Toward P2P-based Multimedia Sharing in User Generated Contents

Harrison Chandler, Haiying Shen, Lianyu Zhao, Jared Stokes
Dept. of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{hchandl, shenh, lianyuz, jbstoke}@clemson.edu

Jin Li
Microsoft Research
Redmond, WA 98052
jinl@microsoft.com

Abstract—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. The server-client model of forums has been used since its creation in the mid-nineties. However, this model has begun to fall short in meeting the increasing need of bandwidth and storage resources as an increasing number of people share more and more multimedia content. In this work, we first investigate the unique properties of forums based on the data collected from the Disney discussion boards. According to these properties, we design a scheme to support P2P-based multimedia sharing in forums called Multimedia Board (MBoard). Extensive trace-driven simulation results utilizing real trace data show that MBoard can significantly reduce the load on the server while maintaining a high quality of service for the users.

Index Terms—User generated content, Forums, Video on demand, Peer-to-peer networks, Distributed hash tables

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. Our trace data shows 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 3rd 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 2GB for normal users. Additionally, the 3rd 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 3rd 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 paper, our contribution is two-fold. First, we present our 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, we propose 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. Our contribution 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 issue. There are already numerous approaches proposed to deal with the problems in the peer-to-peer (P2P) systems. We will investigate how to deal with the side-effects in our future work.

Specifically, we identified the following properties of forum-based multimedia sharing and corresponding design consideration through the analysis of existing forums. We 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 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. We designed MBoard based on our observations from 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.

The rest of this paper is structured as follows. Section 2 presents a concise review of related works. Section 3 briefly presents our analysis on the collected forum activity data from the Disney discussion board, and presents the design details of MBoard. Section 4 reports the simulation experimental results of MBoard. The final section presents a conclusion with a discussion on further work.

2 Related Works

UGC/forum measurement. Ochoa and Duval [8] conducted an in-depth quantitative analysis of 9 popular websites that are based on different types of UGC. They found that UGC production follows "long-tail" distributions and is marked with a strong "participation inequality". Also, they found that not all UGC types follow the inverse power-law distribution, and that large content collections could be dominated by the presence of ultra-productive users. Niu *et al.* [9] conducted a quantitative study of forum spamming using context-based analysis. They also proposed context-based analyses to detect spam automatically and ways to overcome the shortcomings of content-based analyses. To better understand the nature and impact of online content voting networks, Zhu [10] analyzed Digg [11] and provided insight into the design of content promotion algorithms and recommendation-assisted content discovery. Yu *et al.* [12] presented a measurement study of a large-scale VOD system and observed that video session length has a weak inverse correlation with the video's popularity. The authors also introduced a modified Poisson distribution that more accurately models the user arrival rate.

However, few works have conducted a quantitative study of UGC in online forums. This work is the first that quantifies the characteristics of user generated multimedia content in the realm of forums. We find that the P2P model is suitable for multimedia sharing in forums; we then explore the use of peer assisted multimedia sharing within forums. Peer assisted video sharing is widely studied in the area of VOD. Thus, we also present works on VOD in below.

Video on demand. YouTube [13] is a centralized video sharing service, and its operation depends on the support of a huge number of server clusters that cost millions of dollars per day in bandwidth. There are several works on utilizing P2P video sharing to ameliorate the bandwidth cost of YouTube-like services [14]–[16]. Some works focus on the structure of peers to enhance the performance of video sharing in VOD. Zhou [17] proposed a hybrid overlay network protocol that constructs and maintains a tree overlay and a gossip mesh overlay. P2VOD [18] features a single multicast tree with the server at the root position. Chow *et al.* [19] considered one scenario: a source node that wishes to stream an ordered sequence of packets to a collection of receivers, which are distributed among a number of clusters. They studied two data communication schemes, one based on multi-trees and the other based on hypercubes, for solving this question. NetTube [20] groups users that watch the same videos into the same overlay for efficient video sharing on YouTube. BulletMedia [21] combines a traditional overlay mesh approach with a structured overlay. VMesh [22] builds a DHT to enable nodes to quickly find and connect to multiple parents with requested segments. SocioNet [23] is a social-based multimedia access system for unstructured P2P networks. PPLive [6], PPStream [24] and UUSee [25] are popular practical P2P VOD systems. Huang *et al.* [26] discussed the challenges of large-scale P2P VoD systems based on the experience of deploying a real system developed by PPLive. The concept of relying on a distributed system for high quality of service has stimulated the creation of many P2P music and video sharing software products, such as Dalesa [27], WireStack, TorrentSwapper, Qtraxmax and Kazaa Mate [28].

There are other studies on VOD that attempt to improve video sharing performance. PROMISE [29] enables a peer to receive a stream from several parents. iPASS [30] employs a differentiated prefetching design that enables peers with higher contributions to prefetch content at a higher speed. Pussep *et al.* [31] proposed two adaptive server allocation schemes that estimate the capacity and service demand of the system to adaptively optimize allocated resources. Boufkhad *et al.* [32] tried to establish a threshold on the average upload bandwidth of a node above which the system becomes scalable. Feng *et al.* [33] showed how the challenge of minimizing server bandwidth is equivalent to maximizing the system-wide utility when each peer bids for and sells video segments. These studies can be used to complement our work by improving the performance of multimedia content sharing in forums.

Super-peer two-tier structure. MBoard uses a two-tier structure, which has been widely used and studied in P2P networks. In a super-peer structure, peers are divided into two classes: high-capability super-peers that handle search or routing, and ordinary peers that act as their clients. A client performs a search by submitting a search query to its super-peer. Sacha [34] presented an overview of super-peer two-tier topologies. KaZaA [35], Gnutella [36], FastTrack [37], OceanStore [38], [39], eDonkey [40], LST [41], SBARC [42], Skype [43], JXTA [44] and SUPS [45] adopt such a super-peer structure in their designs. The works in [46], [47] build super-peers into a DHT, and the communication between peers located in different groups is relayed by the DHT. Yang and Garcia-Molina [48] examined super-peer networks in detail, gaining an understanding of their fundamental characteristics and performance tradeoffs. Xiao *et al.* [49] presented a work-load model to determine an optimal tier size ratio between lower-tier and upper-tier. The two-tier structure is also used for other purposes, such as file consistency maintenance [50], load balancing [51], P2P live streaming [52], Grid resource discovery, and membership management [53].

3 Forum Behaviors and System Design

3.1 Forum Observations and MBoard Overview

Through analyzing the trace data crawled from the Disney discussion board, we have the following observations that

can answer our three questions in designing a peer-assisted multimedia sharing system. Please refer to Section 6 (in supplemental material) for the details of our data analysis.
(1) Is there an increasing demand for multimedia sharing forums?

Observation(O)1: 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 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.

Due to space limitations, we present the refreshing scheme of MBoard in Section 7 (in supplemental material).

3.2 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], we could 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, we know that 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. In MBoard, we use the Pastry [54] DHT, but any kind of DHT can be used.

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 [55]. When a node u's daily online time exceeds a pre-defined 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.

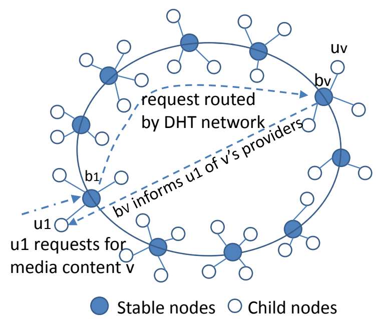


Fig. 1: Two level DHT network.

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 Figure 1. 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.

3.3 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 according to the length distribution in Figure 22 (in supplemental material) and the bitrate of the videos on YouTube [56], 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.

In MBoard, the stable nodes function as brokers to match content requesters and providers. For the media segment v posted by a user u_v, after uploading it to the server, user u_v registers itself as the content provider to v's broker b_v by telling b_v 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, and etc. Using the DHT routing protocol, the request will be forwarded to the broker b_v 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 Figure 1, when user u_1 is watching media segment v, u_1 asks its parent b_1 to send a message to the broker of v to register itself as a media segment v's provider.

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

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 Figure 1, u_1 sends its request for content v to b_1, which further sends it using the DHT routing protocol. The request finally arrives at b_v. Then, b_v looks up for v in its registered media segments. If v is available, b_v returns a number of nodes holding v to u_1. When u_1 finishes downloading the content from one of the content providers, it sends a registration request to register itself as a content holder in b_v. If v is unavailable, then b_v will inform u_1 to fetch v from the server instead.

4 Performance Evaluation

We conducted trace-driven experiments of MBoard on the well-known event-driven simulator PeerSim [57]. The real-trace data was collected over a period of 7 days on DISBoards, consisting of approximately 27000 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, we resort to the statistics derived from 4 million MSN video users' viewing behavior in the trace file collected by Microsoft, as shown in Figure 2. The downloading bandwidth of a node is assigned according to the bandwidth distribution of these MSN video users, as shown in Figure 23

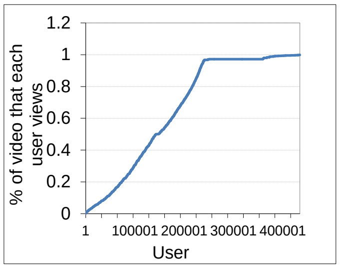


Fig. 2: User viewing behavior.

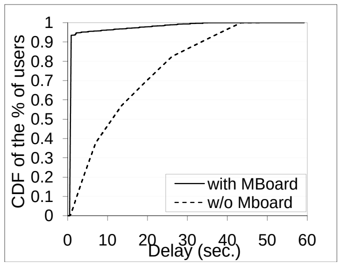


Fig. 3: Playback delay.

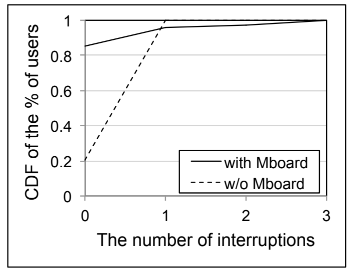


Fig. 4: Playback smoothness.

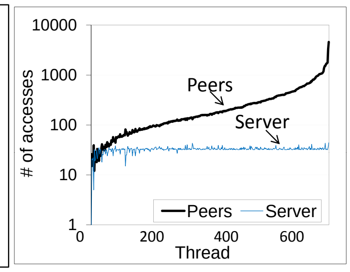


Fig. 5: P2P & server contribution.

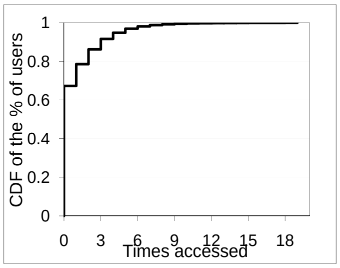


Fig. 6: Load on users.

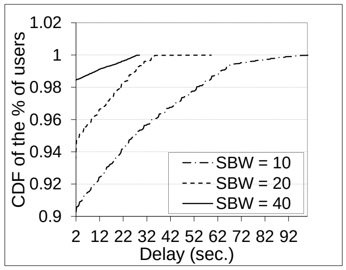


Fig. 7: Effect of SBW on delay.

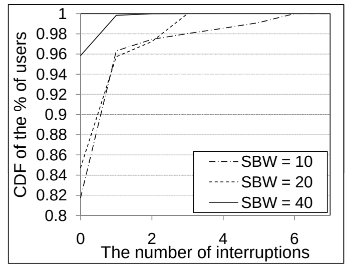


Fig. 8: Effect of SBW on playback smoothness.

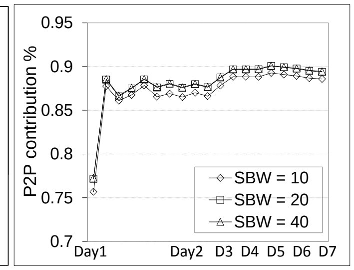


Fig. 9: Effect of SBW on P2P contribution.

(in supplemental material). A user's upload bandwidth is set to 1/3 of its download bandwidth since most users have a DSL Internet connection [58]. 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 is 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 a value equals 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 500kbps 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.

Due to the page limit, we present additional experimental results in Section 8 (in supplemental material).

4.1 The Effectiveness and Efficiency of MBoard

Figure 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. We see that with MBoard, more than 95% of the nodes achieve a very low delay before starting to download video data and 99% of all nodes have a delay under 20 seconds. On the contrary, without peer assistance, only 60% 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. Figure 4 shows the CDF of users' video playback interruptions. Without MBoard, only 20% of nodes have no interruptions, and 80% of nodes have at least one interruption. With MBoard, 85% have no interruptions and only 11% of nodes have at least one interruption. The results are consistent to those in Figure 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. Figure 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,

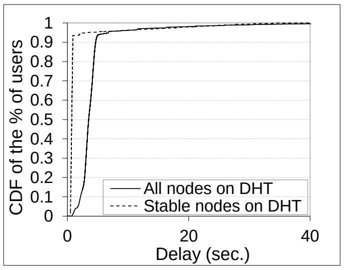


Fig. 10: Effect of stable nodes on delay.

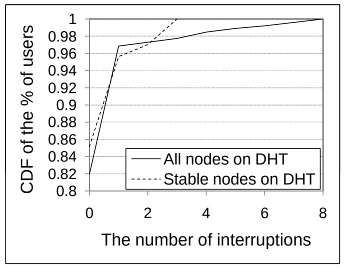


Fig. 11: Effect of stable nodes on playback smoothness.

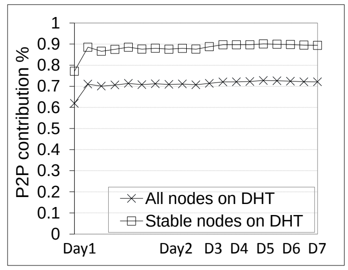


Fig. 12: Effect of stable nodes on P2P contribution.

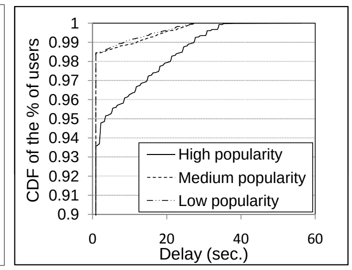


Fig. 13: Effect of popularity on delay.

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 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.

Figure 6 shows the CDF of the percentage of users versus the number of times that videos are requested from different nodes. We see that 90% of all the nodes are accessed 4 times or less but 60% 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 amongst all nodes. This also implies that the absolute number of accesses, even for the nodes with a higher load, is low.

Figure 7 shows the CDF of the percentage of users versus the video playback delay of MBoard with different amounts of server bandwidth (SBW). From this figure, we see that the video playback delay decreases as the server bandwidth increases. For the three bandwidth settings, approximately 90% of users retrieve their videos within 2 seconds. Moreover, we see that the 20Mbps and 40Mbps SBW systems reduce the video retrieval delay significantly. When the server bandwidth is 40Mbps, more than 98% of all nodes have a video playback delay of 2 seconds or less. When the server bandwidth is 20Mbps, 98% 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. Figure 8 shows the CDF of users' playback interruptions with various server bandwidth settings. When the server bandwidth is only 10 Mbps, 82% of users experience no playback interruptions, 15% of users experience 1 playback interruption and 0.8% of users experience a maximum of 6 playback interruptions. When the bandwidth is 20 Mbps, 85% of users experience no playback interruptions, 11% of users experience 1 playback interruption and 2.7% of users experience a maximum of 3 playback interruptions. When the bandwidth is 40 Mbps, 96% of users experience no playback interruptions, 4% of users experience 1 playback interruption and 0.15% of users experience a maximum of 3 playback interruptions. Thus, as SBW increases, more users have no interruptions, most users experience less interruptions and the maximum playback interruptions experienced by users decreases. 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 of users and hence higher playback smoothness. Figure 9 shows how the P2P contribution changes as the server's bandwidth increases. We see that the P2P contribution percentages of SBW=20 and SBW=40 are nearly the same, and they are higher than that of SBW=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 20Mbps, additional server bandwidth does not help to significantly improve P2P contribution. Therefore, the best choice in our experiment is a SBW of 20Mbps, which draws a good balance between performance and bandwidth cost.

4.2 The Effect of Stable Nodes

In this test, we want to show the effectiveness of using stable nodes by comparing stable node settings with all nodes in the DHT. Figure 10 shows the CDF of the percentage of users versus video playback delay when all nodes are on the DHT and when only stable nodes are on the DHT, respectively. When only stable nodes are in the DHT, 30% of nodes have no more 0.6 second delay, 50% of nodes have no more 0.7 second delay, and 93% of nodes have no more 0.9 second delay. While when all nodes are in the DHT, 30% have no more 3 second delay, 50% of nodes have no more than 3.5 second delay, 93% of nodes have no more 5 second delay. In both cases, around 96% of users have delays no more than 10 seconds. Therefore, the delay of using stable nodes is less for most users than placing all nodes in the DHT. This is because the size of the DHT when putting all nodes on the DHT is much larger than only using stable nodes, which increases the number of routing hops and routing delay. The frequent churn also increases the number of routing hops. A similar result is found in Figure 11, which shows the CDF of users' playback interruptions when using stable nodes or using all nodes in the DHT. The figure shows that the maximum number of interruptions is 8 when all nodes are on the DHT, but only 3 when only stable nodes are on the DHT. When only stable nodes are on the DHT, 85% of nodes have no playback interruption, 10.5% of nodes have 1 interruption and 3.4% of nodes experience a maximum of 3 interruptions. When all nodes are on the DHT, 82% of nodes have no playback interruption, 15% of nodes have 1 interruption, and 0.4% of nodes experience a maximum of 8 interruptions. The phenomenon is caused by the same reasons as in Figure 10.

Figure 12 shows the P2P contribution percentage from the first day to the seventh day when all nodes are on the DHT and when only stable nodes are on the DHT, respectively. When only stable nodes are on the DHT, the P2P contribution percentage is 80% on the first day, and it increases to 90% and maintains nearly constant in the remaining days. When all nodes are on the DHT, the P2P contribution percentage is 60% on the first day, and it increases to 70% and maintains nearly constant in the remaining days. In both cases, the percentage

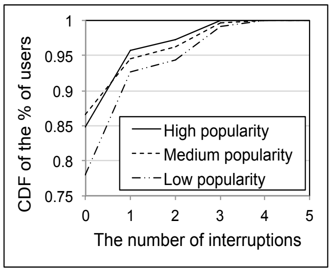


Fig. 14: Effect of popularity on playback smoothness.

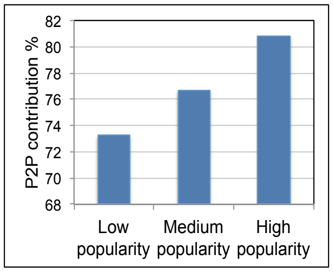


Fig. 15: Effect of popularity on P2P contribution.

is low on the first day because few peers have requested video segments initially. Thus, more users ask the server for video segments. Later, as more and more peers have requested video segments, users can retrieve video segments from their peers, leading to a higher and constant P2P contribution percentage. We also see that the percentage when all nodes are included in the DHT structure is lower than when only stable nodes are included. This is because the increased churn and larger playback delay result in higher data transmission failures when unstable nodes function as brokers in the DHT. Upon failure, the requesters ask the server for the video segments.

4.3 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. We calculated the popularities of the 21 forums and ordered the forums in an ascending order of the popularity. We 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. Figure 13 shows a CDF of nodes' video playback delays for different forum popularities. We see that the low popularity and medium popularity tests have no users with a delay greater than 28 seconds, and over 98% of users have virtually no delay. Furthermore, while the high popularity test has less than 1% of users with the highest delay of 57 seconds, almost 99% have a delay of less than 28 seconds. 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. Figure 14 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% 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. Figure 15 demonstrates the P2P contribution at varying levels of popularity. The P2P contribution in low, medium, high popularity forums is around 73%, 77% and 81%, respectively. As expected, the contribution increases with popularity because there are more online users with the requested videos.

5 Conclusion and Future Work

Most forums presently employ the server-client model, where the server replies requested content to the clients. Our trace data collected from DISBoards shows 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 the trace data from DISBoards, we observed that their large group of users and the 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 a direction to optimize the design of a P2P-based media sharing system. We further propose the MBoard system towards 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. We 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. Our future work lies in deploying MBoard in a real forum to better evaluate its performance. Also, we will study the properties of other online forums where users have very different access patterns and investigate whether MBoard is useful in these forums. Further, we will consider other factors that affect the quality of service such as the formats of the video clips, limited and shaped up-link bandwidths and long network delay in the experimental environment in order to study how these factors influence the performance.

Acknowledgements

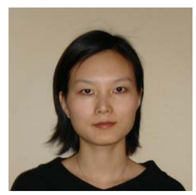
This research was supported in part by U.S. NSF grants OCI-1064230, CNS-1049947, CNS-1025652, CNS-1025649, CNS-1057530 and CNS-0917056, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282. An early version of this work [59] was presented in the Proceedings of INFOCOM'11.

References

[1] "Internet usage statistics," http://www.internetworldstats.com/stats.htm.
[2] "4chan," http://www.4chan.org/.
[3] "Gaia online," http://www.gaiaonline.com/.
[4] "Ultimate guitar," http://www.ultimate-guitar.com/.
[5] "The DIS discussion forums," http://www.disboards.com/.
[6] "PPLive," http://www.pplive.com.
[7] "Google Launches YouTube Video Ads," http://www.readwriteweb.com.
[8] X. Ochoa and E. Duval, "Quantitative analysis of user-generated content on the web," in *Proc. of WebEvolve at WWW*, 2008.
[9] Y. Niu, Y. min Wang, H. Chen, M. Ma, and F. Hsu, "A quantitative study of forum spamming using contextbased analysis," in *Proc. of NDSS*, 2007.
[10] Y. Zhu, "Measurement and Analysis of an Online Content Voting Network: A Case Study of Digg," in *Proc. of WWW*, 2010.
[11] "Digg website," http://www.digg.com.
[12] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems," *Proc. of EuroSys*, 2006.
[13] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of YouTube Videos," in *Proc. of IEEE IWQoS*, 2008.
[14] S. Annapureddy, "Providing Video-on-Demand Using Peer-to-Peer Networks," in *Proc. of IPTV Workshop, WWW*, 2006, pp. 238–247.
[15] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand Be Profitable?" in *Proc. of SIGCOMM*, 2007.
[16] B. Cheng, L. Stein, H. Jin, X. Liao, and Z. Zhang, "GridCast: Improving Peer Sharing for P2P VoD," *ACM TMCCA*, vol. 4, no. 4, pp. 1–31, 2008.
[17] M. Zhou and J. Liu, "A Hybrid Overlay Network for Video-on Demand," in *Proc. of ICC*, 2005.
[18] T. T. Do, K. A. Hua, and M. A. Tantaoui, "P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-To-Peer Environment," in *Proc. of ICC*, 2004.

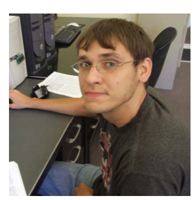
[19] A. L. H. Chow, L. Golubchik, S. Khuller, and Y. Yao, "On the Tradeoff between Playback Delay And Buffer Space in Streaming," in *Proc. of IPDPS*, 2009.
[20] X. Cheng and J. Liu, "NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing," in *Proc. of INFOCOM*, 2009.
[21] N. Vratonjic, D. Kostic, P. Gupta, and A. Rowstron, "Enabling DVD-Like Features in P2P Video-on-Demand Systems," in *Proc. of SIGCOMM Peer-to-Peer Streaming and IP-TV Workshop*, 2007.
[22] W. P. K. Yiu, X. Jin, and S. H. G. Chan, "VMesh: Distributed Segment Storage for Peer-To-Peer Interactive Video Streaming," *JSAC*, 2007.
[23] K. C.-J. Lin, C.-P. Wang, C.-F. Chou, and L. Golubchik, "SocioNet: A Social-Based Multimedia Access System for Unstructured P2P Networks," *TPDS*, 2010.
[24] "PPStream," http://www.ppstream.com.
[25] "UUSee," http://www.uusee.com.
[26] Y. Huang, T. Z. J. Fu, D.-M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, Design and Analysis of a Large-Scale P2P VoD System," in *Proc. SIGCOMM*, 2008.
[27] "Dalesa: the Peer-to-Peer web cache," http://www.dalesa.lk/.
[28] "Downloads Chat, E-mail, & the Web Music & Video Sharing," http://www.pcworld.com/downloads/1457/topic.html?&page=2.
[29] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: Peer-To-Peer Media Streaming Using CollectCast," in *Proc. of ACM Multimedia*, 2003.
[30] C. Liang, Z. Fu, Y. Liu, and C. W. Wu, "Incentivized Peer-Assisted Streaming for On-Demand Services," *TPDS*, 2010.
[31] K. Pussep, O. Abboud, F. Gerlach, R. Steinmetz, and T. Strufe, "Adaptive Server Allocation for Peer-assisted VoD," in *Proc. of IPDPS*, 2010.
[32] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot, "An Upload Bandwidth Threshold for Peer-To-Peer Video-on-Demand Scalability," in *Proc. of IPDPS*, 2009.
[33] A. L. H. Chow, H. Yang, C. H. Xia, M. Kim, Z. Liu, and H. Lei, "EMS: Encoded Multipath Streaming for Real-time Live Streaming Applications," in *Proc. of ICNP*, 2009.
[34] J. Sacha, "Exploiting Heterogeneity in Peer-to-Peer Systems Using Gradient Topologies," in *Doctor of Philosophy (Computer Science)*, 2009, University of Dublin, Trinity College.
[35] "KaZaA," 2001, www.kazaa.com.
[36] "Gnutella," http://www.gnutella.com.
[37] R. Kumar, J. Liang, and K. Ross, "The FastTrack Overlay: A Measurement Study," *Computer Networks*, 2006.
[38] J. Kubiatowicz, Y. C. D. Bindel, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An architecture for global-scale persistent storage," in *Proc. of ASPLOS*, 2000.
[39] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, "Pond: The oceanstore prototype," in *Proc. of FAST*, 2003.
[40] O. Heckmann, A. Bock, A. Mauthe, and R. Steinmetz, "The eDonkey file-sharing network," *GI Jahrestagung*, 2004.
[41] M. Kleis, E. K. Lua, and X. Zhou, "Hierarchical peer-to-peer networks using lightweight superpeer topologies," in *Proc. of ISCC*, 2005.
[42] Z. Xu and Y. Hu, "Sbarc: A supernode based peer-to-peer file sharing system," in *Proc. of ISCC*, 2003.
[43] S. A. Baset and H. Schulzrinne, "An analysis of the Skype peer-to-peer internet telephony protocol," in *Proc. of INFOCOM*, 2006.
[44] L. Gong, "JXTA: A network programming environment," *IEEE Internet Computing*, vol. 5, no. 3, pp. 88–95, 2001.
[45] Y. J. Pyun and D. S. Reeves, "Constructing a balanced, (log(n)/loglog(n))-diameter super-peer topology for scalable P2P systems," in *Proc. of P2P*, 2004.
[46] L. Garcs-Erice, E. W. Biersack, P. Felber, K. W. Ross, and G. Urvoy-Keller, "Hierarchical peer-to-peer systems," in *Proc. of Euro-Par*, 2003.
[47] Y. Li, X. Huang, F. Ma, and F. Zou, "Building efficient super-peer overlay network for DHT systems," in *Proc. of GCC*, 2005.
[48] B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *Proc. of International Conference on Data Engineering (ICDE)*, 2003.
[49] L. Xiao, Z. Zhuang, and Y. Liu, "Dynamic layer management in superpeer architectures," *TPDS*, 2005.
[50] Z. Li, G. Xie, and Z. Li, "Efficient and Scalable Consistency Maintenance for Heterogeneous Peer-to-Peer Systems," *TPDS*, 2008.
[51] H. Shen and C.-Z. Xu, "Hash-based Proximity Clustering for Efficient Load Balancing in Heterogeneous DHT Networks," *JPDC*, 2008.
[52] H. Shen, L. Zhao, Z. Li, and J. Li, "A DHT-Aided Chunk-Driven Overlay for Scalable and Efficient Peer-to-Peer Live Streaming," in *Proc. of ICPP*, 2010.
[53] D. T. C. Mastroianni and O. Verta, "A super-peer model for resource discovery services in large-scale grids," *Future Generation Computer Systems*, 2005.
[54] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location And Routing for Large-Scale Peer-To-Peer Systems," in *Proc. of IFIP/ACM Middleware*, 2001, pp. 329–350.
[55] P. B. Godfrey, S. Shenker, and I. Stoica, "Minimizing churn in distributed systems," in *Proc of SIGCOMM*, 2006.
[56] "Creating Videos for YouTube," http://www.homevideo101.com/creating-videos-for-youtube/.
[57] "The Peersim Simulator," http://peersim.sf.net.
[58] "The Difference Between Upload and Download Speed for Broadband DSL," http://www.broadbandinfo.com/cable/speed-test.
[59] H. Shen, L. Zhao, H. Chandler, J. Stokes, and J. Li, "Toward P2P-based Multimedia Sharing in User Generated Contents," in *Proc. of IEEE INFOCOM*, 2011.
[60] "Big boards," http://www.big-boards.com/.

Harrison Chandler Harrison Chandler is currently working towards his BS degree in Computer Engineering at Clemson University. His research interests include peer-to-peer and file sharing systems.



Haiying Shen Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Clemson University. Her research interests include distributed computer systems and computer networks, with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She was the Program Co-Chair for a number of international conferences and member of the Program Committees of many leading conferences. She is a Microsoft Faculty Fellow of 2010 and a member of the IEEE and ACM.

Lianyu Zhao Lianyu Zhao received the BS and MS degrees in Computer Science from Jilin University, China. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering of Clemson University. His research interests include wireless sensor network, routing protocols, applications and security issues in P2P networks.



Jared Stokes Jared Stokes received the BS degree in Computer Engineering at Clemson University. His research interests include peer-to-peer and file sharing systems.

Jin Li Jin Li is currently a Principal Researcher managing the Multimedia Communication and Storage team at Microsoft Research, (Redmond, WA). He received his Ph.D. from Tsinghua University (Beijing, China) in 1994. After working at USC and Sharp Labs of America, he joined Microsoft Research in 1999. Dr. Li has published in top conferences and journals in a wide area, cover audio/image/video compression, multimedia streaming, VoIP and video conferencing, P2P networking, distributed storage system with erasure coding and deduplication, high performance storage system design. His invention has been integrated into many Microsoft products, such as Microsoft Office Communicator/Lync, Live Messenger, Live Mesh, Windows 7, Windows 8, etc.. He holds 45 issued US patents. He is the lead TPC Chair for ICME 2011.

6 Forum Behavior Analysis

6.1 Background and Measurement Methodology

The Disney discussion forums at DISBoards.com are aimed at letting users share their Disney travel experiences and offer tips to others seeking a Disney vacation. According to Big-boards.com, the Disney discussion forums are the 28th largest forum site on the Internet with more than 34 million posts as of July, 2010 [60].

In a large forum site such as DISBoards.com, as shown in Figure 16, there usually exists a hierarchical structure. A *forum* is the smallest forum unit with one specific topic. In a forum, a *thread* denotes one discussion consisting of the original *post* by the thread creator and *replies* contributed by other users.

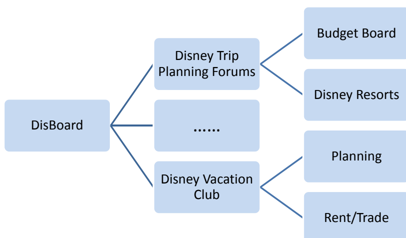


Fig. 16: A forum structure.

In order to quantify forum usage behavior, we randomly selected 21 Disney discussion forums and retrieved user and thread data. The 21 forums along with their thread and post counts are shown in Table 2. The crawl script went through every thread in the chosen forums that had received a post between 5/13/2010 and 6/13/2010; for each post on a thread, the post time and the user ID of the poster were collected. 13,807 threads were crawled in all. This yielded around 27,500 unique user IDs, representing 11.1% of DISBoards registered users. For every user ID, we fetch the statistics from the user's DISBoards page, including the total number of posts and the forum name, date, and time of the 500 most recent posts. We were only able to collect the 500 most recent posts due to DISBoardssearch limits; however, only 4.8% of the users had more than 500 posts. Since we mainly focus on the behaviors of most users with the objective of improving the quality of experience of most users, incomplete sets of posts of 4.8% users will not greatly affect our observations of most users from the trace. To track views on a thread over time, we repeatedly crawled the forums for the desired time increment. Each of the 21 forums was crawled for views on 6/28/2010 and 6/29/2010 to determine views on threads for the one day period. Additionally, we tracked the views on threads in eight forums approximately every half hour from 6/23/2010 to 7/2/2010.

In establishing the design principles of MBoard, we follow the logic of: (1) Is there an increasing demand for multimedia sharing forums? (2) Is P2P model suitable for multimedia sharing in forums? (3) What are the characteristics of forums including user activities and threads that we can take advantage of to optimize our design?

6.2 Is There an Increasing Demand for Multimedia Forums?

Our first question is whether there an increasing demand for multimedia forums? An increasing demand for multimedia sharing will require forums to provide more bandwidth at a low cost, which necessitates the need of MBoard in forums nowadays. Figure 17 (right Y axis) shows the average size of posts including embedded media contents in bytes in a forum for each month from the period of Jan 2005 to Jan 2010. The figure shows a clear trend towards larger forum posts, which rises from 10kb to 80kb per post, supporting the notion that forums will be required to store and deliver more and more content. Additionally, Figure 17 (left X axis) also demonstrates that the trend is due to a greater number of multimedia elements including embedded 3rd party provided content in the posts; thus, providing users with an easier way to upload their content becomes a priority. Although the need for multimedia sharing is currently emerging in forums, most forums only provide very limited attachment size support, typically hundreds of KBs. This is insufficient to display high resolution images, so these multimedia files, including high-resolution photos and video clips, are linked from external 3rd party service providers. Storing and sharing the multimedia files among peers rather than the server can allow users to upload their large multimedia files directly to forums.

Figure 18 shows the estimated storage cost of different forum websites. We found the total number of posts on each forum as of July 26, 2010 from [60]. We multiplied the number of posts in each forum by the average post size (including the size of embedded media content) calculated from the trace data of DISBoards to get an estimate of the storage cost for each forum. The figure shows the storage cost of forums varies from 100GB to 10 TB. Thus, it is a challenge to the servers to serve such a tremendous amount of content to the vast number of users.

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

What is the scale of MBoard deployment? Figure 19 shows the total number of users for each of the 21 DISBoards forums. It shows that the number of users in these forums ranges from 800 to 10,000, and the number is increasing yearly as shown in Figure 20. There could be even a larger number of anonymous users, so the number of users in forums may be very large. Figure 21 shows the number of threads in each forum. We see that the thread count per forum varies widely from forum to forum. Although 60% of the forums have less than 5,000

TABLE 2: List of forums analyzed with their thread and post counts.

Forum	Threads	Posts
Theme Park Attractions	244,799	3,134,586
Disney Resorts	183,318	2,114,282
Disney Restaurants	167,997	1,685,981
Camping at Disney World	114,744	252,074
Transportation	82,259	612,547
DVC-Operations	60,302	600,358
Budget Board	51,440	1,575,845
Disney For Families	51,440	738,883
DVC-Planning	38,073	339,610
Orlando Hotels & Attractions	32,775	212,862
Teen Disney	31,341	1,419,451
DVC-Mousecellaneous	24,703	427,729
Disney Weddings	22,374	320,870
Disney Trip Reports	22,146	1,083,226
Disney World Tips	17,348	187,730
Welcome Board	15,858	162,955
DisABILITIES!	13,626	124,539
Disney for Adults	4,670	110,951
The College Board	3,730	66,812
Gay and Lesbian at Disney	3,526	89,112
Adventures by Disney	1,492	19,083

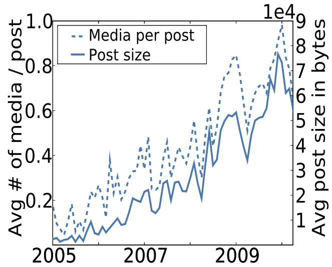


Fig. 17: Multimedia elements.

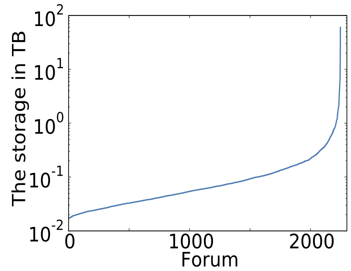


Fig. 18: Burden on the server.

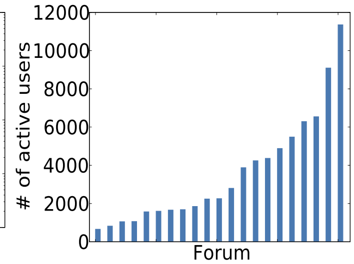


Fig. 19: Number of users.

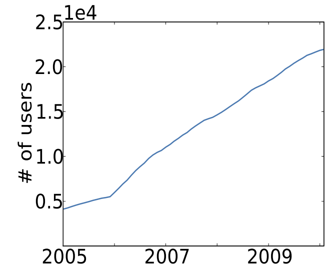


Fig. 20: Number of users.

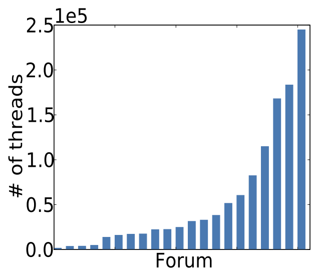


Fig. 21: Number of threads.

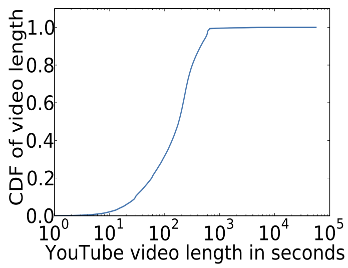


Fig. 22: YouTube video lengths.

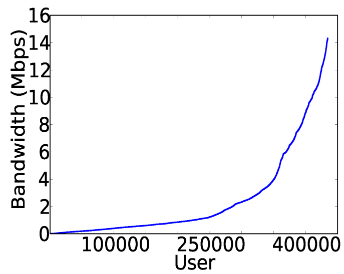


Fig. 23: Bandwidth distribution.

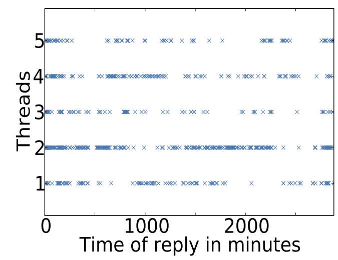


Fig. 24: Time coverage.

threads, the number of threads in popular forums is around 0.1-0.25 million threads. These figures indicate that the scale of a popular forum can be very large in terms of both the number of threads and users, which puts a tremendous strain on centralized servers, making their bandwidth a bottleneck.

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

Therefore, we should resort to a more efficient use of users' resources. The P2P model is a promising method to tackle this problem. With this model, a peer retrieves its interested contents from other peers, and it resorts to the server if no other peer has the contents.

One question is how do we determine the suitable media size limit? Since there is no existing media sharing forum, we resort to YouTube statistics for the answer. Figure 22 shows the Cumulative Distribution Function (CDF) of YouTube video lengths. We see that 80% of the videos are approximately 300 seconds or less. Most of the remainder of the videos posted on YouTube range anywhere from (300,600] seconds. There are only a very small number of videos that have lengths exceeding 600 seconds. This is due to the fact that videos uploaded to YouTube cannot be longer than 10 minutes (i.e. 600 seconds) in length now. Once, certain accounts were given the ability to upload videos exceeding 10 minutes in length, but this privilege is no longer given out [13]. Therefore, there are a small number of videos that exceed 600 seconds in length. We see that the large majority of videos posted on YouTube are short in length and that only 20% of all videos have a length greater than 300 seconds. In a conclusion, most of the user generated contents are shorter than 10 minutes.

6.3 Is P2P Model Suitable for Multimedia Sharing in Forums?

P2P multimedia sharing requires that a certain number of peers exist in the forum to provide contents needed. A large number of users (O2) bring more benefits in using the P2P model, since more peers possessing a content file increases the content availability and downloading efficiency due to a more balanced load distribution. Figure 23 shows the bandwidth resource distribution from a MSN video trace collected by Microsoft, drawn based on the bandwidth of recorded users in an ascending manner. It shows that most users have decent bandwidth for video viewing with about 50% of users having more than 1Mbps and 35% of users having more than 2Mbps. This is important to the practicality of MBoard, since larger bandwidth brings faster multimedia download and sharing. By leveraging the bandwidths of all users in a forum, the load on the server can be greatly reduced.

The P2P model requires the existence of peers constantly viewing threads, so that the peers can help by uploading their watched content to others. Is there sufficient user online activity in a forum to facilitate P2P assistance? We focus on the characteristics of forum-level behaviors in order to prove that forum is suitable in peer-assisted environment. Figure 24 shows the activity time coverage of five randomly chosen popular threads in the Theme Park Forum, a medium sized forum out of the 21 forums. We see that in each time spot in the X axis, there are always some replies in the Y axis direction. It implies that there always exist users every minute in a forum. If we consider all threads in a forum, there would be more users existing in the forum every minute. If we combine these threads' time coverage together, we see there are always some peers available that can be utilized to upload contents to others. This is the foundation for running multimedia forums in a P2P manner.

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.

Then what are the characteristics of threads that MBoard can benefit from? Since all threads receive some views, we use replies to decide whether a thread has been active for some time. Figure 25 is a plot of the percentage of the total threads in each forum that never receive a reply. We see that an average of approximately 5% of the threads created in the forums never receive a reply. This tells us that approximately 5% of the total threads in a forum will fully depend on their server to keep the media content accessible due to the extremely unpopularity. But more importantly, this tells us that approximately 95% of the total threads in a forum will be active for some time, and

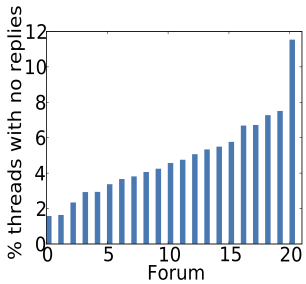


Fig. 25: Percentage of threads that receive no replies in each forum.

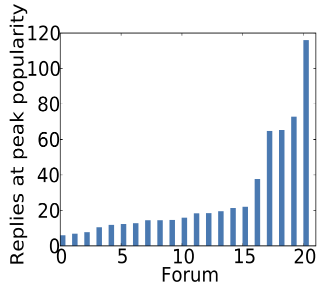


Fig. 26: Total number of thread replies on its peak day.

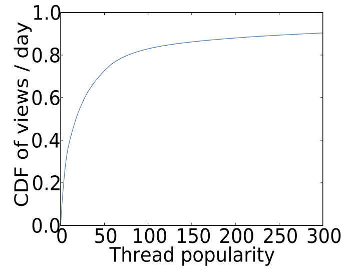


Fig. 27: CDF of the number of thread views per day.

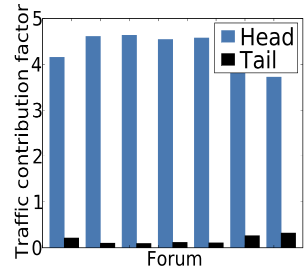


Fig. 28: Head/tail contribution.

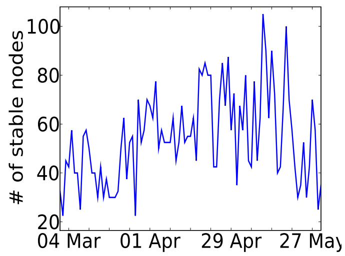


Fig. 29: Number of stable nodes.

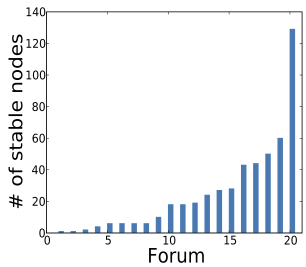


Fig. 30: The number of stable nodes per forum.

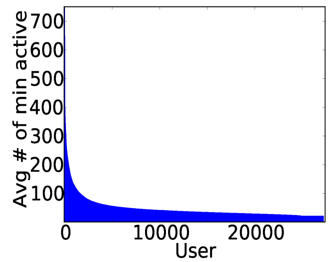


Fig. 31: Daily online time.

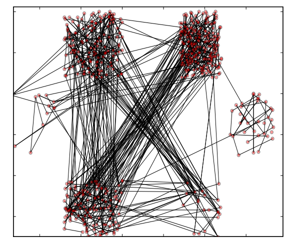


Fig. 32: Connectivity between threads.

thus the media files in those threads can be stored in the P2P network for some period of time.

Figure 26 shows the total number of thread replies on the peak day. From this plot, we see that the average number of replies is somewhere around 30 and the highest number of replies is 120. This gives a good indication of the media content availability on the peak day. Once a thread hits its peak day, its media content will be widely available in the P2P network while it would otherwise exacerbate the load burden on the sever in the server-client model.

Figure 27 is a plot of the CDF of the number of thread views over a 24 hour period in the 21 forums. As we see from the plot, 10% of the total threads have greater than or equal to approximately 300 views. 50% of threads have less than 30 views per day. The middle 40% of threads have median-popularity with a view count between 30 and 300. We call the threads that receive a high number of views *head content*, and the ones receive a low number of views *tail content*. In conclusion, the amount of head content is small, but attracts many more views and results in higher P2P availability than tail content, while content with medium popularity can be made available by P2P to a certain extent.

Figure 28 shows the head/tail content contribution of seven randomly chosen forums. We define the *traffic contribution factor* as $\frac{\textit{total thread accesses}}{\textit{number of threads}}$ and choose a cutoff of head and tail contents at the 20% of the total number of views of all threads. We can see that the head content on average has a contribution factor greater than 4 and contributes more than 80% of all thread viewing traffic, whereas tail content has a very small contribution factor and contributes less than 20% of the traffic.

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.

6.4 What are the Characteristics of the Forums?

We would like to know whether a number of relatively stable nodes are always present in the forums, which can be exploited in the P2P model to enhance media content availability. We regard a *stable node* as a user that posts in one forum at least five times a day. The stable nodes are most likely the administrators and highly active users in the forums. This is an indirect measurement because collectable statistics do not provide the online time of each node.

From Figure 29, which was taken from a relatively popular forum, we see that the number of stable nodes is not constant over time. However, the average number of stable nodes is around 40, with a maximum of more than 100 and a minimum of 25. Figure 30 shows the number of stable nodes per forum. From this figure, we see that the number of stable nodes ranges from [0,130] and 20 out of 21 forums have stable nodes.

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

We consider a user to be online for 10 minutes if he posts/replies a thread. A user is considered to be continuously online for the duration if he continues to post at least once per hour. Figure 31 shows the average number of minutes spent online for 25,000 randomly chosen users. From this figure we see that most users spend an average of about 40 minutes online. Also, there are a small number of users that spend from 2 to 10 hours a day, which also confirms the existence of stable nodes. If we reduce the assumed online time of a user for one posing, we can still see from the figure that there exist relatively stable nodes though the number is reduced.

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.

Another important question is the scope of nodes that MBoard should cluster in order to achieve the best communication efficiency. We should cluster nodes that always visit the same threads. Clustering nodes with less common interested threads will lead to unnecessary P2P structure maintenance cost and lookup cost in a large cluster. When a number of threads share many nodes that always reply and switch between these threads, we say these threads are highly connected by the user thread replying activities. A user may like to view different threads in one forum or switch

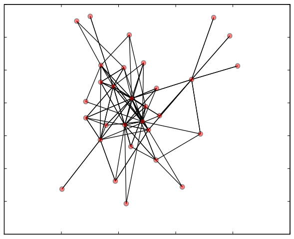


Fig. 33: Connectivity between forums.

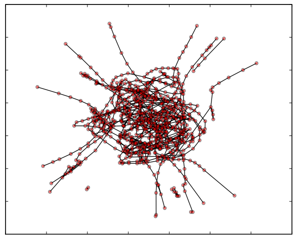


Fig. 34: Thread connectivity by five users' posts within one forum.

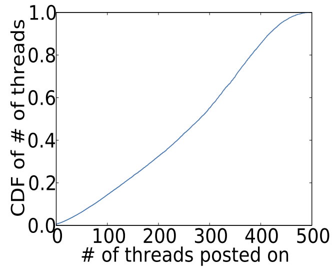


Fig. 35: Num. of threads replied to.

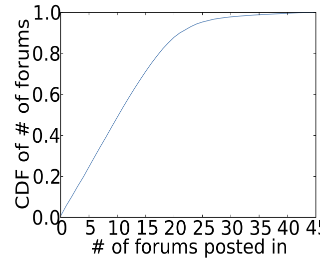


Fig. 36: Num. of forums posted on.

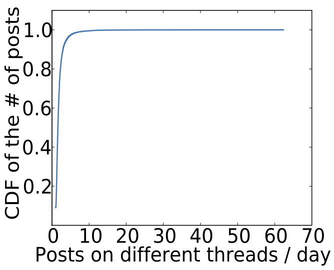


Fig. 37: Posts on diff. threads.

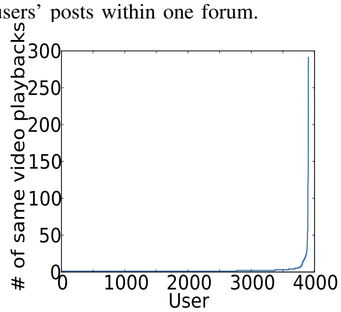


Fig. 38: Playbacks of same video.

between different forums. We analyze the user thread replying activities to see whether some threads within a forum, all threads within a forum, or some threads in several different forums are more highly connected. Based on the observation results, we can know which group of nodes share more similar replying activities and hence should form into a P2P network in the design of MBoard.

Figure 32 shows the activities of 20 randomly chosen users in five forums. Each dot represents a different thread. A link between two or more threads indicates that a user posted a reply on one thread and then went on to the other thread(s) and posted a reply. By grouping the threads based on the forum, we observe that although some forums with more thread posting activities are connected closely, the lines are most densely connected within forums. There are connections between forums, but they are not as dense as those within forums. In the right cluster, the connections are sparse within the forum and it only has two connections with other forums. This is because the threads in this forum are not popular and few users visited the threads. This figure implies that the users switch between threads within a forum more frequently than between forums. Thus, we conclude that it is reasonable for MBoard to group threads by forum.

To confirm this conclusion, we study the connections between forums. Figure 33 shows the connectivity between the different forums. Each dot in this figure represents a different forum while each edge means a user switches from one forum to another. Here, we see that there are quite a few forums that are densely connected. However, there is also a fair number of forums that are not as densely connected. This figure confirms that the clustering feature of forums is not obvious and few users share interests in a number of the same forums. Thus, threads should be grouped by each forum rather than a group of forums.

The next question is whether we can further cluster all the threads within a forum. That is, whether the threads within a forum show a clustering feature. Thus, we study the connections between all threads within a forum. Figure 34 is a plot showing the connectivity of different threads visited by five randomly chosen users within a single forum. It is basically a map of the users' posting activities within a forum. The high weight of the link connecting nodes is caused by two thread visiting patterns: (1) multiple users reply to the same thread and go on to reply to another thread in the same order, and (2) one user replies to a thread, proceeds to reply on another thread, and then returns to reply on the former thread. From this plot, we see that the majority of user activity is tightly clustered in one central location. Some thread reply paths tend to branch out in an area by themselves, but the number of these paths is relatively small when compared to the number of thread reply paths that are tightly connected. These threads are unpopular threads with fewer visits from users. This plot does not exhibit a clustering feature, which means that users in a forum tend to randomly visit all threads rather than preferring a certain group of threads. Therefore, all threads in a forum cannot be further clustered and forming all users in each forum into a P2P network is an optimal method.

Figure 35 shows the CDF of the number of different threads users post on. We see that 20% of users post on less than or equal to approximately 130 different threads. In fact, we see that only 50% of the users post on less than or equal to approximately 275 different threads, and 95% of users post on at least approximately 450 threads. This leads us to the conclusion that a large majority of the users tend to post on many different threads. Figure 36 shows the CDF of the number of different forums users post on. As we see from the plot, approximately 50% of users post on no more than 10 different forums. This increases to approximately 95% of users for no more than 25 different forums. We also see that only 20% of users post on less than approximately 5 different forums. From these observations, we see that the majority of the users tend to post on anywhere from 10 to 20 different forums. This is likely due to the fact that most forums are interest based and users tend to have a certain number of different interests. Therefore, the many threads posted by users are spread among only a few forums. This is a reason that we cannot find nodes sharing interests in several common forums (Figure 33). Thus, enabling nodes in one forum to share contents between each other is an optimal method. The observation results confirm that it is reasonable for MBoard to cluster the threads within one forum together.

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.

In designing a system that uses a P2P structure where media content is pulled from other users who cached the content, we need to answer this question: what is the number of threads that users are interested in every day? Figure 37 shows the CDF of the number of different threads across forums that

segment ID	count down timer	last used time
c9e2101eb463ca44515e08719b71373c	2980000	343920000
24557615bfc6af45b9a29b5078c58828	12500	342720000

Fig. 39: Example of cache content in some node.

users reply to per day. Here we use reply activity to indicate strong interest in a thread. From this figure, we see that 90% of the users reply to approximately 3 different threads per day. Nearly 100% of the users reply to no more than 10 different threads per day. This indicates that most users are actually interested in a small number of threads in a day. Additionally, this implies that the number of pieces of media content cached in users' computers could be small, so the cache burden will not be heavy for users.

Figure 38 shows that users seldom watch the same video again. From the distribution of users' multiple playbacks of one video, we can see that most users do not watch videos they have seen before, meaning the cached videos are seldom used by users for replaying.

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.

7 Refreshing Scheme in MBoard

Based on O2, MBoard uses a refreshing time to discard content indices to content providers periodically to ensure the freshness of the indices and reduce communication cost. MBoard also does not need large cache according to O8.

In order to ensure the validity of the entries in the content table, upon departing, nodes should notify the brokers of the contents they are providing. A provider registered to a broker still may not be available due to a number of reasons: (1) it goes offline; (2) it stops providing uploading service; or (3) it deletes the cached videos. Therefore, the brokers need to update the index information in time in order to ensure that the chosen providers are in service. One way to deal with this problem is to let each node notify its broker before leaving. However, due to the high node join and departure frequency, this will generate a high communication overhead. In order to minimize the overhead, MBoard lets brokers automatically discard the registered indices which were reported a certain time period ago. We call this time period the *refreshing time*, denoted as t_r. MBoard sets t_r equal to the continuous online time of the majority of the nodes.

We assume that each user can tolerate uploading its content for refreshing time t_r. After t_r, users can choose to continue to be in service if they are still online. MBoard can use tit-for-tat to assign more bandwidth to those users who upload more in order to encourage them to contribute their bandwidth for peer assistance. If a segment provider keeps sharing the segments after the refreshing time period, it will register with the broker again to refresh its service. Otherwise, the broker assumes this provider is no longer valid. The broker can assign each piece of the registered content a timer with a refreshing time value. When a piece of content is registered, the timer count-down begins. This piece of content will be automatically marked unavailable after t_r. When this item is registered again, the timer will be refreshed. In this way, MBoard can ensure the availability of providers while reducing communication overhead.

O4 implies that a thread's popularity deteriorates within days, so MBoard does not require users to hold a large cache

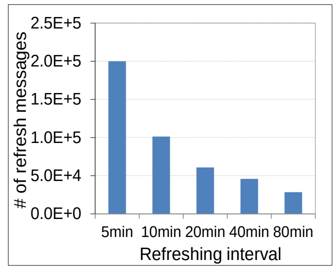


Fig. 40: Refreshing cost.

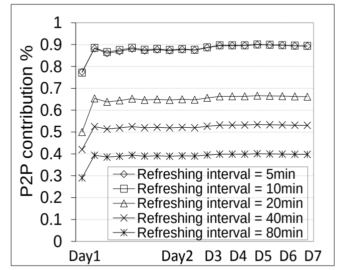


Fig. 41: Effect of refreshing period on P2P contribution.

such as 1GB (the typical cache size of the PPLive [6] client) for the forum content. The cache of each user is organized into a table for easy look up and service refreshing, called the *cache table* in MBoad. A typical user's cache table is shown in Figure 39. In addition to the segmentID, the "count down timer" is set to the refreshing time. When it counts to zero and the user is still online, the service of the segment will be refreshed by re-registering. "Last used time" is used in knocking out the outdated cached items if the cache size limit is reached.

8 Additional Experimental Results

8.1 The Effect of Refreshing Scheme & Service Period

In testing the effect of the refreshing scheme, we assign users different continuous online time according to DISBoards' user activities and our assumption in Figure 31 that if a user replies, its online time is 10 minutes and is increased if he replies again within 1 hour. Figure 40 shows the number of messages generated by service refreshing under different refreshing interval settings. We see that a small refreshing interval produces more messages than a large refreshing interval due to the more frequent communication between nodes and brokers. Figure 41 shows the P2P contribution percentage under different refreshing interval settings. We see that longer refreshing interval leads to lower P2P contribution. The reason that the server has to serve more requests is that most nodes have a short online time, so a long refreshing interval cannot let the broker update the availability information of segment providers in a timely manner. For refreshing intervals equal to 5 and 10 minutes, the P2P contribution remains very high because the majority of nodes have 10 minutes continuous online time. In conclusion, this experiment confirms our design that the optimal refreshing interval setting is equal to the online time of the majority of nodes in the network, which is 10 minutes in this experiment.

A node's service period is the time period it is in service, i.e., the time period it can upload a video after starting to view a video. Figure 42 shows the CDF of the percentage of users versus the playback delay with different service periods. We see that the 3 minute service period leads to a slightly higher delay than other service periods. This is because when the service period is short, the server uploads more content and the queuing delay becomes larger. We also see that other service periods achieve a similar delay distribution. When the service period reaches a certain level, the nodes in service are sufficient to serve the new requests. In all cases, around 90% of users have delays no more than 0.8 seconds, which implies the effectiveness of P2P model for video sharing. The

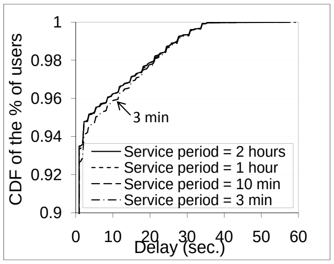


Fig. 42: Effect of service period on delay.

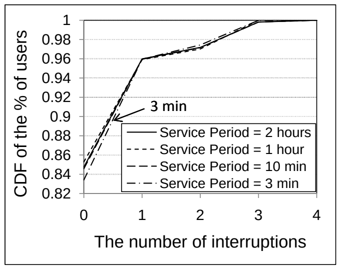


Fig. 43: Effect of service period on playback smoothness.

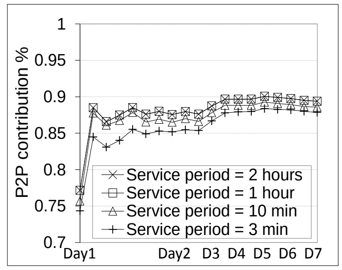


Fig. 44: Effect of service period on P2P contribution.

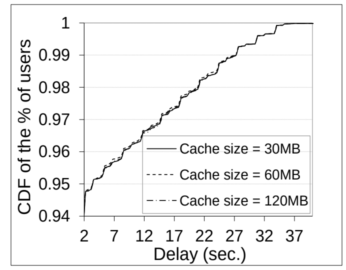


Fig. 45: Effects of cache size on delay.

newly joined nodes take over the videos from the existing nodes (which will not be in service) for sharing, which increases video availability even with a short service period. The delay of the remaining approximate 10% of users exhibits an exponential increase. In more detail, 94%, 95%, 96%, 97% and 99% of users have delay no more than 2 seconds, 5 seconds, 10 seconds, 20 seconds and 50 seconds, respectively. This delay increase is due to the churn nature of forums where users constantly come and leave. A node needs to find video providers again if it has not received response from a video provider who is leaving the system.

Figure 43 shows that the CDF of users' playback interruptions varies slightly with different service periods, with the most interruptions occurring at a service period of 3 minutes. These results are caused by the same reasons as in Figure 42. In conclusion, the length of the service period does not have a great impact on delay or smoothness, and a certain service period decreases the video retrieval delay to a certain extent, with the optimal choice being 10 minutes.

Figure 44 shows the P2P contribution in MBoard over 7 days with different service periods. From this figure, we see that as the service period increases, the P2P contribution percentage increases as well. We see that if users are in service for only 3 minutes, MBoard reaches approximately 85% P2P contribution, while a 2 hour service period leads to approximately 90%. Overall, we see that the longer a user is in service, the larger the P2P contribution. Combining this observation with that of Figures 42 and 43, we conclude that MBoard should require users to be in service for at least 10 minutes (the online time of the majority of nodes in the network) as a contribution for a better user experience in the forum. This is easy to implement since a 10 minute service period is not much longer than the average length of a video.

8.2 The Effect of Cache Size

Figure 45 shows the CDF of the percentage of users versus the video playback delay with different cache sizes in each node. From this figure, we see that more than 95% of users have a delay that falls within the range of (0, 5] seconds, which is a well acceptable time for a user to wait. The rarely seen longest waiting time is around 35 seconds. Since larger cache size increases the video availability in peers, it is very intriguing to see that the effects of cache size on the video playback delay are almost negligible. Recall that users tend to view a small number of threads in a forum in O8 and the number of viewers for popular videos is almost always constant. As a result, some providers for a request always exist even with a small cache size. If there is no provider for a request, such as for unpopular videos, the server provides the service. This result is mirrored in Figure 46, which shows that the CDF of users' playback interruptions barely changes with different cache sizes.

Figure 47 shows the effects of the cache size on the P2P contribution. From this figure, we see that the P2P contribution for a system with cache sizes of 60MB and 120MB are nearly identical. The system with cache size of 30MB has slightly lower P2P contribution. Combining this with the observations in Figures 45 and 46, we see that a larger cache size only improves the performance of the system slightly.

8.3 The P2P Contribution of Head/Tail Content

In Figure 48, *top* $k\%$ refers to the k percent of threads that have the largest number of accesses in our trace period, or the most popular threads. We see that the contribution factor (defined in Section 6.3) of head content is much higher than tail content. Moreover, 5% of the top threads (head content) have a contribution factor of 6, meaning they contribute up to 30% of all accesses, whereas 30% top threads contributes more than 70% of all accesses. Therefore, we conclude that popular threads have very high demand in multimedia forums, and they can benefit greatly from a P2P model. In addition, we know that most of the threads are tail content that generate a small amount of traffic. Since the absolute number of accesses for such unpopular threads is so small, they can be handled easily by the server.

We can see from Figure 49 that the P2P contribution is magnified more in smaller sets of popular threads, with the top 5% content having a contribution of around 98%, while unpopular threads have low P2P contribution. This is because the more popular a thread is, the higher possibility that a user will find other users watching the same thread. This shows that in MBoard, nearly all of the traffic of popular threads is handled by other peers instead of the server. This reduces the server load and the amount of server bandwidth consumed. When considering all traffic, the P2P contribution is around 90%, which also shows the overall P2P efficiency of MBoard remains very high. Additionally, we observe a very slight decreasing trend in the P2P contribution after it reaches its peak on day 3. This is due to the limited period of the trace file. At the end of the trace, there are a few new videos that are requested from the server, but these videos have few chances to be shared in our collected events.

Figure 50 shows a snapshot of the number of providers of each thread collected one hour before the entire experiment finishes when each node's service period equals 10 minutes and 2 hours, respectively. The high results in the figure belong to the 2 hour service period test, which show that the number of available providers of these threads varies from 1 to 3. The number of available providers for the remaining threads in the 2 hour service period test is almost overlapped with all threads

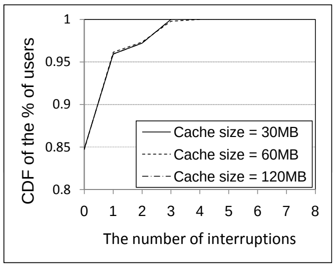


Fig. 46: Effect of cache size on playback smoothness.

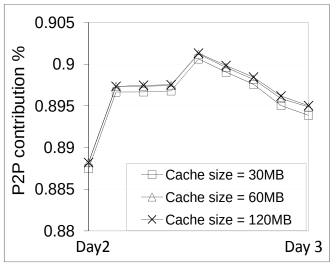


Fig. 47: Effects of cache size on P2P contribution.

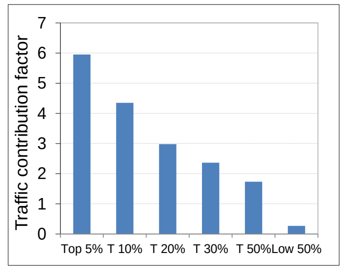


Fig. 48: Traffic contribution.

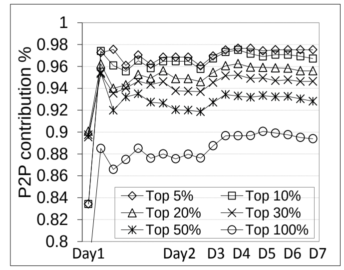


Fig. 49: Head content contribution.

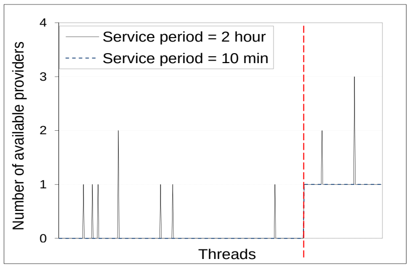


Fig. 50: Number of video providers for different threads.

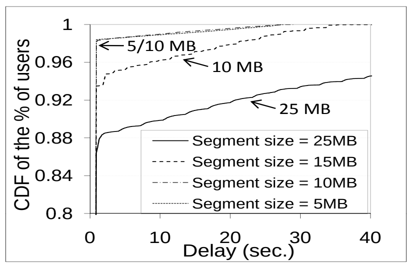


Fig. 51: Effects of segment size on delay.

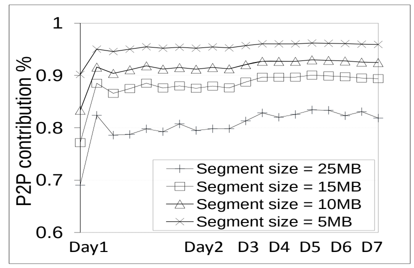


Fig. 52: Effects of segment size on P2P contribution.

in the 10 minute service period test. Specifically, 3/4 of the results are 0 (left of the dotted red bar) and 1/4 of the results are 1 (right of the dotted red bar). This is consistent with our findings in O4 that the majority of threads are not very popular. The overlapping demonstrates that the number of providers does not differ much between the 10 minute and 2 hour service periods. This again supports our conclusion that MBoard can adopt the 10 minute uploading configuration. Several threads in the 2 hour service period test have more available providers, which is caused by its longer service period.

8.4 The Effect of Segment Size

As mentioned in Section 3.3, MBoard sets the segment size to 15 MBytes so that users do not have to split their videos and have lower probability to fail in uploading and downloading a large-size video. In the experiment for "segment size=15mb", the size of videos is generated following the distribution of the embedded 3rd party video sizes in the trace data. In the experiment for "segment size=25mb", we multiply the previously generated video sizes by a factor of $\frac{25}{15}$ in order to ensure that the size of majority videos is 25mb. Thus, the size of each video is also increased.

Figure 51 shows the CDF versus the video playback delay. We see that increasing the segment size increases the playback delay, especially for the users that already have a relatively high delay. This is because as the video size increases, users need more time to download the content, and the users waiting in the queue need to wait longer for them to finish downloading, leading to queuing time increase. Also, downloading larger size videos has higher probability to fail. From Figure 52, we see that the segment size influences the P2P contribution. As the segment size increases, the P2P contribution percentage decreases. This is because larger segments require more time to be downloaded and made available to other peers, which reduces the possibility for a peer to successfully locate a segment provider in the P2P network. We also observe that the P2P contribution percentage is low on the first day and remains nearly constant in the remaining days in all cases due to the same reason explained in Figure 9.