A Proxy-Assisted Adaptation Framework for Object Video Multicasting

Jiangchuan Liu, *Member, IEEE*, Bo Li, *Senior Member, IEEE*, Huai-Rong Shao, *Member, IEEE*, Wenwu Zhu, *Senior Member, IEEE*, and Ya-Qin Zhang, *Fellow, IEEE*

***Abstract*—Video multicast is a challenging problem due to the heterogeneous and best-effort nature of the Internet. In this paper, we present a novel video multicast framework that exploits the potential of object scalability offered by MPEG-4. Specifically, we introduce the concept of object transmission proxy (OTP), which filters incoming streams using object-based bandwidth adaptation to meet dynamic network conditions. Multiple OPTs can form an overlay network that interconnects diverse multicast islands with semi-uniform demands within each single island. We concur with the wisdom that an application best knows the utility of its data. Hence, the bandwidth-adaptation algorithm for the OTPs adaptively allocates bandwidth among video objects according to their respective utilities and then performs application-level filtering based on an effective stream classification and packetization scheme. Extensive simulation results demonstrate that our framework has substantial performance improvement over conventional bandwidth-adaptation schemes. It is particularly suitable for object-based video multicasting where the objects are of different importance.**

***Index Terms*—Object scalability, object transmission proxy, packetization, utility function, video filtering, video multicasting.**

I. Introduction

MULTICAST provides an efficient vehicle for delivering data from a source to a group of receivers. Due to the multireceiver nature of video programs, real-time video distribution has become one of the most important multicast applications. It is also a building block of many existing and emerging Internet applications, such as WebTV and distance learning and thus has recently received a great deal of attention [1].

The heterogeneous and best-effort nature of the Internet, however, makes video multicast a challenging undertaking. Though media scaling methods, like *layering* [2], *stream replication* [3], and *transcoding* [4], have been suggested as solutions to this problem, they often suffer from the coarse adaptation granularity or high computation overheads. Such problems persist in recent proposals on application-layer or overlay multicast [13]. Furthermore, they are generally designed for frame-based video transmission. The emerging MPEG-4 standard supports separate encoding and decoding of atomic units in a scene, called video objects (VOs) [5]. This enables a new type of scaling scheme, namely, *content scalability* or *object scalability*. Such scalability is different from the conventional frame-based layered coding, as objects are independent or partially independent, and yet different from stream replication, as objects are nonoverlapping. Hence, it provides a more flexible and efficient framework for bandwidth adaptation, which has yet to be explored in video multicasting.

In this paper, we propose a novel framework specifically designed for object-based video multicasting. This framework takes advantages of existing approaches and largely mitigates their deficiencies. In particular, we have devised a set of novel mechanisms that leverage the MPEG-4 object scalability. First, we introduce the concept of object transmission proxy (OTP). An OTP filters incoming streams using object-based bandwidth adaptation to meet dynamic network conditions. Multiple OPTs form an overlay network that interconnects diverse multicast islands with semi-uniform demands within each single island. Second, we propose a smart classification, prioritization and packetization scheme for object video streams. This scheme enhances the error resilience of the streams and enables finer-grained rate adaptation. We concur with the conventional wisdom that an application best knows the utility of its data. In our framework, the bandwidth-adaptation algorithm of the OTPs allocates available bandwidth among the video objects according to their respective utilities and then performs application-level packet filtering based on the introduced stream classification and packetization scheme.

The performance of the framework is evaluated through extensive simulations. The results demonstrated that it achieves better bandwidth fairness and perceived video quality than conventional frame-based transmission. The improvement is particularly noticeable when the video objects have uneven utilities, i.e., with different levels of importance to the receivers. Furthermore, its computation overhead is kept at a low level.

The rest of this paper is organized as follows. Section II introduces some related work. Section III presents an overview of the OTP-based adaptation framework. The bandwidth adaptation mechanisms for the OTPs are described in Section IV.

Manuscript received July 7, 2001; revised June 23, 2003. The work of J. Liu was supported in part by a Microsoft Research Fellowship. The work of B. Li was supported in part by grants from Research Grants Council (RGC) under contracts HKUST 6196/02E and HKUST6204/03E, a grant from NSFC/RGC under the contract N_HKUST605/02, and a grant from Microsoft Research under the contract MCCL02/03.EG01. This paper was recommended by Associate Editor C. W. Chen.

J. Liu is with School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: csljc@ieee.org; jcliu@cs.sfu.ca)

B. Li is with the Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: bli@cs.ust.hk).

H.-R. Shao is with the Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139 USA (e-mail: shao@merl.com).

W. Zhu is with Intel Research, Beijing 100084, China (e-mail: wenwu.zhu@intel.com).

Y.-Q. Zhang is with Microsoft Corporation, Redmond, WA 20004 USA (e-mail: yzhang@microsoft.com)

Digital Object Identifier 10.1109/TCSVT.2004.825532

In Section V, we discuss the practical issues for deploying the framework. Its performance is evaluated in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

In a heterogeneous multicast environment, a single transmission rate simply cannot meet the diverse bandwidth demands from receivers. Hence, multirate schemes, like cumulative layering [2], [6] or stream replication [3], [7], are often suggested in this context; various adaptation protocols, such as the receiver-driven layered multicast (RLM) [2], multicast loss delay based adaptation (MLDA) [8], and destination set grouping (DSG) [3], have also been devised to deliver the multirate videos to receivers according to their respective demands. Such adaptation protocols generally rely on an end-to-end service, which is different from the proxy-assisted service as advocated in our framework. Nevertheless, our framework shares many features of them, such as the TCP-friendly bandwidth estimation.

For proxy-assisted transmission, a representative is the source adaptive multi-layered multicast (SAMM) protocol [10]. SAMM assumes that some feedback mergers are placed inside the network; the receiver's bandwidth demands are not routed directly to the sender but to the mergers, where they are merged through a heuristic algorithm before forwarding to upstream nodes. It thus solves the well-known feedback implosion problem [1], but requires a priority-based queue management policy to be implemented at each router for layer filtering. Another example for multirate multicasting that employs feedback mergers is shown in [11], which is very similar to our framework in that optimal bandwidth adaptation is also performed in some intermediate nodes. A significant difference is that their work assumes computation in the network layer. Our framework, however, estimates available bandwidth and performs filtering in the application layer, though the knowledge from the network layer can be helpful. The OTPs in our framework also partition the network into confined regions, mitigating not only the problem of feedback implosion as in [11], but also the problem of bandwidth heterogeneity.

There are also proxies that manipulate the data or control information of video streams [4], [14]. Such proxies, or video gateways, transcode a video stream to a lower rate or a different format to satisfy the requirements from their local regions. Another important proxy functionality is caching, and scalable video caching for heterogeneous receivers has been investigated in [9], [29]. Recently, overlay or application-layer multicast has received much attention. These protocols generalize the application-level proxy functionalities into autonomous end-hosts, and largely overcome the deployment difficulties of IP multicast [13]. Our OTP framework is motivated by them, but also explores the benefit of IP multicast in local regions, as in island multicast paradigm [15]. Moreover, our focus is on utility-based resource allocation for object video streaming, while not overlay structure optimization.

III. Overview of the OTP-Based Multicast Framework

Our adaptation framework is proxy-based and specifically designed for multi-object video multicasting. It uses the object scalability as well as geographical partitioning to overcome the problem of heterogeneity. In this section, we first give an overview of the object scalability and our framework.

A. Object-Based Scalability

The MPEG-4 standard targets a very broad range of applications: from classical video telephony and video conference to applications requiring a high degree of interaction with video contents. To reach this latter goal, a scene in a MPEG-4 video is viewed as a composition of VOs with intrinsic properties such as shape, motion, and texture [5]. The encoding and decoding processes are carried out on the pictures of an object at given time instances, referred to as VO planes (VOPs). The relations of the objects are accomplished by a binary format for scene (BIFS), and a composition of them thus can be performed at the receiver's end based on the BIFS. Since the decoding operation for each object is independent of the others, the control parameters for the video streams, such as the output rate, can be set on an object basis. This is quite different from traditional frame-based coding schemes, such as MPEG–2 or H. 263, and provides a more flexible and efficient vehicle for adaptive video dissemination, as explored in our OTP-based framework.

B. OTP-Assisted Video Multicasting

In our framework, a video server distributes video streams to a set of receivers using the real-time transport protocol (RTP) [18]. Each VO has its own stream and an associated utility function. The object streams belonging to the same scene and the nodes that are interested in the objects constitute a *multicast session*. The control data of the session, such the status of each node, are exchanged through the RTP control protocol (RTCP) [18]. To mitigate the problem of heterogeneity, a set of OTPs are placed inside the network. An OTP is an application-level proxy, which joins the multicast session and performs the following operations:

- monitoring the bandwidth of its upstream link and collecting the bandwidth demands from downstream links;
- for each downstream link, allocating bandwidth among the video objects to maximize their total utility;
- filtering the video object streams according to the allocated bandwidth;
- distributing the filtered streams to downstream receivers or OTPs.

Conceptually, an OTP inside a multicast tree isolates its upstream nodes and downstream nodes by filtering the video traffic. It acts as a receiver to its upstream OTP or the video server and a sender to its downstream OTPs or receivers. We assume that the underlying network supports IP multicast, and all the nodes join the same multicast group; however, in case the IP multicast is not fully deployed in a network, it is also possible to build unicast tunnels among the OTPs. In other words, the OTPs form an overlay or application-layer multicast tree that interconnects diverse multicast islands with semi-uniform demands within each island.

IV. Bandwidth-Adaptation Mechanisms for OTPs

In this section, we first consider how to allocate bandwidth among video objects for a single link case. We then extend the allocation algorithm to accommodate heterogeneous links in a multicast environment.

A. Bandwidth Allocation for Single Link

For a session of N objects, a key problem is how to allocate a limited total bandwidth, called *session bandwidth*, to the objects to maximize their utility. In general, the utility of a video stream is a mapping from its rate to some performance measure, such as perceived video quality. There could be many acceptable definitions of the utility function $u_n(b_n)$ for video object n with bandwidth b_n, especially in the context of multicasting, where the receivers' bandwidth demands are heterogeneous and there is generally a nonlinear relationship between bandwidth and video quality. Furthermore, in our application, the relative importance of the objects can differ in a scene, which needs to be reflected in the utility function as well. Hence, instead of limiting our scope to a specific utility function, we try to devise a general allocation scheme that can accommodate common utility functions.

More explicitly, consider a single link where bandwidth B is allocated to the session; this bandwidth is to be shared by N video objects, $\mathrm{VO}_1, \mathrm{VO}_2, \ldots, \mathrm{VO}_N$. The optimal bandwidth allocation problem can be formally described as follows:

$$\begin{aligned} \mathbf{OPT-ALLOC:}\ max \quad & U(b_1, b_2, \ldots, b_N) = \sum_{n=1}^{N} u_n(b_n) \\ s.t. \quad & 0 \le b_n, \quad n = 1, 2, \ldots, N, \\ & \sum_{n=1}^{N} b_n = B \end{aligned}$$

and the resultant allocation is referred to as an *optimal object bandwidth allocation* for session bandwidth B.

In practice, $u_n(b_n)$ is a discrete function of b_n, because, given a limited set of quantizers, the output rate of a video coder is always discrete. In addition, a small rate adjustment may not be perceived by receivers, and it is thus not necessary to allocate the rate continuously. Therefore, assume that a rate (or bandwidth) is always a multiple of an allocation unit, the above problem is equivalent to the channlized bandwidth allocation problem for multiple video sessions (**OPT-INTER**), as introduced in [19]. Such problem can be solved using an iterative algorithm with time complexity $O(N \cdot B^2)$, which is reasonably low for a limited session bandwidth, as to be examined in Section VI-D.

B. Bandwidth Allocation for Multiple OTPs

The above algorithm can be directly used by the sender to determine the bandwidth allocation for its objects. For an OTP placed inside the multicast tree, however, it would have more than one downstream link,[1] each with a distinct session bandwidth. Hence, each branch should have an allocation commensurate with its own bandwidth. The allocation algorithm should also consider the constraint from the uplink of the OTP. For illustration, in Fig. 1, there is one uplink L_0 and M downlinks $L_1, L_2, \ldots, L_M$ for the OTP. Assume that the session bandwidth of L_0 is B^0, and the corresponding object bandwidth allocation is $(b_1^0, b_2^0, \ldots, b_N^0)$. Clearly, the session bandwidth of link L_i, B^i, $i \in \{1, 2, \ldots, M\}$, should not be higher than B^0. More importantly, any allocation $(b_1^i, b_2^i, \ldots, b_N^i)$ for L_i should satisfy $b_j^i \le b_j^0$ for $j = 1, 2, \ldots, N$, because the filtering operation at the OTP generally reduces bandwidth.[2]

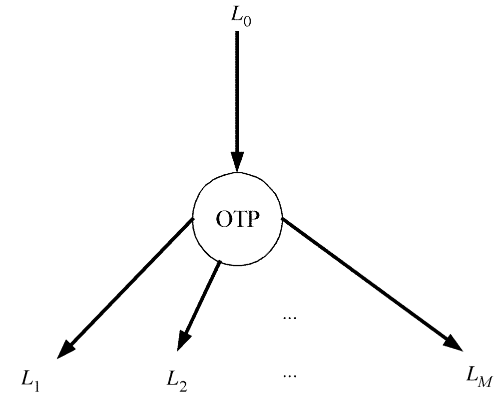


Fig. 1. Uplink and downlinks of an OTP.

We refer to this additional constraint as an *uplink allocation constraint*, and the constrained version of problem **OPT-ALLOC** as **C-OPT-ALLOC**. Clearly, the resultant total utility for **C-OPT-ALLOC** is no more than that for the corresponding **OPT-ALLOC**. Significant utility degradation would also occur. For example, assume the utility of VO_1 is extremely high when its bandwidth is higher than some r, but very low for a bandwidth lower than r. On link L_0, if B^0 is slightly higher than r, to maximize the total utility, most of the session bandwidth is allocated to VO_1, while little to other VOs. Now consider downlink L_i, and assume $B^i < r$. To achieve a high total utility, more bandwidth should be allocated to other VOs. Such allocation, however, is impossible because of the uplink allocation constraint. As a result, the total utility of the VOs could be very low at link L_i. The problem can be more severe when multiple OPTs are concatenated in a path.

To mitigate this problem, one possible solution is to employ a global optimization method for all the links of the multicast tree. This suffers from high computation overheads and is difficult to implement in such a distributed environment. Nevertheless, we find that, for concave utility functions, the optimal total utility with the uplink allocation constraint is the same as that without this constraint. The global optimization is thus unnecessary. More explicitly, we have the following theorem.

Theorem 1: Assume that $(b_1^0, b_2^0, \ldots, b_N^0)$ is an optimal object bandwidth allocation for session bandwidth B^0. If the utility function of each object is concave, then the solution to problem **C-OPT-ALLOC** with session bandwidth $B^i (\le B^0)$ and uplink allocation constraint $b_j^i \le b_j^0$, $j = 1, 2, \ldots, N$, is also a solution to the corresponding **OPT-ALLOC**. That is,

[1]For sparsely distributed OTPs, the overlay path between two OTPs or between an OTP and the sender could involve more than one physical link. Nevertheless, there remains only one bottleneck link in the path. For simplicity, we use term *link* to refer to both cases.

[2]Although it is possible to increase the bandwidth of an object stream in filtering, e.g., using upsampling, such operations indeed do not improve perceived quality. We therefore do not consider them in this paper.

the total utility of the solution to **C-OPT-ALLOC** is the same as that of the optimal object bandwidth allocation for session bandwidth B^i.

Proof: We proof this theorem by contradiction. Assume that the total utility of the resultant allocation for **C-OPT-ALLOC** is less than that for **OPT-ALLOC**. That is, there exists an allocation $(b'_1, b'_2, \ldots, b'_N)$, where $\sum_{n=1}^{N} b'_n = B^i$ and $b'_s > b^0_s$ for at least one $s \in \{1, 2, \ldots, N\}$; for any allocation $(b''_1, b''_2, \ldots, b''_N)$ that satisfies $\sum_{n=1}^{N} b''_n = B^i$ and $b''_n \le b^0_n$, $n = 1, 2, \ldots, N$, we have $U(b'_1, b'_2, \ldots, b'_N) > U(b''_1, b''_2, \ldots, b''_N)$.

Note that there must exists t such that $b'_t < b''_t \le b^0_t$, because $\sum_{n=1}^{N} b'_n = \sum_{n=1}^{N} b''_n = B^i$. Without loss of generality, we assume that s $< t$. Since $(b^0_1, b^0_2, \ldots, b^0_N)$ is the optimal solution to **OPT-ALLOC** with session bandwidth B^0, we have $u_s(b'_s) - u_s(b^0_s) \le u_t(b^0_t) - u_t(b^0_t - (b'_s - b^0_s))$. Otherwise, if $u_s(b'_s) - u_s(b^0_s) > u_t(b^0_t) - u_t(b^0_t - (b'_s - b^0_s))$, we can construct a new allocation $(b^0_1, \ldots, b^0_{s-1}, b'_s, b^0_{s+1} \ldots, b^0_{t-1}, b^0_t - (b'_s - b^0_s), b^0_{t+1}, \ldots, b^0_N)$, where b^0_s and b^0_t in $(b^0_1, b^0_2, \ldots, b^0_N)$ are replaced by b'_s and $b^0_t - (b'_s - b^0_s)$, respectively. This allocation also has a total bandwidth of B^0, but its total utility is $U(b^0_1, b^0_2, \ldots, b^0_N) + [u_s(b'_s) - u_s(b^0_s)] - [u_t(b^0_t) - u_t(b^0_t - (b'_s - b^0_s))] > U(b^0_1, b^0_2, \ldots, b^0_N)$, which contradicts the fact that $(b^0_1, b^0_2, \ldots, b^0_N)$ is optimal. Similarly, we can prove that $u_t(b'_t) - u_t(b^0_t) \le u_s(b^0_s) - u_s(b^0_s + (b^0_t - b'_t))$.

We now show a re-allocation operation for $(b'_1, b'_2, \ldots, b'_N)$ to find contradictions to our initial assumption, as follows.

Case 1: $b'_s - b^0_s \le b^0_t - b'_t$: Consider allocation $(b'_1, \ldots, b'_{s-1}, b^0_s, b'_{s+1} \ldots, b'_{t-1}, b'_t + (b'_s - b^0_s), b'_{t+1}, \ldots, b'_N)$; this new allocation has a session bandwidth of B^i, but its total utility is

$$
\begin{aligned}
&U(b'_1, b'_2 \ldots, b'_N) - \left[u_s(b'_s) - u_s\left(b^0_s\right)\right] \\
&\quad + \left[u_t\left(b'_t + \left(b'_s - b^0_s\right)\right) - u_t(b'_t)\right] \\
&\quad\quad \ge U(b'_1, b'_2, \ldots, b'_N) - \left[u_s(b'_s) - u_s\left(b^0_s\right)\right] \\
&\quad\quad\quad + \left[u_t\left(b^0_t\right) - u_t\left(b^0_t - \left(b'_s - b^0_s\right)\right)\right] \\
&\quad\quad \ge U(b'_1, b'_2, \ldots, b'_N).
\end{aligned}
$$

The first inequality follows that $b'_s - b^0_s \le b^0_t - b'_t$ and $u_t(\cdot)$ is concave, and the second follows that $u_s(b'_s) - u_s(b^0_s) \le u_t(b^0_t) - u_t(b^0_t - (b'_s - b^0_s))$, as proved before.

We re-allocate $(b'_1, b'_2, \ldots, b'_N)$ to this new allocation, i.e., $(b'_1, b'_2, \ldots, b'_N) \leftarrow (b_1, \ldots, b'_{s-1}, b^0_s, b'_{s+1} \ldots, b'_{t-1}, b'_t + (b'_s - b^0_s), b'_{t+1}, \ldots, b'_N)$.

Case 2: $b'_s - b^0_s > b^0_t - b'_t$: Consider allocation $(b'_1, \ldots, b'_{s-1}, b'_s - (b^0_t - b'_t), b'_{s+1}, \ldots, b'_{t-1}, b^0_t, b'_{t+1}, \ldots, b'_N)$. Similar to case 1, we can prove that the total utility of this new allocation is equal to or greater than $U(b'_1, b'_2, \ldots, b'_N)$, according to the concavity of $u_s(\cdot)$ and $u_t(b'_t) - u_t(b^0_t) \le u_s(b^0_s) - u_s(b^0_s + (b^0_t - b'_t))$.

Thus, we re-allocate $(b'_1, b'_2, \ldots, b'_N)$ to $(b'_1, \ldots, b'_{s-1}, b'_s - (b^0_t - b'_t), b'_{s+1}, \ldots, b'_{t-1}, b^0_t, b'_{t+1}, \ldots, b'_N)$.

The above re-allocation operation can be applied to the new $(b'_1, b'_2, \ldots, b'_N)$ iteratively until the resultant $(b'_1, b'_2, \ldots, b'_N)$ satisfies $\sum_{n=1}^{N} b'_n = B^i$ and $b'_n \le b^0_n$, $n = 1, 2, \ldots, N$. This can be achieved in at most $N - 1$ iterations because, in each re-allocation, either VO_s is allocated a bandwidth of b^0_s or VO_t is allocated a bandwidth of b^0_t, and such object will not be involved in the re-allocation operations for the remaining iterations. Since the total utility is nondecreasing in the iterations, we have a contradiction to our initial assumption. **Q.E.D**

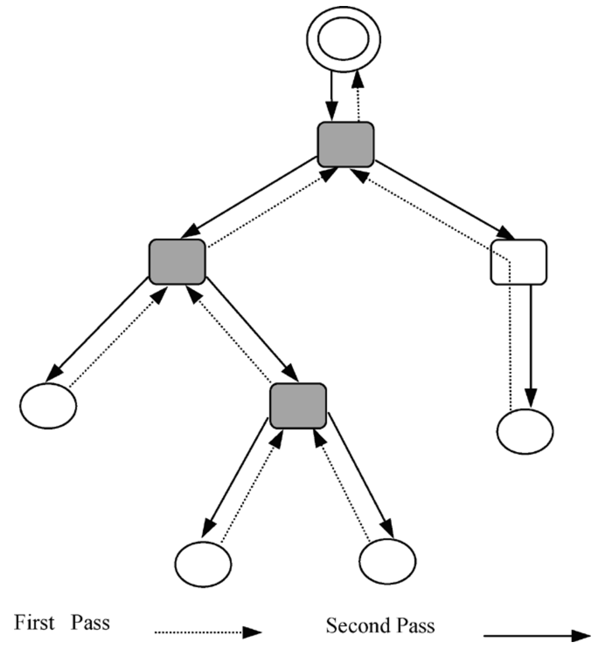


Fig. 2. Two-pass bandwidth allocation.

Since the marginal utility of a video stream generally diminishes when its bandwidth increases, i.e., its utility function is concave, the above theorem directly leads to a two-pass bandwidth-allocation algorithm for an OTP-based application layer multicast tree, as illustrated in Fig. 2.

The first is a bottom-up pass, which determines the request of session bandwidth for each link. In this pass, each receiver monitors its available bandwidth B^{R-AV} and periodically reports it to the upstream OTP as the session bandwidth request for the link between them. Each OTP also monitors the available bandwidth between itself and the neighboring upstream OTP (see Section V-B for details of the method). Assume that this available bandwidth is B^{P-AV}, and the session bandwidth request from downlink L_i is B^i_{req}, $i = 1, 2, \ldots, M$. Since it is no use to have a bandwidth that exceeds the maximum request from the downlinks, the session bandwidth request for the OTP's upstream link, B^0_{req}, is $\min\{B^{P-AV}, \max\{B^1_{\mathrm{req}}, B^2_{\mathrm{req}}, \ldots, B^M_{\mathrm{req}}\}\}$. The OTP then sends this request to its neighboring upstream OTP, and similar procedures continue until the sender is involved.

The second is a top-down process, which determines the exact session bandwidth at each link as well as the corresponding object bandwidth allocation. First, the session bandwidth for the link directly connected to the sender is simply the bandwidth request received in the first pass. The sender calculates the optimal object bandwidth allocation for this session bandwidth without imposing any uplink constraint. It also informs its neighboring downstream OTPs about this allocation. For an OTP, the session bandwidth of its downlink L_i $(i = 1, 2, \ldots, M)$ is set as $\min\{B^0, B^i_{req}\}$, where B^0 is the session bandwidth of its uplink. The OTP then calculates the optimal rate allocation for

TABLE I
STREAM CLASSIFICATION AND PRIORITIZATION

Priority	Category
1	Control data and VO headers
2	IVOP shape and texture DC
3	IVOP texture AC
4	PVOP shape and motion
5	PVOP texture
6	BVOP

this downlink by imposing the constraint of the uplink allocation to the iterative algorithm for problem **OPT-ALLOC** and distributes the allocation information to its downstream nodes. Similar procedures continue until a receiver is reached. In this pass, the utility for each link is optimized with respect to the given constraints, and Theorem 1 guarantees that such local optimization leads to a global optimum.

V. PRACTICAL ISSUES

There are several practical issues to be addressed in our framework. First, how are the object video streams filtered to match the bandwidth allocation? Second, how is the available bandwidth of each link estimated in a dynamic network? Finally, to deploy the framework in a wide-area network, the placement of the OTPs is also a crucial issue. We give detailed discussions on the above issues in this section.

A. *Object Video Filtering*

In this subsection, we describe a simple filtering algorithm based on a smart stream classification and packetization scheme, which leverage the error resilience and concealment tools of MPEG-4. We however stress that our adaptation framework is flexible enough to accommodate diverse filtering mechanisms, such as those based on transcoding or scalable coding.

In MPEG-4, a video stream is structured into video packets (VPs) with individual synchronization markers [5]. Each VP is a self-contained decoding unit and, given some threshold, its size can be well controlled in encoding. Therefore, a VP (or several VPs) is often suggested to serve as the payload for an RTP packet [20]. A data-partitioning mode can be enabled to further separates the shape, motion or texture data in a VP by a DC Marker or Motion Marker. As such, a stream can be effectively resynchronized in the presence of bit errors.

For Internet-based transmission, most errors are caused by packet loss, in which a VP is completely lost. Consequently, the error isolation capability of the DC or Motion Marker becomes useless. Since the data contained in a VP, such as shape, motion, and texture, have different levels of importance to decoding, it is beneficial to classify them in a finer granularity beyond the coarse-grained classification of VOP types (I, P, and B VOPs). Different types of the data thus can be more promptly handled during transmission. To this end, we classify the data in a bitstream into several categories according to VP boundaries as well as DC or Motion Markers and assign priorities to the categories to differentiate their importance, as shown in Table I.

We refer to the content belonging to the same category in a VP as a *data item*. For each video object stream, we use a file to record the indexing information of all the data items, including their categories, starting positions, and sizes. This indexing file is generated during the encoding process or by parsing the bitstream, and classification thus can be performed based on this file.

After the classification and prioritization, the data items are encapsulated into application level packets (ALPs). We limit the size of an ALP to be less than the maximum transmission unit (MTU) of the underlying network. It can thus serve as a transmission unit, e.g., RTP payload. However, for low-rate streams, the size of a VOP itself is relatively small, and the size of a VP thus could be small even if its size threshold is set at a high value during encoding. This is particularly true for PVOPs and BVOPs. Our experiments show that, for the standard video sequence *Akiyo* (CIF 92 Kb/s, 300 VOPs), the average VP size is only 2527 bits for a threshold of 4000 bits. As a result, if each ALP contains only one data item, the packet header overhead (IP+UDP+RTP) is greater than 11%. Hence, for PVOP and BVOP data, we multiplex the data items of the same priority from several VPs into one ALP. This can be viewed as a practical use of the interleaving option in [22]. To reduce losses of important data, we also limit the number of data items in an ALP, specifically, four for PVOP shape and motion items. Our experiments show that, in this case, the average ALP size is 5891 b, and the efficiency is thus improved to 94.8%.

The bistream classification and packetzation process is illustrated in Fig. 3. Given a target rate, the server or an OTP can actively discard some low-priority packets to meet this target. A generic rate adaptation (or filtering) algorithm for a single object is as follows.

Assume there are K priorities (in our classification scheme, $K = 6$), and the rate of priority i is $r_i, i = 1, 2, \ldots, K$. The total rate of the object stream is thus $r^{\text{total}} = \sum_{i=1}^{K} r_i$. If the target rate b of the object is lower than r^{total}, the sender or an OTP needs to discard some packets starting from the lowest priority. To this end, it first finds the critical priority index k for target rate b, where $\sum_{i=1}^{k-1} r_i < b \leq \sum_{i=1}^{k} r_i$. All packets of priorities $k + 1$ through K exceed the target rate and thus need to be dropped directly. The packets of priority k need to be selectively discarded with probability a, where $a = (\sum_{i=1}^{k} r_i - b)/r_k$.

At the receiver's end, standard error-concealment tools can be used to mitigate the effect of packet dropping [5], [21]. We also employ several concealment methods that take advantage of our packetization and rate-adaptation scheme. For example, given that motion data are delivered with a high priority, even if some texture data are lost, the affected blocks can effectively approximated from some neighboring blocks as well as the corresponding block in the previous VOP.

In summary, our proposed filtering algorithm is essentially an extension to the temporal scalability-based filtering [21]. Taking advantage of the data partitioning and error-concealment features in MPEG-4, our algorithm yields finer granularity for rate control than the basic temporal scalability.

B. *Bandwidth Estimation*

Since our framework is designed for video distribution over the best-effort Internet where TCP is the dominant transport protocol, it is necessary to ensure that a video stream will not starve

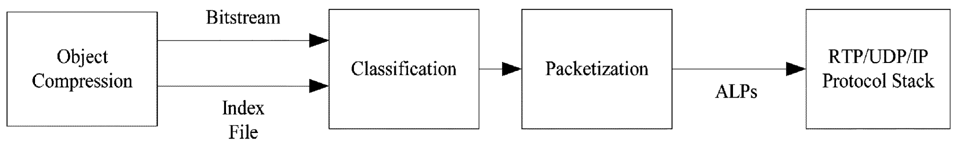


Fig. 3. Diagram of bitstream processing.

its background TCP flows. To this end, we let the available bandwidth B^{R-AV} or B^{P-AV} be estimated as the long-term throughput of a TCP connection, as if the connection is running over the same path. We can therefore control the session bandwidth of the video objects to achieve a fair share with concurrent TCP flows.

There have been many efforts on modeling the equivalent TCP throughput; a general conclusion is that such a model relies on packet size, round-trip time (RTT), and loss event rate [23]. The estimation of RTT needs a feedback loop for request and response. In our framework, we assume that the receivers and OTPs join the same RTP session and each has a unique RTP synchronization source identifier (SSRC) [18]. Each RTCP report of a receiver or an OTP serves as a request for an RTT estimation. The RTCP reports are also used to classify the receivers and OTPs, so that the cluster of receivers that an OTP should serve can be identified, as done in [12]. Such geographical partitioning not only mitigates the problem of heterogeneity, but also suppresses feedback implosion, because the sender or an OTP needs to handle the requests from a confined region only. In addition, we adopt a batched response process to further reduce the number of response packets as well as the overhead of their headers [24]: suppose that an OTP has received K requests with identifiers $\mathrm{SSRC_i}$ and arrival times t_i^{arrival}, $i = 1, 2, \ldots, K$, in time slot $[t, t+T]$. At time $t + T$, it will multicast a response packet to all its neighboring downstream OTPs (or receivers). The packet contains a list of the SSRCs and the corresponding delays t_i^{delay}, where $t_i^{\mathrm{delay}} = t + T - t_i^{\mathrm{arrival}}$, $i = 1, 2, \ldots, K$. When an OTP (or receiver) receives the response packet and its SSRC is in the list, e.g., $\mathrm{SSRC_i}$, it will generate an RTT estimate τ^0 as $\tau^0 = t^0 - t' - t_i^{\mathrm{delay}}$, where t^0 and t' are the current local time and the local time that the request was initiated, respectively.

For the calculation of loss event rate, since the streams of all the objects in a session act as a "single" flow to compete for bandwidth with TCP flows, the loss event rate should be calculated across all of the received object streams. Moreover, in our framework, packet dropping is caused not only by congestion, but also by the filtering algorithm. As a result, we cannot simply distinguish the order of packets from the RTP sequence numbers and calculate loss rate accordingly. To solve this problem, we introduce a *counter* field for this session in each ALP packet. An OTP also maintains a counter for each downlink. When a packet (from any object stream of the session) is to be transmitted, the OTP counter for the corresponding downlink is increased by one and then written to the counter field of the packet. At a downstream node of the OTP, the counter field can be used for loss event rate calculation, because a gap in the counter exactly reflects a packet loss of the session due to congestion at the link from the OTP to the node. We allocate 8 bits to this field, considering that a packet re-ordering or burst loss of more than 255 packets seldom happens [25].

C. Placement of OTP

Finally, the placement of the OTPs is clearly a crucial issue. This indeed a well-studied research problem, and numerous algorithms have been developed in different contexts, such as Web proxing [16], server replicating [26], and enroute data replicating [27]. Given a network topology, these algorithms choose a subset of nodes to place the proxies to maximize (or minimize) some client-perceived measures, such as average delivered bandwidth or latency. Many of these algorithms can be used in our framework, especially those targeting the multicast environment. More importantly, our simulation results suggest that noticeable performance improvement can be achieved even with uniformly placed OTPs.

VI. Performance Evaluation

In this section, we study the performance of our framework through simulations. It is difficult to use a single metric to evaluate such a complex video multicast system; hence, we adopt a set of metrics that have been used in the literature, namely, bandwidth fairness, packet loss rate, and computational overhead. In addition, a weighted function is proposed to assess the overall quality of multiple video objects. We try to identify the tradeoffs among the above performance measures and, for the sake of comparison, we have also simulated other typical video multicast schemes under the same network configurations. Specifically, we present the comparison results for the following two schemes:

Multicast Enhanced Loss-Delay based Adaptation (MLDA) [8], which is an end-to-end layered multicast protocol. In MLDA, the sender encodes a video into some cumulative layers and distributes them over separate multicast groups. Each receiver selectively joins a set of layers according to its available bandwidth. The sender also estimates the minimum and maximum receiver bandwidths periodically through a scalable feedback algorithm, and then uniformly allocates layer rates in this range.

Transcoding Proxy based Multicast (TPM) [4], [14], which is similar to our proxy-assisted framework, but a transcoder is used to filter video streams at each proxy.

Both MLDA and TPM employ frame-based video, not object-based video. In our simulation, the MPEG-4 fine-granular scalability (FGS) [6] tool is used for video layering in MLDA, and a requantization transcoder is used for each proxy in TPM. As in previous studies, the number of layers for MLDA is set to

4. To achieve TCP compatibility, in MLDA and TPM, the available bandwidth is also estimated as the equivalent throughput of a virtual TCP connection running over the same path.

A. Simulation Settings

We evaluate all the schemes using the LBNL network simulator $ns-2$ [28]. The networks for simulation follow the widely used transit-stub (TS) topology model, which tries to reproduce the hierarchical structure of the Internet topology by composing interconnected transit and stub domains. The parameters for this model include: T, the number of transit domains; N_t, the average number of nodes per transit domain; K, the average number of stub domains per transit node; N_s, the average number of nodes per stub domain; E_t, extra transit-stub links; and E_s, extra stub-stub links. We use the *GT-ITM* package [30] with the following setting to generate TS topologies: $T = 2$, $N_t = 3$, $K = 3$, $N_s = 3$, $E_t = 0$, and $E_s = 0$. The total number of nodes in such topologies is 60. Each node represents a FIFO drop-tail router with a queue size of 25 packets, and each edge represents a link of 1-Mb/s bandwidth. Given a topology, we use the following method to produce a network instance for simulation.

Number of proxies: For the TPM and OTP, the number of proxies is either 5 or 10, and the corresponding settings are referred to as TPM-5, TPM-10, OTP-5, and OTP-10, respectively.

Placement of end-nodes and proxies: A video server is attached to a randomly selected node, and an average of five video receivers are attached to each of the remaining nodes. For TPM and OTP, a given number (five or ten) of proxies are then randomly placed to the remaining nodes.

Cross-traffic: TCP Reno connections are uniformly placed between node pairs such that there is on average four TCP connections running over a link, while the minimum and maximum numbers are 1 and 9, respectively. As a result, the expected available bandwidth for a video session is on average 200 Kb/s; yet the minimum and maximum are 100 and 500 Kb/s, respectively.

To mitigate the effect of randomness, we generated ten topologies and ten instances (five of five proxies and five of ten proxies) for each topology using the above method; the results presented in this section are their average. All instances were simulated for 1000 seconds, which is long enough for observing steady-state behaviors. The observation period for the simulation results is from 500 to 1000 s.

A standard MPEG-4 video testing sequence *News* (CIF) is used as the video source. The total rate of this video is set to 512 Kb/s, and the average rates of IVOP, PVOP, and BVOP are 100, 216, and 196 Kb/s, respectively. For TPM and MLDA, there is no classification for the bitstream, and the VPs are directly used as RTP payloads, as suggested in [20]. The payload size is 500 bytes for both TCP and video packets.

For our object-based framework, the scene of *News* consists of five objects, namely, man (VO_1), women (VO_2), screen of dancers (VO_3), caption (MPEG4 WORLD, VO_4), and background (VO_5). The video stream is classified and packetized according to the method described in Section V-A. The size of an ALP is also limited to 500 bytes, and the default bandwidth allocation unit is 1 Kb/s. Let $Q_n(r)$ denote the empirical rate-quality curve of VO_n, with the quality measured by peak signal-to-noise ratio (PSNR) [21]. We use a utility function of the form $u_n = w_n Q_n(r)$, where w_n is a weighting factor that reflects the importance of object i, as suggested in [31]. The default value of w_n is set to 1/5 for $n = 1, 2, \ldots, 5$, i.e., all the objects have the same importance. We also investigate the effects of uneven w_n settings in Section VI-C.

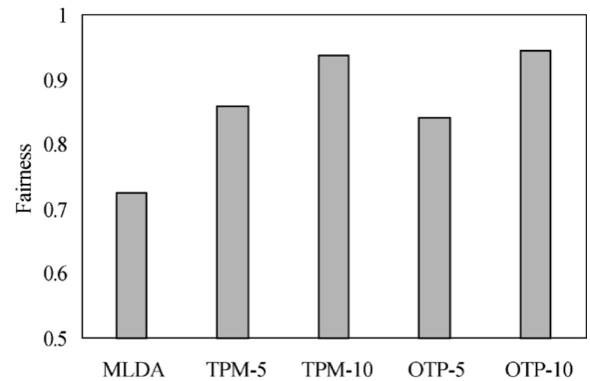


Fig. 4. Comparison for bandwidth fairness.

B. Bandwidth Fairness

In the first set of experiments, we evaluate the fairness of the receivers in a session, specifically, inter-receiver fairness. In an ideal inter-receiver fair distribution, the rate of the video stream delivered to each receiver is commensurate with its own bandwidth demand, regardless of the demands of other receivers in the same session. We use the following function to measure such fairness for each single receiver, as suggested in [7]:

$$F(R) = \frac{\min(R, \hat{B})}{\max(R, \hat{B})}$$

where R is the session bandwidth delivered to the receiver and $\hat{B}$ is the *isolated rate* or the bottleneck available bandwidth for the receiver. Since the available bandwidth is estimated as the equivalent TCP throughput in our study, this function also reflects the degree of TCP-friendliness of the video streams. The fairness value of one is optimal, which means that the receiver perfectly exploits its expected bandwidth, i.e., the video stream(s) fairly shares the bandwidth with TCP flows; other fairness values are between 0 and 1.

We sample a fairness value per second for each receiver. Fig. 4 shows the average fairness values for all the receivers over the observation period (500–1000 s). It can be seen that the proxy-assisted frameworks (TPM and OTP) yield reasonably good fairness values, which are higher than 0.8 for the use of five proxies, and can be improved to more than 0.9 by using more proxies. This is because the proxies enable fine-grained rate adaptation to serve receivers in confined regions. To the contrary, the adaptation granularity of MLDA is at a layer level, which is considerably coarse given that the number of layers is limited in practice. As a result, the fairness value of MLDA is only about 0.7 in our experiment.

TABLE II
PACKET LOSS RATES FOR DIFFERENT PRIORITIES IN THE OTP-BASED FRAMEWORK

Num of OTPs	Loss Rate (%)					
(% of Total Nodes)	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	Priority 6
5 (10%)	0.006	0.02	0.09	0.19	0.31	0.49
10 (20%)	0.003	0.01	0.06	0.13	0.25	0.64

C. Perceived Video Quality

In our framework, a key factor that affects the perceived video quality is packet loss. In Table II, we list the average loss rates for different priorities in our OTP-base framework. Given that these priorities have uneven numbers of packets, we calculate the loss rate for each priority as the number of losses over the total number of packets from the same priority. It can be seen that the distribution is highly skewed; in generally, the data of high priorities are well protected, even if the OTPs are randomly placed and their number is limited. As such, the lost information can be effectively recovered by some error-concealment tools, as introduced before.

We now evaluate the quality of service from an application point of view. Since the objects have different importance in terms of their contributions to the perceptual quality, we use a weighted function to measure the PSNR for each VOP, as follows:

$$\mathrm{WPSNR} = \sum_{n=1}^{N} \left(\frac{w_n}{\sum_{i=1}^{N} w_i} \cdot \mathrm{PSNR}_n \right)$$

where PSNR_n is the PSNR value of object n. In our simulation, we record packet loss and delay information into trace files, and reconstruct the video according to such trace files. The corresponding PSNR values of each object thus can be calculated from the reconstructed and the original VOPs. For the frame-based TPM and MLDA, we can also use the shape mask of each object to distinguish it from others, and calculate its PSNR value accordingly.

We assume that the receivers' interest levels of the video objects follow a Zipf-like distribution [32] and hence set the importance factor for VO_n using the following function:

$$w_n = \frac{\left(\frac{1}{n}\right)^{\theta}}{\sum_{i=1}^{N} \left(\frac{1}{i}\right)^{\theta}}, \quad n = 1, 2, \ldots, N$$

where θ is referred to as a *skew factor*. When θ is zero, all the importance factors are identical (=1/5); this is the setting that has been used in the previous experiments. When θ increases, the importance factors are differentiated, i.e., some take higher values than others. Fig. 5 shows the average WPSNR values of the receivers for different skew factors and adaptation schemes. It can be seen that our scheme achieves remarkably higher WPSNR value than MLDA for any skew factor. This reaffirms our observations in the previous experiment on the degrees of fairness for MLDA and OTP. For small skew factors, the WPSNR of our scheme is lower than that of TPM, because transcoding generally yields better video quality than our filtering algorithm. However, TPM (and also MLDA) reduces video rate

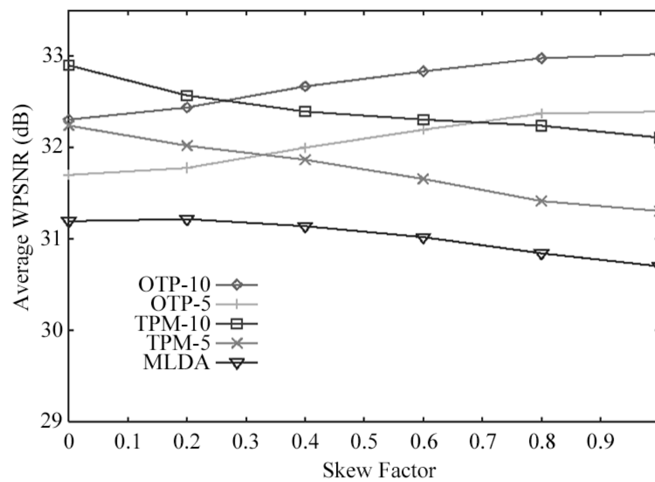


Fig. 5. Average WPSNR as a function of skew factor.

uniformly without differentiating the relative importance of the video objects. As a result, when the skew factor increases, the WPSNR value of TPM (or MLDA) decreases, whereas that of ours remains nondecreasing or even increases. In particular, when the importance factors are highly skewed, the WPSNR of ours is often 1 dB higher than that of the other two schemes. Such improvement, which is noticeable from a video coding point of view, clearly demonstrates the superior of our utility-driven bandwidth allocation algorithm for multi-object multicasting.

D. Computational Overhead

Finally, we evaluate the computational overhead of our framework. Since the filtering operation for an object stream is simply packet scanning, which incurs much lower overheads than conventional transcoding, we mainly focus on the overhead of the algorithm for object bandwidth allocation at each OTP. We implement the allocation algorithm for problem **OPT-ALLOC** using C++ on an Intel Pentium 4, 1.2-GHz PC with 512 MB of memory. Fig. 6 plots the measured computation times for different combinations of the number of objects (N) and the session bandwidth (B)—two key factors that determine the complexity of the algorithm. We find that all the solutions can be computed within a reasonably short time (<40 ms), especially considering that this algorithm is executed infrequently. Note that the same algorithm can be used to solve problem **C-OPT-ALLOC** by imposing a search constraint to the bandwidth of each object. This constraint further reduces the computation time of the algorithm. Moreover, as an iterative algorithm, it gives the optimal allocation for any bandwidth lower than B in some intermediate stage (see [19] for details of the algorithm). Such intermediary results can be saved in a table for future reference.

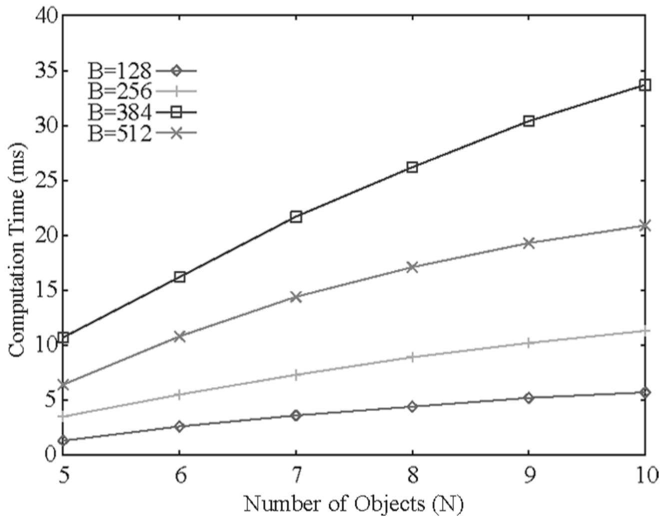


Fig. 6. Computation overhead as a function of the number of objects (N) and normalized session bandwidth (B).

VII. Conclusion

In this paper, we presented a proxy-assisted adaptation framework for multicasting MPEG-4 object video over the Internet. The framework integrates two key components:

1) a bitstream classification, prioritization, and smart packetization scheme; taking advantage of this scheme, the streaming rate of each video object can be adjusted to meet dynamic network conditions at low overheads
2) OTPs, which form an application-layer multicast tree that partitions receivers into confined regions with semi-uniform demands within each region; the OTPs also adopt an optimal bandwidth allocation algorithm that maximizes the total system utility using object scalability.

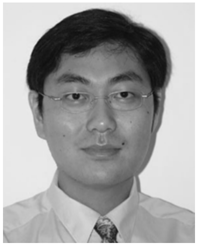
Extensive simulations were carried out to evaluate the performance of our framework. The results demonstrated that it achieves better bandwidth fairness and perceived video quality than conventional frame-based transmission. The improvement is particularly noticeable when video objects have uneven utilities, i.e., with different levels of importance to the receivers. Moreover, its computational overhead is kept at a low level.

Acknowledgment

The authors would like to thank Y. Wu, Princeton University, X. Zhang, The Chinese University of Hong Kong, and M. Chen, University of California at Berkeley, for their implementation work when they were interns in Microsoft Research, Asia. Furthermore, the comments from the anonymous reviewers were a great help for improving the paper.

References

[1] B. Li and J. Liu, "Multi-rate video multicast over the Internet: an overview," *IEEE Network*, vol. 17, no. 1, pp. 24–29, Jan. 2003.
[2] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM' 96*, Aug. 1996, pp. 117–130.
[3] S. Cheung, M. H. Ammar, and X. Li, "One the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM'96*, Mar. 1996, pp. 553–560.
[4] N. Yeadon, F. Garcia, D. Hutchison, and D. Shepherd, "Filters: QoS support mechanisms for multipeer communications," *IEEE J. Select. Areas Commun.*, vol. 14, no. 9, pp. 1245–1262, Sep. 1996.
[5] Overview of the MPEG-4 Standard, Mar. 2002.
[6] W. Li, "Overview of the fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.
[7] T. Jiang, E. Zegura, and M. Ammar, "Inter-receiver fair multicast communication over the Internet," in *Proc. NOSSDAV' 99*, Jun. 1999, pp. 103–114.
[8] D. Sisalem and A. Wolisz, "MLDA: a TCP-friendly congestion control framework for heterogeneous multicast environments," presented at the *8th Int. Workshop Quality of Service (IWQoS)*, Karlsruhe, Germany, Jun. 2000.
[9] F. Yu, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "QoS-adaptive proxy caching for multimedia streaming over the Internet,," *IEEE Trans. Circuits Syst. Video Techno.*, vol. 13, no. 3, pp. 257–269, Mar. 2003.
[10] B. Vickers, C. Albuquerque, and T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," *IEEE/ACM Trans. Networking*, vol. 8, no. 12, pp. 720–733, Dec. 2000.
[11] K. Kar, S. Sarkar, and L. Tassiulas, "A scalable low-overhead rate control algorithm for multirate multicast sessions," *IEEE J. Select. Areas Commun.*, vol. 20, no. 10, pp. 1541–1557, Oct. 2002.
[12] K. Salamatian and T. Turletti, "Classification of receivers in large multicast groups using distributed clustering," presented at the *Packet Video'01*, Seoul, Korea, Apr. 2001.
[13] S. Banerjee and B. Bhattacharjee. "A comparative study of application layer multicast protocols". [Online]. Available: http://www.cs.wisc.edu/suman/pubs/compare.ps.gz
[14] E. Amir, S. McCanne, and R. Katz, "An active service framework and its application to real-time multimedia transcoding," in *Proc. ACM SIGCOMM*, Sept. 1998, pp. 178–189.
[15] K.-W. Cheuk, S.-H. Chan, and J. Lee, "Island multicast: The combination of IP-multicast with application-level multicast," in *Proc IEEE Int. Conf. Communications (ICC)*, Jun. 2004, pp. 1441–1445.
[16] B. Li, M. Golin, G. Italiano, and X. Deng, "On the optimal placement of web proxies in the Internet," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 1282–1290.
[17] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 923–941, Sept. 2000.
[18] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," *RFC 1889*, Jan. 1996.
[19] J. Liu, B. Li, Y.-T. Hou, and I. Chlamtac, "On optimal layering and bandwidth allocation for multi-session video broadcasting," *IEEE Transactions on Wireless Communications,*, vol. 3, no. 3, pp. 656–667, Mar. 2004.
[20] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, "RTP payload format for MPEG-4 audio/visual streams," *RFC 3016*, Nov. 2000.
[21] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Upper Saddle River, NJ: Prentice-Hall, 2001.
[22] C. Guillemot, P. Christ, S. Wesner, and A. Klemets, RTP payload format for MPEG-4 with flexible error resiliency, in Internet Draft, draft-ietf-avt-mpeg4streams-00.txt, Mar. 2000.
[23] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. ACM SIGCOMM*, Aug. 2000, pp. 43–57.
[24] T. Turletti, S. Parisis, and J. Bolot, "Experiments With a Layered Transmission Scheme Over the Internet," INRIA Sophia Antipolis, France, Tech. Rep. 3296, Nov. 1997.
[25] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Trans. Networking*, vol. 7, pp. 277–292, June 1999.
[26] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 1587–1596.
[27] J. Xu, B. Li, and D. L. Lee, "Placement problems for transparent data replication proxy services," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1383–1398, Sept. 2002.
[28] The LBNL Network Simulator, ns-2, S. McCanne and S. Floyd.*http://www.isi.edu/nsnam/ns/* [Online]
[29] J. Liu, X. Chu, and J. Xu, "How to model an internetwork," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1490–1500.
[30] GT-ITM: Georgia Tech Internetwork Topology Models. [Online]. Available: http://www.cc.gatech.edu/projects/gtitm/
[31] J. Ronda, M. Eckert, F. Jaureguizar, and N. Garcia, "Rate control and bit allocation for MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, Dec. 1999.
[32] G. Zipf, *Human Behavior and the Principle of Least Effort*. Reading, MA: Addison-Wesley, 1949.



Jiangchaun Lin (S'01–M'03) received the B.Eng degree (*cum laude*) from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2003, both in computer science.

He is currently an Assistant Professor in the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, and was an Assistant Professor at The Chinese University of Hong Kong from 2003 to 2004. His research interests include Internet architecture and protocols, media streaming, wireless ad hoc networks, and service overlay networks. He is the co-inventor of one European patent and two U.S. patents.

He was a recipient of the Microsoft research fellowship (2000), a recipient of the Hong Kong Young Scientist Award (2003). He won first-class honors in several regional and national programming contests. He serves as TPC member for various networking conferences, including IEEE INFOCOM'04 and INFOCOM'05. He was TPC Co-Chair for the First IEEE International Workshop on Multimedia Systems and Networking (WMSN'05), Information System Co-Chair for *IEEE INFOCOM'04*, and a Guest Editor for *ACM/Kluwer Journal of Mobile Networks and Applications (MONET)*, Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks. He is a member of the IEEE Communications Society, and an elected member of Sigma Xi.

Bo Li (S'89–M'92–SM'99) received the B.S. (*summa cum laude*) and M.S. degrees in computer science from Tsinghua University, Beijing, China, in 1987 and 1989, respectively, and the Ph.D. degree in computer engineering from the University of Massachusetts at Amherst in 1993.

Between 1994 and 1996, he worked on high-performance routers and ATM switches in IBM Networking System Division, Research Triangle Park, NC. Since then, he has been with the Computer Science Department, the Hong Kong University of Science and Technology, Hong Kong, where he is now an Associate Professor. He also holds an adjunct researcher position with Microsoft Research Asia (MSRA), Beijing, China. His current research interests include wireless mobile networking supporting multimedia, video multicast and all optical networks using WDM. He coauthored the first paper on proxy server placement in 1999.

Dr. Li has been a member of the editorial board of or IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, *ACM Journal of Wireless Networks (WINET)*, IEEE Journal of Selected Areas in Communications (JSAC)—Wireless Communications Series, *ACM Mobile Computing and Communications Review (MC2R)*, *SPIE/Kluwer Optical Networking Magazine* (ONM), and KICS/IEEE Journal of Communications and Networks (JCN). He has served as a Guest Editor for the *IEEE Communications Magazine* Special Issue on Active, Programmable, and Mobile Code Networking (April 2000), the IEEE Journal on Selected Areas in Communications Special Issue on Protocols for Next Generation Optical WDM Networks (October 2000), *ACM Performance Evaluation Review* Special Issue on Mobile Computing (December 2000), and SPIE/Kluwer *Optical Networks Magazine* Special Issue on Wavelength Routed Networks: Architecture, Protocols and Experiments (January/February 2002). Currently, he is the lead Guest Editor for a Special Issue of IEEE Journal on Selected Areas in Communications on Recent Advances in Service-Overlay Network, and a Guest Editor for a Special Issue of IEEE Journal on Selected Areas in Communications on Ad Hoc Networks. In addition, He has been involved in organizing over 30 conferences, especially IEEE INFOCOM since 1996.

Huai-Rong Shao (M'00) received the E.Eng and Ph.D. degrees from Tsinghua University, Beijing, China, in 1994 and 1999, respectively, both in computer science.

He is currently a Research Scientist with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. Before joining MERL, he worked with Microsoft Research Asia and Redmond. His interests include adaptive and reliable multimedia communications, QoS provision for the next generation wired and wireless Internet, pervasive computing and collaborative systems.

He is a member of ACM.

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

He joined Communication Technology Lab China as Co-Director in September 2004. Prior to his current post, he was with Microsoft Research Asia, Beijing, China, first as a Researcher in Internet Media Group and later as Research Manager of Wireless and Networking Group. From 1996 to 1999, he was with Bell Labs, Lucent Technologies, Holmdel, NJ, as a Member of Technical Staff during 1996-1999. From 1988 to 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and Chinese Academy of Sciences (Institute of Electronics), Beijing, China. He has published over 180 refereed papers in the areas of wireless/Internet multimedia delivery, and wireless communications and networking. He participated activity in the IETF ROHC WG on robust TCP/IP header compression over wireless links. He is co-inventor of over 20 pending patents. His current research interest is in the area of wireless communication and networking, and wireless/Internet multimedia communication and networking.

Dr. Zhu has been on various editorial boards of IEEE journals such as Guest Editor for the *Proceedings of the IEEE*, Associate Editor for IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, and IEEE Transactions on circuits and Systems for Video Technology. He received the Best Paper Award in IEEE Transactions on circuits and Systems for Video Technology in 2001. He currently is also the Chairman of IEEE Circuits and System Society, Beijing Chapter and the Secretary of Visual Signal Processing and Communication Technical Committee. He is a member of Eta Kappa Nu, Multimedia System and Application Technical Committee and Life Science Committee in IEEE Circuits and Systems Society, and Multimedia Communication Technical Committee in IEEE Communications Society.

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

He joined Microsoft Research China, Beijing, in January 1999, leaving his post as the Director of Multimedia Technology Laboratory at Sarnoff Corporation, Princeton, NJ (formerly David Sarnoff Research Center, and RCA Laboratories). He has been engaged in research and commercialization of MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He was with GTE Laboratories Inc. in Waltham, MA and Contel Technology Center in Virginia from 1989 to 1994. He has authored and co-authored over 200-refereed papers in leading international conferences and journals. He 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. He 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 Visual Signal Processing and Communications Technical Committee of IEEE Circuits and Systems. He serves on the Editorial boards of seven other professional journals and over a dozen conference committees. He has been the recipient of numerous awards, including several industry technical achievement awards and IEEE awards such as CAS Jubilee Golden Medal. He was awarded as the "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 prestigious national award as "The Outstanding Young Electrical Engineering of 1998," given annually to one electrical Engineer in the United States.