ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 3, March 2013

Chunk Scheduling Strategies In Peer to Peer System-A Review

Sanu C , Deepa S S

***Abstract*— Peer-to-peer (P2P) streaming systems have become popular in recent years. Several peer- to-peer systems for streaming have been deployed recently. The challenges for P2P streaming have been on its scalability and video viewing quality. Peer to peer (P2P) streaming tries to achieve scalability and at the same time meet real-time playback requirements. Both require efficient utilization of resources in P2P networks. The most important issue in designing a P2P streaming system is chunk scheduling which should be able to provide high-quality service while maintain a high utilization of system resources. There are different chunk scheduling strategies: Rarest First and Greedy. The former is a well-known strategy for P2P file sharing that gives good scalability, whereas the latter an intuitively reasonable strategy to optimize continuity and startup latency from a single peer's viewpoint. Greedy, while achieving low startup latency, fares poorly in continuity by failing to maximize P2P sharing; whereas Rarest First is the opposite. This highlights the trade-off between startup latency and continuity, and how system scalability improves continuity. Based on this insight, a mixed strategy is proposed that can be used to achieve the best of both worlds. Furthermore, the mixed strategy comes with an adaptive algorithm that can adapt its buffer setting to dynamic peer population. This report presents a most up-to-date survey of modeling work in the area of P2P streaming and chunks scheduling strategies in P2P streaming system.**

***Index Terms*— P2P System, Chunk Scheduling, Continuity, Streaming rate, Start-up latency.**

I. INTRODUCTION

Peer-to-peer (P2P) video streaming have become increasingly popular in recent years since CoolStreaming appeared as the first P2P live video application. A common agreement in this area is that P2P video streaming data will dominate the internet in the near future. Streaming applications have recently attracted a large number of users on the Internet. In 2010, the number of video streams served increased 24.3% to 41.6 billion even without counting the user generated videos. With the fast deployment of high-speed residential access, video traffic is expected to dominate the Internet in near future. To meet the demand of explosively growing multimedia applications, media streaming has been a research topic attracting significant interests over the past two decades. Peer-to-peer (P2P) overlay systems are being recently proposed to stream multimedia audio and video contents from a source to a large number of end users. The ultimate goal of P2P streaming is to satisfy the application requirements of as many end users as possible, with sustainable server bandwidth costs. The technology used before P2P system is client-server architecture in which a bunch of servers with much larger capacity support many other capacity-limited clients..The traditional client-server architecture advocates the use of large data centers to maintain streaming to end users at a large scale. The bandwidth cost on servers increases rapidly as the user population increases, and may not be manageable in corporation with limited resources. This model has several problems. First, the capacity and bandwidth of servers is not infinite, which results in scalability problem. Moreover, the extreme centralization of the system makes it vulnerable to single point of failure.

According to different kinds of service provided, there are two types of file distributions in P2P system: P2P file sharing and P2P file downloading. In either of these systems, peers not only download data from the servers and other peers, but also upload what they have downloaded to those requesting it. This will significantly reduce the bandwidth burdens of the servers and also provide peers with improved quality of service.

P2P streaming is a special case of P2P file downloading. Peer-to-Peer (P2P) stream- ing has become a popular means of distributing real-time online video contents. The distributed nature of the system provides great flexibility, scalability and robustness. The major issue in P2P live streaming is to provide users with real time and fluent playback experience. To maintain the attribute of real time, sometimes peers are even forced to skip certain amount of content to avoid falling behind other peers. Some representative P2P live streaming systems are Cool Streaming, PP Live, PP stream, UUsee , Anysee and Joost etc.

Very recently, P2P VoD (Video-on-demand) becomes the new interest of researchers. Different from P2P live streaming which provide live programs, this new kind of service allows users to watch different videos

Manuscript received March 09, 2013.

***Sanu C .**, Department of Computer Science and Engineering, College of Engineering Trivandrum. Trivandrum, India, 9037096022*
***Deepa S S.**, Assistant Professor, Department of Computer Science and Engineering, College of Engineering Trivandrum. Trivandrum, India,*

programs, or different parts of a certain video, or even allows VCR operations like fast forward and fast backward. This brings much flexibility and convenience to users, while increasing scheduling complexity of the system on the other side, as peers no longer watch the same part of a video. Typical VoD systems are PPLive, Joost, GridCast, PFSVOD, PPStream, UUSee etc

The main operations in P2P streaming are peer selection, load balancing and chunk scheduling. Algorithm such as choke algorithm can be used for peer selection. Load balancing is the process of uniformly distributing load among peers. Chunk scheduling is the most important issue in designing a P2P streaming system. There are three chunk scheduling strategies in P2P streaming. chunk scheduling can improve the quality of experience(QoE).

II. BASIC PEER TO PEER (P2P) STREAMING MODEL

Peer to peer streaming system contains large number of peers. For convenience we consider there are m -number of peers and a server in the system. In the initial condition of the peer to peer streaming only the server is distributing file to peers. The file is divided into small pieces. The small pieces are called chunks. Each chunk has a sequence number starting from 1. Server pushes chunks of content, in playback order, to the peers. Each peer in the system maintain a buffer. The buffer is used to store the chunks of the file .

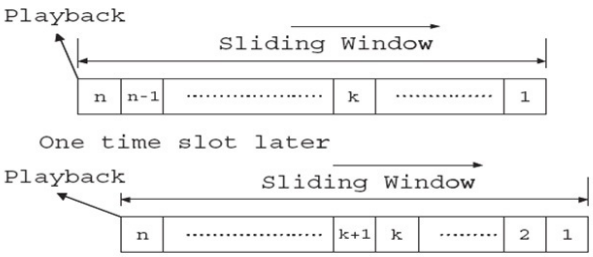

Figure 1: Structure of peer's buffer

Figure 1 shows the structure of the peer's buffer. The length of the buffer is n. Each buffer position is used to store the chunk of the file. Initially the buffer is empty and gets filled by peer to peer streaming protocol, either from the server and from other peers. Time is slotted. In each time slot the server selects a peer and sends chunk to that peer. The buffer position B(1) is used to store the newest chunk that the server is distributing in the current time slot. The buffer position B(n) is reserved for the chunk to be played back immediately. After each time slot, the chunk played back in the previous time slot is removed from buffer B and all other chunks are shifted up by 1.That is the buffer acts as sliding window into the stream of chunks distributed by server. The goal is to ensure B(n) is filled in as many time slot as possible, so as to support the continuous video playback and reduce the frame loss probability.

A. MATHEMATICAL MODEL

Consider a network with M peers trying to stream a live video from a single server. In the initial condition of P2P steaming only the server is distributing the files. The sever divides the files into number of pieces. These are called chunks. Time is slotted. In each time slot server selects a peer from m peers and send a chunk to that peer. The probability of ith buffer position of K peer get filled is 1/m. It is represented by Pk(i) .

$$P_k(i) = 1/M$$

Pk(i) is called buffer occupancy probability of kth peer. The performance would be very poor for large value of M. To improve the performances peer help each other. Each peer select another peer in each time slot and send a chunk. The chunk selection policy can be represented by a probability distribution q. The value of the chunk selection policy is based on three events WANT, HAVE and SELECT. Consider a particular peer k , and assume it selected peer h to download a chunk. The selection of a particular chunk to download is based on the following events.

- WANT(k,i):B(i)of peer k is unfilled; we abbreviate this event as W(k, i).
- HAVE(h,i):B(i)of peer h is filled; we abbreviate this event as H(h, i).
- SELECT(h,k,i): Using the chunk selection strategy, peer k cannot find a more preferred chunk than that of B(i) that satisfies the WANT and HAVE conditions; we abbreviate this event as S(h, k, i).

SELECT events various based on the chunk selection strategies. The chunk selection policy is a combination of these events. Thus, $q(i) = w(k,i) * H(h,i) * s(h,k,i)$.

B. MEASURING METRICS

In this section, we discuss common accepted measuring metrics in P2P video streaming. These metrics are useful in evaluating the system performance.

Continuity: This metric characterizes the key attribute of a P2P streaming system. Continuity is the probability of continues playback. Maintaining continuity is the primary objective for streaming applications. This measures how likely the movie can be played without skips or pauses. It is also a proper measuring metric to evaluate how good a scheduling algorithm is. To evaluate the continuity we define a continuity index which is the ratio of chunks arriving before or on playback deadline over total number of chunks.

Streaming Rate: This metric means the data streaming rate in the system. Maximum average streaming rate and average streaming rate are often derived to demonstrate how efficient the system is. From the user's perspective, this refers to its download rate. A download rate larger enough than the playback rate will be enough to guarantee the fluency of watching experience.

Startup Latency: This metric is specific in P2P video streaming systems. To maintain the fluency of watching experience, end systems must download enough chunks before they can start playback. Some systems set a fixed value for the number of chunks. The time to transmit these chunks is called startup latency. It is the time a peer should wait before starting playback.

C. FUNCTIONS IN (P2P) STREAMING SYSTEM

There are mainly three functions in Peer to Peer (P2P) streaming system which are discussed in the following sections:

- Peer Selection
- Load Balancing
- Chunk Selection

Peer Selection :In a Peer to Peer (P2P) streaming system, a peer may select more than one peer to download a specific file. The peer may obtain different portions of the file from different serving peers. Peer Set Each peer in the Peer to Peer (P2P) streaming system maintains a list of the peer that have the desired portions of the file. This list of peer is called Peer Set. A peer in the Peer to Peer (P2P) streaming system has two state. a. Lecher State b. Seed State In lecher state, peer is downloading the content. But it does not have all the chunks of the file. In seed state the peer has all the chunks of the file. The optimal peer selection problem is to select peer from peer set that have the desired portion of the file. The main goal is the system capacity maximization. Choke algorithm is one of the algorithm used for peer selection.

Load Balancing :For better performance load in each peer must be equal. Load balancing is the process of uniformly distributing load among the peers in the system.

Chunk Selection: Once a peer is selected it needs to determine which chunk to be downloaded from the peer. This is called chunk selection.

III. CHUNK SCHEDULING STRTEGIES

The most important issue in designing a P2P streaming system is chunk scheduling which should be able to provide high-quality service while maintain a high utilization of system re- sources. Once a peer is selected it needs to determine which chunk to be downloaded from the peer. This is called chunk selection. Each peer learns about chunk availability by periodically exchanging its buffer maps with buffer maps of its neighbors in the system. The buffer maps provide the information about the chunk availability.

There are three chunk scheduling strategies

1. Rarest First Strategy
2. Greedy Strategy
3. Mixed Strategy

1. RAREST FIRST(RF) STRATEGY

Rarest First is one of the chunk selection strategy. The buffer map provide information about the chunk availability. In rarest first chunk selection strategy, peer select the chunk that is the rarest in the system. That means , the peer select a chunk that has the least number of copies in the system. Fig.2 shows the rarest first chunk scheduling strategy. In rarest first chunk selection strategy a peer will first select B(1) to download then select B(2) and so on. Rare Chunks present in small number of peers and require long downloading time. Rarest First(RF) strategy reduce the downloading time by increasing the availability of rarest chunk. Selecting the rarest piece helps increase the diversity of chunks, therefore enhances the quality of service of the system. Last chunks are presented in least number of peers. There is a chance of lose of the last chunk by the failure of the peer. By using RF strategy last chunk problem can be eliminated. The main advantage of rarest first strategy is it balance the distribution of chunks among the system.

Figure 2: Rarest First Strategy

2. GREEDY STRATEGY

Greedy is another chunk scheduling strategy. The buffer map provide information about the chunk availability. In greedy strategy peer select pieces according to the closest order. This is also called Sequential algorithm as peers attend to download chunks most close to its playback point. This follows user's immediate interests, while is actually the worst strategy in terms of chunk diversity, thus is only combined with other algorithms together. Fig.3shows the Greedy chunk scheduling strategy. In greedy chunk selection strategy a peer will first select B(n) to download then select B(n-1) and so on. Greedy strategy provides best startup latency.

Figure 3: Greedy Strategy

In this section we discuss chunk scheduling strategies

3. MIXED STRATEGY

Mixed strategy is a combination of Rarest First (RF) strategy and Greedy strategy. It takes advantages of both Rarest First (RF) and Greedy strategies.

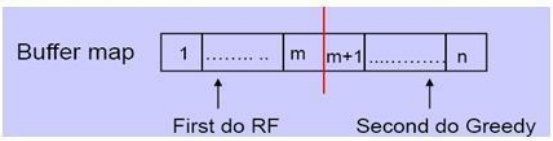

Figure 4: Mixed Strategy

Fig. 4 shows the mixed chunk scheduling

strategy. Each peer in the P2P system maintain a buffer of size n. In mixed chunk selection strategy the buffer in the peer is partitioned into two part. The buffer is partitioned by a point of demarcation m, such that value of m is between 1 and n. i.e1 <= m <= n.Rarest First chunk scheduling strategy is used in the first part of the buffer. Greedy chunk scheduling strategy is used in the second part of the buffer. Rarest First chunk scheduling strategy is used from the buffer position B(1) to B(m). Greedy chunk scheduling strategy is used from the buffer position B(m + n) to B(n). If no chunk can be downloaded using rarest first strategy ,then greedy strategy is used with other part of the buffer. Mixed strategy provides best continuity. Example of streaming application that use mixed chunk scheduling strategy is PPLive.

IV. COMPARISON

There are two performance metrics in P2P steaming system, continuity and startup latency. Continuity is the probability of continues playback. Startup latency is the time a peer should wait before starting playback.First we check the variation of continuity with buffer length. Then we check the variation of continuity with number of peers. Last we check the variation of startup latency with the buffer length.

1. Continuity versus Buffer size

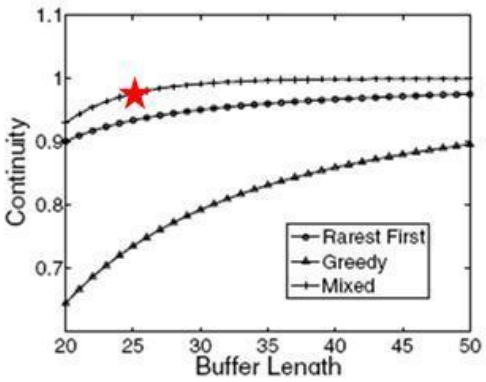

Figure 5: Continuity versus Buffer size

The figure 5 shows the variation of continuity with buffer length in different chunk scheduling strategies .The graph show that continuity increase when buffer size of the peer increase. Mixed chunk scheduling strategy provides best continuity. Rarest chunk scheduling strategy provides better continuity than greedy chunk scheduling strategy.

2. Continuity versus Number of peers

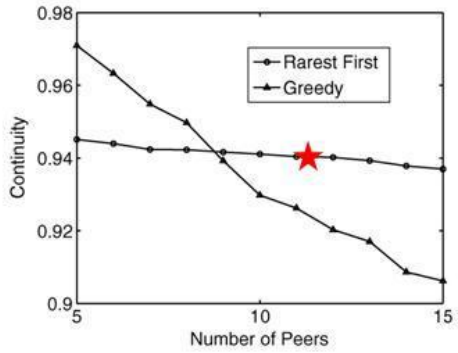

Figure 6: Continuity versus Number of peers

The above figure shows the variation of continuity with increase of number of peers in different chunk scheduling strategies. In case of Rarest First (RF) chunk scheduling there is a slight variation in continuity when the number of peers increase. Greedy strategy provides best continuity when the number of peers is small. When number of peer increase the greedy strategy provides less continuity.

3 .Startup Latency versus Buffer size

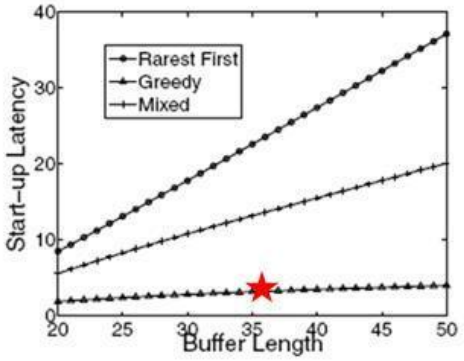

Figure 7: Startup Latency versus Buffer size

Fig.7 shows the variation of startup latency with buffer length in different chunk scheduling strategies Greedy chunk scheduling strategy provide best startup latency when number of peer increase. Mixed chunk scheduling strategy provides better startup latency than Rarest chunk scheduling strategy.

V. CONCLUSIONS

P2P has become an essential part of content distribution and streaming applications on the Internet, with applications ranging from IPTV to video conferencing. Different system models such as structured and unstructured models are discussed and the basic and mathematical models of P2P streaming systems are also explained. Important measuring metrics are also discussed. In terms of these metrics,

fundamental system limits that bound the performance of chunk scheduling strategies in P2P streaming systems are explained. Rarest first strategy is better in continuity. Greedy strategy is the best in start-up latency. Mixed strategy is the best in continuity and better in start-up latency than RF.

ACKNOWLEDGMENT

I would like to express my deep gratitude to my Guide and all my friends.

REFERENCES

[1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 2102–2111.

[2] Yipeng Zhou; Dah-Ming Chiu; Lui, J.C.-S., "A Simple Model for Chunk-Scheduling Strategies in P2P Streaming," *Networking, IEEE/ACM Transactions on* , vol.19, no.1, pp.42,54, Feb. 2011.

[3] J. Mundinger, R.Weber, and G.Weiss, "Optimal scheduling of peer-to peer file dissemination," *J. Sched.*, vol. 11, no. 2, pp. 105–120, 2008.

[4] B. Fan, J. C. S. Lui, and D. M. Chiu, "The design tradeoffs of BitTorrent-like file sharing protocols," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 365–376, Apr. 2009.

[5] Y. Huang, T. Z. Fu, D. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P vod system," in *ACM SIGCOMM*, 2008, pp. 375–388.

[6] D. Qiu and R. Srikant, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," in *Proc. ACM SIGCOMM*, 2004, pp. 367–378.