A Receiver Coordination Protocol for the Efficient Use of Bandwidth in Distributed Multimedia Applications

Hirozumi Yamaguchi Cao Man Letan Akihito Hiromori Takeshi Yamashita
Keiichi Yasumoto† Teruo Higashino Kenichi Taniguchi
Graduate School of Information Science and Technology, Osaka University
Toyonaka, Osaka 560-8531, Japan
{h-yamagu,mlt-cao,hiromori,t-yamast,higashino,taniguchi}@ist.osaka-u.ac.jp
†Graduate School of Information Science, Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan
yasumoto@is.aist-nara.ac.jp

Abstract

In recent distributed multimedia applications such as multi-site teleconferencing, different video streams from different locations are distributed to a number of receivers via multicasting on the Internet, and each receiver simultaneously plays back these videos on his/her terminals. These multicast streams may often be delivered on the same bottleneck link, therefore they are likely to cause competition with each other for the limited bandwidth. In general, layered multicast is considered efficient for avoiding congestion by letting only the affected receivers degrade their receiving rates. However, in case of bandwidth competition by multiple layered multicast streams (inter-stream bandwidth competition), these receivers may choose different streams to be degraded, which may result in slow convergence and low utility of bandwidth. In this paper, we propose a protocol to coordinate those receivers in a distributed manner, for the fast convergence to an optimal layer subscription in the event of inter-stream bandwidth competition, based on application-specific priorities given by receivers to the layers. In our protocol, the number of messages exchanged in the event of congestion is considerably kept low based on tree topology information, obtained by a multicast tree inference technique. Simulation results have shown the effectiveness of our technique on networks with about 200 nodes.

1 Introduction

As the deployment of the high-speed Internet and the computers with highly computing capability, multimedia applications distribute multiple video streams to users on the Internet. For example, in multi-site teleconferencing, each user may watch the video streams from multiple sites simultaneously. The similar situations can be considered in many other applications such as remote lecturing, multipoint live video streaming (baseball etc.). Since each stream consumes much amount of bandwidth, these streams may compete with each other for the limited bandwidth on a single bottleneck link. In such a situation, an appropriate control scheme is desired.

Layered multicast, which uses hierarchical encoding, allows elegant control of transmission rate in heterogeneous environment. In particular, receiver-driven layered multicast (RLM) [6] provides a scalable rate control in end-to-end basis, where receivers who have experienced packet loss decrease their receiving rates by themselves.

We focus on the problem of bandwidth competition by multiple layered multicast streams (called *inter-stream bandwidth competition* hereafter) which is often seen in distributed multimedia applications. Let us assume that two receivers e and f in Fig. 1(a) receive two layered multicast video streams st_a from sender a and st_b from sender b. Let us also assume that each layer consumes 64kbps and thus these two streams compete with each other on link $c-d$ as shown in Fig. 1(b). Both receivers detect congestion via the arrival rates of packets of the two streams, and they try to adapt their receiving rates. Now let us assume that e unsubscribes the second layer of st_a, while f does the second layer of st_b (Fig. 1(c)). In this case, the number of layers of st_a and st_b on the congested link $c-d$ is not decreased, because the second layers of st_b and st_a are still transmitted to e and f, respectively. This leads to the further unsubscription of layers by the receivers like Fig. 1(d), which results in slow congestion avoidance and unnecessary quality degradation (link $c-d$ has capacity to transmit one more layer).

In this paper, for inter-stream bandwidth competition,

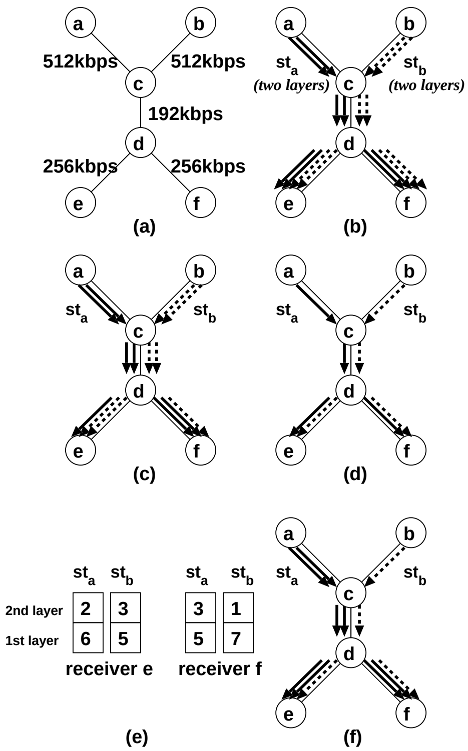

Figure 1. Inter-stream Bandwidth Competition Example

we propose an application-level protocol to coordinate receivers in a fully distributed manner, toward an optimal subscription of layers, based on application-specific priorities given by those receivers to the layers. An example of priorities called utility values is shown in Fig. 1(e), *e.g.* receiver e gives utility value 2 to the second layer of st_a. Given utility values, the optimal subscription of layers is shown in Fig. 1(f) where the sum of the satisfied utility values of two receivers is maximum and bandwidth competition has been dissolved on link $c - d$. In our protocol, assuming that receivers know the tree topology using a tree inference technique, the number of messages exchanged in the event of congestion is considerably kept low using the tree topology. Based on the experimental results using the ns-2 network simulator, our protocol could avoid the congestion within 1 second and prevent unnecessary unsubscription of layers, on networks with around 200 nodes.

1.1 Related Work

Many researches of congestion control at the end level for a single stream of layered multicast have been investigated[8, 9]. In particular, Jagannathan et al.[10] have proposed congestion control algorithm using the knowledge of tree topology. Using a hierarchical (but centralized) architecture, they have discussed the possibility of using tree topology for congestion control. Our approach also relies on tree topology information, however the target situations, goals and approaches are quite different.

Some researches have been dedicated to handle *multiple* streams of layered multicast. Ref. [7] proposes a useful method where the priority of layers of multiple streams are determined by the collection of all users' preferences and a certain set of the layers with the higher priorities are aggregated into one layer. This method can adapt the transmission rate depending on the bandwidth on a bottleneck link. Ref. [11] proposes a receiver-oriented congestion control by receiver coordination. Compared with these approaches, our contribution is that we present a distributed protocol, which uses small number of messages compared with the number of receivers, for optimal subscription of layers in multiple streams' competition. This feature is important in large-scale distributed multimedia applications where the centralized control of the large number of receivers and exchange of the large number of messages between receivers is not practical. Also, optimality is determined based on application-dependent priorities to the subscribing layers, which leads to higher utilization of resources in those applications.

2 Optimal Layer Subscription

We assume that a distributed multimedia application uses layered multicast video streams 1,...,n each of which is transmitted to users $1, ..., m$ (called *receivers* hereafter). For each stream i, i_l denotes the l-th layer of the stream where i_1 is the basic layer and i_{l+1} is used to enhance the quality of video decoded from i_1, ... and i_l. These layers are sent by independent multicast groups. Each receiver can control the number of subscribing layers (*subscription level*) depending on his/her available bandwidth and/or computing capability (CPU power and so on).

We assume that each receiver monitors the packet loss ratios of their receiving streams. Then he/she detects *inter-stream bandwidth competition*, which is caused by competition of more than one layered multicast stream on a single bottleneck link as exemplified in Section 1, by the experience of certain ratios of loss packets (*rate degradation*) on receiving streams within a certain time duration. Two receivers can recognize the sharing of the same inter-stream bandwidth competition if they know that they have experienced rate degradation on the same streams within the same time duration. Note that there may be a case that for two streams, congestion occurs on one stream on a link and also on another stream on different link simultaneously (this is not inter-stream bandwidth competition) and in general it is difficult to distinguish it from inter-stream bandwidth competition at the end level. To filter such congestion, we may be able to analyze jitters or use the sequence numbers of lost packets which characterize path statistics, however, this is out of our scope.

Hereafter, let R denote the set of receivers who have detected the rate degradation of (a part of) their receiving streams at the same time and let V denote the set of all such streams. Let $\downarrow b_j(i)$ denote the degradation rate (bandwidth) of stream i detected at receiver j, at j's subscription level. For example, in Fig. 1(b), if receiver e which receives stream st_a at subscription level 2 (128kbps) detects its rate degradation to 80kbps, $\downarrow b_e(a) = 48$kbps. Also $b(i_l)$ denotes the transmission rate of layer i_l ($b(a_2) = 64$kbps in the same example).

We assume that each receiver j gives application-specific priorities to their subscribing layers as non-negative integer values (a larger value means "more important") [5]. Let $u_j(i_l)$ denotes a value given by receiver j to layer i_l and we call it a *utility value*.

Let $level_j(i)$ denotes the receiver j's current subscription level of stream i, and $opt(i)$ denotes the *optimal layer subscription level* of stream i, which is common to receivers in R. The optimal layer subscription problem for inter-stream bandwidth competition is formulated as a problem to determine $opt(i)$, minimizing the total loss of utility values.

[optimal layer subscription problem]

$$\min \sum_{j \in R} \sum_{i \in V,\ opt(i) < l \leq level_j(i)} u_j(i_l) \tag{1}$$

$$\sum_{i \in V, opt(i) < l \leq \max_{j \in R}\{level_j(i)\}} b(i_l) \geq \sum_{i \in V,} \max_{j \in R} \{\downarrow b_j(i)\} \tag{2}$$

The second inequality means that, the decreased bandwidth on the bottleneck link by receivers' unsubscription of layers must be larger than the lack amount of bandwidth on the link. If receivers in R follow the optimal layer subscription and unsubscribe the layers, the congestion will be dissolved minimizing the loss of utility values.

The problem is a combinatorial optimization problem and we may not find the solution within realistic time for large number of receivers. In such a case, we may use heuristic algorithms, *e.g.* selecting the layer with the minimum utility value per unit of bandwidth iteratively until the required bandwidth is satisfied.

3 Protocol for Optimal Layer Subscription

3.1 Topology Inference of Multicast Tree

Our protocol depends on tree topology information. In this paper, we basically follow the inference technique at the application level presented in [3] to obtain the topology information. Note that network layer support (router support)

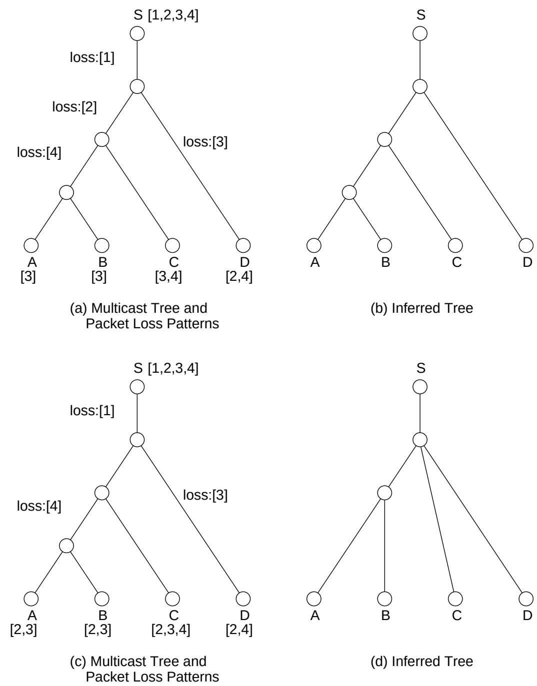

Figure 2. Topology Inference of Multicast Tree

is an alternative approach and in most cases more precise information can be obtained than that by the inference approach. However, our protocol is basically an application-layer protocol, and in this sense, it would be better to incorporate inference-based approaches. Topics in monitoring/measuring multicast trees have widely been investigated and Ref. [2] gives good summarization.

We assume that packets in a basic layer have sequence numbers. Receivers subscribing the basic layer periodically collect the sequence numbers of the lost packets called a *packet loss pattern*.

By collecting the packet loss patterns of the basic layer measured at all the receivers, the topology of a multicast tree can be inferred. Fig. 2 shows an example. In Fig. 2(a), for the packet sequence $1, .., 4$ transmitted by a sender, receivers A and B have received only the packet 3, receiver C has received 3 and 4, and D has received the packets 2 and 4. Knowing that, we can infer that (1) all the receivers share the same bottleneck link where the packet 1 has been lost, (2) the receivers A, B and C share the same bottleneck link where the packet 2 has been lost, (3) the receiver A and B share the same bottleneck link where the packet 4 has been lost, and (4) the receiver D has a bottleneck link where the packet 3 has been lost. According to the above, the tree structure can be inferred as shown in Fig. 2(b). Note that in this case, the inferred tree and the original tree have the same topology, however, it is not true for general cases. For example, if packet loss patterns are measured as shown

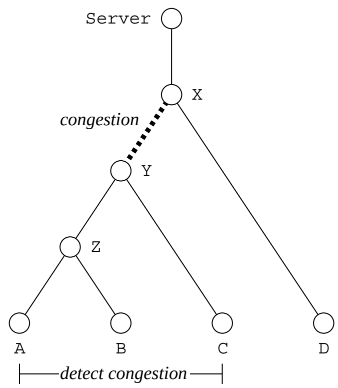

Figure 3. Identifying Receivers Under Congestion: A Simple Example.

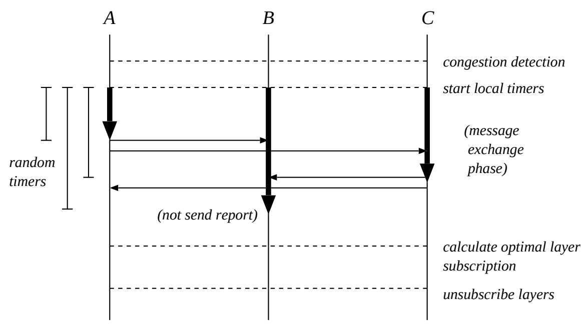

Figure 4. A Message Sequence Chart for the Example in Fig. 3.

in Fig. 2(c), the inferred tree is shown as Fig. 2(d), which is an abstract topology of the original one. We have evaluated, in Section 4, the matching degree between the inferred and the original trees.

3.2 Protocol Description

Our goal is to realize an optimal layer subscription defined in Section 2 in a distributed manner. To this end, all the receivers periodically exchange the information of the receiving streams' information (the subscription level $level_j(i)$ of each receiving stream and the utility value $u_j(i_l)$ for each subscribing layer) as well as packet loss patterns. These messages are called *status reports* and transmitted via a multicast group where all the receivers are the members. Using the packet loss patterns in the status reports, each receiver infers the topology of each multicast tree as described in the previous section. We assume that the transmission rate of each layer $b(i_l)$ is known in advance, for example, at the session initiation phase.

Once inter-stream bandwidth competition is detected by a set R of receivers, each receiver j in R sends messages called *congestion reports* which inform of the rate degradation $\downarrow b_j(i)$ by using the same common multicast group as in the case of status reports. Consequently, receivers in R know the set R of receivers, the set V of streams whose rates are degraded at the receivers in R, the current subscription level $level_j(i)$ of each receiver j in R, utility values $u_j(i_l)$, rate degradation $\downarrow b_j(i)$ of each stream i in V and the transmission rates $b(i_l)$ of the layers. Then the optimal subscription level $opt(i)$ can be determined at receivers in R, and they follow the decision to avoid the competition.

Here, we would like to reduce the number $|R|$ of congestion reports since they are exchanged during congestion and $|R|$ can be large in proportion to the scale of session. In our protocol, using topology information, the number of congestion reports becomes small enough.

A congestion report from a receiver j lets the other receivers in R know (a) the set R and (b) degradation rate $\downarrow b_j(i)$. Here, we assume that the right-hand side of equation (2) that represents the degradation rate of stream i at the bottleneck link can be estimated as follows where $b_{j'}(i)$ is the transmission rate of stream i at receiver j''s subscription level.

$$\max_{j \in R}\{\downarrow b_j(i)\} = \sum_{1 \leq l \leq max_{j \in R}\{level_j(i)\}} b(i_l) * \frac{\downarrow b_{j'}(i)}{b_{j'}(i)}$$

The above estimation allows us to use an arbitrary receiver j''s *degradation ratio* of stream i to estimate the degradation rate at the bottleneck link. Using the above estimation, it is not necessary to receive all the receivers' congestion reports to know $\downarrow b_j(i)$ (only one report is enough for each stream i). Under this assumption, reducing the number of congestion reports depends only on how we identify the set R of receivers without receiving all the congestion reports from receivers in R. We explain how we can do it using tree topology, using a simple example in Fig. 3.

Let us assume that congestion occurs on link X–Y. In this case, receivers A, B and C, who should be the members of R, detect the congestion. Here, since the receivers know the topology of the tree, if the congestion reports from A and C arrive at those receivers, the congestion report from B is no longer necessary to identify R, because those receivers can know that the congestion has occurred above the branch router Y of A and C and that the receivers under the router Y are the members of R. To realize this, we control the timing to send congestion reports. We use a random timer which generates uniform random delay at each receiver. Fig. 4 shows how the receiver B knows that he/she does not have to send his/her congestion report.

3.3 Complexity Analysis

We analyze the amount of status reports and congestion reports.

The analysis of the amount of status reports is very simple. Let us denote the period for sending status reports at a receiver by T_{stat} and the number of receivers by R_{all}. The number of status reports at a receiver per unit of time is R_{all}/T_{stat}.

On the other hand, for the congestion reports, we focus on a path from the server to a receiver (say j). For the receiver j, the necessary and sufficient condition that j does not have to send a congestion report is that the received congestion reports had come to the path from different branch links. For example, considering the path of receiver B in Fig. 3, the congestion reports from A and C had come into the path through the different branch links Z–A and Y–C respectively. In this case, the receiver B knows that congestion occurs at Y or at further place and that he does not have to send his congestion report.

Considering this fact, we analyze the amount of congestion reports. Let us assume that the number of children nodes at each node in the inferred sub-tree rooted at the congestion point is a constant (say K). Thus the number of hops (denoted by H) from the congestion point to a receiver is $H = log_K R$ (R is the set of receivers who have detected the congestion) and the number of the branch links on the path from the server to r is $H * (K-1)$. Let us denote by $N(h)$ the number of receivers contained in the subtree from a branch link which is h hops far from the congestion point. Since $N(h) = K * N(h+1) + 1$ holds, $N(h) \approx K^{H-h}$. If we assume that the random timer at each receiver is set based on the uniform distribution, the ratio of the receivers who had sent congestion reports until at time t is $\frac{t}{T}$ where T is the maximum value for the random timer. Therefore, for a branch link at a router of hop h, the number of congestion reports which had come at time t is $K^{H-h}\frac{t}{T} = \frac{tK^{H-h}}{T}$. Since the average value of h is $\frac{H}{2}$, the average number of congestion reports which had come at time t is $\frac{tK^{H/2}}{T} = \frac{tR\sqrt{K}}{T}$. Here, $P(X)$ is a probability function for the occurrence of phenomenon X. The probability that at least two branch links have congestion reports is:

$$
\begin{aligned}
&1 - P(\frac{tR\sqrt{K}}{T} < 1)^{H(K-1)} \\
&- \quad H(K-1)P(\frac{tR\sqrt{K}}{T} \geq 1)P(\frac{tR\sqrt{K}}{T} < 1)^{H(K-1)-1}
\end{aligned}
$$

Since $P(X)$ is a linear function here, the above is transformed to:

$$
\begin{aligned}
&1 - \left\{\frac{1}{R\sqrt{K}}\right\}^{H(K-1)} \\
&- \quad H(K-1)\left\{1 - \frac{1}{R\sqrt{K}}\right\}\left\{\frac{1}{R\sqrt{K}}\right\}^{H(K-1)-1}
\end{aligned}
$$

This value is almost 1 for a large R, and most receivers can refrain from sending congestion reports.

4 Experimental Results

We have implemented our protocol on network simulator *ns-2* and measured its performance.

4.1 Simulation Setup

The experiments have carried out on tiers networks with 180 nodes. Each network has a WAN where 10 MANs are connected. For each MAN, four or five LANs are connected. The bandwidth of links on WAN, MAN and LAN are 50Mbps, 35Mbps and 10Mbps respectively with 10ms delay and 0.5% of packet loss ratio. In such a network, three multicast servers are located on different LANs, each of which sends four-layered video. The transmission rate of each layer is 0.4Mbps. Also 60 receivers are distributed uniformly on LANs. Multicast trees are shortest path trees. Each receiver inferred trees every 10,000 data packets.

We selected a link on WAN where 9 layers (4 layers of v_1, 3 layers of v_2 and 2 layers of v_3, that is, 3.6Mbps as a total) were transmitted. We reduced the bandwidth of the link to 3.6Mbps and then transmitted an extra UDP stream of 800kbps on the link so that the link becomes a bottleneck link. In this scenario, at least two layers should be decreased on the bottleneck link to avoid congestion.

4.2 Simulation Results

Congestion convergence time: We have measured the time to be taken by congestion avoidance. We focused on a receiver and the packet arrival ratios of three videos v_1, v_2 and v_3, shown in Fig. 5. Note that the maximum value of random timer was set to 0.7s. We can see that the time to avoid congestion is less than 1 second, which is reasonable value for most applications.

The number of congestion reports: We have measured the number of congestion reports varying the number of receivers. Compared with $|R|$, we could achieve much less number of congestion reports.

# of receivers ($\|R\|$)	17	23	28	30
# of congestion reports	5	6	8	8
reduction ratio (%)	71%	74%	71%	73%

The maximum value of random timer vs. the number of congestion reports: The maximum value of random timer, say T, affects the number of congestion reports which are exchanged. If T is small, the number of receivers whose timers expire earlier increases and consequently the number of congestion reports increases, while large T may make congestion avoidance time longer. The results are shown below.

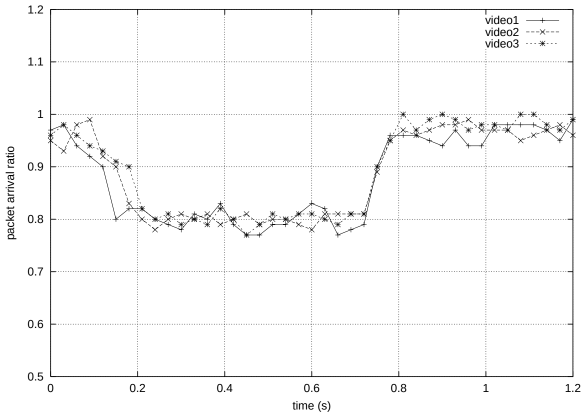

Figure 5. Packet Arrival Ratio Diagram (During Congestion)

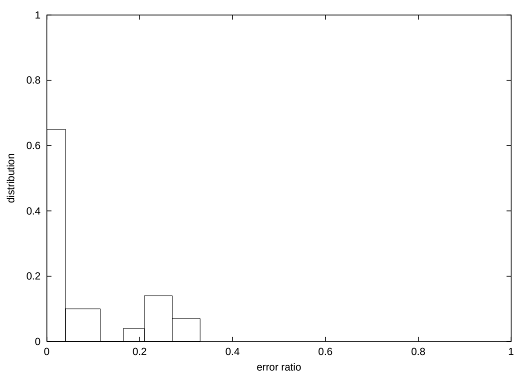

Figure 6. Distribution of Error Ratio from the Original Tree Topologies.

max. value of timer (s)	0.3	0.4	0.5	0.6	0.7
# of congestion reports	9	7	5	6	5

Similarity between inferred and original trees: We have plotted the distribution of error ratios of inference compared with the original tree topologies. 65% of receivers could infer the original topologies, and in the worst cases, the error ratio was 30%.

Receivers' utility: One of our goals is to provide better satisfaction of receivers in terms of their satisfied utility. We have compared our protocol with the method where each receiver autonomously unsubscribes layers.

In our protocol, each receiver decreased at most two layers for the congestion, while in the autonomous method the number of unsubscribed layers was four. This result has shown that higher satisfaction of receivers could be achieved in our protocol.

5 Conclusion

In this paper, we have focused on the problem of bandwidth competition by multiple layered multicast streams which is often seen in distributed multimedia applications, and have provided an efficient solution by an application-level receiver coordination protocol. Based on the experimental results using the ns-2 network simulator, our protocol could avoid the congestion within 1 second and prevent unnecessary unsubscription of layers, on networks with around 200 nodes.

We are planning to conduct further and more detailed experiments for the evaluation and tuning of our protocol, for various types of networks and scenarios. We are also planning to determine a good algorithm for increasing layers. We believe that our coordination protocol is really effective for cooperative layer increasing, and this is part of our future work.

References

[1] C. Diot, W. Dabbous and J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 277-290, April 1997.

[2] K. Samac and K.C. Almeroth, "Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring," *Journal of High Speed Networks*, Vol. 9, No. 3–4, pp. 191-211, 2000.

[3] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Trees and Bottleneck Bandwidths using End-to-end Measurements," *Proc. of IEEE Infocom'99*, 1999.

[4] M. Ghanbari, "Two-Layer Coding of Video Streams for VBR Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 5, pp.771–781, 1989.

[5] N. Shacham and H. Yokota, "Admission Control Algorithms for Multicast Sessions with Multiple Streams," *IEEE Journal of Selected Areas in Communications*, Vol. 15, No. 3, pp. 557-566, 1997.

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

[7] E. Amir, S. McCanne and R. Katz, "Receiver-driven Bandwidth Adaptation for Light-weight Session," *ACM Multimedia*, 1997.

[8] A. Legout and E. Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes," *Proc. of ACM SIGMETRICS '2000*, 2000.

[9] L. Vicisano, L. Rizzo, and J. Crowcroft. "TCP-like Congestion Control for Layered Multicast Data Transfer," *IRTF RM Workshop*, 1997.

[10] S. Jagannathan, K.C. Almeroth and A. Acharya, "Topology Sensitive Congestion Control for Real-time Multicast," *Proc. of the 10th Int. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV 2000)*, 2000.

[11] M. Kawada, H. Morikawa and T. Aoyama, "Cooperative Inter-stream Rate Control Scheme for Layered Multicast," *Proc. of 2001 Symp. on Applications and the Internet (SAINT 2001)*, pp. 147–154, 2001.