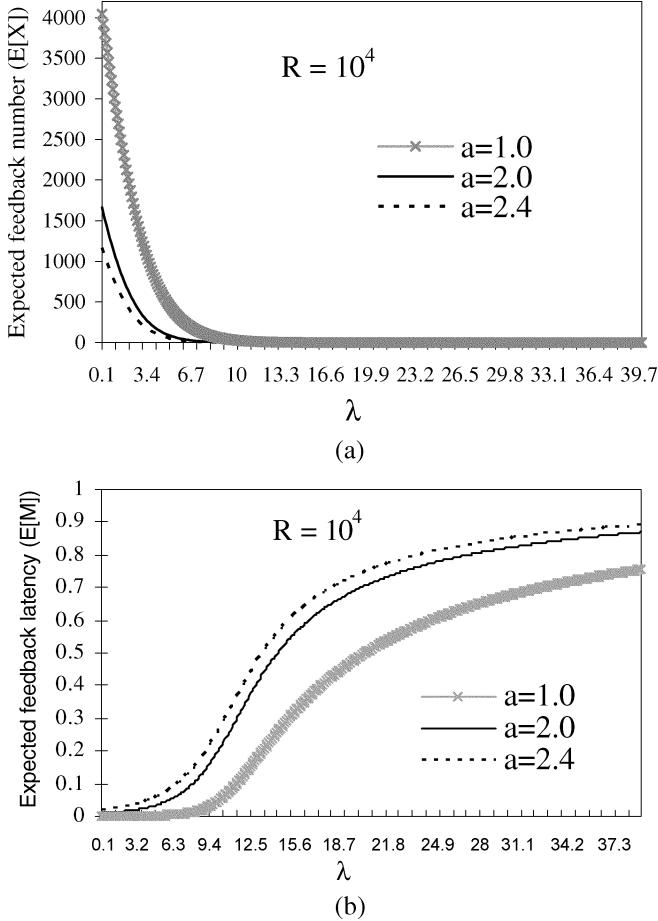


Fig. 8. Performance of scalable feedback scheme. (a) Suppression performance of gamma-distributed timer. (b) Performance of feedback latency.

messages, FBM_{*i*} (z_i, T_A), to the sender, where T_λ is estimated available bandwidth obtained in Section IV.

- 3) Upon receiving the FBM_{*i*}s, the sender estimates the number of receivers given by

$$\hat{R} = \frac{X}{F_Z(z)} \quad (18)$$

where X is the number of received feedback messages, and z is obtained from the receivers' FBM_{*i*}s. To achieve fast convergence and a reasonably smooth estimation, an exponential weighted moving average is used for updating \hat{R} as follows.

$$\hat{R}_n = \begin{cases} 1, & n = 1 \\ (1 - \beta)\hat{R}_n + \beta\hat{R}_{n-1}, & n > 1 \end{cases} \quad (19)$$

where \hat{R}_n denotes the current estimated receiver number, \hat{R}_{n-1} denotes the estimated received number in last time, and β is a weighting parameter that is set to 0.8 in our work.

- 4) The sender calculates the new λ and α for the next feedback request based on \hat{R} , required feedback latency, and the expected number of FBM_{*i*}s.

As in [11], we use Bernoulli random variables x_i to indicate whether an FBM_{*i*} from receiver i is sent ($x_i = 1$) or not ($x_i =$

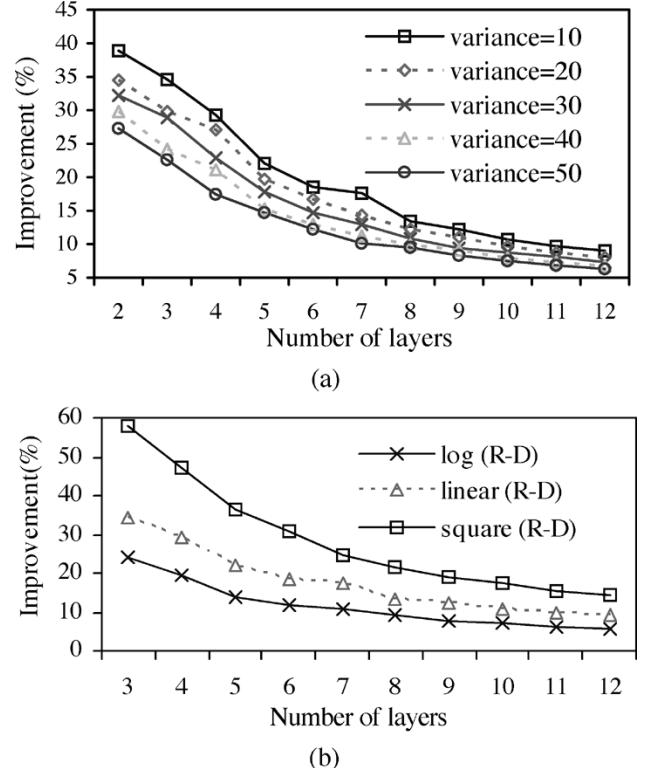


Fig. 9. Performance evaluation in different cases. (a) With different bandwidth variances. (b) With different R-D relations.

0), assuming each receiver i sends feedback independently between $[0, c]$, then, the expected number of FBM_{*i*}s at the sender can be derived as

$$\begin{aligned} E[X] &= \sum_{i=1}^{\hat{R}} E[x_i] \\ &= \hat{R}P(x_i = 1) \\ &= \hat{R} \int_0^c \left[\frac{1}{(e^\lambda - 1)} \cdot \alpha \frac{\lambda}{T^\alpha} x_i^{\alpha-1} e^{\frac{\lambda}{T^\alpha} x_i^\alpha} \right] dx_i \\ &= \hat{R} \frac{e^{\lambda(\frac{c}{T})^\alpha} - 1}{e^\lambda - 1}. \end{aligned} \quad (20)$$

Denote a desired number of feedback messages as N . We can obtain such an N by tuning the parameters λ and α . By solving $E(X) = N$, mathematically we get the expression of λ and α as follows:

$$\lambda = 1.1 \times \ln \hat{R} + 0.8, \quad (21)$$

$$\alpha = \frac{\ln \left[\frac{1}{\lambda} \ln \left[\frac{N(e^\lambda - 1) + \hat{R}}{\hat{R}} \right] \right]}{\ln \left(\frac{c}{T} \right)} \quad (22)$$

where it controls the feedback bandwidth and suppress feedback implosion. The detailed derivation can be found in Appendix.

Fig. 8(a) shows the suppression performance of a gamma-distributed timer with $T = 10c$. It can be seen that through dynamically tuning λ and α on the sender side, we can achieve a small number of expected feedback number, (e.g., $E[X] < 20$) for large number of receivers (e.g., up to 10^4 receivers). In this way, the feedback implosion can be effectively avoided. With

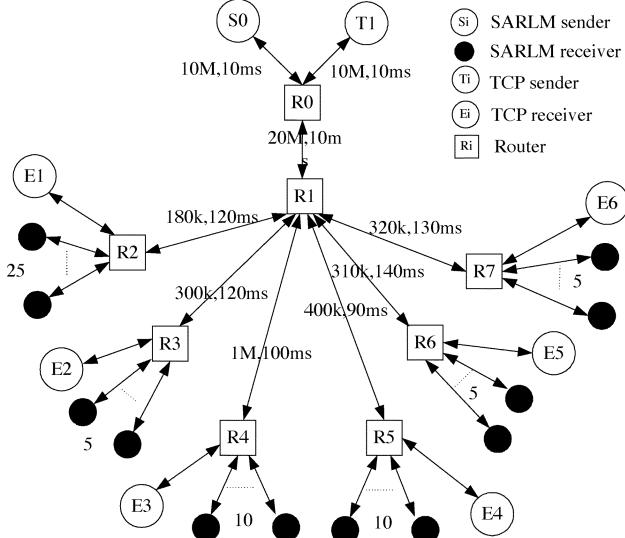


Fig. 10. Simulation Topology 1.

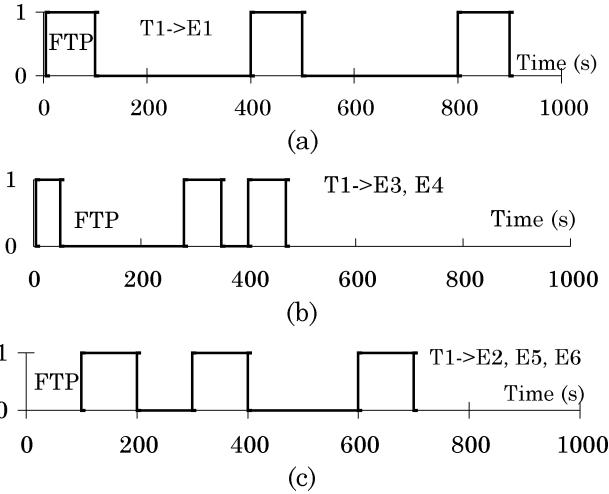


Fig. 11. Background traffic in Topology 1. (a) Background traffic from T1 to E1. (b) Background traffic from T1 to E3 and E4. (c) Background traffic from T1 to E2, E5, and E6.

this scalable feedback scheme, the feedback latency depends on λ and α , which are illustrated in Fig. 8(b).

In summary, by introducing our designed gamma-distributed timer, only a small number of feedbacks are needed to be sent to the sender when the total number of users is very large. Meanwhile, the feedback latency caused by our scheme is rather short. Thus, the feedback implosion can be efficiently avoided.

VI. SIMULATION RESULTS

In this section, we implement our proposed architecture and algorithms. We use network simulator NS2 [21] for our simulation. The purpose of our simulation is to investigate 1) the performance of sender adaptiveness of our SARLM scheme and 2) the performance of our SARLM scheme with layered video codec PFGS.

A. Performance Evaluation of Sender Adaptiveness of SARLM

We assume that there are 100 receivers in total, which belong to five bandwidth categories, ranging from 64–2048 kb/s.

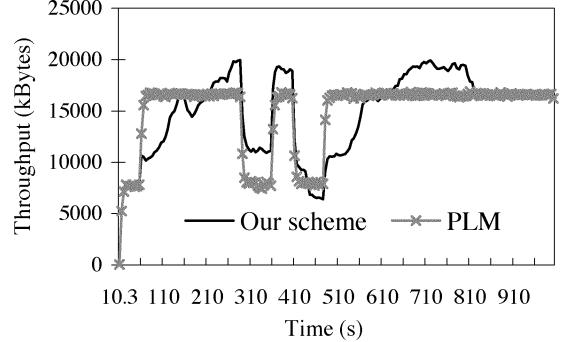


Fig. 12. Total throughput comparison of all receivers in Topology 1.

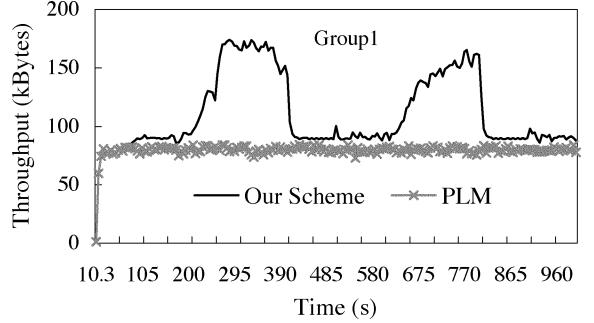


Fig. 13. Throughput comparison of receivers in Group 1 in Topology 1.

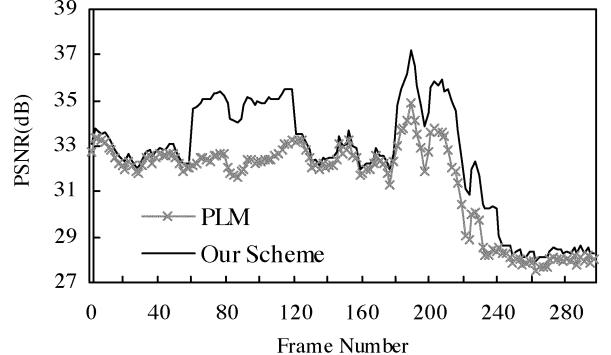


Fig. 14. PSNR comparison of the first receiver of Group1 in Topology 1.

In each category, the numbers of receivers are 5, 5, 20, 30, and 40, respectively. The bandwidth probability density functions of all receivers follow the same Gaussian distribution. The mean values of the distribution vary from time to time, although the variances of the distribution are fixed. In each simulation, we change the mean of the Gaussian distribution randomly and the instant bandwidth of each receiver changes accordingly. Rather than evaluating performance gain using the FEC and pseudo-AQR, which has been investigated in [13], we focus on examining the performance gain in our SARLM with sender-adaptiveness. We compare the overall performance (the sum of received video qualities of all receivers and of all 500 time units per simulation) with the one when the sending data rates are consistently and equally distributed among the range from 64–2048 kb/s. The receiver-driven approach is used in our simulation and the number of multicast groups is the same as our sender-adaptive case. “K-means” classification is used without further local adjustments.

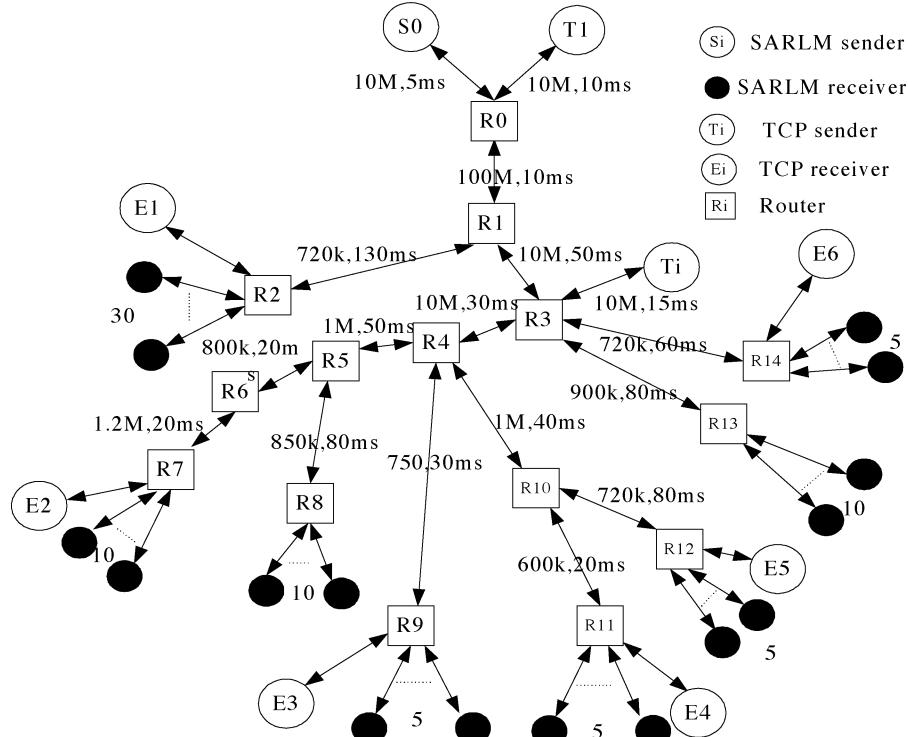


Fig. 15. Simulation Topology 2.

Fig. 9 shows performance evaluation in different cases. In Fig. 9(a), different curves correspond to different variance values of the Gaussian distributions. The variance reflects the degree of the bandwidth fluctuation in each bandwidth category. It can be seen from Fig. 9(a) that the fewer the number of layers, the more the improvements. This is because when the number of the layers increases, the rates of each layer can fit individual receivers better even if the rates are evenly located. In addition, the larger the bandwidth fluctuation of each user category (the closer the overall user distribution to a uniform shape) is, the less the improvements can be obtained. In Fig. 9(b), different curves correspond to different Rate-Distortion relations. It can be seen that the performance improvements are R-D dependent. From Fig. 9(a) and (b), we can see a significant improvement in the small layer-number case. On the other hand, the computational cost of the sender optimization is very low.

B. Performance Evaluation of SARLM With the PFGS Codec

To investigate the effectiveness of our proposed SARLM scheme, in this simulation, we compare the performance of two schemes with PFGS codec as follows: 1) our proposed SARLM and 2) PLM proposed by Legout [3].

We use network simulator (NS-2) in the simulation [12]. The multicast routing protocol used is distance vector multicast routing protocol (DVMRP). All the queues are fairly queued and each router has a queue size of 20 packets. The bandwidth of all the local area access is 10 M. We simulate a dynamic network condition by generating several FTP flows as background traffic. In this simulation, the number of expected feedback messages from the receivers is set to 35. Meanwhile, the one-way delay parameter c is set to 200 ms and update interval T is set to 10 s. The testing video sequence *Foreman* is coded into three layers at 10 f/s in CIF with PFGS codec.

We use two network topology (Fig. 10, Fig. 18) and different background traffic (Fig. 11, Fig. 19) in the following. The first topology, “pure centric topology”, consists of one SARLM source, sixty SARLM receivers, and six varying background traffic. The initial bit rate of each layer in Topology 1 is set to 80, 300, and 580 kb/s, respectively.

Fig. 11 displays the background traffic burst in Topology 1. In this Fig. 1 stands for FTP flow on, and 0 stands for FTP flow off.

Fig. 12 shows the throughput comparison of all receivers between our scheme and PLM in Topology 1. We compute the average throughput comparison ratio for receivers by receiver-number-weighted throughput comparison between our scheme and PLM. From this figure, we can see that the mean throughput of our scheme is 25% higher than that of PLM.

Fig. 13 shows the throughput comparison between our scheme and PLM for receivers of Group 1 in Topology 1. Because the initial rate of video stream is, respectively, 80, 380, and 960 kb/s, the receivers of Group 1 under router R2 of PLM can only join and stay at layer 1 of about 80 kb/s and get low quality of video because of fixed sending rate of each layer. It can be seen that our scheme can achieve higher throughput with different network available bandwidth.

Fig. 14 depicts the PSNR comparison between SARLM and PLM of the first receiver of Group 1. It can be seen that SARLM achieves higher video quality than PLM. More specifically, the average PSNR obtained by this receiver using SARLM is 32.55, while the average PSNR obtained by this receiver using PLM is 31.40.

The second topology, “complex topology”, that is shown in Fig. 15, consists of one sender, 80 receivers, and six varying background traffic flows in this topology. The initial bit rate of

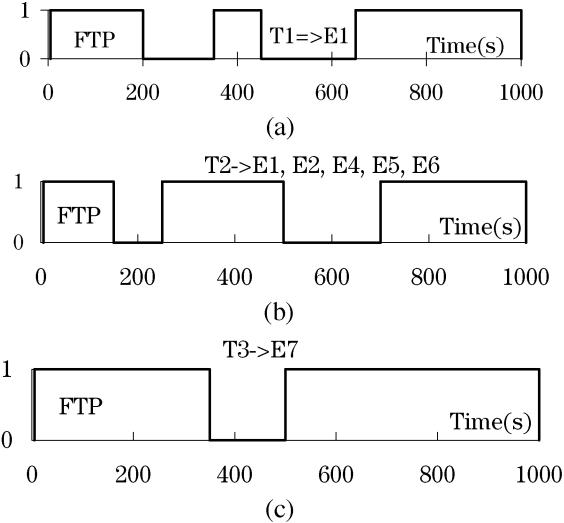


Fig. 16. Background traffic for Topology 2. (a) Background traffic from T1 to E1. (b) Background traffic from T2 to E1, E2, E4, E5, and E6. (c) Background traffic from T3 to E7.

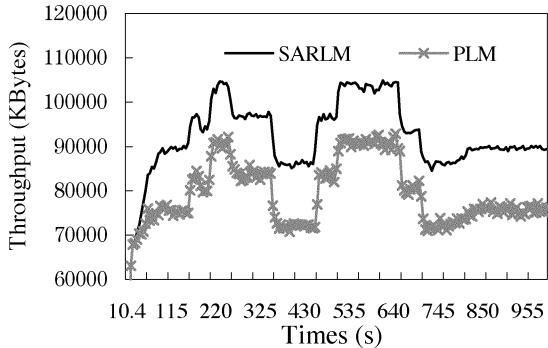


Fig. 17. Throughput comparisons of all the receivers in Topology 2.

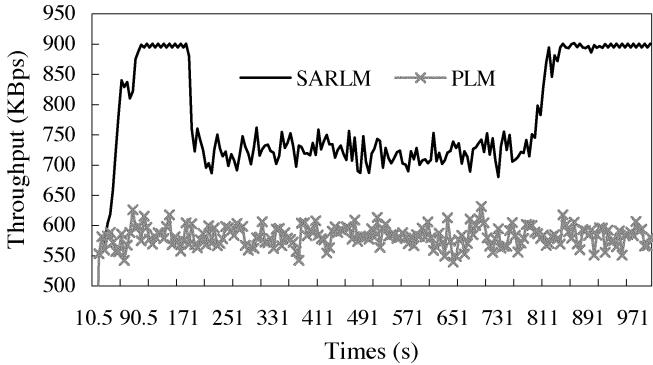


Fig. 18. Throughput comparison of receivers in Group 7.

each layer in Topology 2 is set to 192, 384, and 1024 kb/s, respectively.

Fig. 16 displays the background traffic burst in Topology 2. Fig. 17 shows the throughput comparison of all receivers between our scheme and PLM in Topology 2. From this figure, we can see that the average throughput of our scheme is 40% higher than PLM. This is because the varying network bandwidth can be estimated in SARLM, and the sending rate of each multicast group is adjusted according to the status of each receiver.

Fig. 18 shows the throughput comparison between SARLM and PLM for receivers of Group 7. It can be seen that SARLM

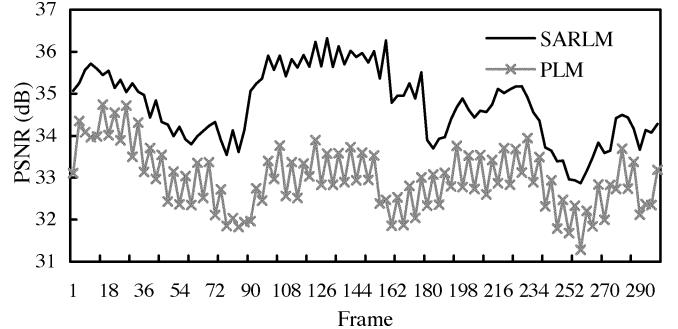


Fig. 19. PSNR comparison of the first receiver in Group 7.

obtains higher throughput than PLM. Notice that there is no background FTP traffic during 140–200 s and 850–1000 s. This bandwidth variation can be measured by SARLM, and the sender adjusts the parameters of the multicast group accordingly. Thus, rather higher throughput can be obtained during those periods of time.

Fig. 19 depicts the PSNR comparison between SARLM and PLM for the first receiver of Group 7. It can be seen that SARLM achieves higher video quality than PLM. More specifically, the average PSNR obtained by this receiver using SARLM is 33.58, while the average PSNR obtained by this receiver using PLM is 32.08.

VII. CONCLUSIONS AND DISCUSSIONS

This paper presents a novel architecture, namely SARLM, for layered video multicast. This architecture is “a loosely closed loop” for the multicast system with the joint control from both sender and receiver.

A mathematical model for layered multicast to receivers with heterogeneous bandwidth and packet loss ratio is introduced on the sender side, and under this model the SARLM scheme can be modeled as a discrete optimization problem. In the SARLM architecture, the different R-D functions and unequal error protections for different layers have been taken into account. Moreover, the sender adjusts its multicast strategy based on analysis of the feedbacks from the receiver side to achieve a better overall system performance.

Each receiver makes autonomous decision on joining or leaving multicast groups according to its estimated network condition. Meanwhile, receivers send feedbacks that contain information about the network conditions in a scalable way to the sender. We demonstrate that our proposed gamma-distributed timer based feedback mechanism has a faster convergence speed than the existing exponential distributed timer-based feedback mechanisms.

The SARLM architecture resides in the application layer, and does not have any special requirement to the network infrastructure. Simulations on different network topologies and network conditions have shown that our system has better adaptive capability to network dynamics over the existing technologies.

ACKNOWLEDGMENT

The authors would like to thank Prof. L. Gao from University of Massachusetts, Amherst, for some of the discussions

while she visited Microsoft Research China, Beijing and Prof. J. Zhang from University of Wisconsin, Milwaukee, for reading this draft.

REFERENCES

- [1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM*, 1996, pp. 117–130.
- [2] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE INFOCOM*, 1998, pp. 996–1003.
- [3] A. Legout, "PLM: Fast convergence for cumulative layered multicast transmission scheme," in *Proc. ACM SIGMETRICS*, 2000, pp. 13–22.
- [4] J. C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *Proc. ACM SIGCOMM*, Aug. 1994, pp. 58–67.
- [5] B. J. Vickers, C. Albuquerque, and T. Suda, "Adaptive multicast of multi-layered video: rate-based and credit-based approaches," in *Proc. IEEE INFOCOM*, 1998, pp. 1073–1083.
- [6] S. Y. Cheung, M. H. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM*, 1996, pp. 553–560.
- [7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. SIGCOMM*, Aug. 1998, pp. 303–314.
- [8] V. Jacobson. (1997) Pathchar. Mesurement tool. [Online]. Available: <ftp://ftp.ee.lbl.gov/pathchar/>
- [9] V. Paxson, "Measurements and analysis of end-to-end internet dynamics," Ph.D. dissertation, Univ. California, Berkeley, Apr. 1997.
- [10] M. Handley and S. Floyd, "Strawman specification for TCP Friendly (Reliable) Multicast Congestion Control (TFMCC)," presented at the *Reliable Multicast Research Group Meeting*, Arlington, VA, Dec. 1998.
- [11] J. Nonnenmacher and E. W. Biersack, "Scalable feedback for large groups," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 375–386, Jun. 1999.
- [12] J. C. Pasquale, G. C. Polyzos, G. V. Xylomenos, and V. P. Kompella, "The multimedia multicasting problem," UCSD/CSE, Tech. Rep. no. CS93-313, Feb. 1996.
- [13] P. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "FEC and pseudo-ARQ for receiver-driven layered multicast of audio and video," in *IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 440–449.
- [14] Study of a new approach to improve FGS video coding efficiency, ISO/IEC JTC1/SC29/WG11, MPEG99/m5583, Dec. 1999.
- [15] Q. Ni, Q. Zhang, and W. Zhu, "SARLM: Sender-adaptive & Receiver-driven Layered Multicasting for Scalable Video," in *Proc. IEEE ICME*, Aug. 2001, pp. 757–760.
- [16] K. Lai and M. Baker, "Measuring bandwidth," in *Proc. IEEE INFOCOM*, 1999, pp. 235–245.
- [17] T. Turletti, S. F. Paris, and J. Bolot, "Experiments with a layered transmission scheme over internet," INRIA Research Rep. no. 3296, Nov. 1997.
- [18] Strawman specification for TCP friendly (reliable) multicast congestion control (TFMCC), M. Handley and S. Floyd. [Online]. Available: <http://www.east.isi.edu/RMRG/newindex.html>
- [19] Q. Guo, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "A sender-adaptive and receiver-driven layered multicast scheme for video over internet," in *Proc. IEEE ISCAS*, 2001, pp. 141–144.
- [20] (1995, Feb.). [Online]. Available: <http://www.cs.cornell.edu/ske-shav/doc/94/2-17.ps>
- [21] The Network Simulator (NS). [Online]. Available: <http://www-mash.cs.berkeley.edu/ns/ns.html>
- [22] G. Cheung and A. Zakhor, "Bit allocation for joint-source channel coding of scalable video," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 340–356, Mar. 2000.
- [23] B. J. Vickers, C. Albuquerque, and T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," Univ. California, Irvine, Tech. Rep. no ICS-TR 99-45, Jun. 1999.
- [24] S. Paul, *Multicasting on the Internet and Its Applications*. Norwell, MA: Kluwer, 1998.
- [25] D. Rubenstein, J. Kurose, and D. Towsley, "The impact of multicast layering on network fairness," in *Proc. ACM SIGCOMM*, 1999, pp. 27–38.
- [26] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for video streaming over the internet, special issue on multimedia over IP," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 339–355, Sep. 2001.
- [27] F. L. Leannec, J. Vieron, X. Henocq, and C. Guillemont, "Hybrid sender and receiver driven rate control in multicast layered video transmission," in *Proc. IEEE ICIP*, Sep. 2000, pp. 532–535.
- [28] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," in *Proc. ACM SIGCOMM*, Cannes, France, 1997, pp. 289–300.
- [29] I. Rhee and S. Joshi, "Error recovery for interactive video transmission over the Internet," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1033–1049, Jun. 2000.
- [30] L. Rizzo and L. Vicisano, "RMDP: an FEC-based reliable multicast protocol for wireless environments," *ACM Mobile Comp. Commun. Rev.*, vol. 2, no. 2, pp. 23–31, Apr. 1998.
- [31] W. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 373–387, Mar. 2001.



Qian Zhang (M'00–SM'04) received the B.S., M.S., and Ph.D. degrees from Wuhan unhan, China, in 1994, 1996, and 1999, respectively, all in computer science.

She joined Microsoft Research, Asia, Beijing, China, in July 1999, as an Associate Researcher in the Internet Media Group. Now, she is the Research Manager of the Wireless and Networking Group. She has published about 90 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is the inventor of about 30 pending patents. Her current research interest includes multimedia delivery over wireless, Internet, next-generation wireless networks, P2P network/ad hoc network. She also participated many activities in the IETF ROHC (Robust Header Compression) WG group for TCP/IP header compression.

Dr. Zhang is a member of the Visual Signal Processing and Communication Technical Committee and the Multimedia System and Application Technical Committee of the IEEE Circuits and Systems Society. She is now the Vice Chair of Membership Development Committee for IEEE Asia Pacific Operations Centre of IEEE Communications Society. She is also a member and Chair of QoSIG of the Multimedia Communication Technical Committee of the IEEE Communications Society. She is now serving as Associate Editor of *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*. She is also serving as Guest Editor for a special issue on wireless video in *IEEE Wireless Communication Magazine*. She has recently received TR 100 (MIT Technology Review) world's top young innovator award. She was also elected as Best Asia Pacific Young Researcher Award by IEEE Communications Society.



Quji Guo received the B.S. degree from the Special Class for the Gifted Young, University of Science and Technology of China, Beijing, China, and the M.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign. He is currently pursuing the MBA degree at Stanford University, Stanford, CA.

He worked for Microsoft Research Asia, Beijing, China, as a summer intern, and Microsoft Corporation, Redman, SC, as a Software Design Engineer.



Qiang Ni received the B.S., M.S., and the Ph.D. degrees from Hua Zhong University of Science and Technology (HUST), Wuhan, China in 1993, 1996, and 1999 respectively.

From 1999 to 2001, he was a Postdoctoral Research Fellow in the Multimedia and Wireless Communication Laboratory, Department of Electronics and Information Engineering, HUST. He spent his visiting research internship at Microsoft Research Asia, Beijing, China, during the year of 2000. In 2001, he joined INRIA Sophia Antipolis, where he is now a Staff Expert Research Engineer at INRIA, Planète group, France. Since 2002 he has been a voting member for the IEEE 802.11 wireless LAN Standard Working Group. His current research interests include QoS enhancements for IEEE 802.11/802.15 wireless LAN/PAN, next-generation high-speed wireless networks, vertical handover and mobility management in mobile networks, and multimedia transmission over hybrid wired/wireless networks.



Wenwu Zhu (S'92–M'97–SM'01) received the B.E. and M.E. degrees from the National University of Science and Technology, Wuhan, China, in 1985 and 1988, respectively, the M.S. degree from Illinois Institute of Technology, Chicago, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1993 and 1996, respectively, all in electrical engineering.

From August 1988 to December 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and Chinese Academy of Sciences (the Institute of Electronics), Beijing, China. He has been working at Intel Communication Technology Lab, China, as Director since September 2004. Prior to his current post, he was with Microsoft Research Asia from 1999 to 2004 as a Research Manager in the Wireless and Networking Group and at Bell Labs, Lucent Technologies, Whippny, Holmdel, and Murray Hill, NJ, as a Member of Technical Staff during 1996–1999. While he was with Bell Labs, he performed research and development in the areas of Internet video, video conferencing, and video streaming over IP networks. He has published over 150 refereed papers in international leading journals and key conferences in the areas of wireless/Internet video delivery, wireless/Internet multimedia communications and networking, and has contributed to the IETF ROHC WG draft on robust TCP/IP header compression over wireless links. He is the inventor of more than a dozen pending patents. His current research interest is in the area of wireless/Internet multimedia delivery and multimedia networking.

Dr. Zhu has served as Guest Editors, respectively, for Special Issues on Streaming Video and Wireless Video in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He also serves a Guest Editor for Special Issue on Advanced Mobility Management and QoS Protocols for Wireless Internet in IEEE JOURNAL SELECTED AREAS IN COMMUNICATIONS. Currently, he is serving a Guest Editor for special issue on Advanced Video Coding and Delivery in the PROCEEDINGS OF THE IEEE. He received the Best Paper Award in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2001. Currently, he serves as Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE TRANSACTIONS ON MULTIMEDIA, respectively. He is now chairman of IEEE Circuits and System Society, Beijing Chapter. He is a member of Eta Kappa Nu, the Visual Signal Processing and Communication Technical Committee, and the Multimedia System and Application Technical Committee of the IEEE Circuits and Systems Society. He is also a member of the Multimedia Communication Technical Committee of the IEEE Communications Society.



Ya-Qin Zhang (S'87–M'90–SM'93–F'98) received the B.S. and M.S. degrees from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1983 and 1985, and the Ph.D. degree from George Washington University, Washington, DC, in 1989, all in electrical engineering.

He is currently the Corporate Vice President of Microsoft, responsible for mobility and embedded system products. He was the Managing Director of Microsoft Research Asia, Beijing, China, from 1999 to 2004. Previously, he was the Director of the Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center and RCA Laboratories). Prior to that, he was with GTE Laboratories Inc., Waltham, MA, from 1989 to 1994. He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He has authored and coauthored over 200 refereed papers in leading international conferences and journals, and has been granted over 40 U.S. patents in digital video, Internet, multimedia, wireless, and satellite communications. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards. He serves on the Board of Directors of five high-tech IT companies and has been a key contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Zhang served as the Editor-In-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from July 1997 to July 1999. He was the Chairman of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems (CAS) Society. He serves on the editorial boards of seven other professional journals and over a dozen conference committees. He has received numerous awards, including several industry technical achievement awards and IEEE awards, such as the CAS Jubilee Golden Medal. He was named Research Engineer of the Year in 1998 by the Central Jersey Engineering Council for his “leadership and invention in communications technology, which has enabled dramatic advances in digital video compression and manipulation for broadcast and interactive television and networking applications.” He recently received The Outstanding Young Electrical Engineer of 1998 award.