GENERALIZED BROADCASTING SCHEME FOR VIDEO-ON-DEMAND SERVICE: PERFORMANCE ANALYSIS

A Dissertation submitted to the
School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi,
in partial fulfillment of the requirements for the award of the degree of

Master of Technology

in

Computer Science and Technology

by

Vinod Kumar

Under the Supervision

of

Prof. Karmeshu

SCHOOL OF COMPUTER AND SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI -110067, INDIA

July 2006

जवाहरलाल नेहरू विश्वविद्यालय

JAWAHARLAL NEHRU UNIVERSITY

School of Computer & Systems Sciences

NEW DELHI- 110067, INDIA

CERTIFICATE

This is to certify that the dissertation entitled "**GENERALIZED BROADCASTING SCHEME FOR VIDEO-ON-DEMAND SERVICE: PERFORMANCE ANALYSIS**" being submitted by Mr. **Vinod Kumar** to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Technology**, is a record of bonafide work carried out by him under the supervision of Prof. Karmeshu.

This work has not been submitted in part or full to any university or institution for the award of any degree or diploma.

Prof. Karmeshu	Prof. S. Balasundaram
(Supervisor)	Dean, SC&SS,
SC&SS, JNU, New Delhi-67	JNU, New Delhi-67

जवाहरलाल नेहरू विश्वविद्यालय

JAWAHARLAL NEHRU UNIVERSITY
School of Computer & Systems Sciences
NEW DELHI- 110067, INDIA

DECLARATION

This is to certify that the dissertation entitled "**GENERALIZED BROADCASTING SCHEME FOR VIDEO-ON-DEMAND SERVICE: PERFORMANCE ANALYSIS**" is being submitted to the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science & Technology**, is a record of bonafide work carried out by me under the supervision of Prof. Karmeshu.

The matter embodied in the dissertation has not been submitted in part or full to any university or institution for the award of any other degree or diploma.

July 2006
JNU, New Delhi

Vinod Kumar
M.Tech, Final Semester,
SC&SS, JNU, New Delhi.

Dedicated

To my

Grand Father

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Prof. Karmeshu, for his guidance and continuous support during my research. His devotion to research and serious attitude toward science and technology has given me great encouragement and inspiration to accomplish this dissertation. It would be hard to imagine finishing this dissertation without his continuous support.

I gratefully acknowledge the unselfish help given to me by Rajesh Vats, who gave me a lot of help in resolving Mathematical problems. Special thanks are reserved for Dilip Kumar, Dinesh Singh, Ashutosh Goel, Mo. Lutfi, Sushil Sharma, Ankur Uniyal and Atul Sridhar for the discussions with me throughout my research work, and for all those moments in which we have enjoyed altogether. I can't forget Rajan Singh, my hostel-mate, who has supported me a lot whenever I got frustrated. My appreciation also goes to all the people in our lab. Their friendship and support made my research environment enjoyable.

Finally, I am most thankful to my family for their unlimited love, care and encouragement. I would not have even come back to school from my previous job without their encouragement and support.

Vinod Kumar

Table of Contents

List of Tables

List of Figures

Abbreviations Used

ATM	Asynchronous Transfer Mode
CHB	Cautious Harmonic Broadcasting
DTR	Disk I/O Transfer Rate
EHB	Enhanced Harmonic Broadcasting
ESEB	Equal Segment-Size Equal Bandwidth
ESUB	Equal Segment-Size Unequal Bandwidth
FB	Fast Broadcasting
Fib-1	Fibonacci minus One
HB	Harmonic Broadcasting
HBO	Home Box Office
I/O	Input/Output
Max	Maximum
MQL	Maximum Queue Length
NVOD	Near Video on Demand
PB	Pyramid Broadcasting
PHB	Poly Harmonic Broadcasting
PPB	Permutation based Pyramid Broadcasting
QHB	Quasi Harmonic Broadcasting
SB	Staircase Broadcasting
SkyB	Skyscraper Broadcasting
TVOD	True Video on demand
USEB	Unequal Segment-Size Equal Bandwidth
USUB	Unequal Segment-Size Unequal Bandwidth
VOD	Video on Demand

Papers Included in the Dissertation

Communicated Paper

- Vinod Kumar, "Fib-1 Broadcasting and Receiving Scheme for Video-on-Demand Service", IEEE Transactions on Multimedia (submitted).

Abstract

In order to address the scalability problem of video-on-demand (VOD) services, several periodic broadcast schemes have been suggested in the literature. These schemes partition a video into segments and repetitively broadcast each segment on a separate channel. We have proposed a new broadcast scheme, named Fibonacci minus One (Fib-1) Broadcasting and Receiving Scheme. This scheme addresses the issue of minimizing the bandwidth requirement with low latency (waiting time). Fib-1 allows the client to download data from N (a positive integer that can be selected) concurrent broadcasting channels, each with a bandwidth of $b/2$, where b (bits/sec) is the display (consumption) rate except the first two channels each with a bandwidth b. Based on the numerical computations we demonstrate that, for realistic sets of parameters, Fib-1 is the more efficient than the Staircase Broadcasting and Receiving Scheme, Fast Data Broadcasting Scheme and other known broadcasting schemes with client bandwidth limitation. Furthermore, it gives a VOD service provider great flexibility and simplicity in implementing VOD services based on the current technologies.

Chapter I

Introduction

Video-on-demand (VOD) provides subscribers the possibility of watching the video of their choice at the time of their choice, as if they were watching a rented video cassette. The reason for VOD not being a commercial success is due to the fact that the technology is still very expensive and accordingly the potential users are unwilling to pay much more for a VOD selection than they are used to pay for a rental video cassette.

The use of multicast or broadcast schemes has shown that the performance of a VOD system can be greatly improved. Broadcasting protocols for VOD aim at efficiently delivering "hot" videos, that are likely to be watched by many viewers. Instead of transmitting one separate data stream to each customer wanting to watch a given video, these schemes require repeated broadcast of the video over several data streams. In this manner no customer will have to wait more than a few minutes before being able to start watching the video.

Two factors can be identified which are critical to the success of VOD services. First, one can conservatively estimate that at least 80 % of all viewers will be ordering the same popular videos which can vary from 10 to 20 [1, 2]. Second, it is very doubtful whether these customers are willing to pay much more for VOD than they now pay for a rental video cassette or a pay-per-view program. Thus any reduction in the cost of distributing popular videos through the use of more efficient broadcasting protocols is certainly desirable. This have a direct impact on the overall cost of VOD and, may result in its successful diffusion among the potential users.

The most important performance index for a broadcasting protocol is the requirement of total bandwidth in order to achieve a given maximum waiting time. This has given impetus to research and the last fifteen years have seen the development of several new broadcasting protocols.

The notable ones are:

1. **Pyramid Broadcasting** protocol (due to Viswanathan and Imielinski [3, 4]),
2. **Permutation-based Pyramid Broadcasting** protocol (due to Aggarwal, Wolf and Yu [5]),
3. **Skyscraper Broadcasting** protocol (due to Hua and Sheu [6]),
4. **Dynamic Skyscraper Broadcasting** protocol (due to Eager and Vernon [7]),
5. **Harmonic Broadcasting** protocol (due to Juhn and Tseng [8]),
6. **Enhanced Harmonic Broadcasting** protocol (due to Juhn and Tseng [9]),
7. **Staircase Broadcasting** protocol (due to Juhn and Tseng [10]),
8. **Fast Broadcasting** protocol (due to Juhn and Tseng [11, 12]),
9. **Cautious Harmonic** and **Quasi-Harmonic Broadcasting** protocol (due to Paris, Carter and Long [13]),
10. **Poly-Harmonic Broadcasting** protocol (due to Paris, Carter and Long [14]),
11. **Pagoda Broadcasting** protocol (due to Paris, Carter and Long [15]),
12. **New Pagoda Broadcasting** protocol (due to Paris, Carter and Long [16]),
13. **Greedy Equal-Bandwidth Broadcasting** protocol (due to Hu, Nikolaidis and Beek [17]), etc.

The above mentioned protocols besides addressing their issues share a common objective of reducing the total bandwidth required to achieve a given maximum waiting time.

We have discussed and examined various broadcast schemes. A new broadcasting scheme with limited client bandwidth is proposed. Further, a comparative study shows the advantages of our new broadcasting protocol.

1.1 Motivation

Video-on-demand (VOD) is concerned with video services such that users can request any video program from a server at any time. VOD can address a variety of applications ranging from

i. Education,
ii. Distance learning,
iii. Entertainment such as movie-on-demand,
iv. Advertising,
v. Home Shopping,
vi. Interactive News.

Our desire to watch has fueled an industry eager to deliver a variety of VOD services. These services enable a subscriber to start the playback of a video of his/her choice at a press of a button. In a futuristic scenario of VOD service, movies are provided to subscribers over a high speed fiber-optic network and advances in networking technologies will contribute to the realization of the VOD service over the Metropolitan Area Network [18]. Video objects are very large even in a compressed form. Typically, one motion picture of 100 minutes duration with NTSC quality video, occupies 40 GB (Giga Bytes) of storage in uncompressed form and 1 GB when compressed according to the MPEG standard [19]. These video objects are available so that the client receives continuous fashion in order to avoid 'zitter'. A number of storage techniques, which ensure continuity of playback, have been considered [20]. To this end designs of a storage server capable of servicing a large number of simultaneous requests have been discussed [18]. These servers support the VOD service within the framework of the client-server paradigm.

In some proposed solutions, it is suggested to periodically broadcast most popular movies (with multiple requests coming in possibly over a short period of time) on the network. In this approach, 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 his/her choice is downloaded from within the broadcasted batch. The access time, defined as the maximum time the client has to wait for the selected movie, becomes independent of the number of clients. However this is not the case for the client-server approach. Thus, the broadcasting solution is appropriate for 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 generally observed that much of the demand (70% - 80%) is limited to few (10 to 20) very popular movies, e.g. the new releases and the top ten movies of the season/year [1, 2]. This explains why the 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 average). Broadcasting one of several techniques that aim at reducing the cost of VOD [21], is clearly not a panacea as it only applies to videos that are likely to be watched by many viewers.

Oracle, a potential VOD service provider has set up a three-tiered video server [30]. The most popular releases will always be loaded in the first tier, the main memory of the computer. Thousands of viewers will have quick access to the most used digital files. The second tier would be kept on 1000 or more hard disks inside the server, containing the next-most-popular movies. The third tier is reserved for lower-demand movies requested only occasionally. There would be a separate machine – a "video jukebox" with tens of thousands of 8-millimeter digital tapes, each containing a single movie. On the basis of request from a viewer for an archived title, a robotic arm would select and load the cassette into the video server's memory bank. According to Viswanathan and Imielinski [3], Frank Capra's Christmas classic of 1946, "It's a wonderful life," [22] is a good example of a movie that might rotate among all the three tiers. There will hardly be any demand for this movie for most part of the year. Therefore, it would be appropriate to store it in the archive on a 8-millimeter tape. A few weeks before Christmas, when the number of requests for it increases, it can be loaded onto hard drives. In case of the really heavy demand as on the Christmas Eve, it can be made part of the first tier.

As expected, most of the requests for the movies are during the prime time (say between 7pm to 11pm on week days and on week-ends). Accordingly, it is obvious to broadcast the movies of the first tier rather than to provide it by the client-server approach. The reason for this is that broadcasting approach scales up well. This way, even if the number of clients (hence the requests) increases 10 folds, all the clients have the same **waiting time** which is the maximum time a client must wait to get to the movie of his/her choice. The VOD service can be provided as follows:

(1) For the popular movies (movies in the first tier), the broadcasting approach can be used during prime time and the rest of the movies are dealt within the client-server approach. The decision as to which movies to broadcast and which ones to manage by the client-server approach is discussed by Imielinski and Viswanathan [23]. They also describe how to allocate optimally the bandwidth for the broadcasting and the client-server approaches.

(2) All other movies are dealt within the client-server framework. In our dissertation, we consider only the broadcasting approach during the prime time, when a few popular movies employ broadcast approach on the network whereas the rest use the client-server approach.

1.2 Broadcasting Protocols

Performance of a VOD system can be greatly improved through the use of multicast or broadcast schemes. Most of the multicast protocols [24, 25, 26] are reactive as they transmit data in response to the user requests. In contrast, broadcasting schemes periodically transmit video segments in a proactive way and thus guarantee service latency within certain time constraint.

One of the limitations of the broadcasting approach is that the access time for a movie can be very large. In the worst case the client has to wait through the entire broadcasted batch to get, say, movie of his/her choice. It is desirable to look for techniques that are simple to use, with a short access time and use the bandwidth efficiently.

This dissertation studies different methods form the perspective of understanding of the broadcasting videos. The idea behind periodic broadcasting schemes is to divide the video into a series of segments and broadcast each segment periodically on dedicated server channel. While user is playing the current video segment, it is guaranteed that the next segment is downloaded in time and can be displayed out in a continuous fashion. User will have to wait for the occurrence of the first segment before they can start playing the video. Therefore, the waiting time is usually the length of the broadcast period of the first segment. The focus of researcher has been to find out the mechanism which divides the video to achieve the lowest server bandwidth while still guaranteeing on time delivery of each segment with least waiting time.

The VOD environment will consist of clients, videos and network (bandwidth). Clients will be interested to view selected video objects in a continuous manner with shortest possible initial delay. It is pertinent to distinguish between the **data-centered** and **client-centered** approaches.

In the **client-centered** approach, a client eventually obtains some dedicated bandwidth. This can be achieved either by making available considerable bandwidth equal to the consumption rate of the video object times the number of clients, or providing less bandwidth for the clients so that to compete for by negotiating with a scheduler. The consumption rate of a video object is equal to the amount of bandwidth necessary to view it in a continuous manner. When a client makes a request to the server, the server sends the requested video object to the client via a dedicated (virtual) channel. Channel is used as a logical term to hide the physical details of the underlying network and can be regarded as an abstraction for any communication medium.

Another approach requires gathering of the requests for each video object over a period of time and multicast the object to the clients. Accordingly, multicast facility as available on modern communication networks can enable the users to share a server stream. For example, if two subscribers make a request for the same video separated by a small time interval, then by delaying the playback for the first request, the same server stream can be used to satisfy both requests [24]. In general, requests by multiple clients for the same video arriving within a short time duration can be batched together and served using a single stream. This is referred to as **batching** [24]. In this batch processing, batches of requests rather than individual requests are satisfied

We briefly discuss the functioning of the scheduled multicast. When a server channel becomes available, the server selects a batch to multicast according to some scheduling policy. In case of policy based on **Maximum Queue Length** (MQL), proposed by Dan, Sitaram, and Shahabuddin [24]. One selects the batch with the largest number of pending requests to serve first. The objective of this approach is to maximize the server throughput. Other scheduled multicast schemes are presented in [24, 26, 27, 28]. These, again, are special case of the user-centered approach.

In the **data-centered** approach, bandwidth is dedicated to video objects rather than to clients such that e ach video object is allocated some bandwidth on a logical channel. This bandwidth is meant for periodic broadcasting of the video object. When a client wants to view a video object, he/she just tunes the channel dedicated to broadcast the video object. In this case all users have a cache, which stores the proper directory with information about video objects, channels, and times. Requests are not sent to the server. The data-centered solution is implemented in the **asynchronous transfer mode** (ATM), using broadcast or multicast facilities and thus, creating "object-based" connections. As noted earlier, the data-centered approach scales much better with the number of clients in contrast to the client-centered approach. The main reason is that we can take advantage of the repetitive requests for the same video object and broadcast it periodically. The bandwidth requirement in the case of client-centered approach is proportional to the number of clients, where as in the data-centered approach, it is proportional to the number of video objects. Thus, with increasing number of clients the client-centered approach will not result in 'scaling'. It is easy to see that for a fixed bandwidth of the channel, the access time for the data-centered approach is independent of the number of clients. This is different in the client-server approach in which the access time for a video object grows with the number of clients. One can establish a cutoff point in terms of the number of clients where the client-centered approach will become inferior to the data- centered approach. For the shake of clarity we consider MAN where the number of clients are very large. In this case the broadcasting approach is better suited to this environment than the client-server approach.

Traditionally, video objects are displayed in a continuous manner with no division of the objects, pipelining or multiplexing. For example, the HBO (Home Box Office) channel broadcasts in its entirety the same set of movies a number of time. No other movie is broadcast till the end of transmission of the movie under consideration. In the broadcasting approach a number of video objects are simultaneously broadcast on the network. Video objects may be broadcast continuously (one after another) or multiplexed (interleaved). Thus the client has to wait until the video object is to be broadcasted. This delay, called waiting time (or access time), is a major parameter of interest which needs to be optimized. The bandwidth required to broadcast the video object is another major metric which is useful for comparing different broadcasting protocols.

As described by Gelman, Kobrinski, Smoot, and Weinstein [29], there are two kinds of VOD services: **true video-on-demand** (TVOD) service, and **near video-on-demand** (NVOD) service. In the former all control functions are provided whereas in the later some control functions are not provided. A true VOD service does not require the user to wait for the video. Since the user cannot watch the video immediately, almost all broadcasting protocols attempt to provide the NVOD service.

In this dissertation, we will focus on the broadcasting schemes for NVOD services.

1.3 Outline of the Dissertation

The remaining part of this dissertation is organized as follows:

Chapter II provides the basic idea of video on demand service, After introduction of basic notations and conventional broadcasting schemes, it also gives the tools used for analysis of the various broadcasting protocols and classifies the broadcasting schemes based on Segment Size and Bandwidth allocated to each segments. Sections 2.5, 2.6 and 2.7 discuss the existing broadcasting protocols as per the classification contained in the section 2.4.

In Chapter III we discuss the proposed scheme and analyze its performance in relation to other broadcasting schemes.

Chapter IV ends with the concluding remarks highlighting some problems which can form part of future enquiry.

Chapter II

Review and Systematic Evaluation of Previous Work

This chapter reviews the existing broadcasting protocols. In Section 2.1 we provide basic notations for studying the broadcasting protocol literature. In Section 2.2 we introduce the conventional broadcasting schemes. In Section 2.3 we provide a generalized analytical approach to studying the VOD broadcasting protocols. In section 2.4 we classify all the broadcasting protocols on the basis of the bandwidth and the segment size. In Section 2.5, we present the previous works in the ESEB broadcasting protocols. Section 2.6 discusses the major broadcasting protocols in the USEB category. In Section 2.7 we present the related research in the broadcasting protocols in the ESUB category.

2.1 Basic Notation

Consider a VOD service to be rendered to a large number of clients on a high speed fiber-optic network. We consider a network (with broadcast capability) which serves a large population of clients 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 at the predefined consumption rate.

Throughout the paper we use the following notations:

b	the video consumption (or display) rate (Mbits/sec);
l	the total size of the video (in Mbits);
S_i	the ith segment;
l_i	the size of S_i (in Mbits);
D	the total length of the video (in seconds);
D_i	the length of S_i (in seconds); such that $D_i = l_i/b$ and $D = l/b$.
W	the client's relative waiting time (i.e. = access time); denotes the upper bound of the ratio over D of the time experienced by the client from the moment the client requested the video until he/she starts viewing it.
M	the number of videos to be broadcasted;
B	the total bandwidth required to broadcast M given videos expressed as a ratio over b; B_i denotes the bandwidth of the ith broadcasting channel as a ratio over b; B and $\{B_i\}$ are dimensionless quantities; In this thesis, the bandwidth will be expressed as either Bb or simply B;
n	the number of segments in a given video;
N	the number of logical channels; the total bandwidth B is divided into N logical channels.

2.2 Conventional Broadcasting Schemes

We describe the following basic types of conventional broadcasting schemes [20, 26]:

2.2.1 **A conventional broadcasting method:** The M video objects are broadcast in their entirety on a single channel. When the last video is broadcast, the cycle is repeated. Fig. 2.1 depicts the situation when total number of movies $M = 10$, length of each movie $D = 120$ minutes and the bandwidth allocated to broadcast all videos on a single channel $B = b$. Thus the maximum waiting time for any video is $W = (M*D)$ minutes = 1200 minutes. In this scheme, there is no need of any buffer for storage purposes, and the bandwidth requirement is minimum.

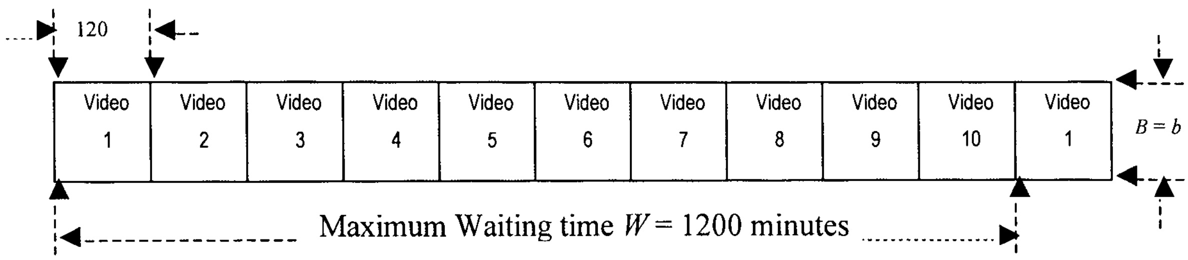

Fig.2.1 A Conventional Broadcasting Method (Adapted from [20])

2.2.2 In this scheme each video is broadcasted periodically on a separate video channel in their entirety. For the purpose of illustration we consider an example. Let the total number of videos M be 10 and the length of each video be 120 minutes. Thus, the total bandwidth required for broadcasting all videos will be $B=10*b$, and the maximum waiting time for any video will be 120 minutes. In this scheme also, there is no need of any buffer, but the waiting time is high as well as the bandwidth requirement is also high.

2.2.3 **Staggered Broadcasting Scheme:** This scheme known as Staggered Broadcasting Scheme [26] is illustrated in Figure 2.2. The physical channel is divided into B logical channels. There are *D/d* logical channels for each video, and a replica of each video is broadcast on a separate logical channel with a phase delay of *d* minutes. On each logical channel, one video is broadcasted periodically at its consumption rate, in its entirety. In this way, any client can start viewing the video in at most *d* minutes (by tuning in to one of the *D/d* logical channels). Each video has a bandwidth of *D/d* (times the consumption rate of a video) allocated to it. So the total required bandwidth is $B = M*D/d$. Any client requesting a video can tune in to the appropriate channel and begin viewing the video in not more than $d = M*D/B$ minutes. Thus the maximum relative waiting time *W* must be *M/B*.

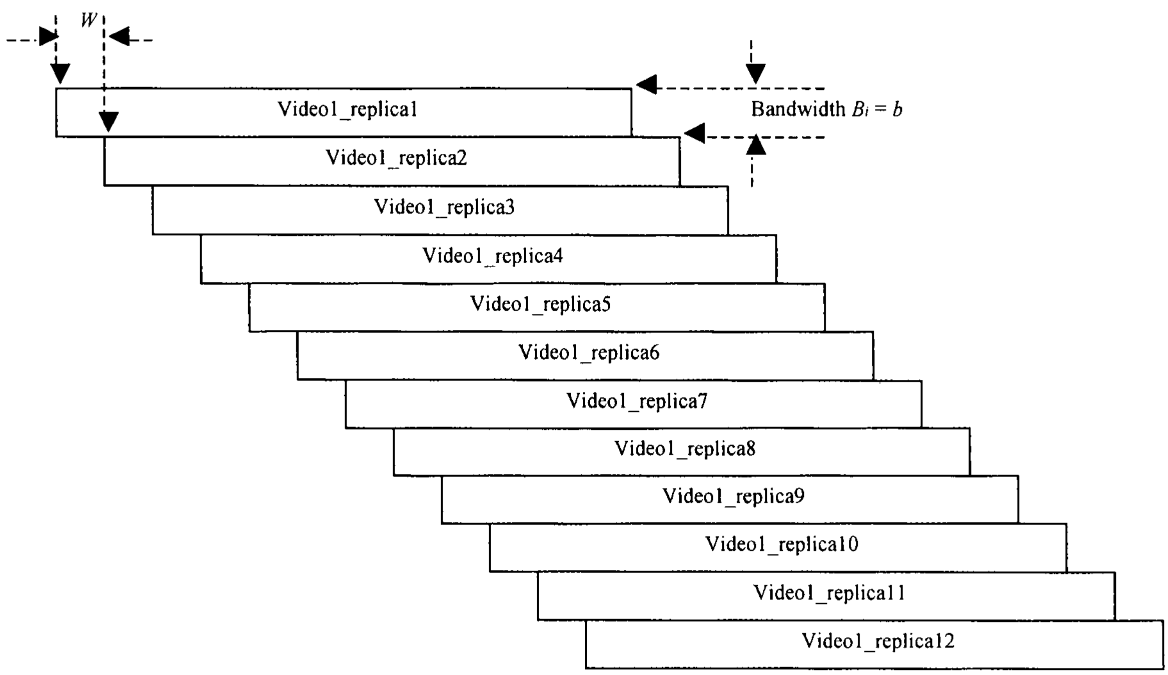

Figure 2.2: Another conventional broadcasting method: staggered broadcasting (Adapted from [26])

2.3 General Analysis Tool

In order to analyze the efficiency of the broadcasting protocols, we observe that there are three very important aspects which we have to take into account: segment size progression, bandwidth allocation for each logical channel (data stream to transmit segments), and the display continuity condition [8, 10].

To visualize these aspects, we introduce a **time-bandwidth map**, whose horizontal axis represents time and the vertical axis represents bandwidth. See Figure 2.3 as an illustration. On the lower part of the map is the **download (broadcasting) plane**. Each logical channel is represented as a horizontal band and its bandwidth is represented by the band's height. These logical channels are piled up on top of each other and the sum of their heights is equal to the server bandwidth. The upper part of the map is the **display plane**. Video segment S_i in this plane corresponds to the broadcast segment S_i in the download plane.

A major design goal of any broadcasting protocol is to maximize the server's bandwidth efficiency, in other words, minimization of the sum of the bandwidths of all logical channels (i.e. the height of the download plane), is done subject to the waiting time (access time) requirement and client's I/O bandwidth and storage requirement.

Before analyzing existing broadcasting protocols, we digress for a moment to show the advantage of separating different videos. In **Pyramid Broadcasting** [3], parts of all videos are broadcasted sequentially in each logical channel. Each channel has bandwidth B/N and contains segments from M videos. Clients have to download video data at the rate of B/N and only $1/M$ of the channel cycle is used for the video they want to view. A slight change to the protocol could be made to reduce the bandwidth: separate the M video segments in one logical channel into M logical channels and let each segment transmit at a lower rate with bandwidth $B/(N*M)$. This slower transmission scheme is adopted in all of the later protocols and contributes to their improved performance.

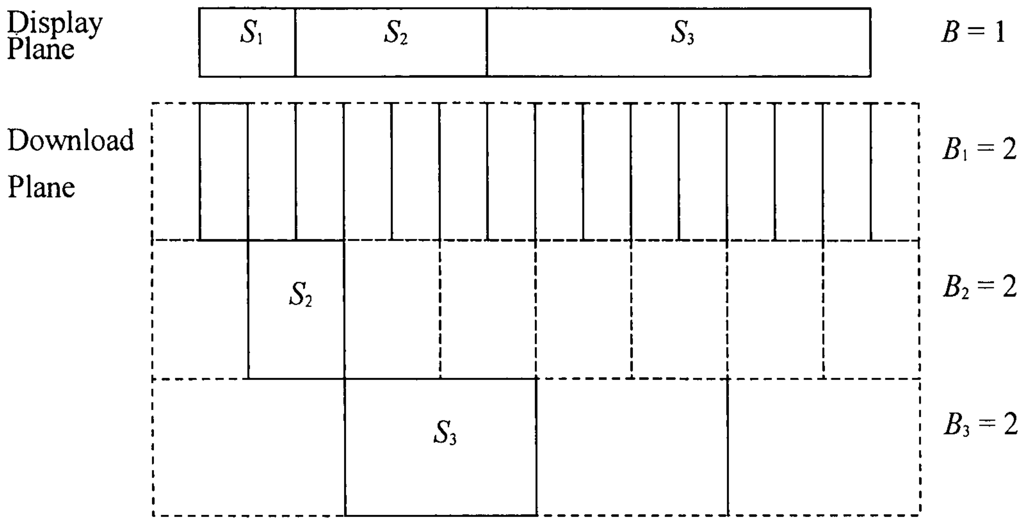

Figure 2.3: Time-bandwidth map

It may be pointed out that we consider each video independently. Each of them is broadcast in its own channels. To mix them with other videos and transmit them serially can result in higher *client* I/O bandwidth and hence higher client storage requirement. Thus, we will consider only one video in all the broadcasting schemes.

After studying all the known broadcasting protocols, we find that the client storage requirement and disk I/O bandwidth are not very crucial with today's technology. The disk storage size and I/O bandwidth are more than enough to satisfy the VOD broadcasting requirements. So we will not discuss the client storage requirement and disk I/O bandwidth requirement in this dissertation.

We will only consider minimizing the broadcasting bandwidth requirement for a given access time. This problem can also be stated differently. Given a certain broadcasting bandwidth, we wish to minimize the client waiting time (access time).

2.4 Classification of Broadcasting Schemes

Most recent research activities in broadcasting schemes for Video on Demand fall into one of the following categories:

- **Equal Segment Size Equal Bandwidth (ESEB)**
- **Unequal Segment Size Equal Bandwidth (USEB)**
- **Equal Segment Size Unequal Bandwidth (ESUB)**
- **Unequal Segment Size Unequal Bandwidth (ESEB)**

These categories are based on the two principles; one is the size of each segment of the video and the other one is the bandwidth allocated to each segment for broadcasting. If the size of any two or more segments are different than it will fall in the Unequal Segment Size category otherwise in the Equal Segment Size category. If the bandwidth allocated to two or more channels are different than it will fall in the Unequal Bandwidth category otherwise in the Equal Bandwidth category. In the next few sections we will review the broadcasting protocols of these categories.

2.5 Equal Segment Size Equal Bandwidth (ESEB) Broadcasting Schemes

This category of broadcasting schemes contains all those schemes in which the video is divided into equal size segments and the bandwidth allocated to each logical channel is also equal. Juhn and Tseng [10, 11] proposed **Fast Broadcasting** in 1997 and 1998.

2.5.1 Fast Broadcasting Scheme:

FB is very similar to Staircase Broadcasting, which was earlier proposed by Juhn and Tseng in 1997 [9], except that FB allows each broadcasting channel bandwidth to be equal to or greater than the consumption rate *b*, and the arrangement of the segments is different. FB modified the Staircase Broadcasting client end receiving algorithm to make it feasible. Figure 2.4 is an illustration of Fast Broadcasting Scheme.

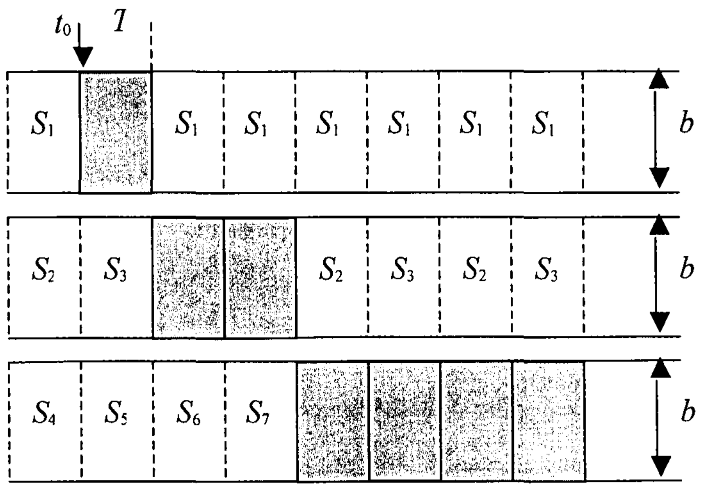

Figure 2.4: Fast Broadcasting Scheme (Adapted from [10,11])

Various steps involved in operation of FB scheme are outlined:

1. The total bandwidth B is divided into K logical channels. B is chosen such that $B = b * K$, b is the consumption rate.

2. At the server end, each video is equally divided into N segments, where

$$N = \sum_{i=0}^{K-1} 2^i = 2^K - 1$$

3. For every $i = 1, 2, \ldots, N$, put 2^{i-1} continuous data segments $\{S_p, \ldots\ldots, S_q\}$ on the ith channel, where $p = 2^{i-1}$, and $q = 2*p - 1 = 2^i - 1$. On this channel, the 2^{i-1} data segments are broadcasted periodically as shown in Figure 2.4.

4. At the client end, begin to download the first data segment (S_1) of the desired video at the first occurrence from the first channel and download other related data segments from channels 2^{nd}, 3^{rd}, ……, concurrently.

5. Right after the client begins to download the data segments, he/she can start to consume the video in the order $S_1, S_2, \ldots, S_N$.

6. Stop downloading after receiving 2^{i-1} data segments from the ith channel.

2.6 Unequal Segment Size Equal Bandwidth (USEB) Broadcasting Schemes

This category of broadcasting schemes contains all those schemes in which the video is divided into unequal size of segments and the equal bandwidth is allocated to each logical channel. Viswanathan and Imielinski proposed a Pyramid Broadcasting for Video on Demand Service [3] in 1995 and Metropolitan Area Video on Demand Service using Pyramid Broadcasting [28] in 1996. In the same year Aggarwal, Wolf and Philip proposed a permutation based pyramid broadcasting scheme for video on demand systems [4].

2.6.1 Pyramid Broadcasting Scheme:

Pyramid Broadcasting scheme [3], proposed in 1995 is a broadcasting protocol which aimes at reducing the access time and the broadcasting bandwidth requirements. We closely follow the work due to Viswanathan and Imielinski [3]. Pyramid Broadcasting multiplexes the videos on the channels in such a way that the clients can start consuming the videos early. Such a situation is realized by breaking up each video into segments of increasing sizes and broadcasting the smallest segment most frequently. The frequency of broadcasting a segment is made to decrease with the increase in its size. The segments are broadcast in such a way that, once a segment has been displayed, the next segment is ready to be displayed, such that one can view the video in a continuous manner. For example, while the first segment is being consumed, the second segment is collected. This process of collecting future segments continues till the whole video is collected. In this pipelining approach, the time to access the video becomes the time to access the first segment. Since the first segment is the smallest, and it is also broadcast most frequently, the access time for the first segment is fairly small. This idea results in a substantial improvement in the access time of the videos, while assuring that the clients can consume the videos continuously (with no disruptions). This is achieved by using storage at the receiving end.

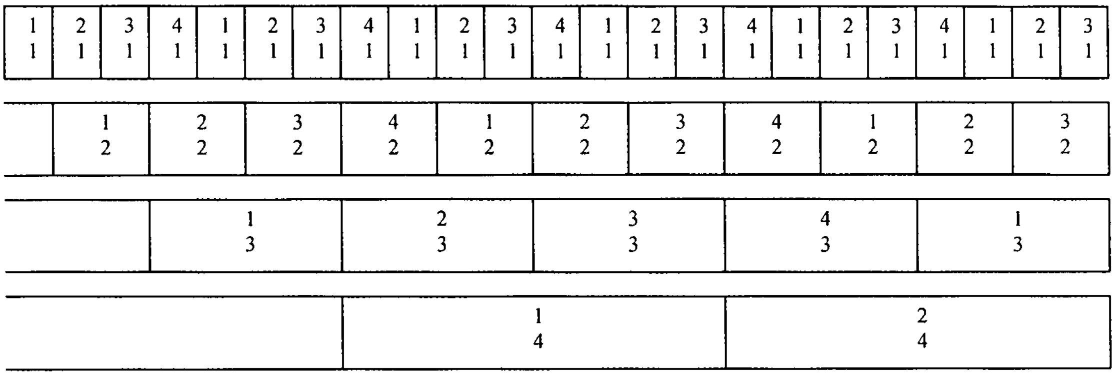

Figure2.5: Pyramid Broadcasting Scheme (adapted from [3])

Figure 2.5 is an illustration of the PB Scheme which specifies the organization of the videos on the logical channels by the server and the access protocol for the clients.

On the server side, broadcasting involves the following steps:

1. The physical channel of bandwidth B is divided into N logical channels, each of bandwidth B/N.

2. Each video of size l is divided into n segments. The data segments are of sizes l_i, $i = 1,\ldots..,N$, where $l = l_1 + l_2 + \ldots + l_N$. The concatenation of all the segments, in order of increasing segment numbers, gives the entire video.

3. The data segments of size l_i of all the M videos are broadcast together on channel i periodically. Within each channel, the first conventional broadcasting method (Section 2.2) is used.

On the client side, the following steps are involved:

1. Begin downloading the first data segment l_1 of the required video at its first occurrence, and start consuming it concurrently.

2.6.2 Permutation-Based Pyramid Broadcasting:

PB guarantees a short waiting time for every video. Therefore, there is a limitation of this scheme such that buffer size which is usually more than 70% of the size of the video must be used at the client end. As a result, it is necessary to use disks to do the buffering. Furthermore, since a very high transmission rate is used to transmit each video segment, an extremely high disk bandwidth is required. To address these issues in conjunction with PB, **Permutation-based Pyramid Broadcasting** (PPB) was introduced in 1996 [4]. PPB has similar features to PB, but the storage and disk transfer rate requirements are each substantially reduced. It was also claimed that the waiting time (access time) of PPB is also substantially reduced, compared to PB.

Based on PB, it is found that PPB further divides each channel into p sub-channels, $p \geq 1$ is an integer and this scheme reduces to PB for $p = 1$.

PPB divides each of the N original channels into $M*p$ sub-channels, each of which has bandwidth $B/(N*M*p)$. Now each segment of each video has p copies, and each copy is broadcast periodically on one of the sub-channels. Thus, these p sub-channels will broadcast identical bit streams. But the bit streams on adjacent sub-channels (modulo p) will have a phase difference of $1/p$. This is similar in concept to the third conventional broadcasting scheme viz. Staggered Broadcasting [4, 26].

On the client side, when the client requests a video, the set-top box at the client end latches onto the beginning of any copy of the first segment (S_1) of the video. As soon as it is transmitted, it starts to consume this first segment. In general, the client latches onto the first copy of segment S_{i+1} which begins after the transmission of segment S_i is completed. So there is no pipelining in PPB. This is another difference between PPB and PB. Thus, there is some latency between the end of reception of one segment and the beginning of the reception of the next segment. During this latency, we have to guarantee that there is no interruption in the display. The idea is to use previously saved portions of the bit stream from a buffer at the client end. The buffer builds up because during times of uninterrupted transmission of a video segment, the rate of transmission is greater than that of consumption. The buffer is used to "bridge" the waiting time (latency) gap before the transmission of the next segment, thus avoiding zitter [4].

In order to prevent zitter, segment S_{i+1} must be accessed before the end of consumption of segment S_i for all $1 \le i < N$. This implies that the sum of the transmission time of the current segment and the maximum access time for the next segment must be less than or equal to the consumption time of the current segment.

Note that the *i*th segment of a video is transmitted on p different sub-channels, which are uniformly spaced with a phase difference of $1/p$ from each other. Thus the maximum access time of a segment is simply $1/p$ times the transmission time of the segment. A viewer who just misses latching onto a segment can latch onto the identical segment lagging behind the missed segment by a phase difference of $1/p$ [4].

The key reason behind this is that the continuity conditions for two broadcasting schemes are different. For PB, the continuity condition is that the current segment display time be greater or equal to the next segment download time, which means that the next segments are already totally in the client's local disk before the client starts to consume it. Whereas, in the PPB, the continuity condition is that the current segment display time is greater than or equal to (should cover) the current segment download time plus $1/p$ of the next segment download time. This means that the download rate of any segment is faster than the display rate. In order to accumulate enough data to ensure the continuous display of the video, PPB actually requires more bandwidth than PB, for any given access time [3, 4].

2.6.3 Skyscraper Broadcasting:

It is known that PPB reduces both disk space and I/O bandwidth requirements at the client end. The disk size, however, is still quite significant due to the exponential nature of the data fragmentation scheme [4]. The size of successive segments increases exponentially, causing the size of the last segment to be very large (typically more than 50% of the video) [5]. Since the buffer sizes are determined by the size of the largest segment, using the data fragmentation method proposed for PB limits the savings that can be achieved by PPB. To substantially reduce the disk costs, and avoid the difficult synchronization problem in the implementation of PPB, **Skyscraper Broadcasting** (SkyB) was proposed in 1997 [5].

Instead of fragmenting the video according to a geometric series, 1, α, α^2, α^3, ..., as in PB and PPB, a series generated by the following recursive function is used in SB:

$$f(n) = \begin{cases} 1 & n = 1, \\ 2 & n = 2,3, \\ 2f(n-1)+1 & n \bmod e4 = 0, \\ f(n-1) & n \bmod e4 = 1, \\ 2f(n-1)+2 & n \bmod e4 = 2, \\ f(n-1) & n \bmod e4 = 3, \end{cases}$$

here n $mode4$ is **n modulo 4**.

The generated series is 1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52, ...

In this way, the segment size progression is much slower than that of PB and PPB, which means the storage requirements at the client end will be greatly reduced. Each segment of a video is broadcasted on a channel of bandwidth b (the consumption rate). The access latency is the maximum access time to the first segment S_1.

2.7 Equal Segment Size Unequal Bandwidth (ESUB) Broadcasting Schemes

This category of broadcasting schemes contains all those schemes in which the video is divided into equal size of segments and the bandwidth allocated to each logical channel is unequal to each other. Juhn and Tseng proposed **Harmonic Broadcasting** (HB) in 1997 [7] and **Enhanced Harmonic Broadcasting** (EHB) in 1998 [8]. In order to solve the continuity problem with Harmonic Broadcasting, Paris, Carter, and Long proposed **Cautious Harmonic Broadcasting** (CHB) and **Quasi-Harmonic Broadcasting** (QHB) in 1998 [12]. Paris, Carter, and Long proposed another broadcasting scheme named **Poly-Harmonic Broadcasting** (PHB) in 1998 [13]. In 1997, Juhn and Tseng proposed **Staircase Broadcasting** [9].

2.7.1 Harmonic Broadcasting:

We closely follow the work due to Juhn and Tseng. They proposed **Harmonic Broadcasting** (HB) in 1997 [7], which greatly reduces the bandwidth requirement for a given waiting time, compared to the Pyramid- based broadcasting schemes.

HB involves the following steps:

1. The video is equally divided into n (=N) segments. The concatenation of all the segments, in the increasing segment orders, constitutes the whole video.

2. The ith segment of video, S_i, is equally divided into i sub-segment(s). Let the i sub-segments of S_i be put on logical channel B_i ($1 \leq i \leq N$). The bandwidth of B_i is $1/i$. Within B_i, the i sub-segments of S_i are broadcasted periodically as shown in Figure 2.6.

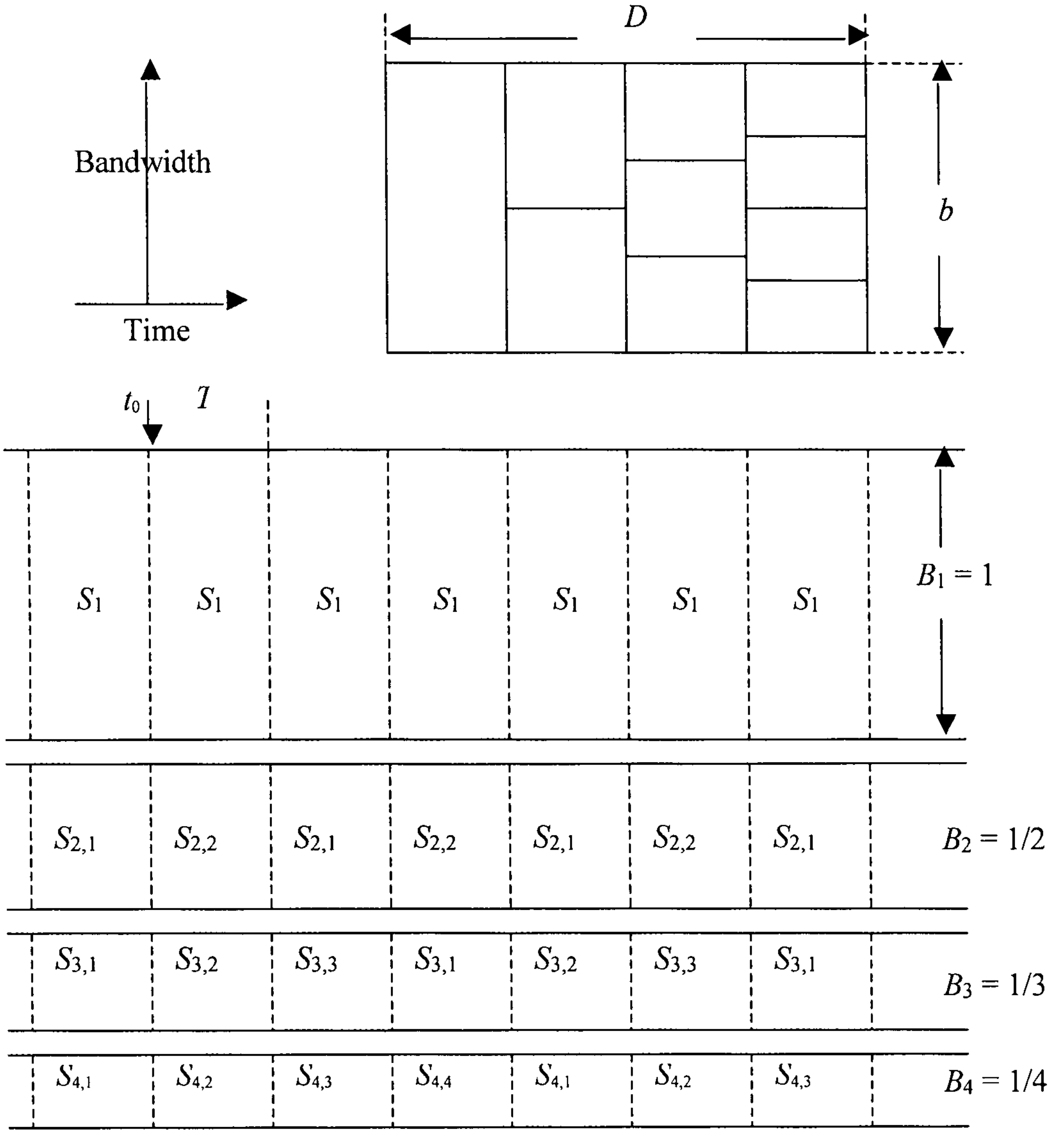

Figure 2.6: Harmonic Broadcasting Scheme(Adapted from [7])

But there is some problem with HB. It does not always work, since Harmonic Broadcasting sometimes fails to deliver all frames on-time. For example, if a request is started at t_0 in Figure 2.6, the first segment S_1 is completely consumed at T. From T to $2T$, the second segment S_2 is supposed to be consumed. But segment $S_{2,1}$ will not finish downloading until time $2T$, although at this point, the whole S_2 is supposed to have been consumed. These problems are resolved in another Harmonic Broadcasting schemes.

2.7.2 Staircase Broadcasting:

In 1997, Juhn and Tseng proposed **Staircase Broadcasting** [9]. Staircase Broadcasting is based on an idea similar to PB. It is very similar to the case where $\alpha = 2$ and $B_i = 1$ in PB. But it doesn't have the same continuity constraints as the original PB.

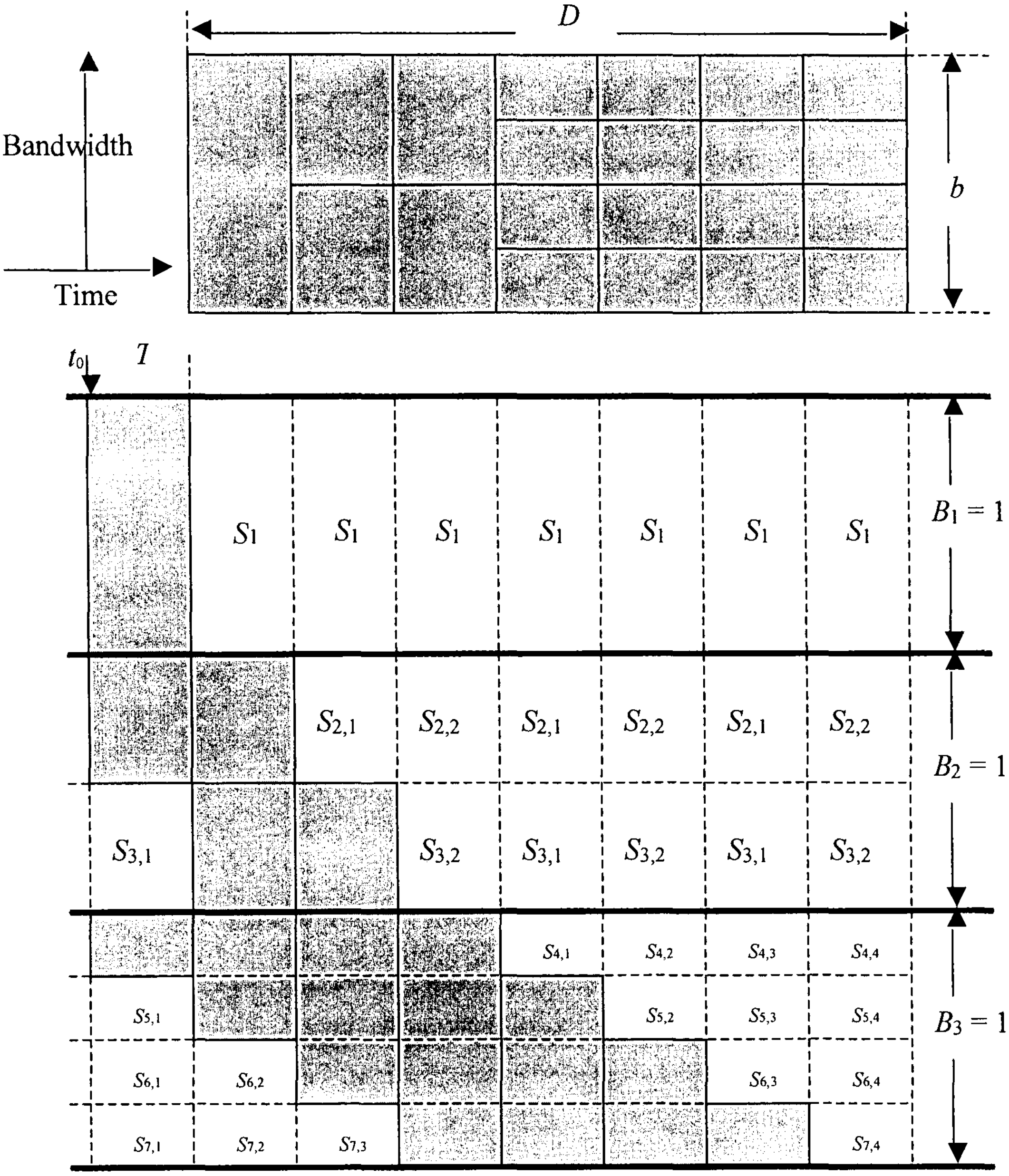

Figure 2.7: Staircase Broadcasting Scheme (Adapted from [9])

advance; this increases the storage requirement at the client. But it is found that Staircase Broadcasting as proposed in the original paper is not correct. It needs some changes in the downloading times by the client, before it could actually work. As a result, the disk space savings claimed by the original paper are also not correct.

For example, in Figure 2.7, according to the original Staircase Broadcasting scheme, if a request starts at t_0, by $2T$, segment S_2 is consumed. From $2T$ to $3T$, the segment S_3 is supposed to be consumed. But $S_{3,1}$ will not finish downloading until time $3T$, while at this point, the whole segment of S_3 should have been consumed. If we change the client's receiving time by, for example, downloading all the segments from the beginning of the consumption of the first segment, then there will be no problem with the continuity in displaying the video.

Chapter III

Unequal Segment Size Unequal Bandwidth Broadcasting Scheme

This category of broadcasting schemes contains the schemes in which the video is divided into unequal size of segments and the bandwidth allocated to each logical channel is also different to each other. This chapter discusses a new scheme, the **proposed** scheme **'Fibonacci minus One (Fib-1) Data Broadcasting and Receiving Scheme'** [31]. Section 3.1 explains the Fib-1 Scheme in details including the segmentation of the video, channel allocation and the receiving of the segments of the video. In section 3.2, the results have been simulated and compared with the other broadcasting schemes. This comparison will show the efficacy of the new scheme proposed in the dissertation.

3.1 Fibonacci minus One (Fib-1) Data Broadcasting and Receiving Scheme

In this section, a new broadcasting protocol is proposed, as follows. Let the length of a video be L minutes and the consumption rate of the video be b. The size S of the video is, thus, $S = L * b$. We want the viewers' waiting time reduced to

$$W_t = L/N_{sb}, \qquad N_{sb} = f_1 + f_2 + \sum_{n=3}^{N}(f_n - 1), \tag{1}$$

where N and N_{sb} are positive integers and $f_1=1$, $f_2=2$. Here N is the number of segments for this scheme i.e. unequal in size, and N_{sb} represents the number of segments of equal-size, equivalent to N.

The Fib-1 broadcasting scheme involves the following steps:

1. The whole video is divided into N number of segments. These segments are of unequal size. The sizes of these segments are kept according to the Fibonacci minus One (Fib-1) series. The sequence so generated is as follows: 1,2,2,3,4,6,9,14,22,35,56,90,155,... The recursive Fib-1 sequence is given by

$$f_n = f_{n-1} + f_{n-2} - 1, \text{ and } (f_1 = 1, f_2 = 2). \tag{2}$$

2. Let S_n be the nth segment of the video. The complete video will be the concatenation $(\bullet)$ of all the segments, in the order of increasing segment number.

$$S = S_1 \bullet S_2 \bullet S_3 \bullet \cdots \bullet S_N \tag{3}$$

3. The nth segment (where $n > 2$) of the video is again sub-divided into i equal-size sub-segments $(S_{n,1}, S_{n,2}, \cdots, S_{n,i})$, where i is determined according to function $2*f_n$. The second segment of the video is sub divided into two sub-segments only. The first segment is not sub-divided at all.

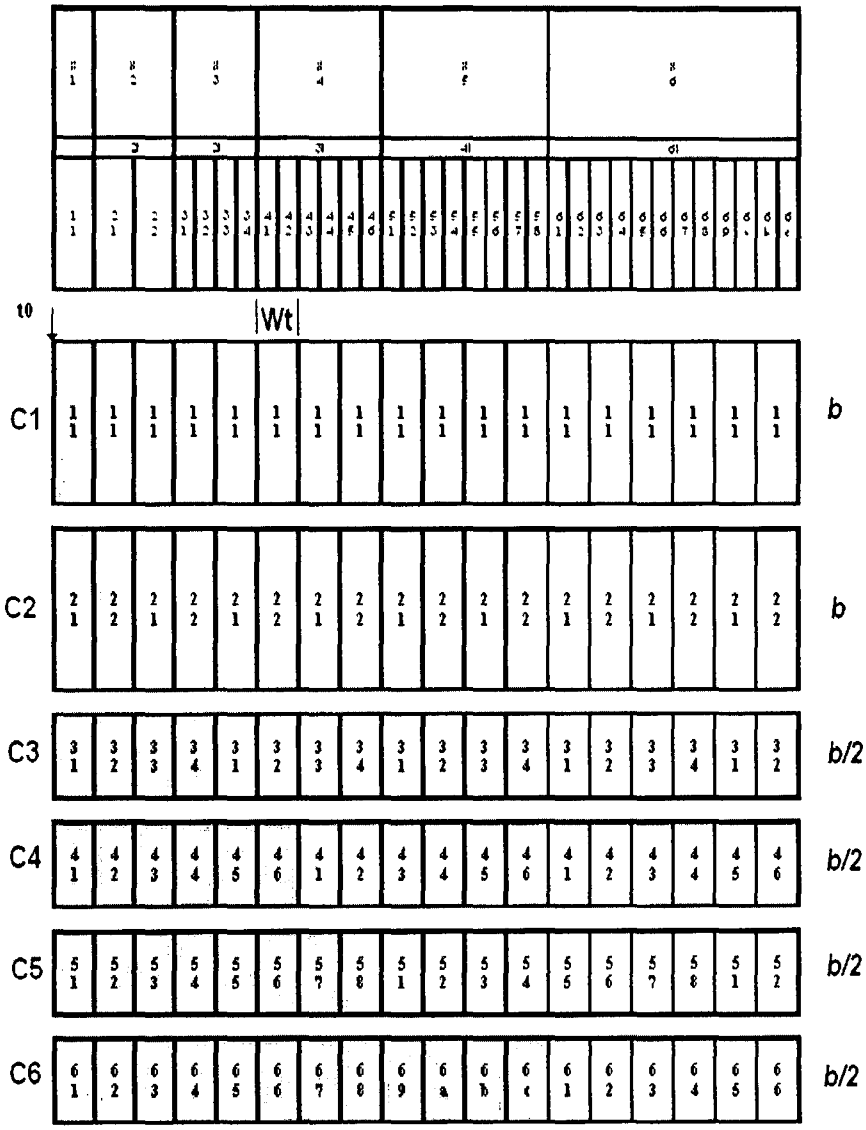

Figure 3.1: Fib-1 Broadcasting Scheme

4. The logical channel C_n broadcasts each of the nth segment(s)/sub-segment(s) sequentially. The first two channels C_n $(n \leq 2)$ have the bandwidth equal to the consumption rate *b* and the bandwidth of all other channels C_n (n > 2) is kept at half of the consumption rate i.e. *b/2*.

5. On each logical channel C_n, the nth sub-segment(s) of S_n are broadcast periodically as shown in Fig.3.1.

Total bandwidth B that is allocated to the video/movie is, thus,

$$B = F(N) * b, \tag{4}$$

where

$$F(N) = \begin{cases} 1, & N = 1, \\ 2 + \dfrac{N-2}{2}, & N \geq 2, \end{cases} \tag{5}$$

where N is the total number of the segments and b is the consumption rate of the video/movie.

At the client side, enough buffer is available to store the parts of the video/movie. For watching a video, the Fib-1 data receiving scheme involves the following steps.

1. Begin to download the first segment of the video from the channel C_1 and consume it with its normal playing speed i.e. b concurrently. At the same time, receive the data segment(s)/sub-segments(s) from other video channels C_n (n > 1) and store them in the buffer.

2. While downloading sub-segments from the C_2 channel, do not store the sub-segments in the buffer if next sub-segments are in order.

3. Only those sub-segments of the video are buffered in the storage disk, which are to be played next and do not exist in the buffer.

4. Play the sub-segments in the increasing number of the sub-segment number. Consume the next segment with its normal speed. Stop downloading the segments/sub-segments from the channels; if all required sub-segments are received form the channels.

3.2 Performance Analysis of the Proposed Scheme

In this section, the performance evaluation of the proposed scheme is done on the following main characteristics of the video broadcasting schemes i.e. waiting time, bandwidth requirements, buffer requirements, and I/O transfer rate of the disk used for buffer.

3.2.1 Bandwidth Allocation vs. Waiting Time

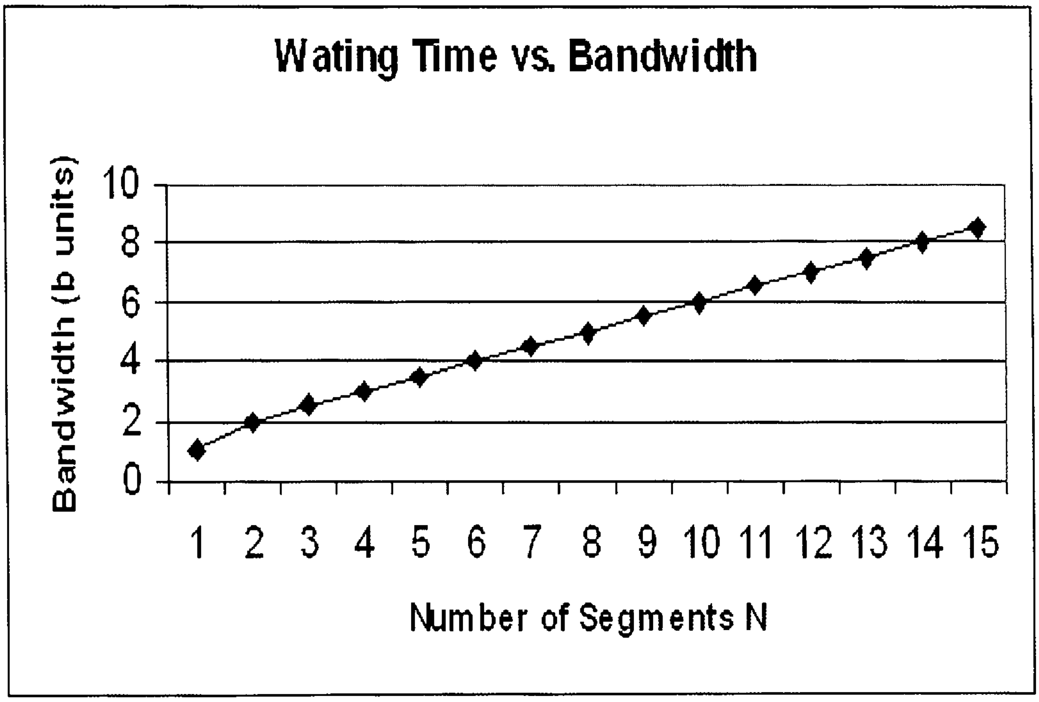

Figure 3.2: Bandwidth vs. Waiting Time

With the help of equation (4), (5), one can easily obtain the relationship between the viewer's waiting time W_t and the bandwidth required to be allocated to the video i.e. *B*. Fig.3.2 shows the relationship between *B* and *N*. For example, If a movie of length 120 minutes is broadcast using this scheme and if we can allocate 7 video channels to it, then we have N=12, and N_{sb}=244. We get a waiting time of 120/244=0.4918 minutes. However, in case of Staircase Broadcasting (SB) Scheme [see 9], if the same movie is broadcast on 7 video channels, then $N_{sb}=2^7-1=127$, and waiting time is 120/127=0.9448 minutes. The Fast Data Broadcasting (FDB) Scheme [see 11] also gives the same waiting time for K=β=7and N_{sb}=127 as SB scheme (i.e. 120/127=0.9448 minutes).

3.2.2 Storage Requirements at the Client Side

At the client end, the proposed scheme needs the storage disk to buffer the portions of the broadcasted video/movie. Let the time to begin to download the segment S_1 from the channel C_1 be t_0. During $\{t_0 + (i-1)W_t,\ t_0 + i * W_t\}$, the sub-segments that are received from the channels $C_2,\ldots,C_N$ need to be buffered. There can be two instances of the received sub-segments from the channel C_2; at one instance, there is no need to store any of the sub-segments and at another instance, both sub-segments are required to be stored. So the second instance is taken into consideration to calculate maximum storage required for the video/movie. Let I_i be the received portions of the video to buffer for $N>1$. Thus,

$$I_1 = I_2 = \frac{N}{2} W_t, \tag{6}$$

$$I_3 = I_4 = \left(\frac{N}{2} - 1\right) W_t, \tag{7}$$

$$I_i = \left[\frac{N}{2} - \left(1 + \frac{x_{\min}}{2}\right)\right] W_t, \text{ for } i > 4, \tag{8}$$

where x_{min} satisfies the following inequality for a given value of $i > 4$,

$$2\sum_{x=1}^{x_{\min}} f_x \geq (i-4), \qquad x=1,2,3,\ldots \tag{9}$$

where f_x is the x^{th} fibonacci number.

Only those sub-segments of the video are buffered in the storage disk, which are to be played next and do not exist in the buffer. If the received sub-segments already exist in the storage disk, then these sub-segments are not buffered in the storage disk. During the same interval, the previously buffered sub-segment(s) of the segment S_i will be consumed.

Let O_i be the consumed portions of the video from the buffer, which will be deleted from the buffer. Thus,

$$O_i = W_t \tag{10}$$

Let D_i be the storage size requirements at time $t_0 + i * W_t$. At $t_0 + W_t$, all portions of the video that come from the channels $C_2,...,C_N$ are required to be buffered. Hence,

$$D_1 = I_1 \text{, and } D_i = D_{i-1} + I_i - O_i . \tag{11}$$

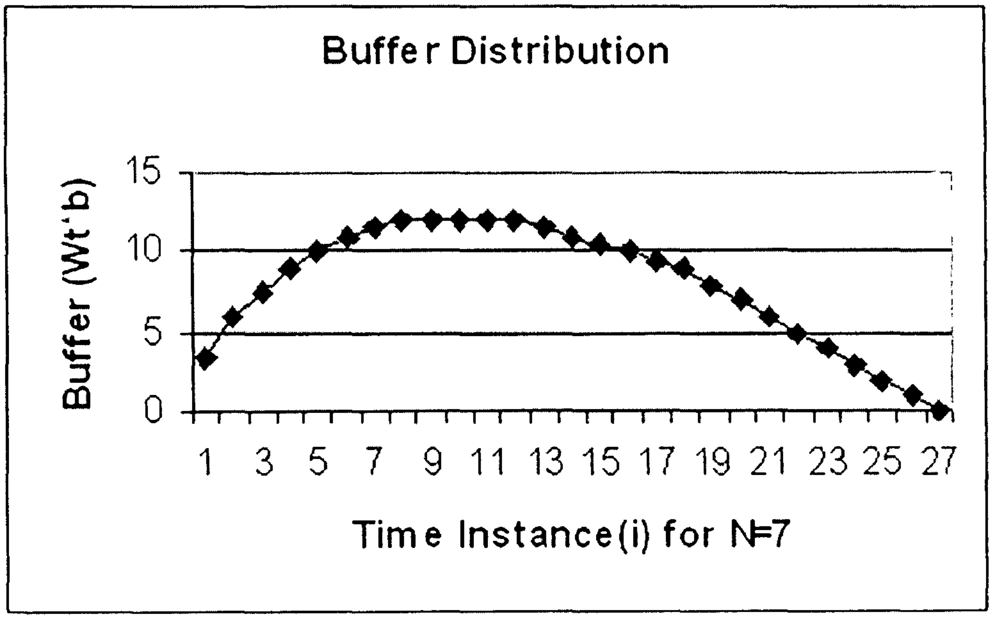

Figure3.3: Buffer Distribution at each instant of time

For the calculation of D_i, the maximum value of D_i gives the maximum storage required at that point. Fig.3.3 shows the required buffer at each instant of time for a video of 120 minutes length (i.e. $27*W_t*b$), which is divided into 7 segments. The maximum buffer required is $12*W_t*b$, which is less than 45% of the video/movie. Fib-1 broadcasting scheme requires buffer space less then 45% of the total video/movie. Fig.3.4 shows the relationship between the storage required and the bandwidth.

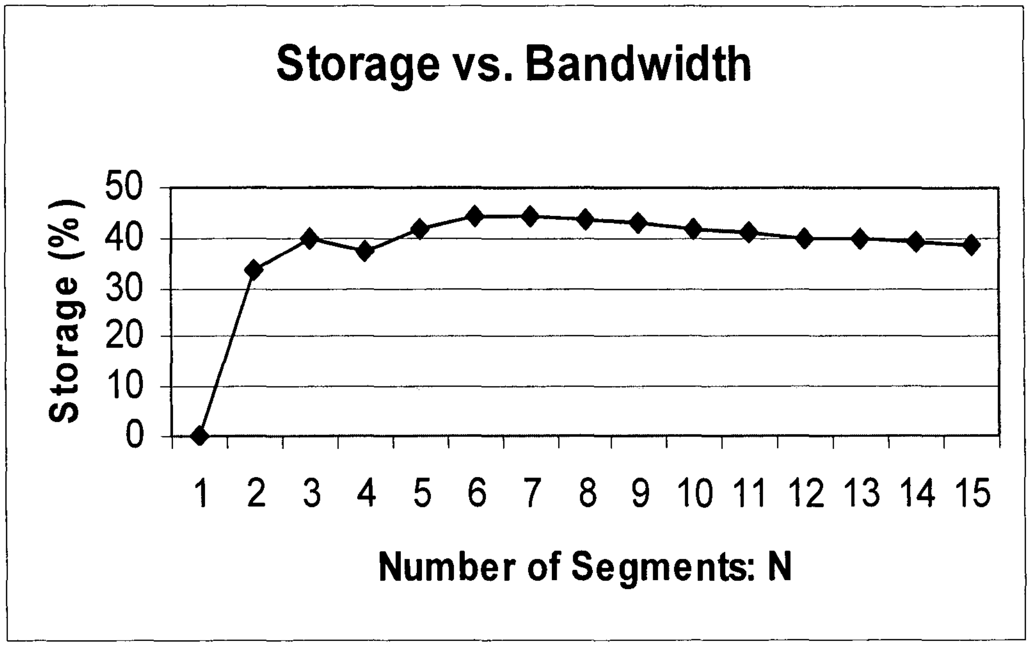

Figure 3.4: Storage vs. Bandwidth

3.2.3 Disk I/O Transfer Rate Requirements

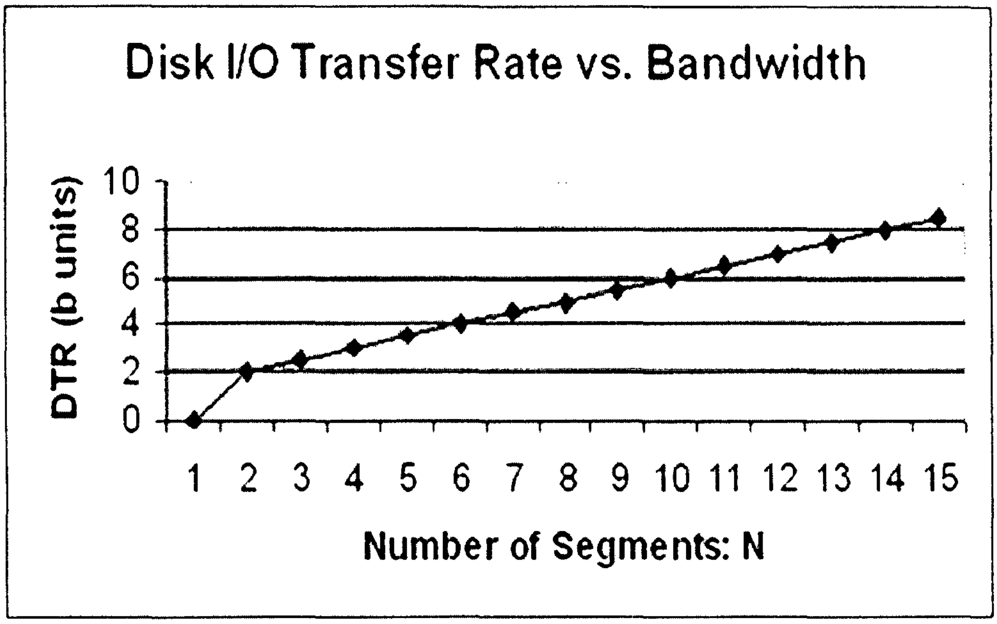

Figure 3.5: Disk I/O Transfer Rate vs. Bandwidth

The required disk I/O transfer rate is the maximum sum of the video data written onto the disk-storage and read from it during a time slot. Let $DTR_{I/O}$ be the disk I/O transfer rate of the disk for a given bandwidth with value of segments $N > 1$.

Thus,

$$DTR_{I/O} = Max(I_i + O_i), \tag{12}$$

I_i and O_i are already defined in the section 3.2.2. Fig.3.5 shows the relationship between disk I/O transfer rate and bandwidth.

3.3 Comparison of the Proposed Scheme with Other Schemes

A simple broadcasting scheme is presented, which is suitable to support popular videos for video on demand service. For a video/movie of length L minutes, this scheme can service the clients at every $W_t = L/N_{sb}$ minutes with N video channels. The storage (buffer) requirements at the client side are less than 45% of the total size of the video/movie. The maximum disk I/O transfer rate requirements are equivalent to the bandwidth.

For a popular video of 120 minutes, if 7 video channels are allocated for broadcast, then we get a waiting time of 0.4918 minutes, the required disk-storage is 4.05 Gbytes and the required disk I/O transfer rate is 70 Mbps.

Table 1 gives the comparative details. For the sake of comparison, we consider a video of length L=120 minutes, S = 9 Gbytes, b=10Mbps and B=4 video channels.

Scheme	**Waiting Time (minutes)**	**Disk Space Required (Gbytes)**	**Disk I/O Transfer Rate (Mbps)**
Harmonic	4.00	3.40	30
Fib-1	6.67	4.05	40
Staircase	8.00	2.10	20
Fast	8.00	4.20	40

Table 1: Comparative details of broadcasting schemes

The above table shows that the harmonic broadcasting scheme is the optimal one, but the proposed fib-1 broadcasting scheme is also performing better in relation to other broadcasting schemes when we study the waiting time.

Chapter IV

Conclusion and Future Work

In this thesis, we studied different broadcasting schemes which can provide video-on-demand services for broadcasting popular videos. This chapter summarizes our contributions and discusses some possible directions for future study.

4.1 Conclusion

We have proposed a new scheme "Fibonacci minus One (Fib-1) Broadcasting and Receiving Scheme", in which segmentation is based on the Fibonacci sequence. We have analyzed and compared this scheme with other broadcasting schemes based on the three characteristics i.e. Waiting Time, Storage Requirement and Disk I/O Transfer Rate. This scheme performs batter in terms of Waiting Time for a given set of parameters. The Disk Space (Storage) requirement and the Disk I/O Transfer Rate requirement is not a major concern due to the technology improvements in these areas. This scheme is easy to understand and implement. It gives greater flexibility to the VoD service providers.

4.2 Future Work

Some important aspects relate to the following:

1. We have shown that broadcasting popular videos can provide near video-on-demand services. Problems with interruption in video data downloading should also taken into account in providing real-time video-on- demand services.

2. A Pertinent question is to provide VCR-like control functions. Such aspects deserve further study so as far as fast forward, fast rewind, pause, etc. are concerned.

These aspects can be taken as a future enquiry.

References

[1] A. Dan, D. Sitaram, and P. Shahabuddin, Scheduling Policies for an On- Demand Video Server with Batching, *ACM Multimedia*, pp. 15-23, San Francisco, CA, October 1994.

[2] A. Dan, D. Sitaram, and P. Shahabuddin, Dynamic Batching Policies for an On-Demand Video Server, *Multimedia Systems*, 4(3): 112-121, June 1996.

[3] S. Viswanathan and T. Imielinski, Pyramid Broadcasting for Video on Demand Service, *IEEE Multimedia Computing and Networking Conferenc*e, Volume 2417, pp. 66-77, San Jose, California, 1995.

[4] S. Viswanathan and T. Imielinski, Metropolitan Area Video-on-Demand Service Using Pyramid Broadcasting, *Multimedia Systems*, 4(4): 197--208, August 1996.

[5] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand Systems, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp.118-126, Hiroshima, Japan, June 1996.

[6] K.A. Hua and S. Sheu, Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems, *ACM SIGCOMM*, pp. 89-100, Cannes, France, September 1997.

[7] D.L. Eager and M.K. Vernon, Dynamic Skyscraper Broadcasts for Video-On-Demand, *Proceedings 4^{th} International Workshop on Multimedia Information Systems (MIS'98),* Istanbul, September 1998.

[8] L. Juhn and L. Tseng, Harmonic Broadcasting for Video-On-Demand Service, *IEEE Transactions on Broadcastin*g, 43(3): 268-271, September 1997.

[9] L. Juhn and L. Tseng, Enhanced Harmonic Data Broadcasting and Receiving Scheme for Popular Video Service, *IEEE Transactions on Consumer Electronics*, vol. 44, pp. 343--346, May 1998.

[10] L. Juhn and L. Tseng, Staircase Data Broadcasting and Receiving Scheme for Hot Video Service, *IEEE Transactions on Consumer Electronics*, 43(4):1110-1117, November, 1997.

[11] L. Juhn and L. Tseng, Fast Broadcasting for Hot Video Access, *Proceedings of Fourth International Workshop on Real-Time Computing Systems and Applications*, pp. 237-243, October 1997.

[12] L. Juhn and L. Tseng, Fast Data Broadcasting and Receiving Scheme for Popular Video Service, *IEEE Transactions on Broadcasting*, 44(1): 100- 105, March 1998.

[13] J.-F. Paris, S.W. Carter, and D.D.E. Long, Efficient Broadcasting Protocols for Video on Demand, *6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '98)*, pp. 127-132, July 1998.

[14] J.-F. Paris, S.W. Carter, and D.D.E. Long, A Low Bandwidth Broadcasting Protocol for Video on Demand, *Proceedings of IEEE International Conference on Computer Communications and Networks (IC3N'98)*, pp. 690-697, 1998.

[15] J.-F. Paris, S.W. Carter, and D.D.E. Long, A Hybrid Broadcasting Protocol for Video on Demand, *Proceedings 1999 Multimedia Computing and Networking Conference (MMCN'99)*, pp. 317-326, San Jose, CA, January 1999.

[16] J.-F. Paris, A Simple Low-Bandwidth Broadcasting Protocol for Video-on-Demand, *Proceedings 8th International Conference on Computer Communications and Networks (IC3N'99)*, pp. 118-123, Boston-Natick, MA, October 1999.

[17] A. Hu, I. Nikolaidis, and P. Beek, On the Design of Efficient Video-on-Demand Broadcast Schedules, *Proceedings of 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MAS-COTS'99)*, pp. 262-269, 1999.

[18] P.V. Rangan, H.M. Vin and S. Ramanathan, Designing an On-Demand Multimedia Service, *IEEE Communications Magazine*, 30: 56-65, July 1992

[19] D.L. Gall, MPEG: A Video Compression Standard for Multimedia Applications, *Communications of the ACM*, 34(4), April 1991.

[20] P.V. Rangan and H.M. Vin, Efficient Storage Techniques for Digital Continuous Multimedia, *IEEE Trans. on Knowledge and Data Engineering*, August 1993.

[21] J.W. Wong, Broadcast Delivery, *Proceedings of the IEEE*, 76(12): 1566-1577, December 1988.

[22] http://in.news.yahoo.com/060616/139/65496.html. Frank Capra's Christmas Classic of 1946,"It's a wonderful life", movie.

[23] T. Imielinski and S. Viswanathan, Adaptive Wireless Information Systems, *Proceedings of the Special Interest Group in Database Systems (SIGDBS) Conference*, pp. 19-41, Tokyo, 1994.

[24] A. Dan, D. Sitaram, and P. Shahabuddin, Scheduling Policies for an On- Demand Video Server with Batching. *Proceedings of ACM Multimedia'94*, pp. 391-398, October 1994.

[25] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, On Optimal Piggyback Merging Policies for Video-on-Demand Systems, *Proceedings of SIGMETRICS '96: Conference on Measurement and Modeling of Computer Systems*, pp. 200-209, Philadelphia, PA, May 1996.

[26] K.A. Hua, Y. Cai, and S. Sheu, Patching: A Multicast Technique for True Video-On-Demand Services, *Proceedings of ACM Multimedia'98*, pp. 191-200, Bristol, UK, September 1998.

[27] K. Almeroth and M. Ammar, On the Use of Multicast Delivery to Provide a Scalable and Interactive Video-On-Demand Service, *IEEE Journal on Selected Areas in Communications*, 14(6): 1110--1122, August 1996.

[28] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, On Optimal Batching Policies for Video-On-Demand Storage Servers, *Proceedings of the IEEE International Conference on Multimedia Systems '96*, pp. 253-258, Hiroshima, Japan, June 1996.

[29] A. Gelman, H. Kobrinski, L.S. Smoot, and S.B. Weinstein, A Store-and-Forward Architecture for Video on Demand Servers, *Proceedings of the International Conference of Comm. '91*, 2: 842-846, Denver Colorado, 1991.

[30] Livingstone, A., "BT's video-on-demand technology trial", *IEE Colloquium on High speed Access Technology and Services, Including Video-on-Demand.*

[31] Vinod Kumar, "Fib-1 Broadcasting and Receiving Scheme for Video-on-Demand Service", submitted to *IEEE Transactions on Multimedia.*