A Scheme to support P2P-Based Multimedia Sharing in Multimedia Board

T.Saisaranya[1], R.Kalaivani[2]

1 Pursuing M.Tech., Computer Science and Engineering, PRIST University, Tiruchirappalli,Tamilnadu.
2 Associate Professor, Department of Computer Science and Engineering, PRIST University, Tiruchirappalli,,Tamilnadu.

Abstract - User-generated content (UGC) is becoming the most popular and valuable information available on the WWW. However, little serious research has been conducted to measure the properties of its production process and also Online forums have long since been the most popular platform for people to communicate and share ideas. Nowadays, with the boom of multimedia sharing, users tend to share more and more with their online peers within online communities such as forums. This model to investigate the unique properties of forums based on the data collected from the Disney discussion boards. According to these properties, to design a scheme to support P2P-based multimedia sharing in forums called Multimedia Board (MBoard). MBoard can significantly reduce the load on the server while maintaining a high quality of service for the users and also this MBoard system toward the application of P2P-based multimedia sharing in forums or other mediums used to deliver user generated multimedia content.

1. INTRODUCTION

With the advent of Web 2.0 applications, user generated content (UGC) such as forums, blogs, and personal websites have become incredibly popular. Online forums produce some of the most highly customized user generated content and play an irreplaceable role in allowing users from across the world to discuss a wide variety of topics and be heard by others. With over 1.8 billion Internet users worldwide,there are literally thousands upon thousands of forums [1]. Some of the most active forums today include 4chan [2], Gaia Online [3], Ultimate Guitar [4], and DISBoards [5]. Forums tend to be for a special purpose, e.g., DISBoards [5] is focused on the discussion of Disney related issues.

In a forum, there are generally two main roles: server and users (i.e., nodes). The server is in charge of providing access to its database for users. The requests of users in a typical forum are for text, corresponding formats, public images (e.g., icons and expression pictures), and attachments. Nowadays, multimedia contents (e.g., images and videos) are shared increasingly in forums as attachments. Indeed, using images to convey the experience of some scenic spot or adding a video to tell a kitten’s story is often much more informative and entertaining than plain text. The trace data show that the tendency to post multimedia items within forums and the number of forum users are growing at a rapid rate. However, currently only those multimedia objects with limited size and resolution are allowed to be uploaded as attachments due to the bandwidth limit of the server in the server-client model. Thus, people have to post multimedia materials such as videos and high-resolution pictures as links to third party service providers such as YouTube. This brings an inconvenience to the forum users. Also, YouTube places limits on the size of video files that users are allowed to upload; the maximum limit for uploads as of April 2010 is 2 GB for normal users. Additionally, the third party services sometimes are not available. For example, YouTube allows the upload of nearly all videos so its service is banned in many countries due to videos of political topics. Another disadvantage in using third party services is the inclusion of embedded commercials [6] since YouTube attempts to profit from commercials embedded in the website and videos themselves [7].

Based on the above, it is beneficial to develop a scheme to enable forums to share multimedia contents in an efficient, low cost and easy-to-use manner. Specifically multimedia content should be shared in a way such that the bandwidth cost will remain within a range acceptable by forum runners and the intensity of server access will not exceed a typical web servers’ capacity. In this model , it has two fold. First, to present the analysis on the collected forum activity data from the Disney discussion board in order to quantify and visualize the forum’s characteristics and establish design principles. Although there are already works on analyzing quantitatively peer-assisted video on demand (VOD) in applications such as YouTube and PPLive [6], this is the first work to quantify this problem in the realm of forums. Second, to present a peer-assisted multimedia sharing system, called Multimedia Board (MBoard), that leverages forum characteristics to provide forums with their own multimedia sharing capabilities in order to reduce bandwidth cost. In this method does not lie in the improvement of existing P2P networks, but adopting existing P2P techniques suitable for forums to improve the performance of multimedia sharing in forums. Admittedly, peer-assisted approaches would bring about side effects such as security issues. There are already numerous approaches proposed to deal with the problems in peer-to-peer (P2P) systems.

Specifically, identified the following properties of forum- based multimedia sharing and corresponding design consideration through the analysis of existing forums. In this method found that P2P-based multimedia sharing is necessary and suitable for forums. The conclusion comes from 1) The daily increasing size of user posts and number of linked multimedia contents. 2) Head content is the content that attracts large amount of viewers but dominates server bandwidth (SBW) consumption. P2P sharing of head content achieves high video retrieval efficiency and meanwhile significantly reduces the server bandwidth consumption. 3) Popular forums tend to have a large number of users that enable P2P sharing. In this technology the MBoard function is based on the trace data. 1) Since nodes within a forum tend to share more similar contents than nodes across several forums, MBoard builds the nodes in one forum into a P2P network. 2) Since certain nodes stay in one forum most of the time, MBoard builds these nodes, termed stable nodes, into a Distributed Hash Table (DHT) to assist in content discovery. 3) MBoard has a refreshing scheme which updates the content index according to the continuous online time of the majority of nodes for content availability updating.

2. RELATED WORKS

2.1 Forum Observations and MBoard Overview

Through analyzing the trace data crawled from the Disney discussion board, we have the following observations that can answer three questions in designing a peer-assisted multimedia sharing system.

1. Is there an increasing demand for multimedia sharing forums?

O1: Observation(O): The size of forum posts and the number of multimedia elements have been rapidly increasing in recent years.

O2: The number of users and threads in a forum can be very large, necessitating a scalable media sharing system.

2. Is the P2P model suitable for multimedia sharing in forums?

O3: There are always some users present in a forum. Moreover, popular threads receive constant views while unpopular threads receive few views in a day.

O4: Most of the threads in the forums are tail content, while a small percentage of the threads in the forums are head content that contribute to most of the traffic, especially during the peak time.

3. What are the characteristics of forums we can take advantage of to optimize our design?

O5: Certain nodes are almost always present in a forum; we call these stable nodes.

O6: According to our assumption that a user is online for 10 minutes if he posts/replies a thread or is continuously online if he keeps posting within 1 hour, users spend 40 minutes online a day on average, while some may spend many hours a day.

O7: Users in the same forum tend to view the same threads but tend to switch to different forums. That is, the thread viewing activities are clustered by forums.

O8: Most users tend to reply to less than 10 threads per day, implying that most users are actually interested in a small number of threads. Therefore, they only need to have a small video cache.

The observations provide guidance to us in designing MBoard as a practical scheme in forums to enable peer assisted multimedia support. O1 and O2 demonstrate the demand of the P2P model in multimedia forums. O2 also shows that forums tend to have a large number of users, which is optimal for the P2P model. The P2P model yields higher efficiency in a larger scale since the content uploading load can be distributed among more content holders. In addition, O3 shows that user activity in popular threads spans over almost all of the time, and O6 shows that users remain online for a certain length of time. These two observations imply that many nodes will remain online for a certain period of time and can be used as P2P nodes for assisting content sharing, especially head content, which makes MBoard theoretically possible. Thus, MBoard employs the P2P model, in which the content information should be stored and retrieved in a P2P manner. This helps to reduce the server bandwidth cost and user waiting time. Specifically, it deals with the following issues:

- Network structure. Based on O7, MBoard builds nodes in one forum into a P2P network. Based on O5, MBoard leverages stable nodes to enhance content discovery efficiency.
- Multimedia content retrieval. MBoard utilizes stable
  nodes to aggregate content indices and ensure efficient discovery of media content providers in highly dynamic environment (O6). Based on O3, MBoard ensures the media content availability of head content.

2.2 P2P Construction

The P2P model has two types: unstructured and structured (DHT). Unstructured P2Ps are mostly gossip and flooding based, which incur large amounts of communication overhead in the network. The typical lookup length of DHTs is log n, where n is the number of nodes in the network. Thus, clustering all the forum users into a P2P network will result in a large network with long searching path lengths. This is because a larger number of nodes leads to longer searching path lengths. On the other hand, clustering on a smaller scale may result in the unavailability of requested content in a P2P network, because a node's requested content may be in a different network. A typical popular website such as DISBoards may consist of a number of forums, each containing thousands of users and threads. Based on O7 that nodes within one forum tend to view similar threads, the nodes in each forum constitute a P2P network in MBoard. Such a design enables a user to find requested content from other users within the forum most of the time while constraining the searching path length by avoiding large-size networks.

Like some practical P2P VOD systems such as PPLive [6], In this method to designate the server to be in charge of helping users locate media content. In such a system, the server manages the indexing of media segments. A media requester asks the server for the providers of its requested media, and media holders report to the server for index updates. However, frequent node joins and departures and media holder creation and removal lead to frequent index updates and server communication. This generates additional load on the server, though it is relatively small compared to that of uploading media. When the server has a limited capacity, this load can be alleviated by enabling nodes to autonomously locate media providers in a distributed P2P manner.

A DHT needs to maintain its topology in churn, where node joins and departures lead to high maintenance overhead and decreased lookup efficiency. O6 implies that nodes are very dynamic in forums. Therefore, DHTs are not an optimal choice. However, the high communication overhead of unstructured P2P makes it a poor choice as well. Through O5, there are a fair number of stable nodes, which remain active in the forum most of the day. Hence, MBoard intelligently forms a certain number of stable nodes into a DHT to assist content discovery by aggregating content indices and matchmaking content requesters to providers. Specifically, MBoard builds a two-tier structure, with the DHT in the upper tier and other nodes connecting to the selected stable nodes in the lower tier. The nodes connected to a stable node are called child nodes of the stable node. Since the selected stable nodes perform media content indexing, they are called brokers.

A DHT uses a consistent hash function to hash the identifier of nodes (e.g., IP addresses) and data objects (e.g., file names) to keys. It has two functions: Insert(key, object) and Lookup(key) to store the object with the key to its owner node and retrieve the object with the key. A node whose key is the closest to the object's key should be its owner node. In a DHT, each node maintains a routing table for log n neighbors. In order for a new node to join in the DHT overlay, it must know at least one other node already within the DHT.

For the best performance, the number of brokers N should not be large in order to avoid long routing latency. On the other hand, N should not be too small in order to avoid generating bottlenecks. To determine N, MBoard can evaluate the number of brokers at different times and use the average value over time.

The server maintains a list of stable nodes that are not selected as brokers and a list of brokers that currently serve in the DHT. The principle of stable node selection is that the longer a node is online daily, the higher probability it has of staying in the DHT. When a node u's daily online time exceeds a predefined threshold, it reports to the server. Then, the server adds node u to its stable node list. The nodes in the stable node list are ordered according to their capacities. The highest capacity stable node becomes a broker if the number of brokers in the DHT has not reached N. Specifically, the server assigns a bootstrap node from its broker list to the highest capacity stable node, and the node joins the DHT using the DHT node join protocol. Each time a stable node leaves the network, the node executes the DHT departure protocol and notifies the server. When a node joins in MBoard, the server randomly picks a stable node and assigns it to the newcomer as their parent. A parent helps its child nodes to send out content requests and receive replies from other nodes. Consequently, a two level DHT structure is formed as shown in Fig. 1.

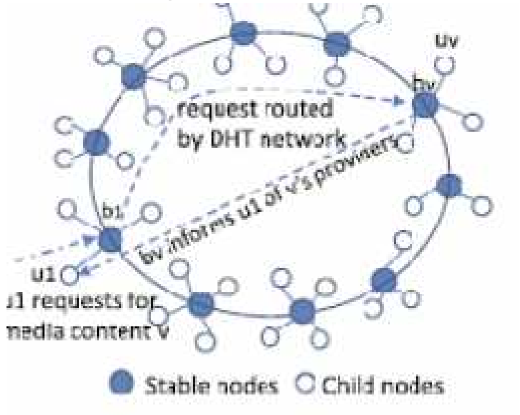


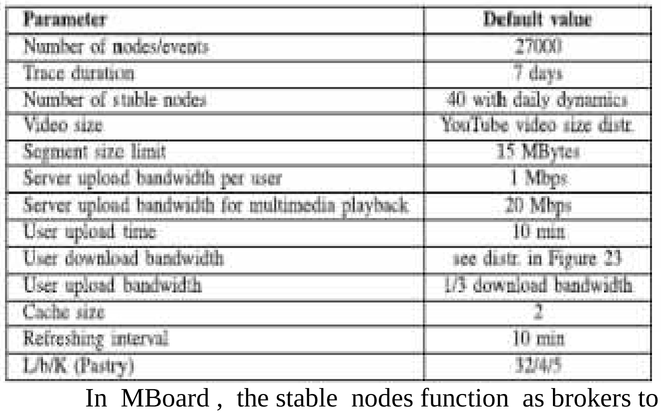
Fig.1. Two level DHT network.

Considering the high dynamism of child nodes, we let child nodes build and maintain connections to their parent. Therefore, the parents (i.e., nodes in DHT) function like brokers without the need to maintain the connection to their child nodes. We can provide incentives such as giving higher priorities to brokers' media requests to encourage stable nodes to function as brokers.

2.3 Multimedia Content Retrieval

When a node is downloading and viewing media content, it can upload the content simultaneously. In order to efficiently share media content, MBoard uses segmented media content to avoid the possibility of downloading failure and enable users to share existing media segments while downloading others. MBoard specifies the segment size as 15 MBytes, the largest size of most media content in YouTube, so that users do not need to split their videos in most cases and the rare long videos are automatically cut into segments by the MBoard client.

TABLE 1
Experiment Default Parameters



Parameter	Default value
Number of nodes/events	27000
Trace duration	7 days
Number of stable nodes	40 with daily dynamics
Video size	YouTube video size distr.
Segment size limit	15 MBytes
Server upload bandwidth per user	1 Mbps
Server upload bandwidth for multimedia playback	20 Mbps
User upload time	10 min
User download bandwidth	see distr. in Figure 23
User upload bandwidth	1/3 download bandwidth
Cache size	2
Refreshing interval	10 min
L/b/K (Pastry)	32/4/5

In MBoard , the stable nodes function as brokers to match content requesters and providers . For the media segment v posted by a user uv, after uploading it to the server, user uv registers itself as the content provider to v's broker bv by telling bv its IP address. Specifically, it asks its parent to send a Insert (key, index) request to the DHT. The key is the consistent hash value of the name of the media segment v, and the index includes the node's IP address, content segment name, etc. Using the DHT routing protocol, the request will be forwarded to the broker bv of segment v. The recipient broker then adds a record in the list of providers for this content segment. When a node is viewing /downloading a multimedia segment from the server or another peer, it also asks its parent to send a Insert(key , index) request to the DHT in order to register itself as the content provider. For example, in Fig. 1, when user u1 is watching media segment v, u1 asks its parent b1 to send a message to the broker of v to register itself as a media segment v's provider.

To retrieve a media segment, a requester asks its parent to send a request Lookup (key). The request will be forwarded to the broker of the segment that holds the registered index of the providers of the segment. The broker looks for the providers of the requested segment and returns a set of the latest registered providers to the requester. The broker returns a number of providers rather than a single provider in order to increase the probability that at least one provider is available. Also, it chooses the latest providers in order to increase the probability that they are still online. The requester then contacts the segment providers for the content. If there is no peer provider, the requester asks the server for the segment. For example, in Fig. 1, u1 sends its request for content v to b1, which further sends it using the DHT routing protocol. The request finally arrives at bv. Then, bv looks up for v in its registered media segments. If v is available, bv returns a number of nodes holding v to u1. When u1 finishes downloading the content from one of the content providers, it sends a registration request to register itself as a content holder in bu. If v is unavailable, then will inform to fetch v from the server instead.

3 PERFORMANCE EVALUATION

The real trace data were collected over a period of seven days on DISBoards, consisting of approximately 27,000 views and more than 700 threads. The trace was collected by constantly monitoring changes in the number of views on a half hour interval in order to determine thread viewing patterns during the tracing period. We assume that thread viewing activity is evenly distributed between two monitoring periods and that each thread has one video in order to simulate a multimedia forum. The default experiment settings are shown in Table 1.

In real life, people do not always watch an entire video. In order to simulate a realistic viewing behavior of users , i.e., to determine what percentage of a video a user typically watches before leaving a thread , this problem has resort to the statistics derived from 4 million MSN video users' viewing behavior in the trace file collected by Microsoft, as shown in Fig. 2. The downloading bandwidth of a node is assigned according to the bandwidth distribution of these MSN video users A user's upload bandwidth is set to 1/3 of its download bandwidth since most users have a DSL Internet connection.

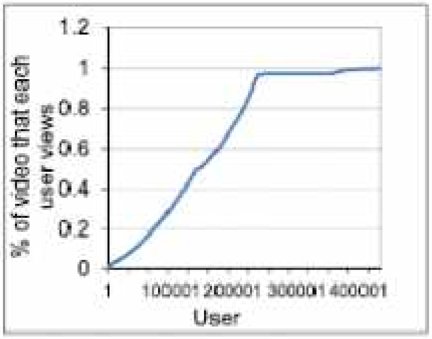


Fig. 2. User viewing behavior.

In MBoard, only the users that have a whole video segment can upload it. Since there is no way to find the number of views of a specific user in a thread, we assume that each thread view is from a unique user. In fact, this assumption compromises the performance of MBoard because otherwise video lookup delay can be reduced using cache if the same user views a thread multiple times. In the experiment, a broker returns all providers of the requested segment to a requester, and the requester randomly chooses a provider to contact until it receives the requested segment. In conclusion, the trace data are a set of events gathered from DISBoards, with each event indicating that a user views a thread by playing the video embedded in the thread at a specific time using the users' bandwidth and video playback percentage information obtained from MSN video.

Routing delay is the total time period for a message to arrive at its final destination. The delay in each routing hop was set to the sum of 0.1 seconds and a value randomly chosen from [0, 0.1] seconds. The latter delay part represents network latency due to different reasons. Queuing delay of a request is the time period that it waits in the queue before being served. The server needs 500 kbps bandwidth cost to serve one video. In the experiment , we are interested in the following metrics:

- Video playback delay. This is the time period a user must wait before the video playback can start, which combines the routing delay and the queuing delay if a user needs to wait for the peers/server for available bandwidth. It shows the delay in retrieving video segments.
- The number of video playback interruptions. This is the number of occasions that the delay in receiving the next video segment is greater than the time needed to watch the previous segment. This metric shows how often a user experiences interruptions during video playback.
- The number of accesses. This is the number of thread content accesses in a specific node or thread. The

former shows the load balance status in MBoard, and the latter shows the popularity of a thread.

- P2P contribution percentage. This is the number of media content accesses assisted by peers over the total number of content accesses. This metric shows the effectiveness of MBoard in reducing the server load.
- The number of refreshing messages. This value is the cumulated number of messages incurred by the refreshing scheme. It shows the cost of the refreshing scheme.

3.1 The Effectiveness and Efficiency of MBoard

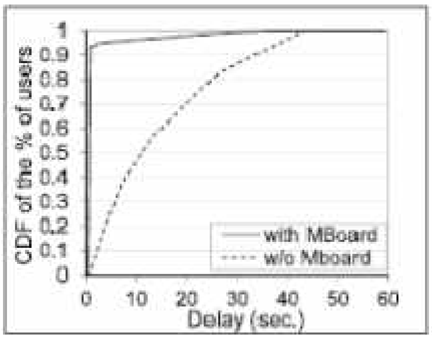


Fig. 3. Playback delay

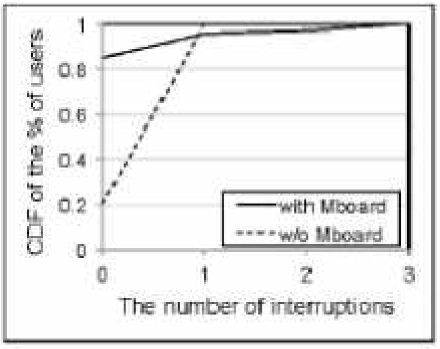


Fig. 4. Playback Smoothness

Fig. 3 shows the CDF of the percentage of users for a forum versus playback delay with and without MBoard, i.e., the traditional server-client model. With MBoard, more than 95 percent of the nodes achieve a very low delay before starting to download video data and 99 percent of all nodes have a delay under 20 seconds. On the contrary, without peer assistance, only 60 percent of all nodes have a delay less than 20 seconds. This is caused by the limited server upload bandwidth. When a large number of requests are sent to the server, most of them have to wait in the queue for processing due to the bandwidth limit of the server. Since MBoard allows nodes to request videos from peers, it achieves a much lower overall delay. Fig. 4 shows the CDF of users' video playback interruptions. Without MBoard, only 20 percent of nodes have no interruptions, and 80 percent of nodes have at least one interruption. With MBoard, 85 percent have no interruptions and only 11 percent of nodes have at least one interruption. The results are consistent to those in Fig. 3; with MBoard the number of interruptions is substantially lower than without. The result implies that MBoard can enhance the users' playback smoothness of the server-client model due to its P2P model.

Recall that in MBoard, if a node cannot find a video segment from peers, it resorts to the server. Fig. 5 shows the number of accesses in each thread contributed by peers and the server. The threads are arranged from left to right in terms of increasing peer contribution, as shown by the bold trend line. A first look tells us that the server's contribution remains constant at around tens of accesses in each thread. On the contrary, peers contribute significantly more than servers. Peers provide up to 700 times more accesses than that of the server on certain threads. These results show that MBoard effectively helps to reduce the amount of stress on

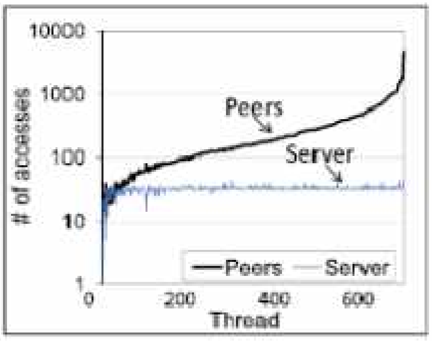


Fig. 5. P2P and server contribution

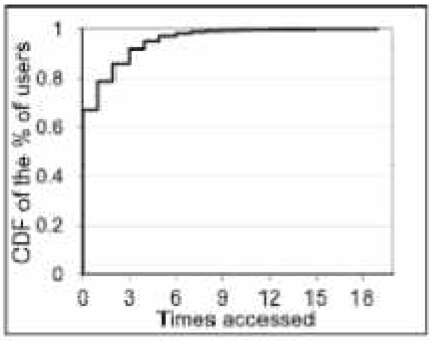


Fig. 6. Load on users

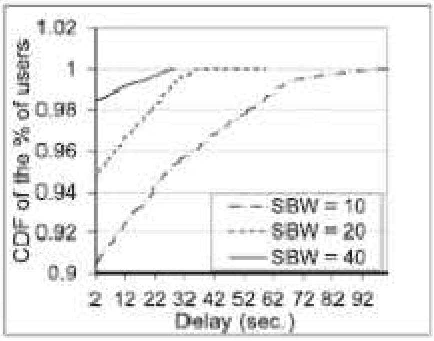


Fig. 7. Effect of SBW on delay

the server. Also, some threads are still served by the server. This is unavoidable since there is a possibility that no peer possesses the requested thread content for unpopular threads or due to peer unavailability, such as after midnight. Fig. 6 shows the CDF of the percentage of users versus the number of times that videos are requested from different nodes. The 90 percent of all the nodes are accessed 4 times or less but 60 percent have been accessed at least once. The remainder of nodes are requested somewhere between (4, 15] times. This shows that in MBoard the load is relatively evenly balanced among all nodes. This also implies that the absolute number of accesses, even for the nodes with a higher load, is low.

Fig. 7 shows the CDF of the percentage of users versus the video playback delay of MBoard with different amounts of server bandwidth. From this figure, we see that the video playback delay decreases as the server bandwidth increases. For the three bandwidth settings, approximately 90 percent of users retrieve their videos within 2 seconds. Moreover, we see that the 20 and 40 Mbps SBW systems reduce the video retrieval delay significantly. When the server bandwidth is 40 Mbps, more than 98 percent of all nodes have a video playback delay of 2 seconds or less. When the server bandwidth is 20 Mbps, 98 percent of all nodes have a video playback delay of 20 seconds or less, which is acceptable. This is because a higher SBW can help to reduce queuing time when there is no peer assistance available. Fig. 8 shows the CDF of users' playback interruptions with various server bandwidth settings.

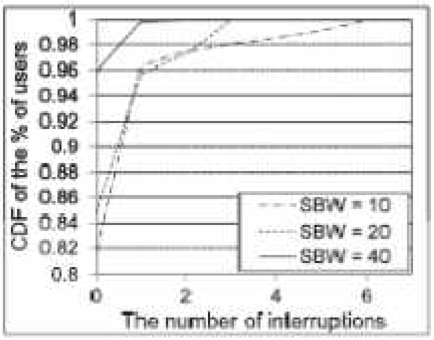


Fig. 8. Effect of SBW on playback smoothness

When the server bandwidth is only 10 Mbps, 82 percent of users experience no playback interruptions, 15 percent of users experience one playback interruption, and 0.8 percent of users experience a maximum of six playback interruptions. When the bandwidth is 20 Mbps, 85 percent of users experience no playback interruptions, 11 percent of users experience one playback interruption and 2.7 percent of users experience a maximum of three playback interruptions. When the bandwidth is 40 Mbps, 96 percent of users experience no playback interruptions, 4 percent of users experience one playback interruption and 0.15 percent of users experience a maximum of three playback interruptions. Thus, as SBW increases, more users have no interruptions, most users experience fewer interruptions and the maximum playback interruptions experienced by users decrease. Again, this is due to the availability of server assistance when no peers are available. Higher SBW enables the server to handle more requests quickly, leading to less video segment waiting time for users and hence higher playback smoothness.

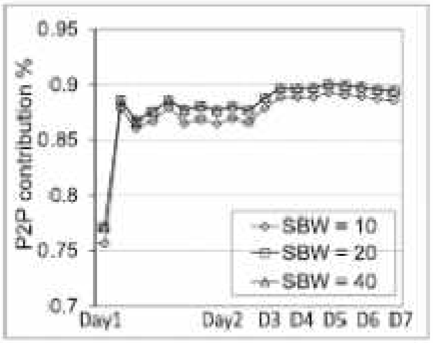


Fig. 9. Effect of SBW on P2P contribution.

Fig. 9 shows how the P2P contribution changes as the server's bandwidth increases. The P2P contribution percentages of SBW ¼ 20 and SBW ¼ 40 are nearly the same, and they are higher than that of SBW ¼ 10. With high SBW, peers are able to initially obtain content faster, and then upload them to other peers. However, when the server bandwidth is over 20 Mbps, additional server bandwidth does not help to significantly improve P2P contribution. Therefore, the best choice in our experiment is a SBW of 20 Mbps, which draws a good balance between performance and bandwidth cost.

3.2 The Effect of Forum Popularity

For this test, we aim to show the effectiveness of MBoard at different levels of forum popularity. The forum popularity is the number of thread accesses during a certain period of time. To calculated the popularities of the 21 forums and ordered the forums in an ascending order of the popularity. In this method to chose the last, two-thirds and one-third popularities in the list as the highest popularity, medium popularity, and low popularity, respectively, and tested the MBoard forum with different popularities. Despite the large increase in traffic in the highest popularity forum over the medium popularity forum and low popularity forum, the video playback delay only increases slightly, due to the scalability of MBoard. .

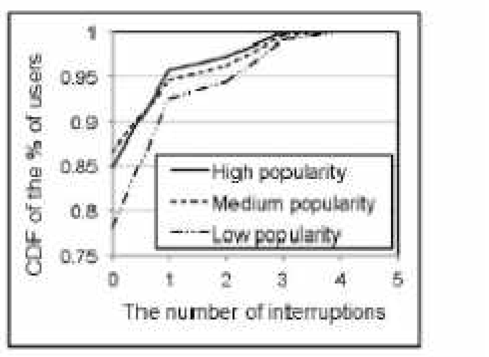


Fig. 10. Effect of popularity on playback smoothness.

Fig. 10 shows a CDF of nodes' video playback interruptions. We see that the number of interruptions is the lowest for the high popularity and medium popularity tests, while the number of interruptions for the low popularity test is still acceptable, with over 78 percent of users experiencing no interruptions. The increase in interruptions for the low-popularity test can be attributed to fewer peers available with a copy of the video due to low popularity.

4. CONCLUSION AND FUTURE WORK

Most forums presently employ the server-client model, where the server replies requested content to the clients. The trace data is collected from DISBoards show the rapid daily growth of user generated media content and users in forums, which becomes a hurdle for forums in meeting user demand due to limited server bandwidth. Through the analysis of trace data from DISBoards, in this method observed that their large group of users and user activity patterns meet the basic environmental requirements of employing a P2P model. Also, the existence of stable nodes, thread characteristics and media content patterns provide us with a direction to optimize the design of a P2P-based media sharing system.To propose the MBoard system toward the application of P2P-based multimedia sharing in forums or other mediums used to deliver user generated multimedia content.

MBoard utilizes a two-tier DHT network to leverage the stable nodes for content discovery in peers and also propose the broker-based content sharing and refreshing schemes to reduce communication cost. Extensive trace-driven experiments prove that MBoard is applicable in today's forum environment. It greatly reduces the load on the server and achieves high P2P sharing efficiency and low playback waiting time. The future work lies in deploying MBoard in a real forum to better evaluate its performance.

REFERENCES

[1] "Internet Usage Statistics. http"//www.internetworldstats.com/ stats.htm,2011.

[2] "4chan," http://www.4chan.org/, 2011.

[3] "Gaia Online," http://www.gaiaonline.com/, 2011.

[4] "Ultimate Guitar," http://www.ultimate-guitar.com/, 2011.

[5] "The DIS Discussion Forums,"http://www.disboards.com/, 2011.

[6] "PPLive," http://www.pplive.com, 2011.

[7] "Google Launches YouTube Video Ads," http://www. readwriteweb.com, 2011.

[8] X. Ochoa and E. Duval, "Quantitative Analysis of User-Generated Content on the Web," Proc. First Int'l Workshop Understanding Web Evolution (WebEvolve '08), 2008.

[9] Y. Niu, Y. min Wang, H. Chen, M. Ma, and F. Hsu, "A Quantitative Study of Forum Spamming Using Contextbased Analysis," Proc. Network and Distributed System Security (NDSS) Symp., 2007.

[10] Y. Zhu, "Measurement and Analysis of an Online Content Voting Network: A Case Study of Digg," Proc. 19th Int'l Conf. World Wide Web (WWW' 10), 2010.