PYRAMID BROADCASTING
FOR
VIDEO ON DEMAND SERVICE

S. Viswanathan *T. Imielinski*

Dept of Computer Science

Rutgers University

New Brunswick, NJ 08903

e-mail:{ vish , imielins }@cs.rutgers.edu

Abstract

The proposed scheme of pyramid broadcasting is a new way of rendering Video On Demand service at metropolitan scale. In pyramid broadcasting, the most frequently requested movies are multiplexed on the broadcast network, resulting in radical improvement of access time and efficient bandwidth utilization. We provide analytical and experimental evaluations of pyramid broadcasting based on its implementation on ethernet LAN, illustrating its advantages.

1 Introduction

Consider a future Video On Demand (VOD) service where movies are provided to subscribers over a high speed fiber-optic network. Advances in networking technologies will contribute to the realization of the VOD service over the Metropolitan Area Network [11] and [13]. Video objects are very large even in compressed form. One motion picture with NTSC quality video which is 100 minutes long, in uncompressed form occupies 40 GB (*Giga Bytes*) of storage and 1 GB when compressed according to MPEG standard [5]. These video objects have to be made available to the client in a continuous fashion in order to avoid "hiccups". A number of storage techniques assuring continuity of playback have been considered in [6] and [12]. Good designs of a storage server capable of servicing a large number of simultaneous requests have been discussed in [13] and [14]. These servers support the VOD service in the client-server approach.

In some proposed solutions a small number of most popular movies (with multiple requests coming possibly over short period of time) is periodically broadcasted on the network [*AT&T Archives*]. In this broadcasting approach of service the user's request for a particular movie does not have to be transmitted to the server. The client just waits until the movie of its choice is downloaded from within the broadcasted batch. In the broadcasting approach the access time, defined as the maximum time the client may have to wait for the selected movie, is independent of the number of clients. This is not the case in the client-server approach. Thus, the broadcasting solution scales very well with growing client population. No explicit request (to the server) is made for any movie or for any control function (stopping, pausing rewinding and fast forwarding). These requests are handled at the client end instead of being handled at the server end.

It is usually the case that most of the demand (80 %) is for a few (10 - 20) very popular movies[1], like the new releases and the top ten movies of the season/year [3]. Oracle (a future VOD service provider) has set up a three-tiered video server. The dozen or so most popular releases will always be loaded into the first tier, the main memory of the computer. This would give thousands of viewers quick access to the digital files that get used the most. The second tier would be kept on 1000 or more hard storage disks inside the server, containing about 200 of the next-most-popular movies. The third tier, reserved for lower-demand movies requested only occasionally, would be a separate machine - a "video jukebox" with tens of thousands of 8-millimeter digital tapes, each containing a single movie. When a viewer requests an archived title, a robotic arm, would grab the cassette and load it into the video server's memory bank. Frank Capra's Christmas classic of 1946, "It's a wonderful life," is a good example of a movie that might rotate among all the three tiers. There will be hardly any demand for this movie for most part of the year except for a stray request from a lonely insomnaic late one june night. So, it would be appropriate to store it in the archive on 8-millimeter tape. It can be loaded onto the hard drives a few weeks before Christmas, when the number of requests for it increases. For the real heavy demand as on the Christmas Eve, it can be made part of the first tier.

Moreover, most of the request for the movies are during the prime time (say between 7 pm to 11 pm on week days and on week-ends). It is a good idea to broadcast the movies of the first tier, rather than to provide it by the client-server approach because broadcasting approach scales up well. This way, even if the number of clients (hence the requests) increases 10 folds, all the clients still get the same access time. The VOD service can be rendered as :

- (1) During the prime time, the broadcasting approach is used for the popul ar movies (movies

[1] This is the reason, outlets of the video rental chain *Blockbusters* have as many as 50 copies of each of the top 20 movies and just 2 copies of the rest (on an average).

in the first tier) and the rest of the movies are provided by the client-server approach.

- (2) During other times, all the movies are provided by the client server approach.

In this paper, we consider only the broadcasting approach during the prime time, when a few popular movies are broadcasted on the network (and the rest is provided in the client-server approach). The problem with the broadcasting approach is that the access time for a movie can be very large - since in the worst case the client has to wait through the entire broadcasted batch to get to it's choice. Techniques that are simple to use, have a fast access time and use the bandwidth efficiently are needed. In this paper, we provide a method of multiplexing the broadcasted movies (or in general, large contiguous objects) in such a way that the resulting access time is radically reduced, compared to a number of current solutions termed *conventional broadcasting.* Although our motivation comes from VOD service, the proposed method can be applied to any large objects which require continuous consumption.

In many applications, which involve dissemination of information to a large number of users, communication network can be treated as a *storage* medium, where frequently accessed data is periodically transmitted [1], [4], [7] and [8]. Such periodic transmission can be viewed as *caching on the network* [7] and [8]. Our proposal can be treated as a data organization and access method for a new, rather non-traditional memory medium - the fast telecommunication network. In fact, we see that in the near future the distinction between communication network and remote storage will be gradually disappearing and the user will be treating the network as one gigantic database from which the data has to be either explicitly requested or it just "comes" if one tunes in at the right time.

Basic Architectures

Our environment will consist of *users*, (video) *objects* and (network and I/O) *bandwidth.* Users will be interested in viewing selected video objects continuously with minimal possible delay.

We distinguish between the *Data Centered* and *User Centered* approaches. In the User Centered approach, a user eventually obtains some dedicated bandwidth. This can be accomplished either by providing a lot of bandwidth, equal to the consumption rate of the object[2] *times* the number of users, or less bandwidth for which the users compete by negotiating with a scheduler. When a

[2]Consumption rate of the video object is equal to amount of bandwidth necessary to continuously view it.

user makes a request to the server, the server sends the requested object to the user via a dedicated (virtual) channel. Such an architecture has been described in [6], [9], [11] and [15].

In the Data Centered approach, bandwidth (channels) are dedicated to *objects* rather than to users. Each object is allocated some bandwidth on a logical channel. This bandwidth is reserved for periodic broadcast of that object. When a user wants to view a video object, he/she just tunes into the channel on which the object is broadcasted (all users have a cache, which stores the proper directory-about which object is broadcasted on which channel and at what time) and after some delay begins the viewing. No requests are sent to the server. Such an architecture has been suggested in the past in [1], [4], [7], [8] and [10]. Data Centered solution can be implemented in ATM using broadcast or multicast facilities and create "subject based" connections.

Yet another approach is to gather the requests for each object over a period of time and multicast the object to the users who had requested that object. This is batch processing, where batches of requests rather than individual requests are satisfied. This again is a special case of the Data Centered approach, where the number of requests against each object is gathered and then the requests are satisfied.

Data Centered approach scales much better with the number of users than the user centered approach. The main reason being, in the Data Centered approach we can take advantage of the *repetitive requests for the same object* (e.g., the same popular movie in the VOD service) and broadcast it periodically. The bandwidth requirement in the case of the User Centered approach is proportional to the number of users, while in the Data Centered approach it is proportional to the number of objects. So, as the number of users increases, the User Centered approach will fail to scale. One can establish a cut off point (the number of users) where the User Centered approach will perform worse than the Data Centered approach. In this paper, we are considering the service being rendered to a massive number of clients and hence the Data Centered approach is better suited.

Traditionally, video objects have been provided in a continuous fashion with no division of the objects, pipelining or multiplexing. For example, the HBO channel broadcasts the same set of movies a number of times on its channel and each time, the movie is broadcasted in its entirety, from start to finish. No other movie is broadcasted till the end of transmission of the current movie. In the broadcasting approach a number of objects are periodically broadcasted on the network. Objects may be broadcasted continuously (one after another) or multiplexed (interleaved) - as we discuss later. In any event, the client has to wait until the video object it wants to view is broadcasted. This delay called the access time, is a parameter that is a major concern to be optimized. In this paper, we provide a method of multiplexing video objects on the broadcast network in such a way that this access time is drastically reduced.

This paper is organized as follows: In section 2, we discuss the basic notions of a broadcasting channel and give some definitions. In section 3, the pyramid broadcasting algorithm is described and its analysis is given. The optimality issues and the storage requirements are discussed in section 3.1. In the section 3.2, the benefits of pyramid broadcasting is compared with that of conventional broadcasting. Section 4 illustrates the pyramid broadcasting algorithm with an example. In section 5, the implementation details of pyramid broadcasting algorithm is discussed. Finally, section 6 discusses the conclusions.

2 Basic Notions

Consider a VOD service to be rendered to a massive number of clients on a high speed fiber-optic network. We consider a network (with broadcast capability) which serves a wide population of users by broadcasting video objects intended for continuous viewing. The client is a multimedia client, which is typically a workstation or a powerful PC with capabilities of receiving and decompressing the digital video, storing it in the secondary storage and concurrently playing it from storage in the predefined consumption rate. This is along the lines of the client in the VOD service to be provided by Silicon Graphics [9], where the client is a modified *indy* workstation with a 100 MHz MIPS R4000 processo r and a Scientific-Atlanta add-in board for decompression and analog signal processing.

Video objects will have a *consumption rate* 'b' bits (frames)/seconds. This means that at least b bits (frames) have to be delivered at the client every second, for continuous viewing of the object. *For simplicity, we will measure the bandwidth of the physical channel B in the multiples of b. Consider M video objects, each of size D.*

The size D of a video object will be measured as the consumption (viewing) time for the entire object. For example, the size of a typical movie is equal to 100 minutes. The bandwidth $B = 10$ indicates that the bit rate of the communication channel is 10 times as much as the consumption rate of a movie. Thus, with $B = 10$, assuming no compression, it will take 10 minutes to physically send the movie over the network. In general, the broadcasting time for an object of size D on a channel with bandwidth B is equal to $\frac{D}{B}$.

Definitions:

The Access Time 'T' for a video object, denotes the upper bound on the time required by the client to begin consuming the video object, from the moment the client needed the object.

Let $\beta = \frac{B}{M}$, *denote the amount of bandwidth that is available per video object.*

We identify two basic types of *conventional broadcasting*:

- (1) The M objects are broadcasted on the channel in their entirety, one after another. Thus the access time for any object is $\frac{M*D}{B}$.

- (2) The network is considered as being divided into B logical channels (see the last paragraph of this section). Replica of each object is broadcasted on different logical channels with a phase delay of d time units. There are $\frac{D}{d}$ logical channels for each object. On each logical channel, one object is broadcasted periodically, in its entirety. The object is broadcasted at the consumption rate. In this way any client can start viewing the object in at most d time units (by tuning into one of the $\frac{D}{d}$ logical channels.) The total available bandwidth B, is divided equally among the M objects. Each object has a bandwidth of $\frac{B}{M}$ allocated to it. Since each channel has a bandwidth equal to the consumption rate of the object, $\frac{D}{d} = \frac{B}{M}$ (note that B is measured in terms of the consumption rate of the object, hence it has no units). Any client requesting an object, can tune into the correct channel and begin viewing the object in not more than $d = \frac{M*D}{B}$ time units. Thus the access time is $\frac{M*D}{B}$.

Both the types of conventional broadcasting have the same access time $T_c = \frac{M*D}{B}$. Note that as the bandwidth B of the network increases, the access time T_c decreases *linearly*. Thus, the conventional broadcasting channel can be viewed as a storage medium which holds M video objects each of size D and provides an access time of T_c. In conventional broadcasting scenario, the client tunes into a specified channel and starts downloading the object. If objects are broadcasted at a bandwidth equal to the consumption rate of the object (as in the second conventional scheme) then, no storage is necessary. The object is consumed directly after being picked up from the channel. If the objects are broadcasted at a bandwidth greater than the consumption rate of the object (the first conventional scheme) then, storage is required at the client. The client while consuming the object, will concurrently store portions of the object that will be consumed in the future.

For illustrating the algorithm, we will divide the *physical* communication channel of bandwidth B into K *logical* channels and explain our scheme as if the channel division took place. In reality the K channels will be time multiplexed on a single communication channel of bandwidth B. Each logical channel will have a bandwidth of $B' = \frac{B}{K}$. The data corresponding to one unit of time in the *physical* channel will be represented in K units of time in one of the *logical* channels. The data corresponding to the t th unit of time on the *physical* channel will be mapped into, $\lfloor \frac{t}{K} \rfloor * K$ thru $(\lfloor \frac{t}{K} \rfloor * K + K)$ units of time in the $(t \textit{ modulo } K)$ th *logical* channel[3]. In the following, we will refer to a logical channel as simply channel.

3 Pyramid Broadcasting

Fiber-optic networks with transmission speed in the tune of *gigabytes* per second already exists, this speed is expected to reach *terabytes* per second in the near future. The consumption rate of a movie with MPEG compression is typically 1.5 Mbps. When a VOD service is planned to be rendered on such a network, the enormous difference between the available bandwidth of the network and the consumption rate of the client (the movie) can be exploited to achieve significant reduction in the access time of the movies (rather than just the linear reduction in the case of conventional broadcasting).

We exploit this difference by multiplexing the objects on the channels in such a way that the clients can start consuming the object early. This is achieved by breaking up the object into a number of segments of increasing sizes. The smallest segment is broadcasted a lot of times. The frequency of broadcasting a segment decreases with the increase in its size. The segments are broadcasted in such a way that, once the first segment is got then, the subsequent segments are got in time, so that the object can be viewed continuously. In this *pipelining* approach, the time to access the object is the time to access the first segment. Since the first segment is the smallest and it is also broadcasted the most number of times, the access time for the first segment is very small. While the first segment is being consumed, the second segment is collected. This process of collecting future segments while consuming the current segment is continued till the whole object

[3]for an illustration of the time division of physical channel into K logical channel, see the Appendix

is collected. The proposed method which is called pyramid broadcasting, results in a substantial improvement in the time required to access the object while assuring that the clients can consume the object in a continuous fashion (with no interruptions). The algorithm describes the organization of the object on the channel by the server and client's access protocol.

On the SERVER SIDE broadcasting involves the following steps :

- The physical channel of bandwidth B is divided into K logical channels, each of bandwidth $B' = \frac{B}{K}$.
- Each object of size D is divided into K segments. The data segments are of sizes D_i, $i = 1, ..., K$, where $D = D_1 \bullet D_2 \bullet \ldots \bullet D_K$. The concatenation ($\bullet$) of all the *segments*, in the order of increasing segment numbers, constitutes the whole object.
- D_{i+1} is made α times the size of D_i, for $i = 1, ..., (K-1)$, where α denotes the ratio between the size of segments between two adjacent channels. Later, we will express α in terms B, M and K.
- Data segments of size D_i are put on channel i. The data segments of size D_i, of all the M objects are broadcasted together in a sequence periodically. Within a channel, the broadcasting is conventional.

On the CLIENT SIDE the following steps are involved :

- Begin downloading the first data segment of the required object at the first occurrence and start consuming it concurrently.
- Download data segment of size D_i in the order $i = 1, \ldots, K$ at the earliest possible time after beginning to consume data segment of size D_{i-1} of the required object.
- Any data segment of size D_i is begun to be downloaded only from the start of that data segment.

Segments are made available to the client in increasing segment sizes, hence the name *pyramid broadcasting.* Notice, that consumption and downloading are done concurrently. The client's storage is used to store the segments that will be consumed in the future.

Alpha :

The parameter α is chosen in such a way that contiguous viewing of a video object is assured. This is derived from the following *Continuity principle (∗)* which assures continuous viewing of an object:

$$consumption\_time(D_i) \;\geq\; access\_time(D_{i+1}, B') \qquad\qquad (*)$$

where $consumption\_time(D_i) = D_i$, by definition. $access\_time(D_{i+1}, B')$ denotes the upper bound (worst case) time to begin downloading the D_{i+1} on the $(i+1)$ th channel which has a bandwidth of B'.

Since $\quad D_{i+1} = \alpha * D_i \quad$ and $\quad access\_time(D_{i+1}, B') \;=\; \frac{D_{i+1} * M}{B'} \quad$ hence, from $(*)$:

$$\alpha \;\leq\; \frac{B}{M * K}$$

Consequently, we set $\quad \alpha \;=\; \frac{B}{M*K} \quad$ since this allows the best bandwidth utilization.

Below we provide the analysis for the access time of pyramid broadcasting and discuss the optimal access time issues and the storage requirement.

3.1 Analysis

Access Time:

The access time T for the object will be equal to the access time of the first data segment of size D_1 which is broadcasted on the first logical channel. This is because, once the first data segment is available, the subsequent data segments are assured to be made available before the previous data segment is consumed. This is done by choosing α as above (i.e., by satisfying the continuity principle).

On the first channel, M segments (the first segment of each object) each of size D_1 are broadcasted. Thus, the time required to broadcast all the M segments is $\frac{D_1 * M}{B'}$. If a client tunes into the first channel then, in the worst case it would have just missed the start of the first segment of the object it wants. In this case, it will have to wait for a duration equal to the broadcast time of all the M first segments. Hence, the access time of the first segment of any object on this channel is:

$$T = \frac{D_1 * M}{B'} = \frac{D_1}{\alpha}$$

$$D_2 = \alpha * D_1$$

This is true for all the channels from 2 through K. Thus,

$$D_i = \alpha * D_{i-1} \qquad i \ = \ 2, \ldots, K$$

$$D_i = \alpha^{i-1} * D_1 \qquad i \ = \ 2, \ldots, K$$

$$D \ = \ D_1 + \ldots + D_K$$

$$D \ = \ D_1 * (1 + \alpha + \ldots + \alpha^{K-1})$$

The solution to this geometric series is :

$$D \ = \ D_1 * (\frac{\alpha^K - 1}{\alpha - 1}) \qquad i.e., \qquad D_1 \ = \ \frac{D * (\alpha - 1)}{(\alpha^K - 1)}$$

$$T = \frac{D * (\alpha - 1)}{\alpha * (\alpha^K - 1)}$$

Notice that K can be expressed in terms of α, B and M since $\alpha \ = \ \frac{B}{M*K}$

Therefore, the access time T depends only on α, bandwidth B and the number of objects M. For $\alpha < 1$ the access time increases, hence for a better access time, K should selected such that $\alpha \geq 1$.

Optimal Access Time:

In this subsection, we discuss access time optimality issues. We assume that the number of video objects M and the duration of each video object D is constant. The access time for an object is dependent on the number of logical channels K, which in turn depends on α. For a constant bandwidth of the physical channel B, we varied the number of logical channels.

Figure 1 ($left$) shows the access time (Y-axis) obtained for different values of α (X-axis). The bandwidth B is $20*M$, indicating that $\beta = 20$. The optimal access time using pyramid broadcasting is obtained approximately for $\alpha = 2.5$.

Figure 1 ($right$) indicates the optimal α (Y-axis) for different β (hence different total bandwidth). For all the bandwidths, the optimal access time using pyramid broadcasting is obtained, for a value of α around the *euler's constant* ($e = 2.72$). We conjecture that the optimal α is such that $\alpha \ \leq \ e$, for the smallest K.

Control Functions : Providing the control functions of stopping, pausing and rewinding is straight forward if the client has enough storage to buffer the entire object. The storage can be in the form

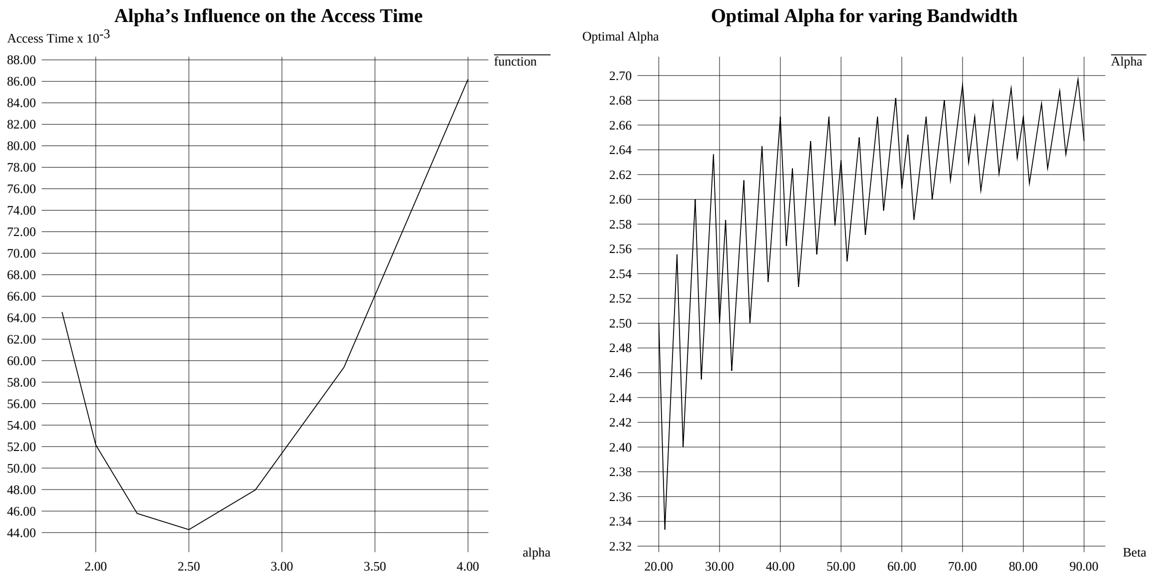

Figure 1: Alpha's Influence

of optical or magnetic disks or in the form of tape. The storage capacity of magnetic disks are predicted to increase with the prices coming down[4] and tapes are cheap. Assuming that the drives are no more expensive than VCRs (which is a kind of tape drive) the solution we provide is viable. The control functions will be handled at the client end. Each control command given by the user is interpreted by the client (machine) and then an appropriate action is taken. Providing the fast forward control function is quite difficult even with storage at the client end. We can try to catch the next data segment and being consuming the object from that point onwards, but then the next segment is not guaranteed to be available the moment the fast forward control command is given. In this sense, our solution is rendering the Near-Video On Demand Service. We provide all the control functions for making it interactive, except the fast forward command.

Storage Requirement :

In this subsection we compute the minimum storage that is required for supporting pyramid broadcasting (with no control functions). The video object has to be received in a continuous manner with no "hiccups". At any point the client will be downloading at most from two consecutive channels and the data segment from one of them will be consumed in parallel. (true because that

[4] There are predictions (on the front page of the Seattle Times on 1992 May 09) that writable optical storage unit with a capacity of 1 GB will be available in the near future. Such a storage unit will have no moving parts and will be in the size of a credit card. This "compact card" with a transfer speed of 45 Mbps is supposed to cost as little as $ 20. No costs regarding the drives for such storage units have been discussed yet

is how the algorithm works.) The storage required to buffer an object of size S time units (we consider the size of an object in units of time) is equal to $S * b$ (i.e., the product of the length of the object and the consumption rate of the object).

Below, we show that the minimum amount of storage required for pyramid broadcasting is $(D_K - \frac{D_K}{B'} + D_{K-1}) * b$. Visualize the storage as a sequence of locations from 1 through $(D_K - \frac{D_K}{B'} + D_{K-1}) * b$. Since the last two channels have the largest data segments (since α is greater than 1), it is enough if we consider the memory required while these two channels are being read. Consider the $(K-1)$ th and the K th channel. The data segment are D_{K-1} and D_K. While D_{K-1} and D_K are being downloaded, D_{K-1} will be consumed in parallel. Let D_{K-1} be stored starting from location $(D_K - \frac{D_K}{B'}) * b$ of the storage and let D_K be stored starting from location 1. D_K will be begun to be downloaded (according to the algorithm) after (or when) D_{K-1} is begun to be consumed. Hence, in the worst case both D_K and D_{K-1} will be begun to be downloaded at the same instant. Both D_K and D_{K-1} will be downloaded at the bandwidth of the channel, i.e., $B' = \frac{B}{K}$. D_{K-1} will be consumed at the rate of b. It will take $\frac{D_K}{B'}$ units of time for the client to consume $\frac{D_K}{B'} * b$ amount of data. That is exactly the time when the information from the K th channel will be written on the storage. Since, the data from $(K-1)$ th channel is being written on the storage beginning at the $(D_K - \frac{D_K}{B'}) * b$ th location, by the time the data being written from the K th channel is almost complete, the data from the D_{K-1} th would have vacated the location from $(D_K - \frac{D_K}{B'}) * b$ thru $D_K * b$. If D_{K-1} is being to be written at a location lesser than $(D_K - \frac{D_K}{B'}) * b$ then, D_K may overwrite some part D_{K-1} before it is consumed. If D_{K-1} is being to be written in a location greater than $(D_K - \frac{D_K}{B'}) * b$ then storage is being wasted.

Thus for the minimum storage, we have to begin writing D_{K-1} beginning at $(D_K - \frac{D_K}{B'}) * b$ and the storage ($Storage$) required for pyramid broadcasting to work correctly is at least $(D_K - \frac{D_K}{B'} + D_{K-1}) * b$

$$Storage \geq (D_K - \frac{D_K}{B'} + D_{K-1}) * b$$

$$Storage \geq \left(\frac{D * (B - MK) * B^{K-2} * (B - K + MK)}{B^K - (MK)^K} \right) * b \qquad (**)$$

$$Storage \approx \left(\frac{D * (\beta^K - K^2 \beta^{K-2})}{\beta^K - K^K} \right) * b$$

As K increases by one, the function on the right hand side of the inequality has the property, such that it goes from $\frac{x}{y}$ to $\frac{x+x'}{y+y'}$, where $y' > x'$. This means that, as K increases the function decreases, implying that the amount of storage required for pyramid broadcasting decreases with an increase in K.

Given a storage of $Storage$ bytes, the selection of K so as to get the minimum access time using pyramid broadcasting can be decided as follows :

- Keep increasing K as long as the Inequality $(**)$ holds, then
- Keep increasing K as long as α is more than e.

At any point two writing and one reading has to be done. If the disk has sufficient bandwidth then the bandwidth of the disk can be time multiplexed between the three activities (all getting the equal bandwidth).

3.2 Benefits of Pyramid Broadcasting

We will be accessing the relative access time benefits of using pyramid broadcasting as opposed to conventional broadcasting.

$$T_{conventional} = \frac{D * M}{B} \qquad \& \qquad T = \frac{D * (\alpha - 1)}{\alpha * (\alpha^{\frac{B}{M\alpha}} - 1)}$$

where $T_{conventional}$ and T denote the access time in conventional broadcasting and pyramid broadcasting respectively.
The access time gain due to the pyramid broadcasting is computed as a ratio of the access time due to conventional broadcasting to that of the access time due to pyramid broadcasting. The gain can then be calculated expressing K in terms of α, B and M. The final formula is presented below:

$$\frac{(\alpha^{\frac{B}{M\alpha}} - 1)}{(\frac{B}{M\alpha}) * (\alpha - 1)} \qquad i.e., \qquad \frac{(\alpha^{\frac{\beta}{\alpha}} - 1)}{(\frac{\beta}{\alpha}) * (\alpha - 1)}$$

In particular, notice that if we increase β by 1 then, we reduce the access time of the pyramid broadcasting method by the factor of $\alpha^{\frac{1}{\alpha}}$ while in case of the conventional method we only obtain an additive improvement.

In terms of the bandwidth gain, we compute the bandwidth reduction in case of pyramid broadcasting as opposed to conventional broadcasting, for a fixed access time. The bandwidth

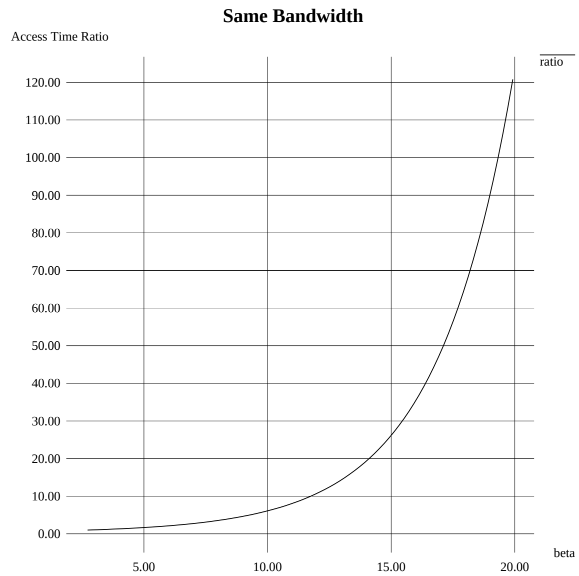

Figure 2: Access Time Comparison

$B_{conventional}$ required for achieving an access time T in conventional broadcasting and the bandwidth B in case of pyramid broadcasting for the same access time are :

$$B_{conventional} = \frac{D * M}{T} \qquad \& \qquad B \leq (M * \alpha * (\log_{\alpha}(\frac{D(\alpha - 1)}{T\alpha} + 1)))$$

The bandwidth gain due to pyramid broadcasting can be calculated to be equal to at least the following ratio.

$$\frac{D}{T * \alpha * (\log_{\alpha}(\frac{D(\alpha-1)}{T\alpha} + 1))}$$

We will be accessing the relative bandwidth gain using pyramid broadcasting as opposed to conventional broadcasting for a fixed access time, by expressing the access time as a percentage of the object size (in time units). The graphs in Figure 2 and 3 show the advantages of the pyramid broadcast versus conventional one in terms of

- Access Time, given the Bandwidth
- Bandwidth, given an Access Time

The value of α is e in all of the following cases.

The graph in Figure 2 illustrates how the ratio of conventional broadcasting access time to pyramid broadcasting access time varies in terms of increasing values of β. The graph illustrates

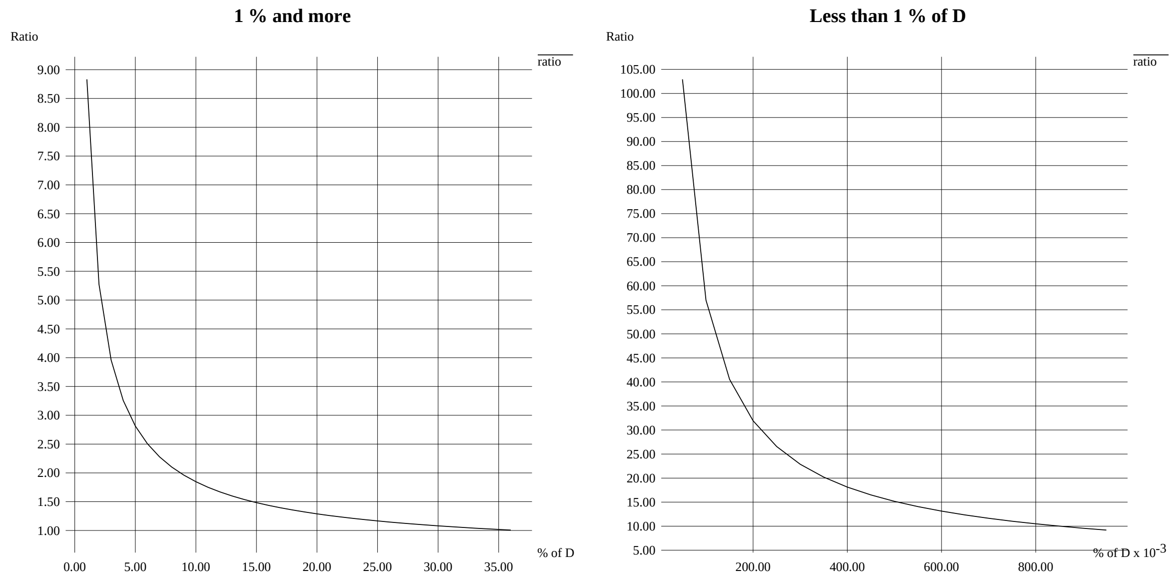

Figure 3: Bandwidth Comparison

that, as β (and hence the bandwidth) increases, so does the benefits. When β is approximately equal to e, there is no benefit. When $\beta = 20$, the the access time we get using pyramid broadcasting is more than 120 times lesser than the access time obtained using conventional broadcasting.

The graph in Figure 3 illustrates how the ratio between the bandwidth needed in case of conventional broadcasting to pyramid broadcasting varies depending on the access time requirements. The access time is measured as a fraction of the duration of the object The graph in Figure 3($left$) shows that when access time is greater than $\frac{D}{\alpha}$ (35%) of the size of D then there is no gain. But as the access time decreases, the gain increases. For an access time of 1% of the size of D, the bandwidth we need using pyramid broadcasting is 10 times less than the bandwidth we need using conventional broadcasting. The graph in Figure 3($right$). illustrates that if the required access time is very small, then the benefits are substantial. As an extreme case consider, an access time of 0.05 percent of the duration of the object then the gain in the bandwidth is over 100 times.

Hence, the benefits of pyramid broadcasting are most impressive in cases when access time requirements are very rigid.

4 Example

In this section, we will illustrate the pyramid broadcasting scheme with α equal to 2. The application considered is the VOD service, where the number of movies (M) in the library is 10. The

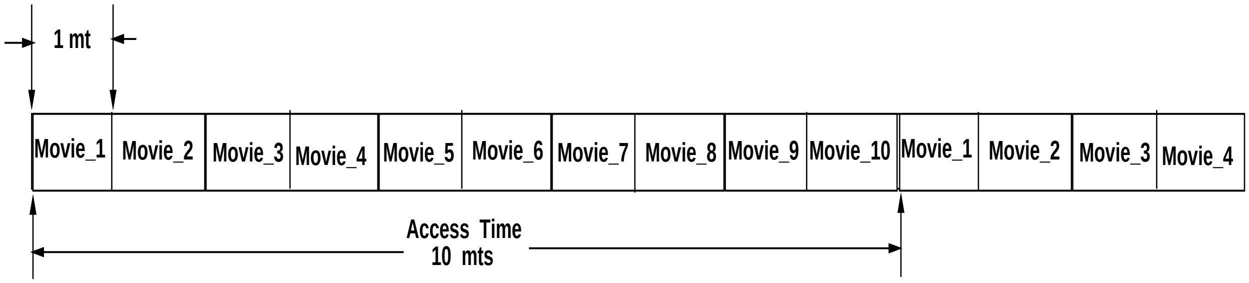

Figure 4: Conventional Broadcasting

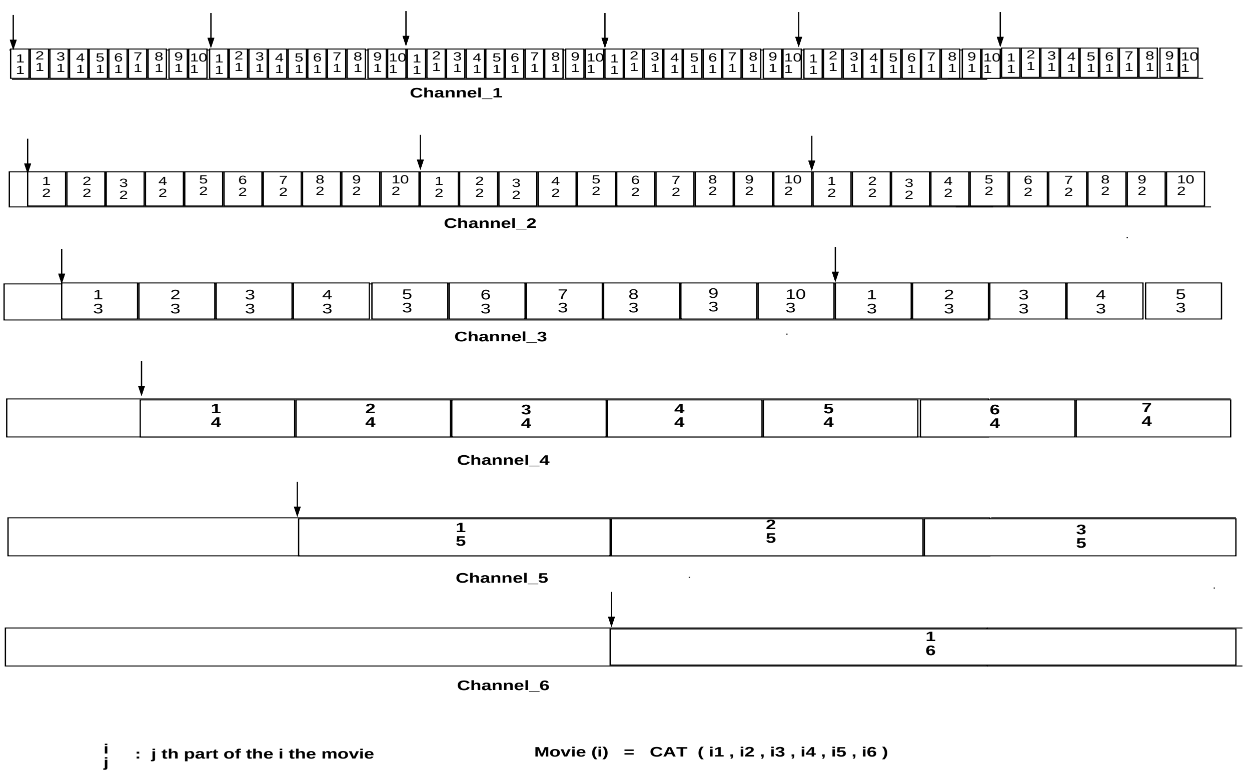

Figure 5: Pyramid Broadcasting

duration of each movie is 2 hrs $(D) = 120\ mts$. The consumption bandwidth of the movie (b) is 50 Mega bits per second ($Mbps$). Let the bandwidth of the broadcast channel be equal to 6 Giga bits per second ($Gbps$). Thus, the channel bandwidth B as a multiple of the consumption rate of the object is equal to $\frac{6\ Gbps}{50\ Mbps} = 120$.

In conventional broadcasting, we periodically broadcast *movie 1, movie 2,..., movie 10* on the channel. As indicated earlier, without the loss of generality, we consider the first conventional broadcasting method. With a bandwidth of 6 $Gbps$, each movie will take 1 minute to be broadcasted. The access time is the total length of the broadcast of the ten movies, i.e., 10 mts. Figure 4 illustrates channel allocation for conventional broadcasting..

We will demonstrate that the pyramid broadcasting will ensure the access time of 1 *minute* for the same channel bandwidth and the same number of movies.
As mentioned earlier $\alpha = \frac{B}{M*K} = 2$, implying that K, the number of channels equal to 6.

Figure 5 illustrates how movies are multiplexed on the channel using the pyramid broadcasting scheme. We divide each movie into 6 *segments*. Each *segment* is double the size of the previous *segment*. The *first segment* is the equivalent of 2 mts of the movie, the *second segment* is the equivalent of 4 mts of the movie, the *third segment* is an equivalent of 8 mts, the *fourth* an equivalent of 16 mts, the *fifth segment* an equivalent of 32 mts and the *sixth segment* being an equivalent of 64 mts of the movie. The concatenation of all the *segments*, in the order of increasing segment number, constitutes the whole movie. Note that the segments add up to 126 mts and hence, we can broadcast a longer movie (with the same benifits as given below).

In figure 5 each data segment is denoted by a rectangle. The upper number inside the rectangle denotes the movie number and the lower number refers to the segment number within a movie.

The bandwidth of each channel is 1 $Gbps$ i.e., $B' = 20$. The size of the first segments is 2 mts each. There are 10 movies and the first segment of each movie will be broadcasted in a duration $\frac{2\ mts}{20} = \frac{1}{10}\ mts$. The first segments of all the 10 movies will be broadcasted in 1 minute. Thus, the access time for any movie is 1 minute.

With a bandwidth of 6 $Gbps$, the access time using pyramid broadcasting is 1 minute whereas the access time using conventional broadcasting is 10 mts. Thus pyramid broadcasting results in an access time that is 10 times lesser.

To achieve an access time of 1 minute using conventional broadcasting , we need a bandwidth

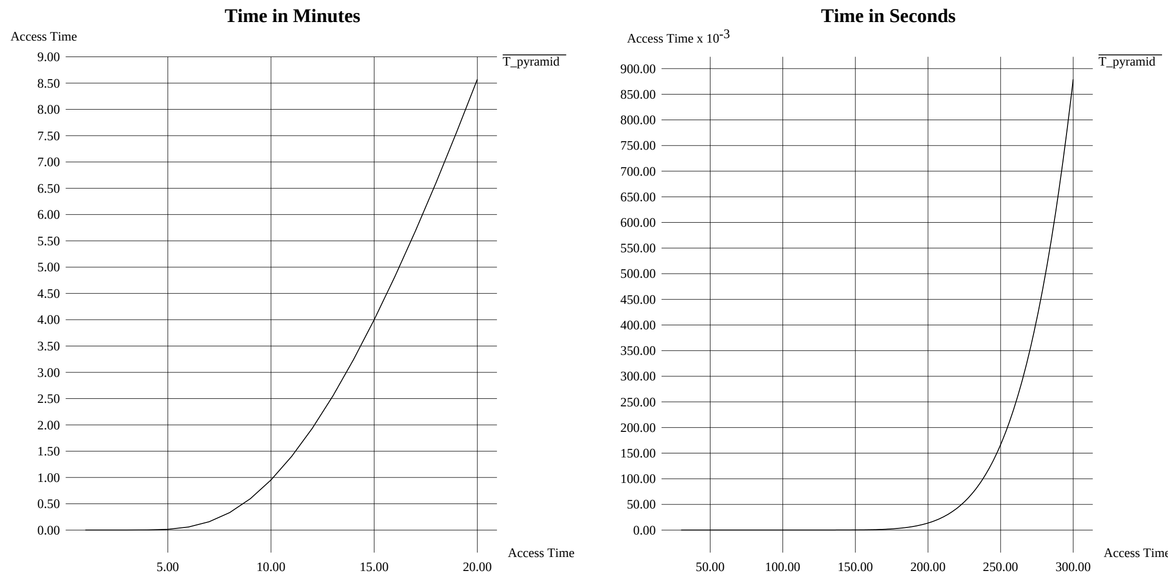

Figure 6: Access Time Benefits

of $B = D * M$. That is, we need a bandwidth of 60 $Gbps$. Using our method we got this access time using a bandwidth of just 6 $Gbps$, saving a bandwidth of 54 $Gbps$.

4.1 Illustration of Benefits

In this subsection, we illustrate the benefits of using pyramid broadcasting in the above scenario.

The graph in Figure 6 illustrates the benefits of pyramid broadcasting in terms of the access time. The x-axis denotes the access time in conventional broadcasting. The y-axis denotes the access time we get if we use pyramid broadcasting. The bandwidth is the same in both the case. The graph in Figure 6($left$) illustrates the benefits of pyramid broadcasting for large access times. As stated earlier, the access time gain is substantial when the required access time is low. We illustrate this in the graph in Figure 6($right$).

The graph in Figure 7 illustrates the bandwidth benefits of pyramid broadcasting. The x-axis denotes the bandwidth used in the conventional broadcasting in bits per second. The y-axis denotes the bandwidth for pyramid broadcast for the same access time as in conventional broadcasting. It is clear that as the bandwidth increases, so does the benefit.

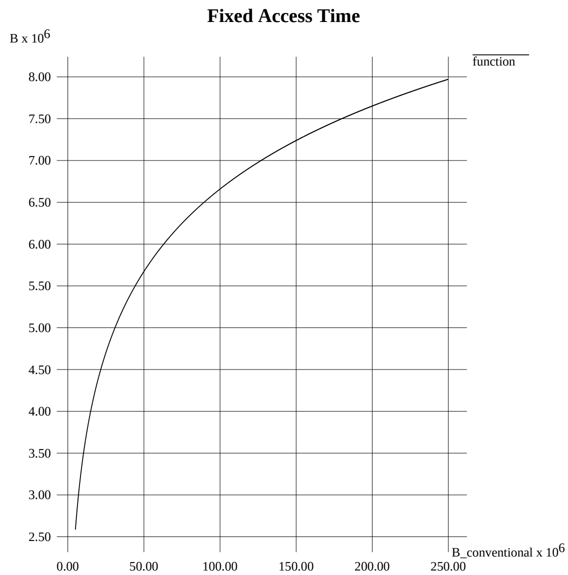

Figure 7: Bandwidth Benefits

5 Implementation

We have implemented pyramid broadcasting over an ethernet LAN. *Sun* workstations connected to the LAN served as the server and the clients. The local disks were used as the buffer for storing the future data segments. A Tcl/Tk based interface was developed for getting the input from the users. For comparison, we implemented both the pyramid broadcasting algorithm and the conventional broadcasting algorithm. Audio and video clips (38.4 Kbps with MPEG compression) were used as video objects. The broadcasting (both conventional and pyramid) algorithms were implemented using sockets.

All the video (audio) clips were of the same size. The user could select video clips or songs from the menu driven interface. The selection should be homogeneous i.e., all video clips or all audio clips. The number of channels to be opened and the total bandwidth (B) is also given as an input by the user.

Pyramid Broadcasting :

SERVER : The control files for storing the header that will be sent out with the data on each channel, is created first. Alpha (α) is computed using the input parameters and each object is divided into data segments of the required size, to be put on each channel. The actual broadcasting on the network is done using sockets. Corresponding to each channel a broadcasting process is forked

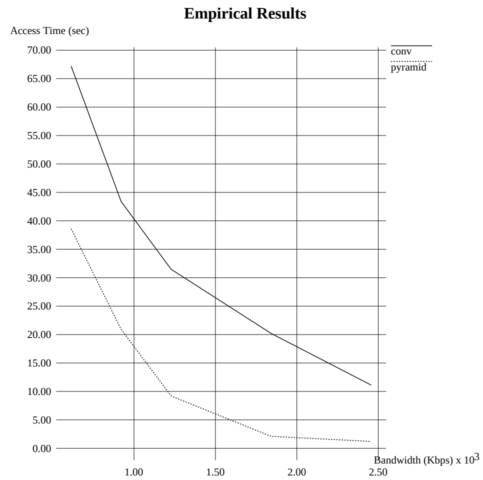

Figure 8: Empirical Results

which in turn opens the socket and write the corresponding data segment (of all the selected object) into it, periodically. While sending the data, flow control is done using the 'usleep' command after a fixed number of packets.

CLIENT : As soon as a client interface is opened on a machine the server writes the control information (the socket ids to listen to and the order in which to listen) in a data file. Alternatively we could have a control channel where the directory is broadcasted periodically and the client listens to this socket id when it comes up. Corresponding to each channel a file is opened to store the data segment corresponding to that channel. Two processes are forked, one reads the current socket (channel) and the other reads the subsequent socket. The parent opens a pipe into the (play / mpeg_play) process and writes the data from the files corresponding to each socket into the pipe.

Conventional Broadcasting:

The second conventional broadcasting strategy was implemented.

SERVER : The input information regarding the number of channels to open, the video /audio objects and the overall bandwidth is got from the interface. Depending on the bandwidth the phase delay for the broadcast of each object, on each channel is determined.

Again the data is broadcasted using sockets. For each channel to be opened a process is forked which opens a socket and writes the objects (in their entirety) one after the other into the socket. Objects are written on the first socket immediately, in the second socket some junk packets are written for a duration equal to the phase delay and then the objects are written one after the other.

The junk packets at the beginning, for each socket is proportional to the phase delay for the objects on that channel. The flow control is again implemented as in the pyramid server.
CLIENT : The client has the control information about which socket ids to listen to and what are the phase lag in the broadcasting of the objects in each socket. Depending on the current time and the time at which the object is broadcasted in each socket, the socket on which the object is going to come next (this gives the minimum access time) is decided. The datafile (to store incoming data) are opened and this data is piped to play / mpeg-play process.

We performed a number of experiments with different values of bandwidth. The bandwidth was varied from 614 $Kbps$ to 2456 $Kbps$. Four video clips with a consumption rate of 38.4 $Kbps$ were broadcasted. The flow controlling mechanism that we used, gave the average bandwidth and it fluctuated with the number of other users (processes) in the machines. The bandwidth that we used had to be around 15 % more than that required theoretically. Otherwise, a some of the initial packets of the segments were missed and consequently the future segments were not made available to maintain the continuity of the audio/video clips. We attribute this to the variable load on the Sun machines and on the network. Hence, we had to use a bandwidth ranging from 706 $Kbps$ to 2824 $Kbps$, instead of 614 $Kbps$ to 2456 $Kbps$. The empirical results are shown in the graph of Figure 8. For a bandwidth of 2456 $Kbps$ the access time according to conventional broadcasting was 11.12 seconds and the access time according to pyramid broadcasting was 1.21 seconds. The access time due to pyramid broadcasting is 9 times smaller than the access time due to conventional broadcasting. If the video clips are 100 minutes long instead of just 300 seconds as in our experiments, the access time translates to 24.2 seconds due to pyramid broadcasting and 222.4 seconds due to conventional broadcasting.

6 Conclusions

Most of the users in a Video On Demand service offered at a metropolitan level, want the same few popular movies at about the same time. The companies which are going to provide VOD service in the future, have different ways of handling the most popular movies. We have looked at an alternative model for providing the most popular movies to the clients. Providing these few movies in a broadcasting approach will be a good idea, since it will scale up with the number of clients. Fiber-optic networks with transmission speed in the tune of *gigabytes* per second already exists, this speed is expected to reach *terabytes* per second in the near future. The consumption rate of a movie with MPEG compression is typically 1.5 Mbps. When a VOD service is planned to be rendered on such a network, the enormous difference between the available bandwidth of the network and the consumption rate of the movie can be exploited to achieve significant reduction in the access time of the movies, rather than just the linear reduction in the case of conventional broadcasting.

In pyramid broadcasting, we exploit this difference by multiplexing the objects on the channels in such a way that the clients can start consuming the object early. This is achieved by breaking up the object into a number of segments of increasing sizes. The smallest segment is broadcasted a lot of times. The frequency of broadcasting a segment decreases with the increase in its size. The segments are broadcasted in such a way that, once the first segment is got then, the subsequent segments are got in time, so that the object can be viewed continuously. In this *pipelining* approach, the time to access the object, is the time to access the first segment. Since the first segment is the smallest and it is also broadcasted a lot of times, the access time for the first segment is very small. While the first segment is being consumed, the second segment is collected. This process of collecting future segments while consuming the current segment is continued till the whole object is collected.

Pyramid broadcasting provides much better access time compared to conventional broadcasting. As the available bandwidth increases, the improvement in access time is *exponential* as opposed to just *linear* improvement in the case of conventional broadcasting. The larger the bandwidth of the channel, the better the access time due to pyramid broadcasting. As the access time requirement decreases, the bandwidth requirement in the case of conventional broadcasting increases *linearly* while the bandwidth requirement in the case of pyramid broadcasting increases only *logarithmically*.

Pyramid broadcasting can be used both for periodic broadcast of hot spots (objects in demand) of data as well as for the organization of large multicasted collections of video objects. Pyramid broadcasting can also be used for multicasting (sending the object to a group as a whole, instead of sending it to each member of the group separately) video objects that are on demand. This will result in substantial reduction of the access time and the bandwidth requirement as compared to the traditional way of multicasting.

References

[1] T. F. Bowen et.al., "The DATACYCLE Architecture,' ' Comm. of the ACM, Vol 35, No. 12, December 1992, pp. 71 – 81.

[2] David R. Cheriton,"Dissemination-Oriented Communication Systems", Stanford University, Tech. Rept. 1992.

[3] Don Clark and Evan I. Schwartz in "The Wall Street Journal- November 10 1993" and "The New York Times" respectively.

[4] G. Herman et al., "The datacycle architecture for very large high throughput database systems," in Proc ACM SIGMOD Conf., 1987, pp 97-103.

[5] Didier Le Gall, "MPEG : A Video Compression Standard for Multimedia Applications", Communications of the ACM, Vol. 34, No.4, April 1991.

[6] S Ghandeharizadeh and L. Ramos, "Continuous Retrival of Multimedia Data Using Parallelism", IEEE Tran. on Knowledge and Data Engineering, Vol. 5, No. 4, August 1993.

[7] T. Imielinski, S. Viswanathan and B. R. Badrinath, "Indexing on Air," ACM SIGMOD - International Conference on Management of Data - May 1994.

[8] T. Imielinski, S. Viswanathan and B.Badrinath, "Power Efficient Filtering of Data on Air", Fourth Intl Conference on Extending Database Technology -Cambridge, UK - March 1994.

[9] Larry Press, "The Internet and Interactive Television", Communications of the ACM, December 1993, Vol. 36, No 12 - pp 19 - 24.

[10] J C. Pasquale, G C. Polyzos, E W. Anderson and V P. Kompella, "The Multimedia Multicast Channel", in Proceedings of 3 rd Intl Workshop on Network and Operating System Support for Digital Audio and Video, November 1992.

[11] W.D. Sincoskie, " System Architecture for a Large Scale Video On Demand Service", Computer Networks and ISDN systems, 22 1991.

[12] P. Venkatan Rangan and Harrick M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia", IEEE Transactions on Knowledge and Data Engineering - August 1993.

[13] P. Venkatan Rangan, H M. Vin and S. Ramanathan, "Desigining an on-demand multimedia service," IEEE Comm Mag., Vol. 30, pp. 56-65, July 1992.

[14] Harrick M. Vin and P. Venkatan Rangan, "Desigining a Multi-User HDTV Storage Server," IEEE Journal on Selected Areas in Communication, Vol. 11, January 1993.

[15] S B. Weinstein, "Telecommunications in the Coming Decades", IEEE Spectrum, Nov 1987.

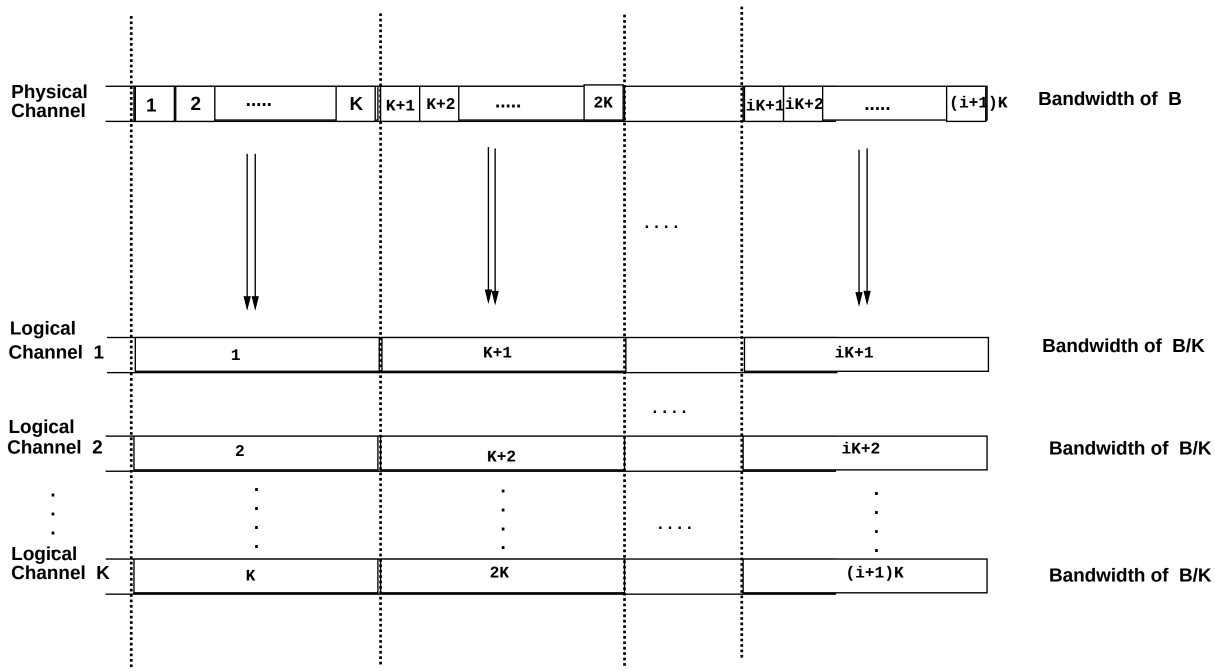

Figure 9: Physical channel to Logical channels

Appendix

Dividing a Physical channel into Logical channels

Figure 9 shows the division of physical channel into logical channel. The physical channel is the first channel from the top. This channel has a bandwidth of B. This channel can be considered to be made up of K logical channels, each with a bandwidth $\frac{B}{K}$. Data that is represented in one unit of time in the physical channel, is represented by K units of time in one of the logical channels. The data corresponding to the i th unit of time on the *physical* channel will be mapped into, $\lfloor \frac{i}{K} \rfloor * K$ thru $(\lfloor \frac{i}{K} \rfloor * K + K)$ units of time in the $(i \ modulo \ K)$ th *logical* channel.

We can re-map the logical channels into the physical channel by the inverse function: data from time units $i * K$ thru $(i+1) * K$ from the j th logical channel is mapped into the $(i * K + j)$ time unit in the physical channel. It has to be observed that, when the logical channels are re-mapped into physical channels then, the data segments will be physically divided into further equal fragments of unit time. But logically they are still an entity.