Toward P2P-based Multimedia Sharing in User Generated Contents

Haiying Shen, Lianyu Zhao, Harrison Chandler, Jared Stokes
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{shenh, lianyuz, hchandl, 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 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 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.**

I. Introduction

With the advent of Web 2.0 applications, user generated content 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], [2]. Some of the most active forums today include 4chan [3], Gaia Online [4], Ultimate Guitar [5], Something Awful [6] and DISBoards [7].

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. For example, YouTube, which employs the server-client model, spends anywhere from \$83 million to \$380 million per year on bandwidth, storage, and software [8], an infeasible amount for forums. 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. Additionally, embedding content on 3rd party services limits the forum administrator's full control and authority on the forum contents. 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 [9].

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 [9], this is the first work to quantify this problem in the realm of forums. Second, we propose a scheme to provide forums with their own unique multimedia sharing capabilities called Multimedia Board (MBoard), which is a peer-assisted multimedia sharing system that produces low bandwidth cost.

Specifically, we identified the following properties of forum-based multimedia sharing and corresponding design consideration through the analysis of existing forums. We found that forums are suitable for P2P-based multimedia sharing. 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 and dominates bandwidth consumption. P2P sharing of head content can achieve high efficiency in video retrieval. (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 in one forum tend to share similar contents than nodes in another forum, MBoard builds the nodes in one forum into a P2P network. (2) Since there are always a number of stable nodes in a forum, MBoard builds the 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 II presents a concise review of related works. Section III presents our analysis on the collected forum activity data from the Disney discussion board. Section IV presents the design details of MBoard. Section V reports the simulation experimental results of MBoard. The final section presents a conclusion with a discussion on further work.

II. Related Works

Although we are exploring the use of peer assisted multimedia sharing within forums, the manner of media sharing is similar to P2P VOD. Although YouTube [10] is a centralized video sharing service provided by Google, it is currently the largest video sharing website in the world where people can upload videos freely. Its operation is based on the support of a huge number of background server clusters which cost millions of dollars per day in bandwidth. YouTube attempts to profit from commercials embedded in the website and videos themselves [11]. There are several exploits in utilizing peers to ameliorate the bandwidth cost for YouTube-like services [12]–[15]. An early work called nVOD [12] incorporates an unstructured network that can make best use of network resources while providing high-quality service. It also features network coding to better utilize network resources. Huang *et al.* [13] proposed the utilization of peer assistance in MSN Video (now Bing video), in which clients also share and relay traffic. It aims to maintain the quality of user experience while simplifying protocols such as using single uploading (i.e., peers can only upload one file at a time) and no cache support. GridCast [14], another deployment of peer assisted VOD service, identifies that the single uploading scheme leads to idling in the P2P networks and that multiple video caching can better reduce the server load. On the other hand, NetTube [15], which is based on the current YouTube service, attempts to identify the users that watch the same video in order to group them into the same overlay to share the video. Moreover, it utilizes the existing related video list in YouTube to help nodes prefetch certain videos in order to reduce the waiting time before playback. There is also work on the analysis of the user behaviors in P2P VOD systems such as [16], which reveals a poisson distribution of user arrival rates and an inverse correlation between video watching time and video popularity. In the area of online forum analysis, Zhu [17] studied the Digg [18] online content voting network by providing observations on user digging activities and content network rating issues. It shares similarity with the analysis of online forum user behaviors in this paper.

Most popular practical VOD systems, such as PPLive [9], PPStream [19] and UUSee [20], are based on the tracker service and program source provided by centralized servers and uploading contribution from peers. Huang *et al.* [21] provided an insightful design analysis of the PPLive application for future system design. In these applications, servers are in charge of both providing video source and helping users locate video resources, while most videos are shared by users' uploading capacity.

To our knowledge, this is the first work that quantifies user generated content characteristics in the area of forums, as well as explores the possibility of implementing a multimedia sharing service within forums that utilizes peer assistance.

III. Forum Behavior Analysis

A. 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 Bigboards.com, the Disney discussion forums are the 28th largest forum site on the Internet with more than 34 million posts as of July, 2010 [22].

In a large forum site, 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. In order to quantify forum usage behavior, we randomly selected 21 Disney discussion forums and retrieved user and thread data between 5/13/2010 and 6/13/2010. 13,807 threads were crawled in all. This yielded around 27,500 unique user IDs, representing 11.1% of DISBoards registered users. To track views on a thread over time, we repeatedly crawled the forums for the desired time increment. Each of the twenty one 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 principle 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?

B. Is There an Increasing Demand for Multimedia Forums?

Figure 1 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 1 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.

Observation(O)1: The number of multimedia elements has been rapidly increasing in recent years.

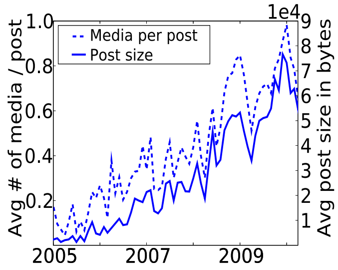

Fig. 1: Multimedia elements.

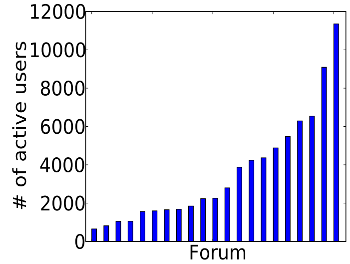

Fig. 2: Number of users.

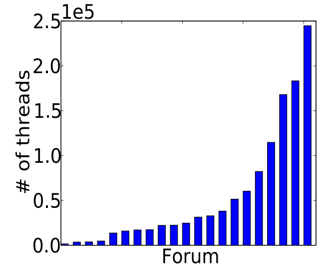

Fig. 3: Number of threads.

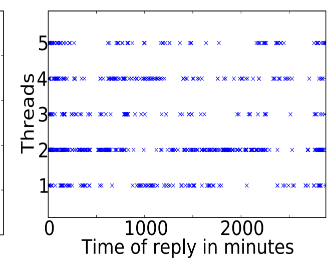

Fig. 4: Time coverage.

What is the scale of MBoard deployment? Figure 2 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. There could be even a larger number of anonymous users, so the number of users in forums may be larger. Figure 3 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 threads, the number of threads in popular forums may be 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.

C. Is P2P Model Suitable for Multimedia Sharing in Forums?

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 the forums to facilitate P2P assistance? Figure 4 shows the activity coverage of five randomly chosen popular threads in the Theme Park Forum, a medium sized forum out of the 21 forums. It shows that threads receive user replies for most of the day. 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.

We call the threads that receive a high number of views *head content*, and the ones receive a low number of views *tail content*. Figure 5 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 contributes to most of the traffic.

D. 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 at least 5 times a day, which 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 6, 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 max of more than 100 and a min of 25.

O5: A number of stable nodes are essentially always present in a forum.

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 7 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.

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 maintenance cost and lookup cost in a large cluster. To cluster nodes in a forum into a cluster, we need to consider whether the threads within each forum are highly connected measured by the user thread replying activities.

Figure 8 shows the activities of 20 randomly chosen users in five forums. Each dot represents a different thread. A link

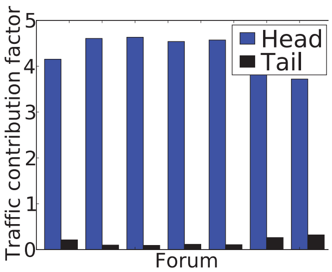

Fig. 5: Head/tail contribution.

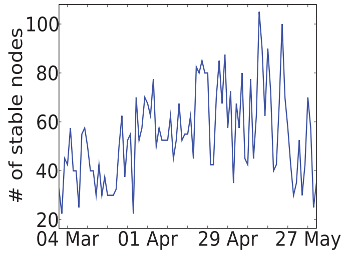

Fig. 6: Number of stable nodes.

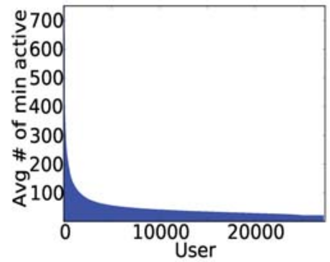

Fig. 7: Daily online time.

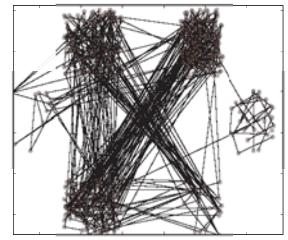

Fig. 8: Connectivity between threads.

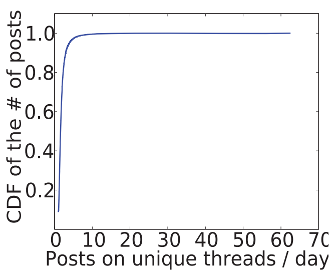

Fig. 9: Posts on different threads.

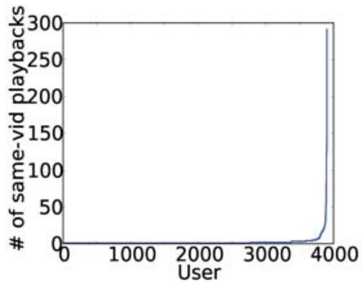

Fig. 10: Playbacks of same video.

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. Thus, we conclude that it is reasonable for MBoard to cluster threads by forum.

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 9 shows the CDF of the number of different threads across forums that 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. 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 10 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 seldomly 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.

IV. System Design

MBoard is designed to function as a download accelerator to compliment the existing forum server. MBoard clients need to form a specific network structure and follow the content retrieval primitives.

A. Overview

The observations in Section III 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, which 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.
- Refreshing scheme. Based on O2, MBoard uses a refreshing time to discard indices to existing content providers periodically to ensure the freshness of the indices and reduce communication cost. MBoard also does not need large cache according to O8.

B. P2P Construction

For the scale of network clustering, clustering all the forums into a P2P network will result in a large network with long latency, while clustering on a smaller scale will result in unavailability of requested contents in a cluster. 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, MBoard constitutes the nodes in each forum into a network. Such a design enables a user to find the requested content from other users within the forum most of the time. On the other hand, MBoard constrains the latency by avoiding building a large-size network.

The P2P model has two structures: unstructured and structured P2P (DHT). Unstructured P2Ps are mostly gossip and flooding-based, which incur large amounts of communication overhead in the network. 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, DHT is not an optimal choice. However, high communication overhead also makes unstructured P2P not a good choice. Through O5, we know that there are a fair number of *stable nodes* that remain active in the forum most of the day. Hence, MBoard intelligently forms 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 stable nodes in the lower tier. The nodes connected to a stable node are called *child nodes* of the stable node. Since stable nodes perform media content indexing, they are called *brokers*.

A DHT uses a consistent hash function [23] 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 DHTs, 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 [24] DHT.

For the best performance, the number of stable nodes N should not be large to avoid long routing latency. On the other hand, N should not be too small to avoid generating bottlenecks. To manage the stable nodes, the server maintains a list of them. To determine N, the administrator of the forum can analyze the number of stable nodes as we did in Figure 6 and use the long term average value.

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 [25]. When a node u's daily online time exceeds a pre-defined threshold, it reports to the server for the promotion to a stable node on the DHT. The server promotes u to a stable node if the number of stable nodes has not exceed N. Then, the server assigns a bootstrap node from its stable node list to the newcomer, and the node joins in the DHT using the DHT node join protocol. Each time a stable node leaves the network, in addition to executing the DHT departure protocol, it also notifies the server.

When a node joins in MBoard, the server randomly picks a stable node and assigns it to the newcomer as their *parent*. A parent helps its child nodes to send out content requests and receive replies from other nodes. Consequently, a two level DHT structure is formed as shown in Figure 11. Considering the high dynamism of child nodes, we let child nodes build and maintain connections to their parent, so that parents do not need to maintain connections to their child nodes.

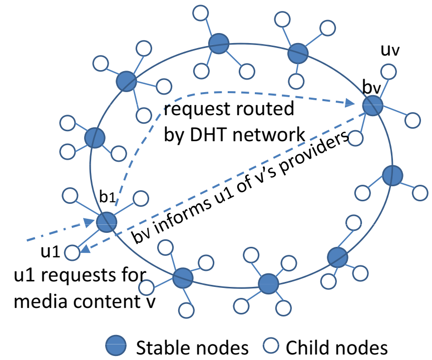

Fig. 11: Two level DHT network.

C. 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 and the bitrate of the videos on YouTube [26], 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. When a node is viewing/downloading a multimedia segment, it asks its parent to send the `Insert(key,index)` request to DHT in order to register itself as the content provider. The key is the consistent hash value of the name of the media segment, and the index includes the node's IP address, and content segment name, etc. Using the DHT routing protocol, the message will be forwarded to the broker of this segment. The recipient broker then adds a record in the list of providers for this content segment. For example, in Figure 11, 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 retrieve v's providers.

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 freshest 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 freshest 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 11, 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

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

Fig. 12: Example of cache content in some node.

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.

D. Refreshing Scheme

A provider registered to a broker will not be in service forever 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 server again to refresh its service. Otherwise, the broker assumes this provider is no longer valid. In this way, MBoard can ensure the availability of providers while reducing communication overhead.

O4 shows that a thread's popularity deteriorates within days, so MBoard does not require users to hold a large cache such as 1GB (the typical cache size of the PPLive [9] 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 12. 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.

V. Performance Evaluation

We conducted trace-driven experiments of MBoard on the well-known event-driven simulator PeerSim [27]. 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 I.

TABLE I: 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 dist.
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	Microsoft MSN video trace
User upload bandwidth	1/3 download bandwidth
Cache size	2
Refreshing interval	10 min
L/b/K (Pastry)	32/4/5

In real life, people sometimes do not finish watching the 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 13. The downloading bandwidth of a node is assigned according to the bandwidth distribution of these MSN video users. A user's upload bandwidth is set to 1/3 of its download bandwidth since most users have a DSL Internet connection [28]. 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 the experiment, we are especially 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 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 exhibits the cost of the refreshing scheme.

A. The Effectiveness and Efficiency of MBoard

Figure 14 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

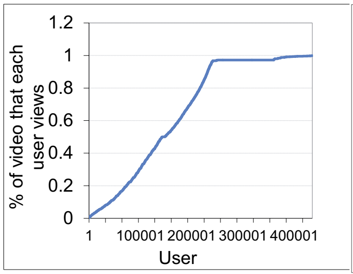

Fig. 13: User viewing behavior.

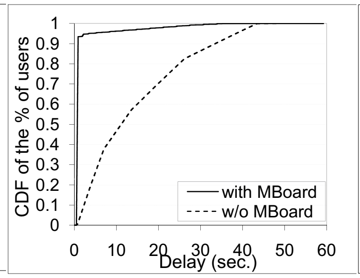

Fig. 14: Playback delay.

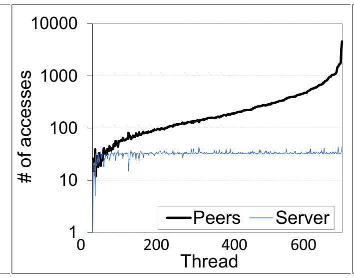

Fig. 15: P2P & server contribution.

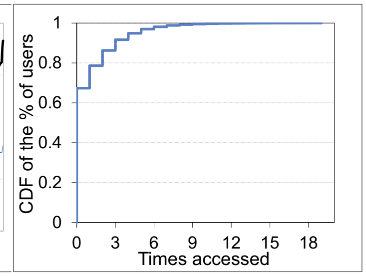

Fig. 16: Load on users.

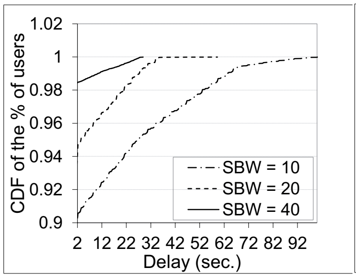

Fig. 17: Effect of SBW on delay.

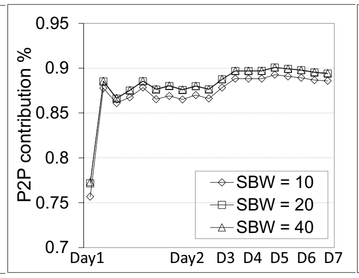

Fig. 18: Effect of SBW on P2P contribution.

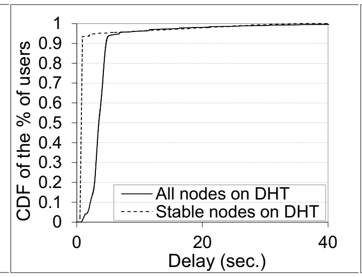

Fig. 19: Effect of stable nodes on delay.

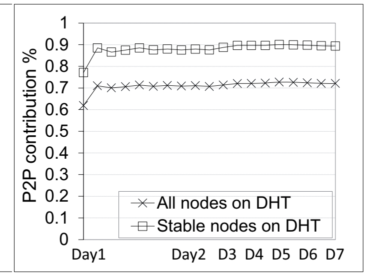

Fig. 20: Effect of stable nodes on P2P contribution.

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, MBoard achieves a much lower overall delay.

Recall that in MBoard, if a node cannot find a video segment from peers, it resorts to the server. Figure 15 shows the number of accesses in each thread contributed by peers and the server. A first look tells us that the server's contribution remains constant at around tens of accesses in each thread. On the contrary, peers contribute significantly more than servers. Peers provide up to 700 times more accesses than that of the server on certain threads. These results show that MBoard effectively helps to reduce the amount of stress on 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 16 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 17 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 is reduced as the server bandwidth increases. However, for the three bandwidth settings, approximately 90% of users retrieve their videos within 2000 milliseconds. Moreover, we see that the systems with 20Mbps and 40Mbps SBW reduce the video retrieval delay significantly. 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 times when there is no peer assistance available. Figure 18 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.

B. *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 19 shows the CDF of the percentage of users versus video playback delay. We see that the delay of using stable nodes is less for most users than putting all nodes in the DHT. This is because the size of the DHT when only using stable nodes is much smaller than putting all nodes on the DHT, which decreases the number of hops and reduces routing delay. The frequent churn also increases the number of routing hops. Figure 20 shows that the P2P contribution 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 results in higher data transmission failures.

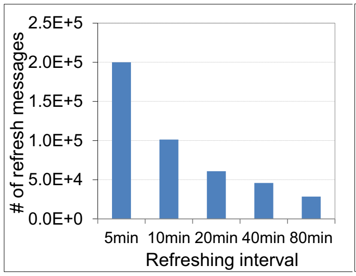

Fig. 21: Refresh message cost.

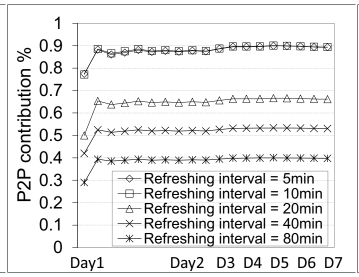

Fig. 22: Effect of refreshing period.

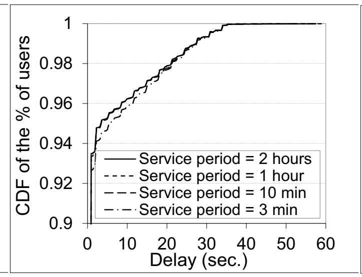

Fig. 23: Effect of service period.

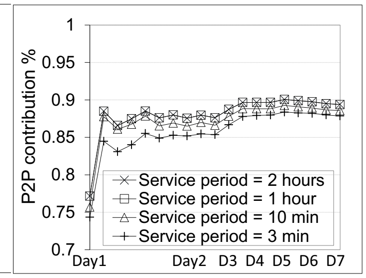

Fig. 24: Effect of service period.

C. 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 7 that if a user replies, its online time is 10 minutes and is increased if he replies again within 1 hour. Figure 21 shows the number of messages generated by service refreshing under different refreshing interval settings. We see that a small refreshing interval generates more messages than a large refreshing interval due to the more frequent communication between child nodes and brokers. In addition, the cost decreasing rate slows down as the refreshing interval doubles because only a small number of nodes have long continuous online time, and only they refreshes service in long refreshing interval settings.

Figure 22 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.

Figure 23 shows the CDF of the percentage of users versus the percentage of users with different service periods. A node's service period is the time period it is in service. We see that the 3 minute service period leads to a slightly higher delay. This is because when the service period is short, the server uploads more content and the queuing delay becomes larger. The slight delay increase is due to the nature of forums where users constantly come and leave. Thus, 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. 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 conclusion, the length of the service period does not have a great impact on delay, and a certain large service period decreases the video retrieval delay to a certain extent, with the optimal choice being 10 minutes.

Figure 24 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 Figure 23, we conclude that MBoard should require users to be in service for at least 10 minutes 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.

D. The Effect of Cache Size

Figure 25 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.

Figure 26 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 Figure 25, we see that a larger cache size only improves the performance of the system slightly.

E. The P2P Contribution of Head/Tail Content

In Figure 27, *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 III-C) 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

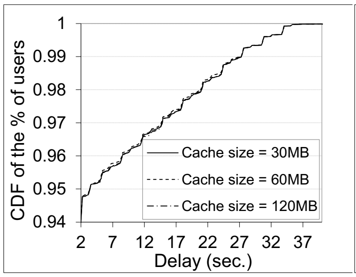

Fig. 25: Effects of cache size.

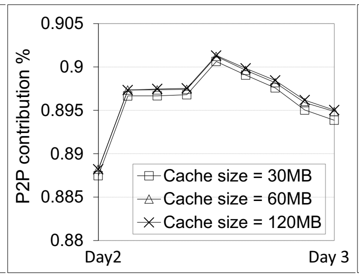

Fig. 26: Effects of cache size.

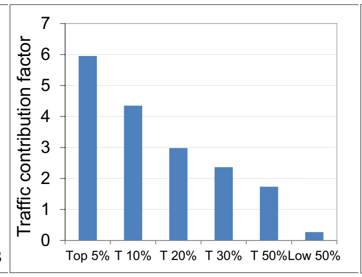

Fig. 27: Traffic contribution.

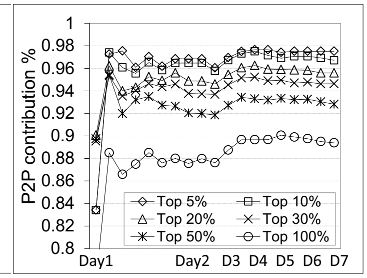

Fig. 28: Head content contribution.

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 28 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 no chance to be shared in our collected events.

VI. 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 period-based 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.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-OCI 1064230, CNS-1049947, CNS-1025652, CNS-1025649, and CNS-0917056, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282.

References

[1] "Internet usage statistics," http://www.internetworldstats.com/stats.htm.
[2] "Internet forums," http://en.wikipedia.org/wiki/List_of_internet_forums.
[3] "4chan," http://www.4chan.org/.
[4] "Gaia online," http://www.gaiaonline.com/.
[5] "Ultimate guitar," http://www.ultimate-guitar.com/.
[6] "Something awful," http://www.somethingawful.com/.
[7] "The DIS discussion forums," http://www.disboards.com/.
[8] "How much money is youtube burning?" http://www.independent.co.uk/life-style/.
[9] "PPLive," http://www.pplive.com.
[10] X. Cheng, C. Dale, and J. Liu, "Statistics and Social Network of YouTube Videos," in *Proc. of IEEE IWQoS*, 2008.
[11] "Google Launches YouTube Video Ads," http://www.readwriteweb.com.
[12] S. Annapureddy, "Providing Video-on-Demand Using Peer-to-Peer Networks," in *Proc. of IPTV Workshop, WWW*, 2006, pp. 238–247.
[13] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand Be Profitable?" in *Proc. of SIGCOMM*, 2007.
[14] 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.
[15] X. Cheng and J. Liu, "NetTube: Exploring Social Networks for Peer-to-Peer Short Video Sharing," in *Proc. of INFOCOM*.
[16] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems," *Proc. of EuroSys*, 2006.
[17] Y. Zhu, "Measurement and Analysis of an Online Content Voting Network: A Case Study of Digg," in *Proc. of WWW*, 2010.
[18] "Digg website," http://www.digg.com.
[19] "PPStream," http://www.ppstream.com.
[20] "UUSee," http://www.uusee.com.
[21] 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.
[22] "Big boards," http://www.big-boards.com/.
[23] F. 180-1, "Secure Hash Standard," *U.S. Department of Commerce/N.I.S.T., National Technical Information Service*, 1995.
[24] 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.
[25] P. B. Godfrey, S. Shenker, and I. Stoica, "Minimizing churn in distributed systems," in *Proc of SIGCOMM*, 2006.
[26] "Creating Videos for YouTube," http://www.homevideo101.com/creating-videos-for-youtube/.
[27] "The Peersim Simulator," http://peersim.sf.net.
[28] "The Difference Between Upload and Download Speed for Broadband DSL," http://www.broadbandinfo.com/cable/speed-test.