

HSM: A Hybrid Streaming Mechanism for Delay-Tolerant Multimedia Applications

1556168082

1556168082

1556168082

ABSTRACT

This paper proposes a novel *Hybrid Streaming Mechanism* (HSM) for enhancing performance of streaming services in *delay tolerant* multimedia applications. Delay-tolerant applications are a special class of on-demand multimedia applications where clients request the start of play back at a time specified by $(t + dt_i)$ where t is the current time and dt_i is the delay tolerance acceptable to client i . Examples of delay-tolerant applications include universities offering their courses to a set of global subscribers and service providers streaming movies requested by their clients.

In a content dissemination network, typically a central server at the source streams content in response to client requests. We term this as *Pure Streaming Mechanism* (PSM). Considering that in a dissemination network controlled by a Content Service Provider (CSP), the backbone links are highly provisioned, using a streaming server at the source leads to underutilization of these links. Also the links are occupied for the duration of play out of the multimedia content. In contrast, HSM allows streaming from strategically chosen intermediate nodes to which the content is dynamically transferred from the source, using FTP (File Transfer Protocol). As FTP uses the entire link bandwidth, it frees up the high bandwidth links faster for servicing requests from other clients sharing these links, increasing the efficiency of service. Central to HSM are the strategies used for selecting appropriate intermediate nodes as *streaming points* to enhance the following performance parameters: (i) number of serviced clients, and (ii) delivered stream rates at clients.

Simulation results demonstrate that by leveraging the delay tolerance of clients, higher stream rates are delivered at clients. By combining the FTP and streaming mechanisms intelligently, HSM performs better than PSM servicing on an average 40 percent more client requests. In HSM, transferred contents are temporarily cached at the streaming points. Such temporary caching further enhances the performance of HSM as requests for the same content are serviced from the cache.

Keywords Delay Tolerance, Pure Streaming, Hybrid Streaming, Caching, Streaming Point, Heterogeneous Networks, Multimedia Dissemination.

1. INTRODUCTION

Streaming media is expected to become one of the most popular types of web content in the future. Typically, a central server handles streaming services, is responsible for servicing all client requests. We term this *Pure Streaming Mechanism* (PSM), where streaming capability is available only at the source. In a large-scale network with a large number of concurrent client requests, streaming from only one source has been proven to be inefficient because of the limitation of streaming server capacity and link bandwidth constraints in the network [4] [10] [3]. To improve the performance of the streaming services as well as to serve more clients, many techniques have been developed, including content replication, and resource sharing [6] [9] [7] [5][18][1][3][14]. Content replication is an efficient way to increase the number of serviced clients, reduce network traffic and workload at central server. However such techniques require large storage, since typically the content is downloaded in advance in anticipation of client requests.

In this paper, we present a *Hybrid Streaming Mechanism* (HSM) to increase the efficiency of a Content Service Provider (CSP) by using a combination of FTP and streaming mechanisms in the context of a special class of on demand applications termed *delay-tolerant* multimedia applications [15]. In delay-tolerant applications, clients request for the multimedia content, specifying their requirements, stream quality- -a minimum rate at which they want to receive the stream, and delay tolerance - - the time they will wait for the play out of the stream. Applications in distance education and corporate training where multiple clients at different time slots access the same contents are typical examples that fall in this category of applications. Note that mechanisms proposed in the literature to efficiently serve requests for multimedia content assume that play out at clients start immediately [5][2][18][1][8].

In HSM, data flow is divided into two parts: (i) a FTP flow from source to strategically chosen intermediate node(s) and (ii) a streaming flow from that node to client(s) in the sub trees rooted at that node. We model the content dissemination network as a tree, with the source as the root and clients as leaves. Intermediate nodes in the tree serve as relay nodes. In HSM, streaming capability is available at strategic relay nodes and the mechanism invokes the capability at the appropriate relay node(s). Streaming content

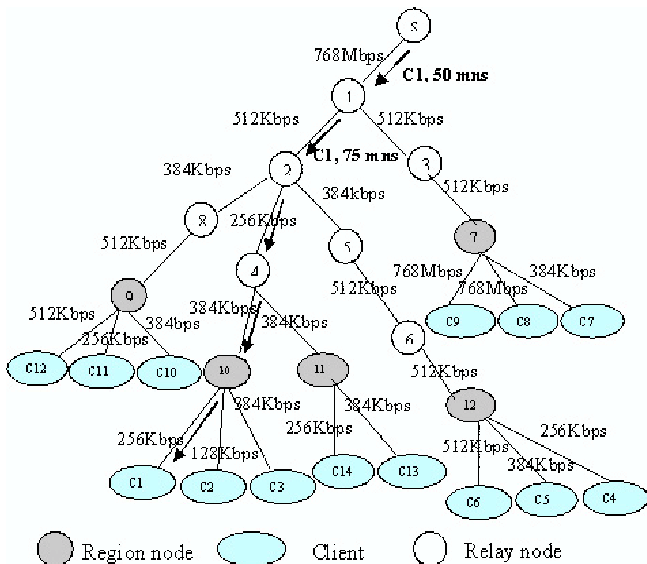


Figure 1: A simple tree network topology

is dynamically downloaded from source to the relay node, which is temporarily stored at that node.

We validate HSM through simulation-based comparative analysis between performance of PSM and HSM. Performance parameters used are: (i) the number of serviced clients, and (ii) the stream rate that each client receives. Our simulation results show that a CSP can service more clients using HSM. Using the delay tolerance requirements of clients, the stream quality at the clients can be enhanced.

The contribution of this paper are: We propose a novel mechanism, HSM, to increase the efficiency of streaming services in a content dissemination network. We use the FTP mechanism to leverage highly provisioned links in the backbone of a dissemination network and use streaming capabilities at strategic relay nodes to service clients with enhanced quality using their delay tolerance. HSM performs 40 percent better on the average when compared with PSM, as it uses bandwidth efficiently to service concurrent requests.

In the next section, we present a motivating example explaining the details of PSM and HSM and discuss related work in the area. We present the HSM algorithm in section 3 and discuss the main modules of the algorithm. We present our experimental analysis in the following section, comparing the performance of PSM and HSM. A simple cost-benefit analysis is outlined in Section 5 to justify the role of HSM in increasing revenues for a CSP. Section 6 presents the conclusions of our work.

2. MOTIVATION AND RELATED WORK

We present an example to illustrate the differences in PSM and HSM and motivate the need for the proposed hybrid mechanism in section 2.1. Related work in the area is discussed in section 2.2.

2.1 An illustrative example

A simple network model in Figure (1) represents a heterogeneous dissemination network as a tree, with the source S at the root and the clients C_1, C_2, \dots, C_{14} at the leaves. All

Table 1: Details of client requests

Clients	Request time (Minutes)	Requirements (delay-tolerance, rate)	PSM	HSM
C1	0	(30,256)	Served	Served
C14	10	(60,256)	Not served	Served
C6	75	(30,256)	Not served	Served
C9	75	(15,480)	Not served	Served
C12	75	(30,256)	Not served	Served

Table 2: Details of client C1

C1's path (Links)	Link band -width (Kbps)	C1's stream rate (Kbps)	Available bandwidth (Kbps)		Link busy period (Minutes)	
			PSM	HSM	PSM	HSM
S-1	768	320	448	0	120	50
1-2	512	320	192	0	120	75
2-4	256	320	0	0	150	150
4-10	384	320	64	64	120	120
10-C1	256	320	0	0	150	150

other intermediate nodes serve as *relay* nodes. A node that directly serves a group of clients is termed a *region node*. We use the term *region* to refer to the sub tree that contains the region node and the clients it serves. For example in Figure (1), the network has 5 regions with nodes 7, 9, 10, 11, and 12 serving as region nodes. We refer to the network from S to the region nodes as the *backbone* of the content dissemination network.

Let us assume that at time zero, client C1 joins the network. Client C2 joins 10 minutes later and clients C6, C9, and C12 join 75 minutes later. Table 1 gives the details of clients requesting the stream.

We illustrate the difference between PSM and HSM by considering the above arrival pattern, as discussed below:

Case 1: PSM

In a streaming application where clients do not tolerate any startup delay, the weakest link in a client's path dictates the encoding rate of the stream to provide loss-free transmission to that client.

In PSM, the source node contains the only streaming server in the network that processes all the requests from clients. In this mechanism, there may be underutilization of backbone bandwidth in case the delivered stream rate at the client is less than bandwidths of links in the backbone. This is because according to the streaming property, the streaming server sends only the amount of data equivalent to streaming encoded rate to the client irrespective of the available link bandwidth in the path. For example if the streaming object is encoded at 256Kbps, only 256 kb is sent by server to client every second even when the link bandwidth is greater than 256 Kbps. In delay-tolerant applications, the delivered stream rate at a client can be enhanced using buffers in the nodes in the path of the client [15].

Let us consider client C1 which requests the stream at $t=0$. Let the play out duration of the stream be 2 hrs. We

first calculate the delivered stream rate at C1, considering its delay tolerance. C1 gets 320 Kbps. (The formula used is derived in Section 3.2). When the streaming server is placed at the source, stream flows from S to C1 along the path (S-1-2-4-10-C1). The server sends the stream encoded at 320 Kbps, which occupies the path for 2 hrs, the play out duration. Table 2 shows the available link bandwidths in the path of C1 when it is being serviced using PSM.

C14 joins the network at time $t=10$. Since C14 shares links (S-1-2-4) with C1, its request cannot be serviced. Client C6 joins network at $t=75$. It shares links (S-1-2) with C1. Given its requirements, C6 can get only a stream rate of 240 Kbps. Since this rate is below C6's minimum required rate, request from C6 is also rejected. Similarly, clients C9 and C12 also get rejected. Thus, using PSM, only one out of five clients is serviced by the CSP.

Case 2: HSM

HSM allows selected relay nodes with streaming capability to stream the content instead of central server (source). When a request arrives at the central server, it determines the stream rate that can be provided to the client given the client's delay tolerance requirement and the location of the streaming server, termed *streaming point*. The central server then starts sending data by using FTP to the chosen streaming point and allows it to serve the clients. As FTP uses the entire bandwidth for transferring data, the links between the central server and the streaming point are fully utilized. As a result, these links are freed earlier compared with PSM. In HSM, the data sent by source to the streaming point is cached at that node for a period equivalent to the streaming duration, in the interest of future requests for the same content. We term this period *Time To Live of the Content* (TTLC). TTLC at a relay node is extended when a new request is made for the same content before the original TTLC expires. For a detailed discussion refer to section 3.4.2.

Consider the same scenario presented in Table 1 with HSM. As before, the delivered stream rate at C1 is 320 Kbps. But now we choose node 4 as the streaming point (Details of streaming point selection are presented in Section 3.3). FTP mechanism is used to transfer data from the source to the streaming point along the path (S-1-2-4). Table 2 shows the available link bandwidths in the path of C1 when it is being serviced using HSM.

C14 joins the network 10 minutes after C1. Since C14 shares links (S-1-2-4) with C1, it is not possible for C14 to initiate a new stream from S. However, since C14 is requesting for the same streaming object, as the object is being cached at node 4, its request can be serviced from node 4. C14 gets the stream at 320 Kbps which is greater than its minimum rate requirement. Note that C14 doesn't join C1 ongoing stream, the new stream rooted at node 4 is sent to C14. Clients C6, C9, and C12 join the network at time $t=75$. At $t=75$, C1's transmission across links S-1 and 1-2 are finished and the links become free. All three clients C6, C9, and C12 get serviced with a stream rate of 480Kbps, their streaming points being at nodes 5, 1, and 8 respectively. As a result, under HSM all 5 clients are serviced.

The above example demonstrates that HSM performs better than PSM in terms of number of serviced clients. This is

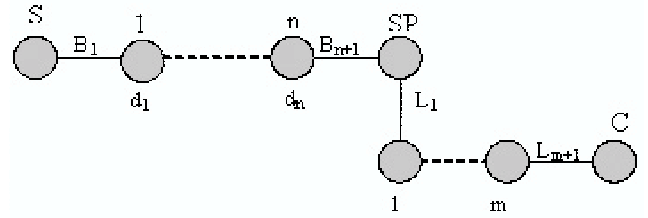


Figure 2: Example used for deriving the expression

because in HSM, links from the source to the streaming point are freed sooner than PSM. Another important feature of HSM is that the content is temporarily cached at the streaming point and requests from other clients in the sub tree can be serviced from the cache. This property allows HSM to improve not only the number of serviced clients but also reduce the traffic in the network.

2.2 Related work

Most of the research in the area of multimedia dissemination treats delivery of multimedia as real time application [2][18], which can tolerate a small delay for the purpose of solving the delay jitter problem. Mechanisms proposed in the literature focus on minimizing this start up delay [2]. When we consider multimedia delivery over the Internet, there were reasons for using streaming with minimal start up delay: (i) caching or buffering the content was high due to the size of multimedia files: many mechanisms, [17][13][9][7] have been proposed for efficient content management (ii) price of Internet connection was high: many mechanisms have been proposed for effective use of bandwidth, including [4][11][6][3][16][14].

In the recent literature, an optimal chaining scheme proposed in [19] for Video-on-Demand applications uses the concept of collaborative networks. In this mechanism, clients store fragments of streaming content shared between them. This mechanism is more applicable for P2P and not for CSP.

Multi-path routing for video delivery over bandwidth-limited networks [20] is another mechanism in which quality of streaming service and the number of serviced clients are improved as data is sent faster than one way routing. In this mechanism, links are freed sooner allowing other clients to get the services. But this scheme has a drawback of high computational overhead when the number of client requests increases, as the streaming server performs the scheduling function also. In the recent times with the drop in the prices of memory and connectivity, and abundance of network bandwidth, clients demand convenience while accessing content.

Today's content dissemination networks exhibit heterogeneous characteristics, as networks have combinations of satellite, terrestrial, and Internet links from the source to the clients. Our work focuses on such heterogeneous networks and explores ways of combining different mechanisms for effective and efficient content dissemination, when clients specify their delay tolerance.

We present HSM, a simple hybrid streaming mechanism in the context of delay tolerant applications, considering the current trend in multimedia dissemination focusing on users convenience, to maximize bandwidth utilization.

3. WORKING OF HSM

The algorithm used in HSM is presented in section 3.1. We present the major components of HSM : (i) Expressions used in HSM(3.2), (ii) Streaming point selection strategies(3.3), and (iii) Temporary cache management policies(3.4).

Assumption are as follows:

- Links have dedicated bandwidth provisioned for the given application.
- The selected intermediate nodes have the streaming capability.
- Multicasting is supported in the given network.

3.1 HSM algorithm

The top-level algorithm used in HSM is presented in Table 3.

Table 3: HSM,top-level algorithm

<p>Algorithm: When a client's request arrives /*client specifies minimum rate required and its delay tolerance*/ Determine the stream rate delivered to client considering its delay tolerance /* Equation (2) from section 3.2 is used*/ If stream rate <client's minimum rate requirement Reject request Else If the link is free /*Perform Streaming point selection*/ if stream rate <= the weakest in the path from source to region node . Use Stategy 1 for streaming point selection /*refer to Section 3.3.1*/ Else . Use Stategy 2 for streaming point selection /*refer to Section 3.3.2*/ End . Transfer the content by using FTP from source to selected streaming point(SP) and stream from SP . Find the time to transfer the contents to SP /*Equation (1) from section 3.2 is used*/ Else If the same content is already cached at streaming point . Accept request and stream from cache . Update TTL /*refer to Section 3.4.2*/ Else Reject request End End End End</p>

3.2 Expressions used in HSM

In this section we first derive the expression for time to transfer the file from source to streaming point and then the expression for delivered stream rates at clients given their delay tolerance values. The notations using for deriving the expressions are given in table 4.

Table 4: Notations

SP	Streaming Point
d_i	Queuing delay at node i
D_q	Total queuing delay in the path
SD	Streaming Duration
T_{ftp}	Time to transfer a file using FTP
T_w	Time to transfer a file across the weakest link
B_i	Link bandwidth at link i
SR_i	Stream rate of client i
L_{min}	Minimum link bandwidth in the path
C_i	Client number i
CS_g	Cache size at streaming point g

3.2.1 Time to transfer file using FTP

With reference to Figure(2), let there be n relay nodes 1, 2, ..., n from source S to streaming poing SP. Let B_1, B_2, \dots, B_{n+1} be the link bandwidths from source S to node 1, node 1 to node 2, ..., node n to SP, respectively. Time to transfer the file across the weakest link (T_w) from S to SP is given by:

$$T_w = \frac{file\ size}{Min(B_1, B_2, \dots, B_{n+1})}$$

Let d_1, d_2, \dots, d_n be the queuing delays at nodes 1, 2, ..., n respectively. And D_q is the total queuing delay in the path. Assuming that propagation delay is negligible and there is no other competing traffic, the total queuing delay is given by:

$$D_q = \sum_{i=1}^n d_i$$

Total time to transfer file from S to SP is:

$$T_{ftp} = T_w + D_q \quad (1)$$

3.2.2 Equation for delivered stream rate at a client

In delay-tolerant applications, clients specify two parameters: minimum rate at which they want to receive the stream and their delay tolerance, time they can wait to receive the stream. The delivered stream rate(SR) at client i is given by the expression:

$$SR_i = \frac{CD_i * L_{min}}{SD} + L_{min} \quad (2)$$

We derive the expression with reference to Figure (2): let there be m relay nodes 1, 2, ..., m from SP to Client C_i . let L_{min} be the minimum of link bandwidths L_1, L_2, \dots, L_{m+1} , in the path between SP and client C_i . When the stream is encoded at L_{min} , client receives it without any loss. Let CD_i be the client delay tolerance of C_i . C_i waits for time CD_i before the play out starts. However, during this waiting time, an amount of data can be streamed to C_i given by: $L_{min} * CD_i$. The amount of extra data that C_i gets per second is $\frac{L_{min} * CD_i}{SD}$. Thus, the delivered stream rate at C_i is $\frac{CD_i * L_{min}}{SD} + L_{min}$.

3.3 Streaming Point Selection

In HSM, a selected relay node serves as the steaming point for all the clients in its sub tree instead of the central server. Thus the streaming point selection strategy is an important part of HSM. In sections 3.3.1 and 3.3.2 we present two selection strategies based on the following criteria: (i) streaming point should help to improve the number of serviced clients and /or (ii) the position of the streaming point should help to improve the stream rate for other requests which come

from the region serviced by that streaming point. Examples to illustrate these strategies are presented in 3.3.3.

3.3.1 Strategy 1: At a relay node which has the most connected links (Improving the number of serviced clients)

This strategy is used in a network topology where *all* the links in the path from source to the region node are provisioned with high bandwidth. Suppose the clients in this network have very low bandwidth connections to the region node (the *last mile* problem). In such a case, if the delivered stream rate at a client is less than or equal to the weakest link in the backbone, we select the relay node with the maximum number of out going links as the streaming point. A step-by-step approach to find out the streaming point in strategy 1 is presented below :

```

When client arrives
  Find the client's stream rate
  Find the weakest link in the path from source to region node
  of the client.
  If the client's stream rate is less than or equal to weakest link.
    Choose the node with the maximum out-going links
    as a streaming point.
  End
End

```

The reasoning for this strategy is as follows:

- When the stream rate is less than or equal to bandwidth of the weakest link in the path from the source to region node, the stream will flow without introducing any delay up to the region node. Hence, any node in the client's path can be chosen as the streaming point.
- However, when the relay node with most out going links is chosen, more clients can be serviced concurrently.

3.3.2 Strategy 2: At a node below the weakest link in the path (Enhancing the stream rates)

In this strategy, any node below the weakest link in the path from source to the region node serving the client is chosen as the streaming point. A step-by-step approach to find out the streaming point in strategy 2 is presented below.

```

When client arrives
  Find the client's stream rate
  Find the weakest link in the path from source to region node
  of the client .
  If the client's stream rate is greater the weakest link.
    Choose the node below the weakest as streaming point.
  End
End

```

The reasoning for this strategy is as follows:

- The weakest link in a client's path uses up most of the client's delay tolerance.
- When the client's delivered stream rate is greater than the weakest link rate up to the region node, the streaming point is chosen below this weak link. This enables service of other requests from clients in the sub tree made within TTL to get the stream at that rate, as the stream's flow is not subjected to the weakest link. This strategy may improve the stream rates for the clients.

Table 5: Details of client requesting the stream

Clients	Request (Minutes)	Client requirements (delay-tolerance,rate)	Service strategy
C2	0	(90,128)	Strategy 1
C4	15	(30,128)	
C11	15	(30,128)	
C14	15	(60,128)	
C1	0	(90,256)	Strategy 2
C3	100	(30,256)	
C13	110	(30,256)	

3.3.3 Example

Consider the simple network model in Figure (1). We present two scenarios to illustrate the streaming point selection strategies. Table 4 gives the details of clients requesting the stream.

Illustration of strategy 1: Considering the clients in Table 5(strategy1). Client C2 specifies a delay tolerance value of 90 minutes. Stream rate that can be delivered to this client is 224 Kbps. This rate is less than the weakest link (256 Kbps) in the path from source to the region node serving this client. As per strategy 1, we choose the streaming point at node 2. To validate this idea, we study two different cases: first we consider node 4 as the selected streaming point for client C2. Requests of clients C4, C14, C11 arrive at 15 minutes after C2 when the link (S-1-2) is being occupied by C2. Hence requests from C4 and C11 get rejected. Only C14's request can be serviced from the cached content in node 4. Now we consider the same scenario with node 2 as the streaming point. In this case client C4 and C11 can be serviced concurrently.

Illustration of strategy 2: Considering the clients in Table 5 (strategy2). Client C1 allows delay tolerance of 90 minutes. The delivered stream rate at C1 is 448 Kbps. This rate is greater than the weakest link in the path from source to the region node serving this client. According to Strategy 2, we choose the streaming point at node 4. All the clients C1, C3, and C13 get 448 Kbps. This is because the content requested by C1 is stored at node 4 and since C3 and C13 are requesting the same content before the TTL expires, both these clients can be served by node 4. The stream flowing from node 4 is not subject to the weakest link in the path. Note that if we stream from the source or any node above node 4, the delivered rates at C1, C3, and C13 are 448Kbps, 320 Kbps and 320 Kbps respectively, as the weakest link rate is 256 Kbps.

3.4 Cache Management at Relay Nodes

In HSM, when the content is transferred from the source to the streaming point, it is temporarily cached. We need to determine the memory requirement for the cache and a mechanism to manage the cache. In Section 3.4.1, we present the cache requirement at a selected streaming point in the network based on the link bandwidth and the number of links connected to that point. We present a simple cache management mechanism with very little overhead in Section 3.4.2.

3.4.1 Memory Requirement at Relay Nodes

We derive the formula for cache size at a streaming point by finding the maximum amount of data that can flow through each sub tree originating at the streaming point. A step-by-step approach is presented below:

- Find the number of sub trees rooted at the streaming point. Let this be N.
- Find the number of regions in each sub tree. Let this number be R.

- For each sub tree
 - Find the weakest link for region j , $Bmin_j$.
 - Find the max of weakest link across all regions R , $W_i = \text{Max}(Bmin_j)$, $j=1, \dots, R$ and $i=1, \dots, N$
 - The maximum amount of data that can flow in the sub tree i is $W_i * SD$, where SD is the stream duration.
- Let $FSMax$ be the maximum file size across all content files stored at S . Since clients in the regions can specify delay tolerance, we must find the largest file size that need to be cached. Thus, the cache size for a sub tree is given by: $\text{Max}(W_i * SD, FSMax)$
- The cache size at streaming point in node g , considering all the sub trees is given by:

$$CS_g = \sum_{i=1}^N \text{Max}(W_i * SD, FSMax) \quad (3)$$

Cache Memory requirement at SP is bounded by CS_g

Example: With reference to figure(1)

Let $SD=2$ hours (7200 seconds)

$FSMax=1$ GB.

Let node 2 be the chosen streaming point.

We calculate the cache size at node 2 as follows:

Number of sub trees rooted at node 2: $N=3$

1. For sub tree 1, Number of regions $=R=1$; $Bmin1 = \text{Min}(384, 512)=384$; $\text{Max}(Bmin1) = 384$.

2. For sub tree 2, Number of regions $=R=2$; $Bmin1 = \text{Min}(256, 384) = 256$;

$Bmin2 = \text{Min}(256, 256)=256$;

$\text{Max}(Bmin1, Bmin2) = (256, 256) = 256$.

3. For sub tree 3, Number of regions $=R=1$; $Bmin1 = \text{Min}(384, 512, 512)=384$;

$\text{Max}(Bmin1) = 384$.

Cache size at node 2, $(CS_2) = \text{Max}(384 * 7200, 1GB) + \text{Max}(256 * 7200, 1GB) + \text{Max}(384 * 7200, 1GB) = 7.36$ Gb

3.4.2 Time To Live of Content (TTLC) in the cache

In HSM, content is temporarily cached at the chosen streaming point. We use a simple method to determine the value of Time To Live of Content such that the cache management has no additional overheads.

Let us consider a client i with delay tolerance CD_i requesting for a stream with duration SD . The client's transmission starts at $time = t_0 + CD_i$. The client finishes its transmission at $(t_0 + CD_i + SD)$. Hence the stream needs to be active for the duration $CD_i + SD$. We choose this value as the TTLC for the stream in the cache at the streaming point. When multiple clients access the same stream at the same time, we choose the maximum of the delay tolerance values of the clients in the above expression.

$$TTLC = CD_i + SD \quad (4)$$

Note that the TTLC can not be less than $CD_i + SD$ because if it is, the content will be removed before client finishes the stream.

When there is a new request (k) for the same stream before the TTLC expires, it is extended to $T_c + CD_k - (T_c - t_k) + SD$, where T_c is the TTLC of the current content, t_k is the time when client k 's request arrives and CD_k is the delay tolerance of client k .

Example: Let client C1 with 30-minute delay tolerance is requesting a stream with 2 hours duration. TTLC for this stream is $T=30+120=150$ minutes. Let a new request for the same stream come from client C2 at $t=90$. C2's delay tolerance is 90 minutes. The extended value of TTLC for the stream is: $150+90-(150-90)+120=300$. Thus, the content is alive till $t=300$ minutes.

4. PERFORMANCE EVALUATION

In this section we present the results of simulations evaluating the performance of PSM and HSM, using *Matlab*. Our objective is to

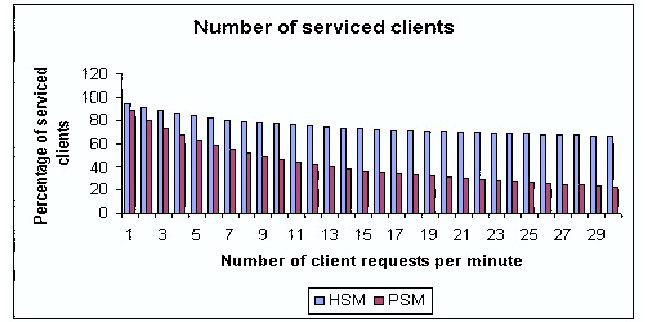


Figure 3: Number of serviced clients with identical client requirements (Class 1)

compare the performance of the two mechanisms under various network topologies and client requirements. The following performance metrics are used: (i) the number of serviced clients and (ii) percentage improvement of client stream rates as compared with their minimum rate requirement.

We define the parameters used in the simulations in Section 6.1 and present details of our experiments in Section 6.2. In Section 6.3, we discuss the results of our simulations. We present a case study on the Gnutella Peer like Network [12] in Section 6.4.

4.1 Simulation parameters

The following parameters remain same across all our experiments: (i) Multimedia play out duration is set to 2 hours (ii) Without loss of generality, queuing delay and propagation delay are set to zero (iii) Period over which client arrivals are monitored, termed *observation period* is set to 4 hours (iv) Arrival rate of the client requests is varied from 1 to 30 per minute.

Performance of HSM depends on the network topology, links characteristics, and client requirements. In order to study the impact of these parameters on the performance of HSM, we have taken 100 different topologies that fall into two classes:

- Class 1: The first 50 topologies are in Class 1; these topologies have high bandwidth links from source to region node. The bandwidths are chosen randomly from the range (256 Kbps – 768 Kbps).
- Class 2: Next 50 topologies are categorized into Class 2 topologies that have low bandwidth (weak links) in the upper part of the network from source to region node. The bandwidths are chosen randomly from the range (128 Kbps – 256 Kbps).

Each topology have total number of nodes in the range 100 to 500, where the number of nodes is selected randomly.

4.2 Details of experiments

Experiments 1 (Class1): Set the client's minimum rate to 128 Kbps and delay tolerance values to 30 minutes for all clients. We find the number of serviced clients and the percentage improvement of clients' stream rates under PSM and HSM.

Figure (3) shows that when the request rate increases, the number of serviced clients decreases for both the mechanisms. This is as expected. However, the decrease is more pronounced in PSM compared to HSM. While the number of serviced clients gradually decreases in HSM, it drops more rapidly in PSM. Also, the difference between the number of serviced clients in PSM and HSM keeps widening as the number of requests increases. The results given in Figure (4) show that whenever the number of client requests increases, the percentage improvement decreases for both the mechanisms. From this graph, we see that PSM services clients with better stream rates compared with HSM. However, PSM rejects 78 percent of client requests compared with 30

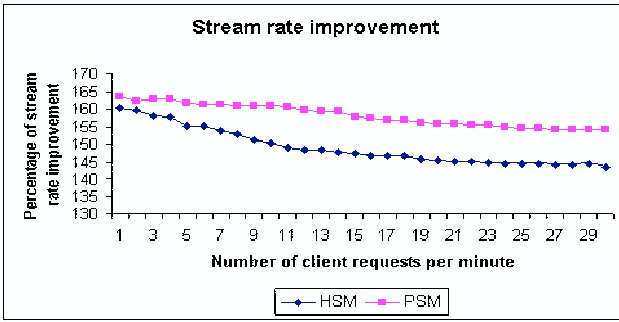


Figure 4: Percentage of stream rate improvement with identical client requirements(Class 1)

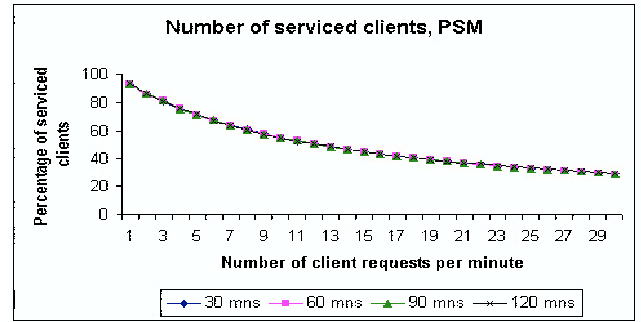


Figure 7: Impact of client delay tolerance values on PSM(Class 1)

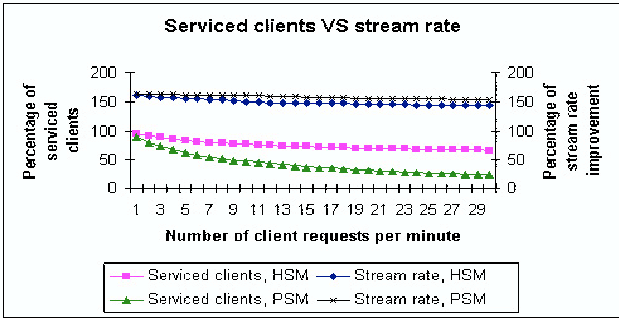


Figure 5: Percentage of serviced clients VS. percentage of stream rate improvement(Class 1)

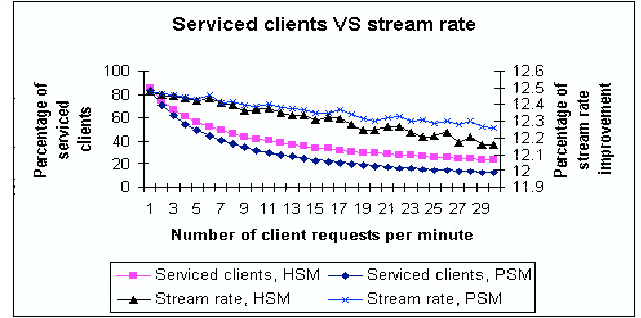


Figure 8: Percentage of serviced clients VS. percentage of stream rate improvement(Class 2)

percent of requests rejected by HSM. In order to observe both the parameters - - number of serviced clients and stream rate improvement at the clients - - we present Figure (5). In this figure, X-axis represents the number of client requests per minute, Y-axis (on the left) represents the percentage of clients serviced, and Y-axis (on the right) represents percentage of stream rate improvement.

Experiments 2(Class 1): In this set of experiments, we use Class 1 topologies to evaluate the impact of clients' delay tolerance on the number of serviced clients using PSM and HSM. We set the clients' minimum rate to 128 Kbps. We run four experiments: for each experiment the delay tolerance values of all clients are equal. The values of delay tolerance chosen for experiments are 30, 60, 90, and 120 minutes respectively.

Results in Figure (6) demonstrate that as clients' delay tolerance increases, the performance of HSM gets better. When the

client delay tolerance is equal to the streaming duration, HSM services nearly 100 percent of the clients' requests. In the case of PSM, as shown in Figure (7), clients' delay tolerance has very little effect on the number of client requests serviced.

Experiments 3(class 2): In this set of experiments, we use Class 2 topologies to evaluate the performance of PSM and HSM. We set the clients' minimum rate to 128 Kbps and delay tolerance values to 15 minutes for all clients. We find the number of serviced clients and the percentage improvement of clients' stream rates under PSM and HSM. Figure (8) displays the number of serviced clients and the percentage of stream rate improvement under HSM and PSM.

Results in Figure (8) demonstrate that, in class 2 network topology, HSM still performs better than PSM in term of number of serviced clients, but only marginally. It also happened to client stream rates, where PSM performs slightly better than HSM. In general, HSM performs well for class 1 topologies.

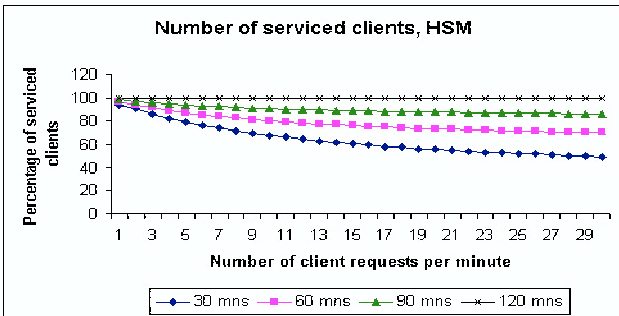


Figure 6: Impact of client delay tolerance values on HSM(Class 1)

4.3 A Case study: Gnutella Peer Network

In this section, we present a case study on the Gnutella Peer Network[12]. We simplify the original network topology to a tree-based network by removing cycles in the topology. Our approximated Gnutella Peer Network backbone contains 510 nodes. In this simulation, we set the clients' minimum rate to 128 Kbps and delay tolerance values to 30 minutes for all clients. We observe the number of serviced clients and the percentage improvement of clients stream rates under PSM and HSM. Figure (9) displays the number of serviced clients and the percentage of stream rate improvement under HSM and PSM. With the given results, we observe that HSM perform better than PSM in term of number of serviced clients, but less in term of percentage of stream rate improvement. Note that when the number of client requests reaches 30 per minute, HSM performs 20 percent better than PSM. While PSM rejects more client requests, it provides stream rates that are on average 5 percent better than HSM.

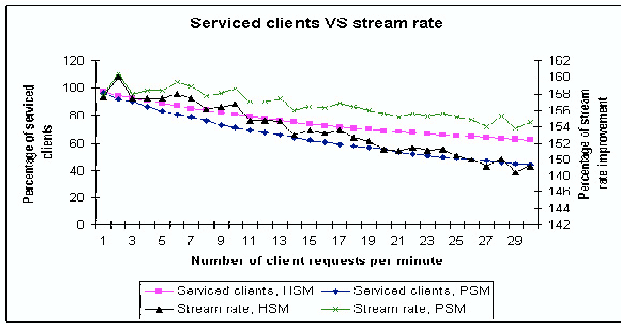


Figure 9: Percentage of serviced clients VS. percentage of stream rate improvement (Gnutella)

4.4 Analysis of results

Performance of HSM depends on the following factors: (i) network topology with specific link bandwidths, and (ii) clients' requirements. We observe that for Class 1 topologies where the link bandwidths are provisioned such that the upper links from the source to the regional nodes have high bandwidth, HSM is a better scheme as the available bandwidth can be better utilized with this mechanism.

In class 2 topologies links from the source to the region nodes have low bandwidths. In this case, using FTP to transfer the file to a relay node does not provide any advantage, as the time for transferring the file is same even when streaming server is placed at the source. The only advantage of HSM is that by choosing a streaming point appropriately, requests from clients for the same content can be serviced from the cached contents. Thus, we observe only marginal improvement in the number of clients serviced with such topologies.

To summarize, HSM works well with Class 1 topologies because of the impact of FTP property used in HSM. If the dissemination network falls in Class 2 category, PSM is preferred as the costs involved in enabling relay nodes with streaming capability may outweigh the benefits.

5. COST-BENEFIT ANALYSIS

In this section we present a simple analysis the trade-off between cost of putting streaming capability at relay nodes and the benefit from improved client services. Our goal is to find the break-even point, time taken for the CSP to cover the cost of putting streaming capability at the relay nodes by the revenue from improved client services.

Cost of providing streaming capability at the relay nodes is given by: $N * C$, where N is the number of relay nodes with the streaming capability and C is the cost per streaming server.

Let N_h be the number of serviced clients per unit time when HSM is used. Let N_P be the number of serviced clients per unit time when PSM is used. Note that when PSM is used only one streaming server is placed at the source. By using HSM, the increase in number of clients serviced is given by $(N_h - N_P)$. The additional revenue the CSP makes by servicing these clients is given by $P * (N_h - N_P)$, where P is the price a client pays for the service. The break-even point in unit time is given by:

$$Y = \frac{N * C}{P * (N_h - N_P)} \quad (5)$$

Example: Consider a network with 20 selected relay nodes with streaming capability. If we use PSM, we serve 200 out of 400 clients per day using only one central server. If we use HSM, 300 out of 400 clients are served per day. Suppose one streaming server costs 2000 dollars and a client pays 4 dollars for the service.

$$Y = \frac{20 * 2000}{4 * (400 - 200)} = 100 \text{ days.}$$

6. CONCLUSION

Leveraging clients' delay tolerance can be used to deliver better stream rates to clients, even when links are constrained in their path from the source. Typically in a content dissemination network controlled by a CSP, weak links are at the edge of the network closer to the clients. By using a combination of FTP and streaming mechanisms, provisioned links, the CSP's backbone can be fully utilized, serving more client requests when compared to a centralized server handling all the streaming requests. HSM, the proposed hybrid streaming mechanism uses this idea to improve the performance and hence the revenue for a CSP. We have shown that by intelligently choosing an appropriate relay node as streaming point, on the average 40 percent more requests can be serviced using HSM as compared with PSM. In HSM, The transferred content is cached temporarily at the streaming point, used to service future requests for the same content. This feature further enhances the performance of HSM when class 1 topologies with highly provisioned backbone are used in the simulations. With class 2 topologies having weak links in the backbone, we observe only marginal improvement in the performance of HSM resulting from additional requests serviced from the caches at streaming points. Our on-going research includes efficient utilization of resources such as buffers and transcoders to maximize revenues for CSP in delay tolerant multimedia applications.

7. REFERENCES

- [1] S.-J. L. Bo Shen and S. Basu. Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks. *IEEE Transaction on Multimedia*, 6(2):375–386, June 2000.
- [2] P. A. Corma J. Sreenan, Jyh-Cheng Chen and Narendran. Delay reduction techniques for playout buffering. *IEEE Transaction on Multimedia*, 2(2):88–97, June 2000.
- [3] G. C. Danjue Li, Chen Nee Chuah and S. B. Yoo. Muvis: Multi-source video streaming service over wlangs. *KIC*, 2003.
- [4] W. Z. Y.-Q. Z. a. J. M. P. Depeng Wu, Yiwei Tomas Hou. Streaming approach over internet: Approaches and directions. *IEEE Transaction on circuit and system for video technology*, 11(3):282–300, March 2001.
- [5] P. Frossard and O. Verscheure. Batched patch caching for streaming media. *IEEE COMMUNICATION LETTER*, 6(4):159–161, April 2002.
- [6] W. W. Guang Ming Su. Efficient bandwidth resource allocation for low-delay multiuser video streaming. *IEEE Transaction for Circuits and Systems for Video Technology*, 15(9):1124–1137, September 2005.
- [7] B. L. J. Xu and D.L.Li. Placement problem for transparent data replication proxy service. *IEEE Journal on Selected Areas in Communications*, 51(6):1383–1398, 2002.
- [8] X. C. Jiangchuan Liu and J. Xu. Proxy cache management for fine-grained scalable video streaming. *IEEE IFOCOM*, 2004.
- [9] F. Y. C. Keqiu Li, Hong Shen. A multimedia object placement solution for hybrid transparent data replication. *Japan Advanced Institute of Science and Technolog and University of Hong Kong*, 2005.
- [10] S. S. Kien. Hua, Ying Cai. Multicast technique for true video-on-demand services. *ACM Multimedia*, pages 191–200, 1998.
- [11] D. K. Mohamed M. Hefeeda, Bharat Bhargava. A hybrid architecture for cost-effective on-demand media streaming. *Department of Computer Science, Purdue University, West Lafayette*, October 2003.
- [12] G. of Cyberspace Directory. <http://www.cybergeography.org/>.
- [13] D. R. P.Krishnan and Shavitt. Caching location problem. *IEEE/ACM Trans. Networking*, 8(5):795–825, October 2000.

- [14] W. Z. Qian Zhang and Y.-Q. Zhang. Resource allocation for multimedia streaming over the internet. *IEEE Transaction on Multimedia*, 3(3):339–355, September 2001.
- [15] R. removed for the purpose of anonymous review.
- [16] R. removed for the purpose of anonymous review.
- [17] Y.-S. M. Sang-Ho Lee, Kyu-Young Whang and I.-Y. Song. Dynamic buffer allocation in video-on-demand systems. *IEEE Transactions on knowledge and data engineering*, 15(6):1535–1551, 2003.
- [18] J. R. Subhabrata Sen and D. Towsley. Proxy prefix caching for multimedia streams. *IEEE Transaction on Multimedia*, pages 1310–1318, 1999.
- [19] C.-L. C. Te-Shou Su, Shih-Yu Huang and J.-S. Wang. Optimal chaining scheme for video-on-demand applications on collaborative networks. *IEEE Transactions on multimedia*, 7(5):972–980, October 2005.
- [20] S.-H. G. C. Victor O.K, Li Jiancong Chen. Multipath routing for video delivery over bandwidth-limited network. *IEEE Trans*, 22(10):1920–1932, 2004.