

Scalable and Adaptive Multicast Video Streaming for Heterogeneous and Mobile Users

Li Lao¹, Jun-Hong Cui², M. Y. Sanadidi¹, and Mario Gerla¹

llao@cs.ucla.edu, jcui@cse.uconn.edu, medy@cs.ucla.edu, gerla@cs.ucla.edu

¹ Computer Science Department, University of California, Los Angeles, CA 90095

² Computer Science & Engineering Department, University of Connecticut, Storrs, CT 06269

Abstract—To provide scalable multicast video streaming services for heterogeneous and mobile users, we design a protocol called SAMP. It is based on a multicast service overlay model. By employing layered encoding, aggregated multicast, and user clustering, SAMP can handle heterogeneity, scalability and mobility very effectively. We conduct simulations and the results show the promising performance of SAMP.

I. INTRODUCTION

With the rapid development of video compression techniques, wireless network technologies and high storage devices, video streaming has received increasing popularity in wired and wireless networks. To provide efficient video delivery among a group of users, multicast has been used as one effective solution. However, the stringent requirements (such as high bandwidth and low delay) of video streaming applications, the heterogeneity of client resources, and the user mobility issue pose several key challenges for video multicast with last-hop wireless mobile users.

First of all, due to both technical and market reasons, IP multicast has been retarded from wide deployment in the Internet. This reality calls for an alternative multicast service model implemented at higher layers in the protocol stack. Second, video streaming applications often span hundreds or even thousands of clients (e.g., Internet TV and large video conferences), and it is also likely that there are simultaneously numerous video applications running in the same network, thus a video multicast scheme should be scalable to large groups as well as a large number of groups (the latter is referred to as “state scalability” issue in the literature). Third, due to the diverse bandwidth capacities and computing capabilities of end users, a video multicast scheme should provide adaptive data rates to receivers in order to maximize the system throughput while efficiently utilizing network resources. Finally, as the wireless technologies become more mature, it is conceivable that more and more wireless users will take part in multimedia distribution. However, most video multicast schemes designed so far concern wired and static users only [18], [17], [20], [16], [2]. How to handle user mobility rapidly and efficiently is still an open issue.

To address the above issues, in this paper, we present a protocol called **SAMP** (Scalable and Adaptive Multicast streaming Protocol). SAMP is based on a service model called **multicast service overlay** [14], which consists of *overlay proxies* deployed strategically in the network. This

service model circumvents the deployment difficulties of IP multicast. SAMP has a two-tier architecture, in which overlay proxies form a multicast service overlay network (MSON) as the backbone logical domain, and end users subscribe to appropriate proxies in access domains. Inside the MSON, SAMP employs an *aggregated multicast* approach to alleviate the state scalability problem. Outside the MSON, end users are grouped around proxies into “clusters”, and the proxies can naturally handle the handoff scenarios of mobile users. To accommodate heterogeneous users, SAMP adopts the layered encoding technique, which encodes video data into layers and allows end users to adjust receiving rate adaptively by adding or dropping encoded layers. We conduct extensive simulations to evaluate the performance of SAMP.

The rest of this paper is organized as follows. Section II reviews background and related work. In Section III, we present our protocol, SAMP, in detail. We then evaluate the performance of SAMP in Section IV, and finally conclude our paper in Section V.

II. BACKGROUND AND RELATED WORK

A. Multicast Models

The concept of IP multicast was first proposed by Deering more than a decade ago [7]. However, it encounters many critical problems, such as the requirement of global deployment of multicast routers, a lack of scalable inter-domain routing protocols, the state scalability issue, and the absence of appropriate pricing models. Thus, it has not been widely deployed in the Internet.

To resolve these issues, two alternative solutions, application layer multicast and overlay multicast, have been proposed. In application layer multicast, multicast functionalities are solely implemented in end hosts (e.g., Yoid [9], ESM [5] and NICE [1]). Though it has high deployability, this approach has an inherent limitation in supporting large-scale multicast groups [15]. Overlay multicast, on the other hand, relies on specially deployed overlay proxies as well as end users to support multicast (ScatterCast [4], RMX [3], OverCast [13], and TOMA [14], to name a few). Overlay multicast is easier to deploy than IP multicast (since it does not require router support) and it can handle large groups very well; but it inherits the multicast state scalability problem from IP multicast (which will be discussed in the following subsection). Considering the trade-off between the two alternative approaches, SAMP

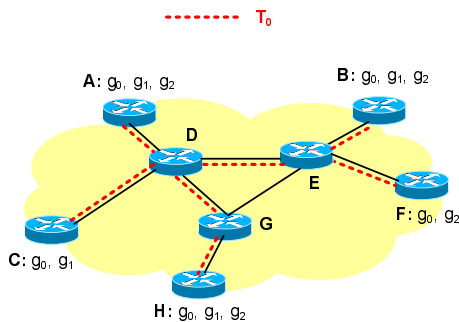


Fig. 1. An illustration of Aggregated Multicast. Groups g_0 , g_1 and g_2 can use an aggregated multicast tree T_0 .

adopts the latter one, overlay multicast, while addressing the state scalability issue.

B. Multicast State Scalability

The state scalability issue has puzzled the multicast research community for a long time. In conventional IP multicast, multicast routers need to maintain multicast forwarding state for every group going through them. Thus, when there are large numbers of groups in the network, this state maintenance requirement translates into a large amount of memory for storing multicast state and correspondingly long packet lookup time. In addition, routers need to send join and leave messages when establishing or destroying multicast trees and exchange refresh messages periodically to manage multicast trees. As a result, when the number of co-existing groups increases, the induced control overhead will grow rapidly. Similar to IP multicast, overlay multicast also suffers from the state scalability problem due to the use of proxies.

Recently, an effective approach called **aggregated multicast** [8] has been proposed to address this issue. It decouples the concept of delivery trees and that of multicast groups: the relationship between groups and trees is not one-to-one any more; instead, many groups may share one delivery tree (called aggregated tree) if they have a similar set of receivers. In this way, the number of multicast trees can be reduced. Further, multicast forwarding entries and control overhead can be significantly decreased. Fig. 1 illustrates how groups are “matched” to trees. The matching between a group and a tree can be either a *perfect match* when every tree leaf is a group member (e.g., T_0 is a perfect match for g_0), or a *leaky match* when some tree leaves do not correspond to group members (e.g., T_0 is a leaky match for g_1 and g_2). Clearly, leaky match has the advantage that more groups can share one tree, even though some bandwidth is wasted for packet delivery to non-member nodes. We employ aggregated multicast in SAMP to improve its scalability to a large number of groups.

C. Layered Encoding

Layered encoding is a popular technique to solve the heterogeneity issue in video streaming with multiple users. The source encodes data streams into multiple layers, each carrying a subset of the original information. Receivers subscribe

to a subset of the layers based on their computation and communication power. There are generally two approaches for video layered encoding, namely, *cumulative* and *non-cumulative*. In the *cumulative* approach, the base layer contains the basic information necessary for decoding, and other layers contain additional information for enhancing the video quality. Receivers need to subscribe to the layers in order from low to high. On the other hand, in the *non-cumulative* approach, all the layers are independent of each other, and receivers can subscribe to any set of layers. It is important to note that layered encoding further aggravates the multicast state scalability problem, since one multicast session now consists of multiple layers (which are equivalent to multiple groups), and more multicast state needs to be maintained for the same number of sessions.

In the literature, many layered multicast protocols have been proposed, including Receiver-driven Layered Multicast (RLM) [18], Layered Video Multicasting with Retransmissions (LVMR) [17], Receiver-driven Layered Congestion Control (RLC) [20], Packet-pair Layered Multicast (PLM) [16], and Fine-Grained Layered Multicast [2]. However, these protocols mainly focus on the congestion control issue for wired multicast. In contrast, our work strives to address all the major challenges mentioned in Section I.

III. PROTOCOL DESCRIPTION

The design goal of SAMP is to provide efficient, scalable and adaptive services for video streaming applications with heterogeneous mobile users. In this section, we first provide an overview of the SAMP architecture, then we present the detailed design of SAMP and discuss how SAMP addresses the key challenges.

A. Overview

SAMP is based on the multicast service overlay model proposed in [14]. In this service model, overlay service providers play the key roles: they deploy overlay proxies, lease network resources (such as link bandwidth) in bulk from network service providers, manage overlay resources, and provide multicast services to clients. Thus, it provides incentives for overlay service providers to deploy multicast services, and circumvents the deployment issues of IP multicast.

The network architecture of SAMP is a two-tier structure, with a multicast service overlay network (MSON) advertised as the backbone domain. Inside the MSON, overlay multicast trees are constructed for video streaming among overlay proxies. Outside the MSON, video servers connect to some proper proxies and “inject” video data to the overlay using unicast. For clients, they subscribe to MSON by choosing appropriate proxies with good performance and receiving video from the proxies by unicast or multicast (we use unicast in this paper for illustration). In this way, each overlay proxy is able to handle tens or hundreds of users; thus, SAMP is envisioned to scale to very large groups.

Fig. 2 shows an example of the SAMP architecture for a single multicast session. In this architecture, the video server

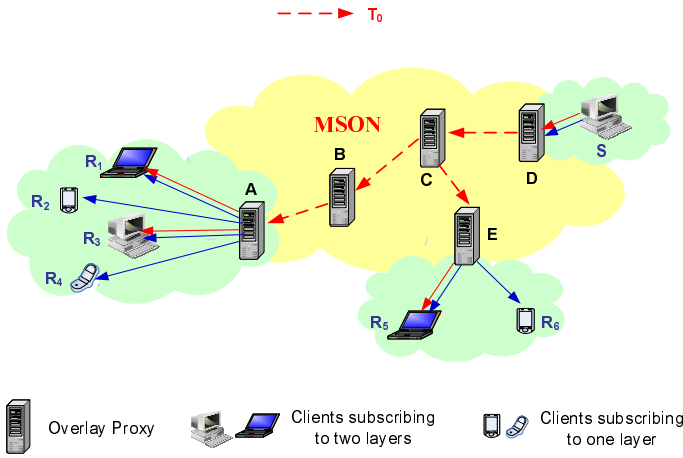


Fig. 2. An example of SAMP.

S is connected to a source proxy D , and receivers R_1, R_2, \dots, R_6 are connected to receiver proxies A and E ¹. When the server S begins to transmit data, it first sends packets to the source proxy D . When the receivers R_1 to R_6 join the network, they first contact a bootstrap node (advertised by the MSON provider) to obtain the addresses of participating receiver proxies (i.e., A and E). According to the measured latencies to the proxies, the receivers select their own closest proxies, to which they initiate join requests. In the example, R_1 to R_4 select A whereas R_5 and R_6 select E . The receiver proxies A and E then request video data from the source proxy D , and cooperatively build a source-rooted multicast distribution tree T_0 on top of the overlay network for data deliver. After receiving data packets from source proxy D via T_0 , proxies A and E perform transcoding and deliver the packets to end users via separate unicast connections. The detail of SAMP will be discussed in the following subsections.

B. Layered Encoding at Servers

To accommodate users with different types of network connections and computing equipments, video data are encoded into layers by servers and transmitted in layers to overlay proxies. For the purpose of illustration, we assume that cumulative layering is used in SAMP. Receiver proxies initially subscribe to the base layer, dynamically adjust their sending rates to users, and add or drop layers in the overlay accordingly. For example, in Fig. 2, receivers R_3 and R_5 have higher capacities and subscribe to two layers, while the remaining receivers only have enough capacity for one layer. Consequently, proxies A and E subscribe to the source proxy based on the highest layer their receivers request, i.e., two layers in this case. Note that, since unicast is used between receivers and proxies, the receivers can adapt their rates based on end-to-end rate control mechanisms such as TFRC[10] and VTP [21], or available bandwidth estimation tools such as IGI [11], pathload [12] and Spruce [19].

¹For simplicity, the applications are assumed to be single-source based. SAMP can be easily extended to support multiple-source applications.

C. MSON Management

Inside the MSON, source proxies send data to receiver proxies via multicast delivery trees. As mentioned earlier, it is very likely that there are large numbers of video streaming sessions running in the MSON. Thus, overlay proxies need to maintain a large amount of multicast forwarding state and produce numerous control messages for multicast tree management. In addition, for each session, the video server employs layered encoding to handle heterogeneous users, i.e., each session forks multiple channels (we refer to a session encoded in one layer as a channel). This makes the state scalability issue even worse since there are more multicast groups (each channel corresponds to one group) to manage.

To solve this issue, SAMP adopts aggregated multicast, simplifying the management of multicast trees and reducing the control overhead of setting up and destroying multicast trees. This approach becomes especially powerful in the scenario of layered multicast. Traditionally, packets for each layer (i.e., each channel) are dispatched on a separate multicast tree, even if they belong to the same session. With aggregated multicast, the layers within a session are more likely to be aggregated onto the same tree, since the members subscribed to these streams tend to be strongly correlated. To a great extent, the additional complexity of tree management brought by layering video streaming is effectively controlled by aggregated multicast. For instance, in Fig. 2, the two multicast channels have the same source and receivers inside the MSON. Hence, it is sufficient to use one aggregated multicast tree T_0 for both channels (or groups).

In SAMP, the major component of aggregated multicast, tree management, is implemented in the source proxy of an aggregated tree. Upon the receipt of a join message from a receiver proxy, the source proxy will conduct group-tree matching (i.e., mapping incoming groups to existing trees) [6] and establish or tear down trees when necessary. In Fig. 2, when the server S sends packets to the source proxy D , D encapsulates the packets using the aggregated tree ID and dispatches the packets onto the aggregated tree. When the packets exit at proxies A and E , the proxies decapsulate the packets and deliver them to end users.

D. End User Clustering

In access domains, end users in proximity are connected to proxies; thus, they form local clusters around proxies. By grouping end users into “clusters”, proxies handle the handoff scenarios of mobile users smoothly. During a video streaming session, the handoff may occur either vertically or horizontally. In the former case, a mobile user switches wireless interface and wireless network technology. In the latter case, the user changes the base station it connects to. In both scenarios, when there are no overlay proxies, the user has to send a new request to the video server and set up a new connection with it. In SAMP, the mobile user simply notifies its proxy directly and updates its connection with the proxy. This could significantly reduce the handoff latency and minimize service interruption, especially when the proxy is already connected to the source

proxy. Again, let us use Fig. 2 to give an example. Assume receiver R_1 moves to the position of R_5 . Without SAMP, R_5 needs to contact source S and wait for the data to be delivered from S . In contrast, with SAMP, R_1 only needs to contact proxy E and receive data directly from it.

E. Group Member Dynamics

In reality, group members dynamically join and leave groups. When a member joins a group, it first selects a nearby receiver proxy based on measurement. If the selected proxy has not joined this group before, it forwards the request message to the source proxy and requests the base layer of the video stream. In this way, the receiver proxy connects to the multicast tree. After the receiver proxy establishes a new connection with the end user, the user is ready to receive video data relayed from the receiver proxy. The member leave process can be accomplished in a similar fashion.

By using aggregated multicast, member join/leave procedure is simplified and expedited. For example, when the group is mapped onto an existing tree, no multicast tree setup is required. In addition, an aggregated tree is removed only when all the groups mapped to it terminate. This feature reduces control overhead and decreases the join latency.

In summary, SAMP use the following techniques to support large-scale video streaming with heterogeneous and mobile users: layered video encoding to handle users with different computation and communication capabilities; aggregated multicast to improve the state scalability exacerbated by layered multicast and reduce the management overhead of multicast trees; end user clustering to support large groups and gracefully handle the handoff scenarios of end users.

IV. PERFORMANCE EVALUATION

We implement a prototype of SAMP in the NS-2 simulator and conduct simulation studies to evaluate its performance. In this section, we present results for three metrics: end-to-end delay, handoff overhead, and tree management overhead.

In the simulations, we use a real network topology, AT&T backbone topology. This network consists of 9 backbone routers, 9 gateway routers and 36 remote access routers. We select the 9 backbone routers as overlay proxy nodes, because they tend to have higher node degree and they are usually located in the center of the network. To match the overlay topology with the underlying network topology, an overlay link is established between two proxies if their network-layer shortest path does not go through other proxies.

For each multicast session, members are randomly attached to remote access routers with a uniform probability, and a source is randomly selected from the members. The total number of members in a session (i.e., session size) is varied from 100 to 300. We assume all the members are capable of moving. When a user moves, it randomly selects an access router as its destination.

For layered encoding, we fix the number of layers within a multicast session to 5 (unless otherwise specified). We assume

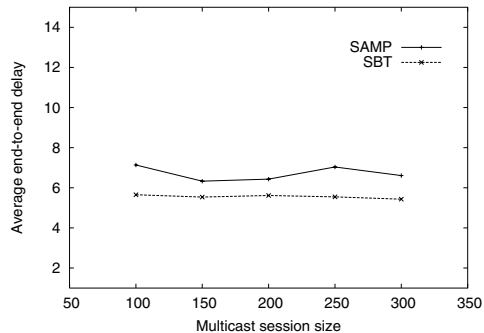


Fig. 3. Average end-to-end delay.

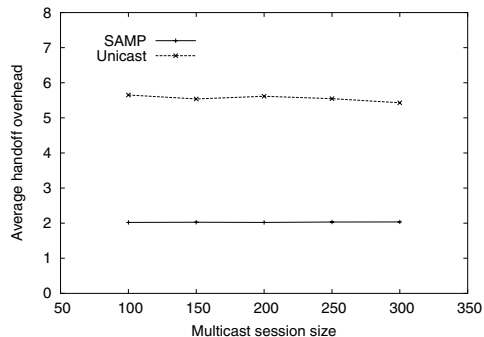


Fig. 4. Average handoff overhead.

a fixed percentage of members (80% in the simulations) that join a layer l ($l < 5$) will also join the higher layer $l + 1$.

End-to-end Delay For video streaming applications, one of the most critical performance metrics is end-to-end delay; thus, we first evaluate the average delay from a source to receivers. We use the number of hops to approximate the end-to-end delay², and the results are plotted in Fig. 3. For the purpose of comparison, we also include the delay of an IP multicast protocol called source-based tree (SBT). From the figure, it is clear that the end-to-end delay of SAMP is comparable to that of SBT. This result demonstrates the promising performance of SAMP, considering that it only requires a small number of proxies to support multicast rather than all the routers in the whole network.

Handoff Overhead Recall that when a mobile user encounters a vertical or horizontal handoff, the handoff overhead and latency can be reduced in SAMP, since the user only needs to send a join request to its nearby proxy instead of the remote source. In this set of simulations, we use the number of hops that a join message traverses to evaluate the handoff overhead.

Fig. 4 depicts the average handoff overhead of each member for SAMP and unicast. In all the cases, SAMP reduces the average handoff overhead by approximately 60%. As session size increases, the average handoff overhead remains fairly stable, since members are randomly distributed in the network. Nonetheless, it is worth pointing out that the total handoff

²We assume the last hop is not congested, as the users continue receiving video streams from the source.

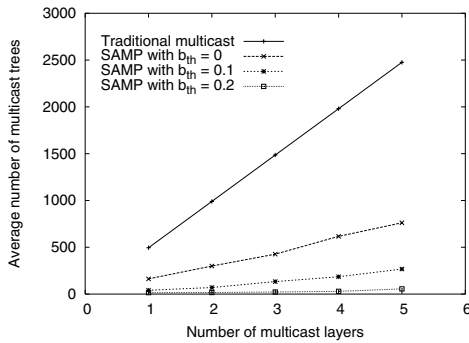


Fig. 5. Average number of multicast trees.

overhead saving is indeed proportional to the session size.

Multicast Tree Management Overhead By using aggregated multicast, SAMP can reduce multicast tree management overhead even when layered encoding technique is adopted. To quantify the overhead reduction, we evaluate the total number of multicast trees needed when there are a large number of sessions in the network. In this set of simulations, the number of sessions is fixed to 500, and the number of layers for each session is varied from 1 to 5 to test its impact on the performance of SAMP. Note that aggregated multicast uses a parameter called bandwidth waste threshold (or b_{th}) to control the trade-off between the degree of aggregation and the bandwidth waste. Intuitively, when more bandwidth waste is allowed (i.e., b_{th} is higher), we can use a smaller number of large trees to cover many small groups, thus reducing the number of trees and improving tree aggregation.

As shown in Fig. 5, SAMP is very effective in reducing the total number of multicast trees when there are multiple layers in each session. For example, when the video stream is encoded into 5 layers, traditional approach would require a total number of approximately 2476 trees. By contrast, in SAMP, only 761 trees are needed to handle all the sessions even when there is no bandwidth waste (i.e., $b_{th} = 0$). As the threshold b_{th} is further increased to 0.3, the number of trees is significantly reduced to 56.

To summarize, our simulation results indicate that SAMP can significantly reduce the handoff overhead of mobile users and tree management overhead of layered multicast, while maintaining a comparable performance to IP multicast in terms of end-to-end delay.

V. CONCLUSIONS

We have proposed an overlay multicast video streaming protocol, SAMP, to provide scalable and adaptive services to multi-user video-streaming applications. In SAMP, we adopt three key techniques, namely, layered encoding to accommodate heterogeneous multicast receivers, aggregated multicast to efficiently manage multicast trees in presence of multiple layers within each session, and user clustering to effectively handle the handoff scenarios of mobile group members. Our simulation study demonstrates that: 1) SAMP can achieve low

end-to-end delay that is comparable to IP multicast; 2) it can significantly reduce handoff overhead for mobile users; 3) in comparison with traditional multicast schemes, it can provide very efficient tree management for layered multicasting.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation for the support of Grant No. 0435515 and Grant No. 0435230.

REFERENCES

- [1] S. Banerjee, C. Kommareddy, and B. Bhattacharjee. Scalable application layer multicast. In *Proceedings of ACM SIGCOMM*, Aug. 2002.
- [2] J. Byers, M. Luby, and M. Mitzenmacher. Fine-grained layered multicast. In *Proceedings of IEEE INFOCOM*, 2001.
- [3] Y. Chawathe, S. McCanne, and E. Brewer. RMX: Reliable multicast for heterogeneous networks. In *Proceedings of IEEE INFOCOM*, Mar. 2000.
- [4] Y. Chawathe, S. McCanne, and E. A. Brewer. *An Architecture for Internet Content Distributions as an Infrastructure Service*, 2000. Unpublished, <http://www.cs.berkeley.edu/yatin/papers/>.
- [5] Y. Chu, S. Rao, and H. Zhang. A case for end system multicast. *Proceedings of ACM Sigmetrics*, June 2000.
- [6] J.-H. Cui, L. Lao, M. Y. Sanadidi, and M. Gerla. Dynamic on-line group tree matching for large scale group communications: A performance study. In *Proceedings of 10th IEEE Symposium on Computers and Communications (ISCC)*, June 2005.
- [7] S. Deering. Multicast routing in a datagram internetwork. *Ph.D thesis*, Dec. 1991.
- [8] A. Fei, J.-H. Cui, M. Gerla, and M. Faloutsos. Aggregated Multicast: an approach to reduce multicast state. *Proceedings of Sixth Global Internet Symposium(GI2001)*, Nov. 2001.
- [9] P. Francis. Yoid: extending the internet multicast architecture. <http://www.aciri.org/yoid/docs/index.html>.
- [10] M. Handley, S. Floyd, J. Padhye, and J. Widmer. Tcp friendly rate control (tfrc): Protocol specification. *IETF RFC 3448*, Jan. 2003.
- [11] N. Hu and P. Steenkiste. Evaluation and characterization of available bandwidth probing techniques. In *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, number 6, Aug. 2003.
- [12] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. In *Proceedings of ACM SIGCOMM*, Aug. 2002.
- [13] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O. Jr. Overcast: Reliable multicasting with an overlay network. In *Proceedings of USENIX Symposium on Operating Systems Design and Implementation*, Oct. 2000.
- [14] L. Lao, J.-H. Cui, and M. Gerla. TOMA: A viable solution for large-scale multicast service support. In *Proceedings of IFIP Networking*, May 2005.
- [15] L. Lao, J.-H. Cui, M. Gerla, and D. Maggiorini. A comparative study of multicast protocols: Top, bottom, or in the middle? In *Proceedings of the 8th IEEE Global Internet Symposium (GI'05) in conjunction with IEEE INFOCOM'05*, Mar. 2005.
- [16] A. Legout and E. Biersack. PLM: Fast convergence for cumulative layered multicast transmission schemes. In *Proceedings of ACM Sigmetrics*, June 2000.
- [17] X. Li, S. Paul, and M. Ammar. Layered video multicast with retransmission (lvmr): Evaluation of hierarchical rate control. In *Proceedings of IEEE INFOCOM*, Mar. 1998.
- [18] S. McCanne and V. Jacobson. Receiver-driven layered multicast. In *Proceedings of ACM SIGCOMM*, Sept. 1996.
- [19] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC)*, Oct. 2003.
- [20] L. Vicisano, J. Crowcroft, and L. Rizzo. TCP-like congestion control for layered multicast data transfer. In *Proceedings of IEEE INFOCOM*, Mar. 1998.
- [21] G. Yang, M. Gerla, and M. Y. Sanadidi. Adaptive video streaming in presence of wireless errors. In *The 7th IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS)*, Oct. 2004.