2004 International Conference on Image Processing (ICIP)

A PEER-TO-PEER VIDEO-ON-DEMAND SYSTEM USING MULTIPLE DESCRIPTION CODING AND SERVER DIVERSITY

Xiaofeng Xu, Yao Wang, Shivendra S. Panwar, Keith W. Ross
Polytechnic University, Brooklyn, NY 11201, USA
Email: xiaofeng@vision.poly.edu,{yao,panwar,ross}@poly.edu

ABSTRACT

We propose a video streaming system that uses many ordinary computers (peers) as servers, which is less costly and more scalable than using a single or a few dedicated servers. To circumvent frequent peer disconnects and limited peer up-link bandwidth, each video is coded into multiple descriptions, which are distributed over multiple peers. The system serves a client request by streaming multiple descriptions of the requested video from separate peers. Using the MD-FEC coding scheme, and simple models for peer disconnects, reconnects and for searching replacement serving peers after a peer disconnect, we simulated video streaming in such a system. Our results show significant benefits in using a large number of descriptions and serving peers.

1. OVERVIEW OF THE PROPOSED SYSTEM ARCHITECTURE

In a conventional infrastructure-based video-on-demand system, compressed videos are stored in one or a few servers. One problem with this architecture lies in the tremendous cost for setting up the servers, which must have very large storage space, ultra-fast bandwidth, and high reliability. To alleviate the above problem, we propose to use many ordinary computers as servers and store the video content over these computers. In the context of peer-to-peer (P2P) networking, these computers are called peers or nodes. Each node has a certain storage space and uplink bandwidth reserved for participating as a serving node. Compared to the central server approach, the proposed system architecture requires very low initial set-up cost and can be more scalable and reliable.

To circumvent frequent peer disconnects and limited uplink bandwidths at the peers, we propose to encode each video into multiple descriptions and to place each description on a different server. When a client wants to see a video, multiple peers act as servers, each sending a different description of the video to the client. After a short initial delay, the client combines, decodes, and displays the video as the multiple descriptions are being delivered. When a serving peer disconnects in the middle of a streaming session, the system will look for a replacement peer storing the same video description and which has sufficient uplink bandwidth. Figure 1 illustrates the proposed system architecture. The server selection and replacement can be done in either a centralized manner (for example, by a central server governing all peers in the system) or decentralized manner (for example, by the requesting client).

The benefit from coding a video into multiple descriptions and distributing these over separate nodes are several fold: 1) When a serving peer goes down, it only causes temporary loss of a single description, which has limited impact on received video quality. 2) Each description has a rate much lower than the total rate of a video, thus reducing the required uplink bandwidth at peers. 3) Splitting a video file across multiple nodes helps to reduce the load on each serving node, which is an important design factor in a P2P application. 4) Finally, from the service provider point of view, it is desirable not to store any video entirely in one node, to prevent the node from having illegal access to this video.

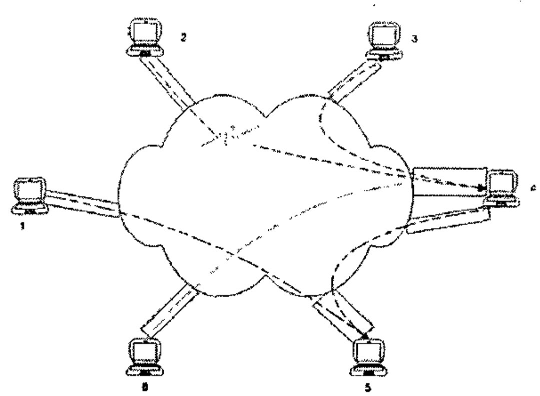

Figure 1. Proposed System Architecture: In this illustrative example, two simultaneous streaming sessions are requested from node 4 and node 5. The system initially selects nodes 2 and 3 to serve node 4's request, and select nodes 4 and 1 to serve node 5's request. After Node 3 goes down, the system finds node 6 as a replacement. We use a fat pipe to indicate the downlink of each node, and a thin pipe to illustrate the uplink of each node. A node can simultaneously function as a server and client (node 4), or as a server or client only.

Recently, several video multicast systems making use of peer coordination have been studied [1-8]. But in these systems, the video content is stored in a central server, the peers merely help to relay a video originated from the server, by forming one or several multicast trees, and peers only help each other when they watch the same video. In order to reduce quality degradation due to peer disconnect, several groups assumed that the media is coded into multiple descriptions and different descriptions are sent from separate multicast paths [5-8]. However, performance evaluation with actual coded bitstreams was not considered, except the work in [5].

The work in [9] considers on-demand media streaming. As with our proposed system, media streams are stored in many peers, and a client streams different parts of a media from multiple peers simultaneously. However, this project so far has not considered actual media coder design. It assumes that a pre-coded media (by some standard codecs) is split into multiple segments. Their work focused on how to assign media segments to supplying peers (i.e. how many segments each peer sends) and how to determine the channel coding rate for protecting the assigned segments. The work in [10] considered streaming multiple description video from multiple (two in their implementation) servers in a infrastructured-based content-delivery network and evaluated the benefit of integrating multiple description coding with server diversity.

The remainder of this paper is organized as follows. Section 2 describes our network models and FEC-based multiple description source coding (MD-FEC) scheme. Section 3 presents our simulations results and investigates the impact of description number and network parameters on video quality. Section 4 concludes the paper and suggests directions for future work.

2. SYSTEM MODELS AND ASSUMPTIONS

2.1 Network Model

We assume that there are a total of N homogeneous nodes (peers), each with storage capacity C, uplink bandwidth R_u, and downlink bandwidth R_d. Note that C and R_u represent the storage space and uplink bandwidth reserved for the peer-to-peer video streaming service by each node. As shown in Figure 1, each node is connected to a backbone network that is assumed to have infinite bandwidth and no congestion (and hence no packet loss and negligible delay). The communication channel (in terms of delay, bandwidth, loss) between two nodes is only limited by the uplink and downlink (bandwidth and availability) of the two nodes to/from the backbone.

2.2 Node ON/OFF models

We assume that each participating peer alternates between "on" and "off" status. The ON time is how long a node would stay in the network as a server node. We model it as an exponentially distributed random variable $T_{on} \sim \lambda e^{-\lambda t}$. Similarly, the OFF time is the time period when a node cannot provide the streaming service to others. We model it by another exponentially distributed random variable $T_{off} \sim \beta e^{-\beta t}$. For every node, we generate a sequence of $T_{on}(1)$, $T_{off}(1)$, $T_{on}(2)$, $T_{off}(2)$,..., based on the above models. This sequence represents the time intervals during which the node is on or off.

2.3 Video request and serving node selection

The number of new requests within a given time interval (e.g., 1 second) is modeled by a Poisson arrival process. When a node requests a video, we assume that a central manager will try to find M serving nodes that have the M descriptions of the video and each node also has sufficient available uplink bandwidth to serve one description. The central manager chooses M serving nodes based on two criteria: (1) the chosen nodes are those who have the maximum uplink bandwidth among all ON nodes in the network; (2) Each node serves only one description.[1] Since we enforce that a node must download different descriptions from M distinct nodes, the entire service load is evenly distributed among all nodes. Also, the disconnect of one chosen serving node in the middle of the streaming session will affect only one description.

2.4 Modeling of description loss rate

As described in Sec. 2.1, we assume that the uplink and downlink of each node have negligible bit error rate and packet loss rate. Furthermore, we assume a node will request a streaming video only if it has sufficient unused downlink bandwidth, and a node is chosen for serving a description only if it has sufficient available uplink bandwidth. Under these assumptions, there will be no loss incurred by congestion at the serving node or receiving node. Once a node has been chosen to deliver a description upon the start of a streaming session, the packets in this description are delivered error free, until the node becomes unavailable. The node disconnect rate is λ according to section 2.2. Once a node becomes unavailable, the server will try to find a replacement node that stores the same description and has sufficient available uplink bandwidth. Assume that the time to find a replacement node is an exponential random variable, with mean $1/\gamma$. For a given substream, the receiving node will see alternating periods of receiving the stream and receiving nothing. We can model the above process as a two state (receiving vs. non-receiving) Markov process. The transition probability from receiving to non-receiving is λ; the transition probability from non-receiving to receiving is γ. The steady state probability in the non-receiving state is $\varepsilon = \frac{\lambda}{\gamma + \lambda}$, which is the probability that a description is lost at any time instant.

Recall that we assume the M descriptions of a given video are stored in M different nodes. Assuming the disconnect and finding replacement events at different nodes are independent, the probability of missing m out of M descriptions at any time is:

$$P(m, M) = \binom{M}{m} \varepsilon^m (1-\varepsilon)^{M-m} \qquad (1)$$

2.5 The MD-FEC video coder

One simple way of generating M equally important and equal rate descriptions is by combining a scalable coder and forward error correction (FEC) codes [10]. In the most general case of this coder, in each time interval T_{seg} (usually a group of frames or GOF), one partitions all the coded bits for this interval (R bits) into M groups with the m-th group ending at R_m bits (m = 1,2, ..., M). Then each group of R_m-R_{m-1} bits is further divided into m sub-groups of equal length. Next a Reed-Solomn (M,m) channel code is applied across the m subgroups, to generate M sub-groups. Description m for this time interval is formed by concatenating subgroups m from all M groups and is put into a separate packet.

[1] Unless less than M nodes are available that have sufficient available bandwidth, in which case some nodes may serve more than one description.

With the MD-FEC scheme, receiving m descriptions will enable correct decoding of up to R_m bits. The total rate after using FEC depends on the rate partition $(R_1,R_2,...,R_M)$. In our implementation, we obtain $p(m,M)$ either based on the model in (1) or from network simulations. Given $p(m,M)$ and $D_s(R)$, we use the algorithm in [11] to find the optimal rate partition that minimizes the expected distortion for a given total rate.

3. SIMULATION RESULTS

We simulated the proposed peer-to-peer video streaming system, using the node ON/OFF model and the replacement time model, described in Sec. 2. At present, we have not explored how to distribute descriptions from multiple video sources with different popularity over all participating nodes. The present simulation considers one video source only and every node stores all descriptions of this video. We varied the number of descriptions and the mean replacement time, to see the impact of these parameters on the received video quality.

3.1 Simulation setting

(a) Video Data: We coded the "Foreman" video sequence in CIF (352x288) resolution into a scalable bit stream using the MPEG-4 FGS codec [12], at a base layer rate of 150 Kbps. Each GOF has a duration of 1 second and comprises of 15 frames. The output bits from each GOF are converted to M descriptions using the MD-FEC method, where M is varied from 4 to 64. The rate partition for a particular network setting is calculated based on the measured $P(m,M)$ from the network simulation. The total rate of a video after MD-FEC is constrained to be 768 Kbps. (The video source rate is below 768 Kbps and depends on $P(m,M)$).

(b) Network settings: Our simulated network has N=500 nodes. The uplink bandwidth at each node is R_u=250 Kbps. The average on/off time ($1/\lambda$ or $1/\beta$) of a node is 20/5 minutes. The arrival rate of new requests is 1 per second. The replacement time ($1/\gamma$) is varied from 10 seconds to 1 minute. These parameters are chosen so that the network is *not overloaded* in the sense that M distinct serving nodes can be typically found for each new request. The total simulation time is 1800 minutes. We assume the length of each streaming session is 20 minutes. Although the length of a typical video file in video-on-demand applications can be much longer, we chose a shorter time, so that within our network simulation time, more streaming sessions can be invoked.

3.1. Validation of the Model for $P(m,M)$

Figure 2 shows the measured distribution $P(m,M)$ from our simulated network and the one calculated using the model in Equation (1). We can see that when the network is not overloaded, the model describes the actual distribution accurately.

3.2. Impact of the description number

We first examine how the video quality varies with the number of descriptions, when the average on/off time set to 20 and 5 minutes, respectively, and the average replacement time is 1 minute. Figure 3 shows the average PSNR for M= 4, 8, 16, 32, 64, respectively. From this figure (the curve for replacement time=60 sec.), we can observe that the PSNR improves as the number of descriptions increase, but the improvement gradually saturates as M becomes very large (beyond 32).

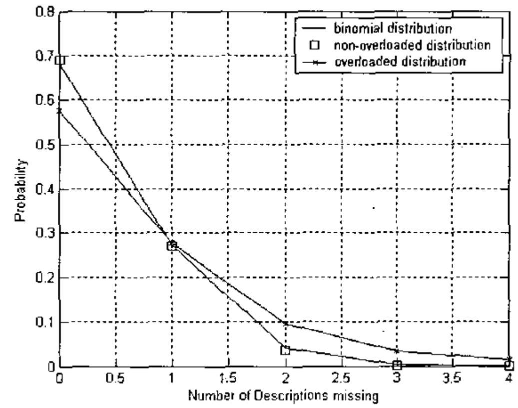

Figure 2. The measured and modeled P(m,M) for M=4, Replacement Time=60 Sec.

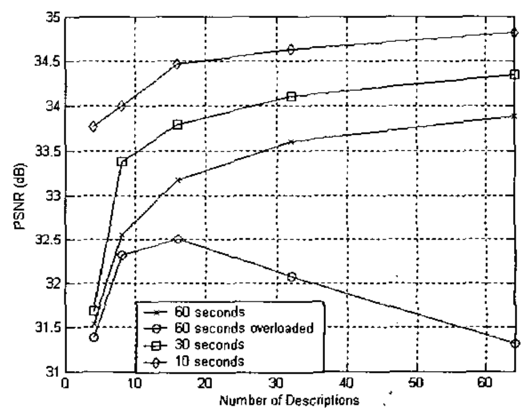

Figure 3. Average PSNR vs. Number of Descriptions vs. Replacement Time. The PSNR is averaged over all nodes over the entire simulation time

The improved performance with increasing M can be attributed to two factors: 1) The system can exploit the benefit from path diversity more fully, and 2) The MD-FEC can apply unequal error protection to each GOF more effectively. For example, when M=4, only four levels of redundancy are possible and the possible redundancy is only the mixture of these four. But when M=8, eight levels of redundancy are possible. The total allocated redundancy (=(total rate)/(source rate)) reduces (hence the video source rate increases) as M increases. Figure 4 shows the total redundancy for different values of M.

3.3. Impact of Replacement Interval

The results so far are for a fixed replacement time (1 minute). Figure 3 also examines the impact of replacement time on video quality. We can see that the average PSNR increases as the replacement time becomes shorter. This is as expected, as the replacement time is the time interval a lost description due to a node disconnect stays unavailable.

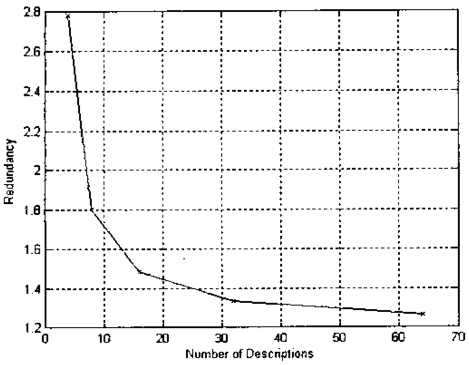

Figure 4 Redundancy vs. Number of Descriptions

3.4. Overloaded Case

To examine the system behavior when the network is overloaded, we change the average on/off time ($1/\lambda$ or $1/\beta$) of a node to 20/20 minutes. The longer off time leads to fewer available nodes in the network on average. A new requesting node sometimes has less than M descriptions available, even at the beginning time. Also, a node losing descriptions in the middle of a streaming session may not find replacements for a long time (much longer than the mean replacement time $1/\gamma$). In both cases, the requesting nodes that require additional serving nodes are put in a queue where they are sorted by the number of missing descriptions. The nodes that have the largest number of missing descriptions will be served first if a new serving peer becomes available.

As shown in Figure 3, the PSNR drops significantly compared to the non-overloaded case under the same replacement time. Also, increasing the description number M does not always improve the video quality. This is because, during the overloaded period, a streaming client often receives two or more descriptions from the same node (using the server selection scheme implemented in our current system). The disconnection of this node will cause significant degradation of video quality. Because the loss events of any two descriptions are not independent, the distribution $P(m,M)$ does not follow the binomial model closely, as shown in figure 2. As a result, compared to the non-overloaded case, the probability of losing more descriptions increases. In fact, the probability of losing all descriptions is non-negligible. (Figure 2 shows the case for $M=4$. Similar trends have been observed for larger M).

4. CONCLUSIONS AND FUTURE WORK

We described a video-on-demand system using peers as servers. By using multiple description coding and streaming different descriptions of a requested video from separate peers, the system is resilient to data losses due to frequent and unpredictable peer disconnects. We have investigated the impact of the number of descriptions and the replacement interval on the received video quality. We have found that, when the network is not overloaded, increasing the number of descriptions can improve the system performance significantly due to greater path diversity and flexibility in redundancy allocation by MD-FEC.

The present network simulations are based on simple models of peer behavior (disconnects and reconnects). For a more realistic evaluation of the system performance, simulations based on actual peer-availability traces are needed. In overloaded situations, an admission policy that governs the minimal number of available descriptions necessary before starting a session, and the associated blocking probability needs to be studied. Furthermore, a more judicious resource management strategy for allocating the resources (mainly the uplink bandwidth) from available serving nodes among requesting nodes is warranted. Finally, the video placement problem, which deals with how to distribute the descriptions from different videos among peers, is still an open question.

5. REFERENCES

[1] Y. Chu, S. Rao, S. Seshan and H. Zhang, "A Case for End System Multicast", *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, vol. 20, pp. 1456-1471, Oct. 2002.

[2] H. Deshpande, M. Bawa, and H. Molina. "Streaming Live Media over a Peer-to-Peer Network", *Stanford Database Group Technical Report (2001-20)*, Aug. 2001

[3] D. A. Tran, K. A. Hua, and T. T. Do. "ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming". In *Proc. IEEE INFOCOM 2003*, San Francisco, CA, USA.

[4] Y. Guo, K. Suh, J. Kurose, D. Towsley "P2Cast: Peer-to-peer Patching Scheme for VoD Service" *Proc. the 12th World Wide Web Conference (WWW-03)*, May 2003.

[5] V. Padmanabhan, H. Wang, P. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking", *NOSSDAV '02*.

[6] A. Rowstron and A. Singh, "Splitstream: High-bandwidth content distribution in a cooperative environment," Proc. 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), (Berkeley, CA, USA), Feb. 2003.

[7] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," IEEE Journal on Selected Areas in Communication (JSAC), vol. 20, pp. 1489-1499, Oct. 2002.

[8] J. Guo, Y. Zhu, and B. Li, "Codedstream: Live media streaming with overlay coded multicast," Proc. SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2004), (Santa Clara, California), pp. 28-39, Jan. 21-22, 2004.

[9] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: peer-to-peer media streaming using collectcast," Proc. ACM Multimedia 2003, (Berkeley, CA), Nov. 2003.

[10] J. Apostolopoulos *et al.*, "On Multiple Description Streaming in Content Delivery Networks," in *Proc. IEEE INFOCOM*, June 2002.

[11] R. Puri and K. Ramchandran. "Muliple description source coding through forward error correction codes". *Proc. Asilomar Conference on Signals, Systems, and Computers*, Asilomar, CA, Oct. 1999. IEEE.

[12] MoMusys code, "MPEG4 verfication model version 18.0" *ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio*, January 2001.