Peer-Owl: An Adaptive Data Dissemination Scheme for Peer-to-Peer Streaming Services*

Xiaofei Liao and Hai Jin

Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China
{xfliao,hjin}@hust.edu.cn

Abstract. More and more researchers have put their emphases on peer-to-peer streaming services, which can provide massive and cheap video-on-demand services. Data dissemination is one of the most important open problems when providing peer-to-peer streaming services, including live streaming and video-on-demand. Considering the performance and fault-tolerance of streaming systems, content should be distributed and replicated onto peer nodes according to some kind of strategy. In this paper, a novel data dissemination scheme is proposed. According to the proposal, media files are recoded into injected objects with several segments according to the characteristics of VCR operations' frequency, without changing the media compressing formats. Then, a peer who wants to get one media file can acquire the corresponding segmented object files from many source peers, not from only one peer. The new scheme does not need any extra network bandwidth. Results from simulations have proved that our Peer-Owl scheme has good performances.

1 Introduction

In the last few years, due to the increasing demand for streaming services on the Internet, many studies are being undertaken to find an efficient scheme. The new network model, peer-to-peer, can aggregate abundant resources from thousands of computers and address the above problems. Now many researchers provide new overlays on p2p networks to support file-sharing services, game services, especially the media streaming services [6].

But the peer-to-peer streaming networks have many open problems [1], such as fault-tolerant schemes and data replication schemes. The key to a p2p system, and one of the most challenging design aspects, is efficient techniques for data dissemination. There are several advantages using a good data dissemination scheme. First, services can be more stable than ever because there are many supplying peers to support the same media objects in p2p networks. Second, QoS can be improved. Third, load balance performance also can be improved. Hot movies have enough replicas, which are distributed onto almost all peers. Requests for the same hot movie can spread around the p2p networks.

Some researchers put their emphasis on the data dissemination schemes, but not for streaming services and only to support data sharing overlays. Other researchers

* This paper is supported by National 863 Hi-Tech R&D project under grant 2002AA1Z2102.

H. Jin, Y. Pan, N. Xiao, and J. Sun (Eds.): GCC 2004, LNCS 3251, pp. 479–486, 2004.

focus on data dissemination of the streaming services. But they only support the broadcast services and living streaming services, not on-demand services. It is obvious that data dissemination schemes in different streaming patterns, including living streaming and on-demand streaming, are not the same.

In [2], a novel data-splitting scheme *Owl* has been proposed to improve the performance of cluster video servers. In this paper, we extend the idea of *Owl* scheme to p2p environments. We provide a data dissemination scheme, called *Peer-Owl*, for p2p streaming services. The new scheme can encode movie objects into segments and transmit them onto ultra-peers when a new stream is created. Without any extra network bandwidth, the scheme can obtain a high success ratio of data dissemination because it takes full consideration of the characteristics of the frequency of VCR operation and the roles of ultra-peers.

The paper is organized as follows. We present some statistics about p2p streaming network and describe some properties of streaming services in section 2. Section 3 focuses on the method of how to segment media objects. In section 4 *Peer-Owl,* a novel data dissemination scheme for p2p streaming networks, will be described in detailed. Section 5 focuses on the methodology and results of simulations. In section 6 we survey some related works. Finally, section 7 closes with conclusions and future works.

2 Modeling Peer-to-Peer Streaming Networks

There are several models to describe the p2p networks, such as Erdos-Renyi (ER) model, BA model [1], and EBA model [7]. But when we consider a data dissemination scheme for a p2p streaming network, it is difficult to find a compatible model to describe our needs.

2.1 Topological Properties of Large-Scale P2P Networks

We look at two aspects of a p2p network: network topology, distribution of degrees of all peers. All these topological properties of p2p networks are based on Gnutella networks. All experimental data are calculated from our crawler on Gnutella network and the source codes are rewritten from *Limewire* open source client. Our crawler can discover over ten thousands peers and their connections. There are three data sets, calculating with three different time lengths, 20 minutes, 30 minutes and 40 minutes, called Data<200404031026>, Data<200404031426> and Data<200404022126>, respectively. All our conclusions are from these data sets. To describe a stable topological graph of p2p network, we only run our crawler for no longer than 40 minutes.

From Tab.1, we can conclude that though the number of ultra-peers is little, most edges are connected with them.

We have made the following conclusions. First, ultra-peers take more resources, such as routing paths, in hand than that of leaf peers. Second, ultra-peers are the most important agents for almost all shortest paths of pairs of peers. If we want to provide an effective data dissemination scheme, it is important to make more replicas onto ultra-peers.

Table 1. Statistical results about degrees of Gnutella networks. Properties measured are: the number of peers (or called nodes) *v*; total ultra-peers number *uv*; total edges number *e*; the number of edges connected between two ultra-peers *u2u*; the number of edges connected between one ultra-peers and one leaf peer *u2l*; the number of edges connected between two leaf peers *l2l*.

Items	Data<200404031026>	Data<200404031426>	Data<200404022126>
v	17395	28119	38115
uv	3302	5287	6257
e	20426	35181	52531
u2u	2803	4649	5788
u2l	17425	30163	46128
l2l	198	369	615

2.2 Topological Properties of Large-Scale P2P Networks

Streaming services based on p2p networks have some characteristics that other applications overlays do not have, such as file-sharing overlays and data storage overlays. VCR operations will make great impact on data dissemination. When one requesting peer accepts the video-on-demand service, it should store the media data onto its own storage systems and its neighbors'. But if the requesting peer does seek operation, media data accepted before will be dropped. To make the VCR operations frequency clear, we give two definitions about *MTTVCR* (called Mean Time To VCR) and *ttvcr* (called time to VCR) as in Equation 1. In Eq.1, t_{i+1} means the time position of the current VCR operation and t_i means the time position of last VCR operation.

$$MTTVCR = \frac{\sum ttvcr_i}{n}; ttvcr_i = t_{i+1} - t_i; \tag{1}$$

In the subsection, we use the log files from the video server located in the CCRNC (*Center China Regional Network Center*) of CERNET (*China Education and Research Network*). Log files of the video server record historical data. Figure 1(a) shows the *ttvcr* (time to VCR) of different users who choose the movie "*Thelma & Louise*" in March 17, 2001. The value of *MTTVCR* means that there are about three or four VCR operations in viewing one movie with 120 minutes length. Figure 1(b) shows the number of VCR operations in different time range. It is obvious that most VCR operations have been done in the first 20 minutes.

We can make the following conclusions about dada dissemination for p2p streaming services: first, media data should be cut into segments according to the above statistical results of VCR operations; second, when one requesting peer is accepting media data, it should store media onto local storage systems to build new replicas.

3 Media Encoding and Decoding

We have proposed a novel data splitting scheme, called *Owl*, to improve the performance of clustered video servers [3]. In clustered video servers, media data should be

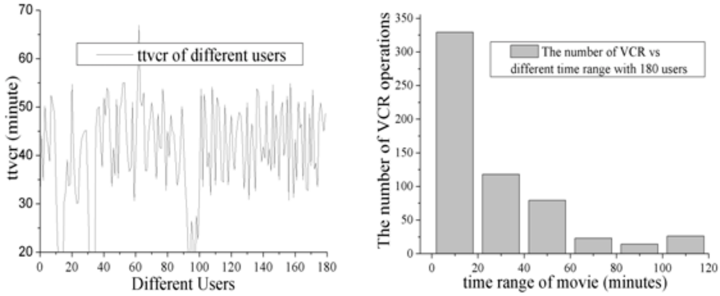

Fig. 1. (a) Length of *ttvcr* of different users; (b) the number of VCR operations *vs* different time range.

cut into many segments and distributed onto all nodes because of VCR operations. The scheme supports load-balancing and fault-tolerant services in a thin granularity

In our p2p streaming networks, there are two types of media objects (also called media files): one is called seed objects; the other is called infected objects. The number of seed objects is little. But after the evolution of the p2p streaming network with our data dissemination schemes, there are many infected objects, which are built from the segmentation operation. Each infected object is a fragmentized replica.

An injected object is got after one seed object is encoded. One injected object is composed of several clips and each clip has an added head with the information *begin time* and *end time*. The clip content is the same as that of one seed object. Each clip length is according to the statistical results from subsection 2.2. For example, the first clip length is about 20 minutes; the lengths of other clips are the same 40 minutes.

4 *Peer-Owl*: An Adaptive Data Dissemination Scheme for P2P Streaming Services

The new data dissemination scheme is based on on-demand streaming services in P2P networks. To design an efficient data dissemination scheme, there are several points need to be considered. First, no extra network bandwidths are needed. It is important to make full use of the bandwidth from connections of streaming services. Second, it is not necessary to run the data dissemination services all the time. The data dissemination services are only invoked when new on-demand stream is built. Third, ultra--peers are the most important agents of building the injected objects to increase the number of replicas. All ultra-peers of the path from the supplying peer to the requesting peer should do encoding operations. In this section, some basic concepts will be described first; then the scheme *Peer-Owl* will be described in detailed.

4.1 Basic Concepts

One streaming path from the supplying peer to the requesting peer is defined as follows:

$$P_{s \to r} = \{s, A_1, A_2, \ldots A_K, r\} \quad (2)$$

There are *m* ultra-peers in the streaming path. Each ultra-peer A_i' ($1 \le i \le m \le k$) is the agent to retransmit media data to the requesting peer. In fact, the streaming path is a path from the temporary multicast tree rooted at the supplying peer. Then the set of candidates of building injected objects is described as following:

$$C_{s \to r} = \{A_1', A_2', \ldots A_m'\}, A_m' = r \quad (3)$$

The success ratio $<P>$ for one streaming path can be defined as following:

$$P_i(A_i') = \frac{S_i}{TS_i}, \; < P >= \frac{\sum_i P_i(A_i')}{|C_{s \to r}|} \quad (4)$$

In Eq.4, S_i defines the size of injected object on peer A_i'; TS_i means the size of total accepted media data including the dropped part. $P_i(A_i')$ represents the success ratio of peer A_i'. The parameter $<P>$ displays the efficiency of data dissemination on one streaming path.

4.2 Principles of Scheme *Peer-Owl*

Our scheme *Peer-Owl* can b described as the following steps.

First, when one peer sends out a request for one movie with ID *name*, p2p networks will execute the search request with ID *name* among all seed objects and injected objects from all peers. Searching results are used to build a new task list with the consideration of fault-tolerance. When the requesting peer connects to one supplying peer selected from the task list, a new streaming path is formed.

Second, from the streaming path, we get the candidates set $C_{s \to r}$ according to Eq.3. Each peer from $C_{s \to r}$ will determine whether it is the same media data. Peers with *no* answers will prepare to do the encoding operations.

Third, for those peers with encoding operations, each peer accepts media data from the same streaming path and builds injected objects. Clips with length not longer than the default value will be dropped when a new VCR operation is coming. Otherwise, the clips must be stored into the injected objects. When no media data are coming, encoding operations are finished and p2p networks obtain a new replica.

At the fourth step, peers building new injected objects will release the new replicas with time length out. Then other peers can access the new injected objects.

At the last step, a streaming path can be destroyed.

5 Methodology and Simulation Results

To make a good and fair simulation, it is necessary to consider several problems. One is to build a realistic environment; second is to focus on several metrics acknowledged by researchers in the fields. From section 2, we can make some assumptions. First, it is obvious that the percentage of ultra-peers to all peers on one path is about

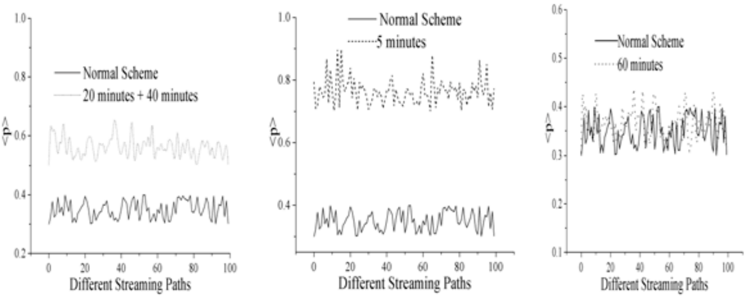

Fig. 2. (a) Success Ratio: Normal Scheme *vs Peer-Owl* with the first Clip's length 20 minutes and other clips' length 40 minutes; (b) Success Ratio: Normal Scheme *vs Peer-Owl* with the clip's length 5 minutes; (c) Success Ratio: Normal Scheme *vs Peer-Owl* with the clip's length 60 minutes.

0.865. When selecting two peers *s, r* randomly, there is a streaming path $P_{s\rightarrow r}$ between them. Then the number of ultra-peers on the path can be described as $0.865 * |C_{s\rightarrow r}|$. Second, the characteristics of the frequency of VCR operations can be described in mathematical model. For one requesting peer, the probability of VCR operations on one time position can be described as following:

$$f = \frac{1}{t} * c ; \sum_{t} \frac{1}{t} * c = 1 ; t = 10 * \tau, 1 <= \tau <= 11 \qquad (5)$$

There is an assumption that one movie is only 120 minutes long. After computing, *c* equals to 3.3. When the frequency of VCR operations and the selection of streaming paths have been decided, we can build a static environment to simulate the data dissemination. In our simulation, we select one pair peers from Data<200404031026> for 100 times and generate 100 different streaming paths randomly.

In the following simulations, there are several analyses in Fig.2. In these figures, there are four types of schemes: normal scheme without movies segmentations, *Peer-Owl* scheme with the first clip's length 20 minutes and other clips' length 40 minutes, *Peer-Owl* scheme with the clip's length 5 minutes, *Peer-Owl* with the clip's length 60 minutes.

From these figures, the success ratio of the third scheme is better than that of other three types of schemes. The success ratio of second scheme is middle in the four schemes. The normal scheme and the fourth scheme have almost the same performances of success ratio. But in the third scheme, clip size is too small and the length of tasks list is too longer. Systems need too resources to process the fragmentized injected objects.

6 Related Works

Most researches about data disseminations focus on mobile ad-hoc networks and structured p2p networks. Its main functions are to support the popular data sharing and improve the efficiency of p2p services.

Popular file sharing and living streaming are the primary topics [5] when researchers mention the data dissemination and prefacing techniques. It is obvious that these two services have much comparability. Because of no VCR operations to interrupt the normal data streams, data transmission of these two types of services are stable and consecutive. When systems begin data streams, data channels will be created and keep stable. A useful way to make data dissemination with the help of Tornado coding [3][4] is provided. The paper first proposed a new model Street-and-Building to describe the mobility of peers in mobile ad-hoc networks. Then, it provided a data dissemination protocol to disseminate Tornado encoded file segments (packets). But the beautiful scheme only supports file-sharing services. The middleware only supports file downloading services.

Researches on the statistics of VCR operations for on-demand streaming services are not hot topics. Intra-movie skewness from logs of traditional video servers is studied [9]. In fact, due to users' interest, there are some VCR operations when the movie is being viewed. From these logs, we can make some interesting conclusions, described in subsection 2.2. These conclusions will be helpful in designing a good data dissemination scheme.

Media objects segmentation techniques are mentioned at [4][6]. Tornado coding use the coding methods to support file sharing in normal and p2p environments [4]. A stream is segmented into *blocks* only for the efficient use of storage space [6] and the retrieval of necessary parts of a media stream. To apply segmentations of media objects in data disseminations for on-demand streaming services is one contribution.

7 Conclusions and Future Work

In this paper, we first give some statistics from Gnutella networks and traditional video servers; then a novel data dissemination scheme for P2P streaming networks is presented. From simulations, the success ratio of our *Peer-Owl* scheme is good according to the frequency of VCR operations. But the scheme *Peer-Owl* has great rooms to improve. In the future work, our focus will be concentrated on the media data splitting scheme for P2P networks, the bulk data transmitting technique and other open issues.

References

1. L. Barabasi and R. Albert, "Emergence of scaling in random networks", *Science*, Vol.286, pp.509, 1999.
2. E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky, "SETI@home: Massively Distributed Computing for SETI", *Scientific Programming*.
3. H. Jin and X. Liao, "Owl: a new multimedia data splitting scheme for cluster video server", *Proceedings of 28th Euromicro Conference*, 2002, pp.144-151.
4. J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed content delivery across adaptive overlay networks", *Proceedings of ACM SIGCOMM 2002*, August 2002.
5. J. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: Using tornado codes to speed up downloads", *Proceedings of IEEE INFOCOM'99*, pp.275–83, March 1999.

6. N. Leibowitz, M. Ripeanu, and A. Wierzbicki, "Deconstructing the Kazaa Network", *Proceedings of 3rd IEEE Workshop on Internet Applications (WIAPP'03)*, 2003, Santa Clara, CA.
7. Q. Wang, Y. Yuan, J. Zhou, A. Zhou, "Peer-Serv: A Framework of Web Services in peer-to-Peer Environment", *Proceedings of the Fourth International Conference on Web-Age Information Management (WAIM 2003),* Springer-Verlag, 2003.
8. S. Wu and H. Jin, "Symmetrical Pair Scheme: a Load Balancing Strategy to Solve Intra-movie Skewness for Parallel Video Servers", *Proceedings of 16th International Parallel and Distributed Processing Symposium (IPDPS'02)*, Florida, USA.