GRIDMEDIA: A MULTI-SENDER BASED PEER-TO-PEER MULTICAST SYSTEM FOR VIDEO STREAMING*

Meng Zhang†*, Yun Tang*‡*, Li Zhao, Jian-Guang Luo and Shi-Qiang Yang*

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
{†zhangmeng00, ‡tangyun98}@mails.tsinghua.edu.cn

ABSTRACT

We present a novel single source peer-to-peer multicast architecture called GridMedia which mainly consists of 1) multi-sender based overlay multicast protocol (MSOMP) and 2) multi-sender based redundancy retransmitting algorithm (MSRRA). The MSOMP deploys mesh-based two-layer structure and groups all the peers into clusters with multiple distinct paths from the source root to each peer. To address the problem of long burst packet loss, the MSRRA is proposed at the sender peers to patch the lost packets by using receiver peer loss pattern prediction. Consequently, GridMedia provides a scalable and reliable video streaming system for a large and highly dynamic population of end hosts, and ensures the quality of service in terms of continuous playback, bandwidth demanding and low latency. A real experimental system based on GridMedia architecture has been constructed over CERNET [9] and broadcasting TV programs for seven months [10]. More than 140,000 end users have been attracted with almost 600 simultaneously being online at Aug, 2004 during Athens Olympic Games.

1. INTRODUCTION

With the explosive growth of multimedia services and applications over the Internet, there has been much work in recent years on the topic of streaming video to a large population of users. Although IP multicast is probably the most efficient vehicle, its deployment remains limited due to certain practical issues. As remedies to IP multicast, some research pioneers advocate application layer multicast or overlay multicast and their honorable work includes NARADA [3], HMTP [4], NICE [5], ZIGZAG [6], CoopNet [7], SplitStream [8], to name a few.

However, besides adequate bandwidth and stringent playback continuity for comfortable quality of streaming service, if we take frequent entry and departure of these autonomous end hosts into account, these overlay structures still have some limitations. NARADA [3] and some other tree or mesh-based tree overlay structures involve a tree for data delivering, which is vulnerable with the high churn rate of nodes and enforces enormous delay to those with more depth. NICE [5], ZIGZAG [6], etc. perform distributed algorithm and adopt hierarchical clustering structure. Apparently there are particular internal nodes with heavier load and its failure or departure would have grievous impact to large numbers of descendants. Meanwhile, there is only one supplier for each peer which could not meet the bandwidth demanding for video streaming. In CoopNet [7], the source root takes responsibility to gather information of all nodes for structure maintenance, so it relies on a powerful root node and offers low scalability. SplitSteam [8] advocates the use of multiple distribution trees to evenly distribute forward load across the nodes by making a node a leaf in all but one tree and assumes that uninterested nodes will be available to forward traffic.

Accordingly, GridMedia is motivated by various issues when we face the scenario to provide a scalable and resilient large scale video streaming system and ensure agreeable quality for end users. Our main contributions are two folders. 1) We adopt mesh-based two layer overlay structure to provide multiple distinct paths from the root to each receiver peer, which ensures the demanding bandwidth for video streaming and alleviate network congestion with help of transmission algorithm. 2) To address the problem of long burst packet loss caused by network congestion and peer failure, we propose a redundancy retransmission algorithm based on multi-sender. The collaboration among multiple senders patches the episode of lost packets and more importantly, the redundancy introduced by this algorithm is upper-bounded by link loss ratio. The rest of this paper is organized as follows. Section 2 gives an overview of the overlay protocol MSOMP. Section 3 details the transmitting protocol MSRRA and presents a numerical analysis, followed by the experimental system and statistical result in Section 4. Finally, we conclude this paper in Section 5 and discuss some future work.

2. MSOMP IN GRIDMEDIA

In this section, we describe the Multi-Sender based

*Supported by the National Natural Science Foundation of China under Grant No. 60273008 and No. 60432030

Overlay Multicast Protocol (MSOMP) to present the mesh-based two layer structure in GridMedia. Due to page limitation, key design ideas are briefly highlighted and more details could refer to our technical report [1]. In the following, end host, peer and node could be used alternately unless declared explicitly.

As a single source streaming overlay protocol, MSOMP originates from the streaming server which is a node at the root. And we employs a well-known Rendezvous Point (RP) to help each new joining peer bootstrap, which is similar to other decentralized application layer multicast protocol. Besides, MSOMP will utilize the existing IP multicast service which is available in many LANs. Here define a term *IP Multicast Domain (IMD)* which is a local network of any size that supports IP multicast. An IMD could be a single host, a LAN, etc. In each IMD, there is a header peer which is responsible for disseminating streaming content to other peers in the same IMD. As soon as the header leaves, a new header will be elected to replace the original role. What MSOMP should do is to connect these IMDs by unicast tunnel altogether.

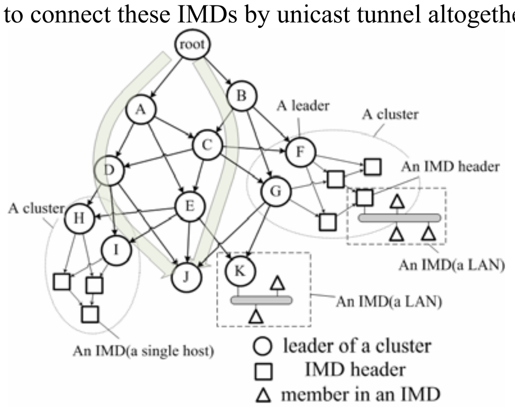


Fig. 1 GridMedia architecture based on MSOMP

In MSOMP, all the peers self-organize into a two-layer structure. At first these peers are grouped into clusters in terms of End-to-End Delay (EED) to form the lower layer. Note that the size of each cluster has upper bound. Then with one or several leaders in each cluster, all the leaders construct the backbone of the overlay to build the upper layer. MSOMP provides each leader with multiple parents to receiver distinct streams simultaneously. To make stress at each peer lower, an upper bound of out degree is assigned at each node. Moreover, MSOMP uses a heuristic scheme to induce all the participants self-organize into a mesh with low delay to the root. Fig. 1 illustrates such architecture.

As mentioned previously, since all the participants form a mesh structure, it is more robust than tree based data delivery. Any single host failure or single link switching will not destroy the connectivity of the overlay. Meanwhile, there are usually multiple independent paths from the root to a single node, as illustrated in Fig. 1, so we could use particular transmitting algorithm to patch one sender's data with the other's, which depresses the network congestion. Evidently, the more parents each host has and the more leaders each cluster has, the more robust the overlay is. However, there is a delay tradeoff for each host.

Fig. 2 shows the system diagram of a peer, including three key modules: (A) inter-cluster overlay manager maintaining the structure between clusters, (B) intra-cluster overlay manager taking charge of the structure inside a cluster and (C) streaming manager scheduling the streams by resilient transmission algorithm.

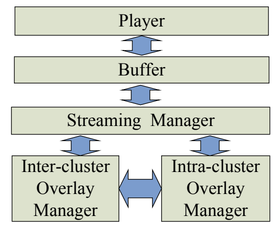


Fig. 2 A diagram of peer

More discussion about the construction and refinement of the overlay, including node join, departure and recovery could be found in technical report [1].

3. MSRRA IN GRIDMEDIA

After constructing overlay for data dissemination, it comes to the question how to schedule distinct streams felicitously with low redundancy and packet loss. Here we would explain and analyze Multi-Sender based Redundancy Retransmission Algorithm (MSRRA) to emphasize how to address the issue of packet loss.

3.1. Overview of MSRRA

In peer-to-peer streaming multicast, peer failure and link switch operation due to peer departure usually cause long burst loss of packets. However, the two main traditional channel coding techniques, for instance FEC (Forward Error Correction) and ARQ (Automatic Repeat reQuest) mechanisms are not suitable for error control in peer-to-peer streaming multicast since FEC is not efficient in burst loss environments while ARQ may bring cumulative delay in hop-by-hop transmission. To solve the problem of long burst loss of packets with low delay, we propose a retransmitting algorithm MSRRA with a small quantity of redundancy. Before demonstrating this algorithm, here first introduce a term called Retransmit Delay (RD). The RD of a packet is the cumulative delay caused by retransmission from the root to the peer. MSRRA uses receiver peer loss pattern prediction which can efficiently reduce the RD of packets, and the redundancy of MSRRA is upper-bounded.

In MSRRA, each receiver peer obtains streaming packets simultaneously from multiple senders. Every sender peer transmits part of the streaming content. As soon as there is congestion occurring on one link, the receiver will take notice of this congestion by a timeout *T*. Subsequently it delivers an NACK to notify other senders who will continuously patch the episode of lost packets until there is an arrival of ACK. Obviously each sender peer should be provided with knowledge of packets delivered by itself as well as packets sent by other senders.

The following instance will demonstrate MSRRA in detail. For simplicity, in Fig. 3 there are two sender peers,

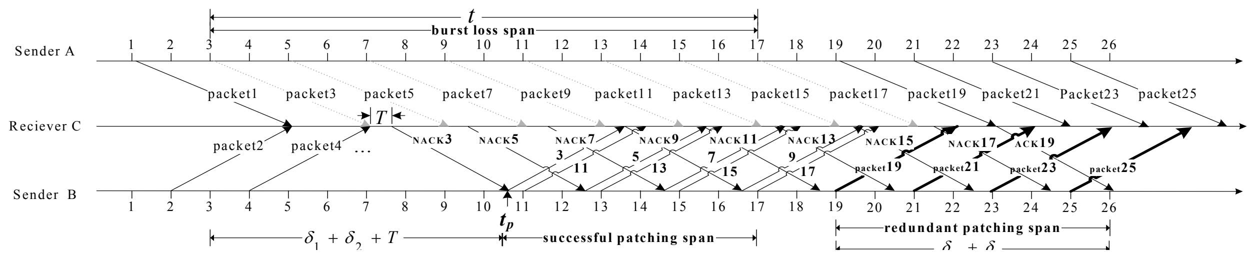


Fig. 3 A case for MSRRA

A transmitting packets with odd sequence numbers and B forwarding packets with even sequence numbers, while C is the receiver peer. The delay between A and C is 3 time units and the delay between B and C is 2 time units. Here only illustrates the case that the packets transmitted by sender A and the patching packets delivered by sender B. Due to routing congestion between sender A and receiver C, packets with odd sequence numbers from 3 to 17 are lost. Receiver C detects the loss of packet 3 by a timeout T, and then delivers an NACK to sender B. When sender B receives the NACK of packet 3 at time t_p, it enters a *patching session*. Then sender B will continuously forward the packets received after t_p which should have been delivered by sender A (the packets with odd sequence numbers from 11 to 25) till it is notified of an ACK of packet 19 which terminates the session, as shown in the Fig. 3. But the packets before t_p (sequence number 3, 5, 7, 9) should be retransmitted exactly when their NACKs arrive. In this case, the redundancy of retransmitting packets are the packet 19, 21, 23, 25 that are patched in a span during which packet 19 is delivered from sender A to the receiver and its ACK comes to sender B.

3.2. Numerical Analysis of MSRRA

In this part, we will analyze MSRRA with some mathematical models and compute two metrics. One is the ratio of number of the packets with RD to that of the total packets, represented by r_d and the other is the ratio of number of the redundant packets to that of the total packets represented by r_r.

To simplify the scenario of MSRRA, we assume that there are only two senders, each of whom delivers half of the constant bit rate (CBR) stream. The EED between any two nodes is symmetric and each packet has the same size. In addition, it is assumed that the congestion does not occur at the entry gateway of the receiver.

The network channel can be well-approximated by a simple model of two-state Markov continuous-time chain [2], in which the channel is supposed to have only two states – the good state and the bad state. The parameter λ represents the transition rate from the good state to the bad state, while μ is defined to be the transition rate from the bad state to the good state. The average durations in good state and bad state are $1/\lambda$ and $1/\mu$, respectively. And the stopping time at each state has an exponential distribution with parameter λ and μ, respectively.

In the case of two senders, let λ_1, μ_1 and λ_2, μ_2 denote the parameters of the two channels. the current state of each channel could be represented by a state pair (s_1, s_2), where s_1, $s_2 \in \{g, b\}$. Then we could figure out the following composite channel model, as depicted in Fig. 4.

When the current state is (g, b), the sender A will evoke the patching session, and vice versa. The stationary probability of each state is:

$$\pi_{gg} = \mu_1\mu_2 / \Delta, \quad \pi_{bg} = \lambda_1\mu_2 / \Delta$$
$$\pi_{gb} = \mu_1\lambda_2 / \Delta, \quad \pi_{bb} = \lambda_1\lambda_2 / \Delta$$

where $\Delta = \lambda_1\lambda_2 + \lambda_1\mu_2 + \mu_1\lambda_2 + \mu_1\mu_2$

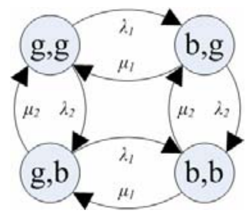


Fig. 4 Composite channel model for two senders

Let δ_1 denote the EED between sender A and the receiver, while δ_2 denote the EED between sender B and the receiver, and let $\delta=\delta_1+\delta_2$. The interval between two packets is τ. The receiver will send an NACK if a packet doesn't arrive within a time of T. Since each sender delivers half of the stream, the average loss ratio is $r_{loss} = (\pi_{bg} + \pi_{gb}) / 2 + \pi_{bb}$.

Now we could compute the average ratio of number of the packets with RD to that of the total packets delivered by all senders in a patching session of sender B. Here use $E(r_{d1} \mid s_1 = b, s_2 = g)$ to denote this, so

$$E\left(r_{d1} \middle| s_1 = b, s_2 = g\right) = \int_0^{\delta+T} 1/2 \cdot \mu_1 e^{-\mu_1 t} dt + \int_{\delta+T}^{+\infty} \frac{\lceil (\delta+T)/2\tau \rceil}{2\lceil t/2\tau \rceil} \mu_1 e^{-\mu_1 t} dt$$

Likewise, $E(r_{d2} \mid s_1 = g, s_2 = b)$ is calculated, and then r_d could be computed as:

$$r_d = E(r_{d1} \mid s_1 = b, s_2 = g)\, \pi_{bg} + E(r_{d2} \mid s_1 = g, s_2 = b)\, \pi_{gb}$$

Similarly, the average ratio of number of the redundant packets to that of the total packets delivered by all senders in a patching session of sender B could be represented by $E(r_{r1} \mid s_1 = b, s_2 = g)$:

$$E\left(r_{r1} \middle| s_1 = b, s_2 = g\right)$$
$$= \int_T^{\delta+T} \frac{\left[(t+\delta)-(\delta+T)\right]/2\tau}{2\lceil t/2\tau \rceil} \mu_1 e^{-\mu_1 t} dt + \int_{\delta+T}^{+\infty} \frac{\delta/2\tau}{2\lceil t/2\tau \rceil} \mu_1 e^{-\mu_1 t} dt$$

Likewise, $E(r_{r2} \mid s_1 = b, s_2 = g)$ is calculated, and

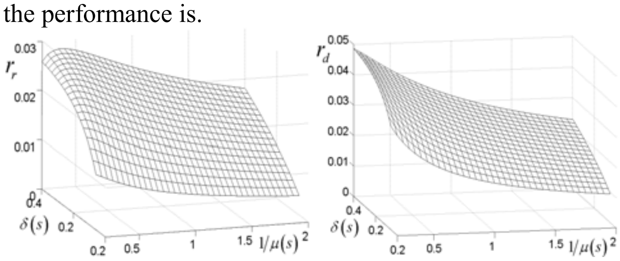
$$r_r = E(r_{r1} \mid s_1 = b, s_2 = g)\, \pi_{bg} + E(r_{r2} \mid s_1 = g, s_2 = b)\, \pi_{gb}$$

The following inequation indicates that the redundancy of MSRRA could not exceed link loss ratio

$$E\left(r_{r1}\right) < \int_T^{\delta+T} 1/2 \cdot \mu_b^A e^{-\mu_b^A t} dt + \int_{\delta+T}^{+\infty} 1/2 \cdot \mu_b^A e^{-\mu_b^A t} dt < 1/2$$

So $r_r = E(r_{r1})\, \pi_{bg} + E(r_{r2})\, \pi_{gb} < r_{loss}$

Fig. 5 shows the numerical result of MSRRA in the scenario of two senders. Let the NACK timeout T be 30 ms and the sending rate be 100 packets per second, τ=10ms. Assume that the two channels have the same parameters, and the average loss ratio of the two channels is both 5%. The average bad time μ_1, μ_2 varies from 10 ms to 400 ms, and the sum of the EED of the two channels δ varies from 60 ms to 400 ms. Fig. 5(a) shows the ratio of number of the packets with RD to that of the total packets r_d, while Fig. 5(b) shows the redundancy ratio r_r. In Fig. 5(a), we can observe that there are few redundant packets when the average bad time is very short because when an NACK comes, an ACK will arrive immediately to terminate the patching session. Note that the redundancy ratio doesn't exceed the loss ratio 5%. We can also discover from Fig. 5(b) that the longer the average bad time is, and the better the performance is.



b) The ratio of the redundant packets to the total packets

(a) The ratio of the packets with RD to the total packets

Fig. 5 Numerical analysis of MSRRA (channel loss ratio 5%)

In fact, MSRRA achieves splendid performance with low redundancy and overhead in the scenario of high churn rate and link switch operation in peer-to-peer multicast.

4. EXPERIMENTAL RESULT OVER CERNET

An implementation of GridMedia over CERNET [9], called *GridMedia Player v1.0*, was released on May 20, 2004, [10]. It has been broadcasting TV programs for seven months during which several significant and exciting events were live broadcasted. The encoding rate approaches 500kbps and more than 140,000 person-times have been logged. To evaluate continuity of playback, we define *continuity index* as the ratio of number of packets that arrive before or on playback deadline to the total number of packets.

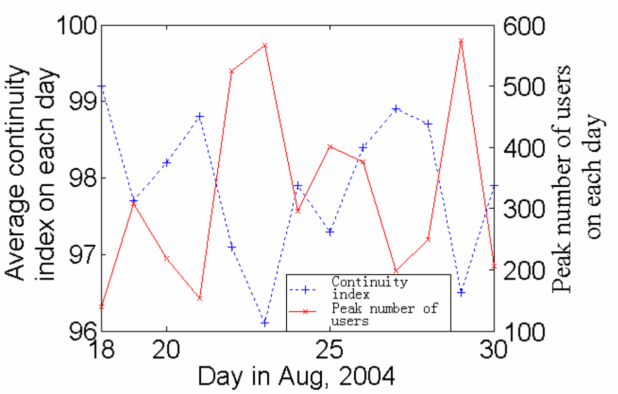


Fig. 6 Average continuity index and peak number of users during Athens Olympic Games

Fig. 6 depicts two statistical metrics during the period of Athens Olympic Games, the left y-axis demonstrates average continuity index on each day while the right y-axis describes the peak number of users simultaneously online on each day. We could figure out that not only were almost 600 simultaneous sessions attracted for particular content with reference to its popularity, but the average continuity index always keeps excellent record to be beyond 96%, which confirms the visual quality during viewing.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a peer-to-peer based multicast architecture called GridMedia which is designed especially for large scale video streaming with quality requirement in terms of real time, low latency and bandwidth demanding. This architecture mainly contains the overlay protocol and the transmitting algorithm. The overlay protocol MSOMP advocates mesh-based two layer structure, which makes the upper layer much more robust than traditional tree-based scheme and ensures the demanding bandwidth. The MSRRA algorithm efficiently relieves the impact of nodes failure, network congestion and link switch operations. The redundancy of MSRRA would not exceed link loss ratio. Now we are anatomizing log information and try to demonstrate user characteristics to guide the improvement to such system. We are also about to do some experiments on the platform of Planet-Lab.

6. REFERENCE

[1] Technical Report, "GridMedia- Large Scale Video Streaming System", Department of Computer Science and Technology, Tsinghua University.

[2] T. Nguyen, A. Zakhor, "Multiple Sender Distributed Video Streaming", IEEE Transactions on Multimedia, vol. 6, no. 2, pp. 315-326, April 2004.

[3] Y. Chu, S. G. Rao, and H. Zhang. "A case for end system multicast". In Proceedings of ACM SIGMETRICS, pages 1–12, June 2000.

[4] B. Zhang, S. Jamin, and L. Zhang, "Host multicast: a framework for delivering multicast to end users," in Proceeding of IEEE INFOCOM, 2002.

[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in Proceeding of ACM SIGCOMM, 2002.

[6] D. A. Tran, K. A. Hua, "A Peer-to-Peer Architecture for Media Streaming", IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, January 2004.

[7] V. N. Padmanabhan, H. J. Wang, P. A. Chou,"Distributing streaming media content using cooperative networking", Proc. NOSSDAV'02, USA, May 2002

[8] M. Castro, P. Druschel, A.-M. Kermarrec, "SplitStream: high-bandwidth multicast in cooperative environments," in Proc. ACM SOSP'03, Oct. 2003.

[9] The China Education and Research Network (CERNET), "http://www.edu.cn/HomePage/english/cernet/index.shtml"

[10] Our website: "http://www.gridmedia.com.cn"