Chapter 2

MULTIPLE SERVERS RETRIEVAL STRATEGY : DATA PARTITIONING APPROACH

2.1. Introduction to Multiple Server Technology

In Section 1.2.3, we have seen how to retrieve a long duration media document form a single-server system. The recommended portions considered the link bandwidth between the client and the server as well as the playback rate of the media at the client end. This is a typical service style of a networked client-server system. We have also seen how a single server serves more than one client and uses an admission controller to maximize the number of clients it can serve in Section 1.2.3. However, in this analysis, we have not taken into account the non-zero network delays. In contrast, the formulation of Section 1.2.3 focused on the impact of disk systems and also showed how storage specifications affect the admission control and service quality. However, several real-time applications demand the response of the media servers to be as quick as possible in order to make the underlying mission successful, hence network delays cannot be ignored. In this chapter, we will introduce a novel retrieval strategy that particularly suits long duration media such as feature-length movies (typically longer than 60 and usually around 90 minutes in duration) for Video-on-Demand (VoD) or Movie-on-Demand (MoD) applications[1]. Below we will see the need for such a multiple server technology for modern day high-bandwidth applications.

With the increasing popularity of multimedia services on network based environments, there is a continuous thrust in achieving an optimized design for multimedia servers and network service providers. The main attraction of these services is that viewing and presentation control is handed over to the user, in

[1]Hereafter we will use VoD and MoD interchangeably in this book.

contrast with conventional video broadcast services as cable TV. A particular movie can not only be made available to a user at his/her convenience (as with video cassettes), but the user can also have complete control on all aspects of the different media involved (audio, video), as for example the physical layout of the screen. Also, with an increase in demand for a particular movie, depending on the popularity profile, viewing cost per user can also be reduced considerably when clever placement of movies at strategic locations on the network is carried out. The effort for the development of such systems and services would be futile without the availability of high performance computers and high speed fiber optic networks that offer the capability of supporting such demands.

The above mentioned application and other (futuristic) applications like MoD, collaborative video editing and synthesis of multimedia objects and other network based distributed applications, will be attractive only when the available network bandwidth and other necessary resources are cleverly utilized. While using state-of-the-art technology to realize such applications is always a solution, the established -network mostly- and slowly changing infrastructure leaves no alternative but to carefully plan and utilize what is currently available. Owing to the continuous thrust in developing multimedia services on network based environments, the service providers situated at geographically large distances can co-operate and share their documents in order to serve their local subscribers. Once a document is available locally, in turn, each service provider can choose the appropriate admission control and scheduling algorithms to maximize their servicing capabilities. We believe that this multi-tier service architecture, provides an elegant solution for developing such multimedia services, since the data sizes that are involved are very large. Another motivation underlying this support is from monetary aspects. It will be prohibitively costly if one tries to replicate all the media documents across all the sites, to provide efficient service. Of course, replication of the documents may take care of the performance to certain extent, although monetarily such a scheme may not be desirable.

Also, applications like virtual multimedia conferencing or virtual group discussions, will be attractive and viable, only when the on-line discussions are captured and presented without any temporal delays. If however, the subject of discussion is currently not available at the local service provider, then the corresponding document would have to be downloaded from a remote site in which it is available. Since the sites are typically geographically apart and downloading would involve the transmission of large volumes of data, it would take quite an amount of time before the entire document is downloaded and the discussion at the local site initiated. This waiting time may be annoying to end users, especially if the discussion depends on making time critical decisions.

An illustrative example presented in the next section clearly describes the motivation for this research. In this chapter we precisely address the problem of minimizing this waiting time or the access time for similar and allied network based multimedia services. The key idea is to divide the media document into several disjoint portions, which are then retrieved from several servers, taking into account each client-server connection bandwidth. Since we partition the document, this approach shall be referred to as *data partitioning approach*, hereafter. Also, this data partitioning approach, to a large extent presents a unified theoretical framework that suits most of the existing network-based distributed multimedia services and also opens a new avenue for the researchers in this field. The theoretical framework is accompanied by an in-depth analysis of the behavior of the proposed strategies that leads to the design of more refined approaches in the chapters that follow. Whenever appropriate, we also make comparisons with the Single Server Retrieval Strategy (SSRS), that bring out the true potential of the suggested multiple server strategies.

2.1.1 Network architecture

We envision a network consisting of a set of service providers serving each locality (see figure 2.1)[2]. Each service provider has a directory facility, which registers the available documents at various sites. Whenever a user requests a movie, this directory service will produce a list of servers that can supply the requested multimedia document. Thus, if the requested multimedia document is not locally available, the service provider will request the other service providers to upload that document to its local site. Thereafter, whenever a request for this document arrives, the local server can use the stored document. This service provider could be the multimedia server itself, if it has adequate resources for supporting this service. In such a large network, the user requests may originate anywhere, and servicing these requests should incur the minimum possible delay. If not, such a multimedia service becomes less attractive. A survey [16] on monetary issues provides a good hope of realizing such multimedia applications on network based environments and opens avenues to pursue further research in this domain.

In the case of networks that span a large area wherein the server sites are geographically distributed, communication from one site to the other will incur a finite amount of non-zero delay. One of the attractive features of such multimedia services on networks lies in keeping up the promise of a smooth presentation without any audio-visual discontinuities. Unless clever strategies are adopted in retrieving the video blocks amidst the presence of these communication delays, this objective may not be met. We model the communication delay as a quantity that is directly proportional to the length of the video data that is

[2]Figure 4.1 presents a slightly more detailed version.

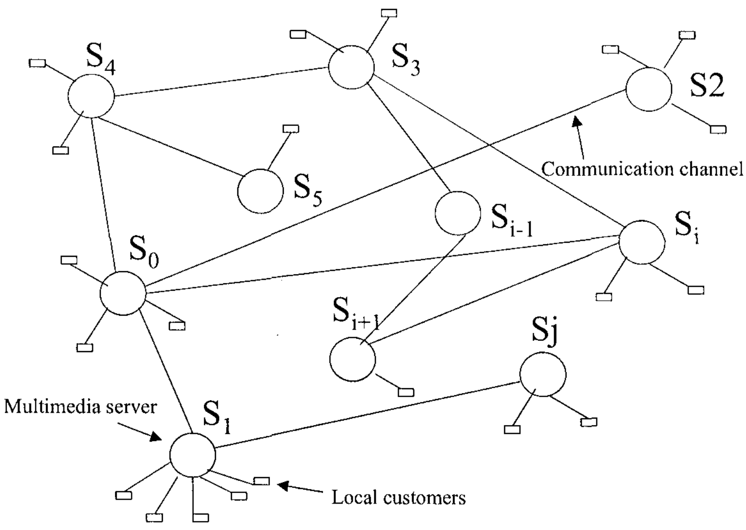

Figure 2.1. Networked Multimedia Servers Servicing Customers

carried over that established communication path (virtual or circuit switched) to the service provider. This is unlike the model proposed for pyramid broadcasting technique[3], in which the video blocks in the successive retrievals are of increasing sizes. In this multiple server technology, the available communication bandwidth and the display/playback rate of the video clip are the two major system parameters that are considered, and this chapter focuses in the design and analysis of retrieval strategies that minimize the access or the wait time of the users.

2.1.2 Distinct advantages in using Multiple Server Retrieval

We now highlight some of the key inherent advantages in using a multiple server approach. Since this strategy primarily involves more than one server for retrieving the document and rendering the VoD/MoD service, this strategy, and hence the technology associated with this service infrastructure is referred to as *Multiple Server Retrieval*(MSR) strategy, hereafter in this book. A MSR scheme inherently subsumes the following advantages. Firstly, on a network-based service rendering environment, if a single server system, however sophisticated it may be (in terms of speed and capacity) is used there is a continuous "work pressure" that is enforced on the system. For instance, when there is

[3]See the bibliographic notes for pyramid broadcasting.

a continuous demand for a long duration video retrieval by several clients, a significant amount of the time is spent in servicing these requests, while some small number of requests demanding short services may undergo long waiting times. By employing a MSR strategy, the work pressure can be balanced among the servers. Secondly, by using a MSR strategy, even low-bandwidth or heavily-loaded servers, that may not be usable on their own, can now be significantly contributing to a group of several servers that upload a movie. Thirdly, considering fault-tolerance aspects, which are treated in Chapter 7, even under server/link failures, the workload imbalance can be gracefully taken care of by the remaining servers, in a multiple server environment. Since multiple servers are engaged in the retrieval process, failure of one or more servers, will allow the service to continue without any interruption so long as there is at least one server operational. In fact, with a clever design of a retrieval strategy, the clients will continue to view the presentation while a certain number of servers may "die" and come back to "life" after some time. In contrast, with a conventional system, the clients will most probably need to be rescheduled at the expense of their presentation continuity. Also, as shown in rigorous simulation studies in the literature [105, 11], scalability of the physical system and heterogeneity of the system, can be easily accounted in the design, as the size of the portions retrieved from each of the servers depends on the available bandwidth and playback rate of the movie. In effect, a MSR strategy has a natural load balancing capability built-in its design. Each server participates according to its available capacity and/or its connection bandwidth to the client, collectively offering a service far superior than anything it could offer on its own. Finally, from service provider's perspective, since each server, on the whole, is engaged only for a short while in retrieving a portion of the media document, the number of clients that can be entertained simultaneously can be potentially maximized. Thus MSR offers a clear win-win situation for both the customers and the service providers.

2.2. Problem Definition and Preliminary Remarks

In this section, we present the problem more formally, describe the network architecture that is considered, and introduce the necessary definitions, notations and terminology.

The network model consists of a pool of N multimedia servers each serving their respective customers (see figure 2.1). The requests for viewing a movie of a long duration (typically of 100 to 120 minutes) arrive at these servers from its local customers. These servers are by and large, powerful workstations with sufficient amount of bandwidth capacity and memory space to serve a maximum number of users concurrently by employing efficient admission control algorithms. Upon an arrival of a request, the server seeks the requested multimedia document. We interchangeably use the terms service providers and servers as per the context. If the document is available locally, then usual retrieval and presentation techniques as described in the so far literature can be employed to serve the request. However, if the requested document is not available, then the server with its directory service facility, a kind of look-up table procedure, determines the server sites at which the requested multimedia document is present. It then obtains a set of server addresses from which the document may be retrieved. The requested multimedia document is then retrieved by employing a MSR strategy demonstrated through the following motivating example. We introduce the necessary notations and terminology in the example for the ease of understanding.

2.2.1 Motivating example

Consider a scenario in which a requested multimedia document is not available locally at a server denoted as, S. Let the requested multimedia document be present at the sites S_0, S_1, and S_2. Let the total size of the movie requested be $L = 1GB$. Further, let the channel bandwidths in terms of the time delay encountered per unit load transfer, measured in *seconds per unit load*, between each of these servers to S be denoted as bw_i $i = 0, 1, 2$. It may be noted that we are referring to the inverse of the channel bandwidth, however, we will continue to use the terminology *bandwidth* while referring to this quantity. Let these quantities(expressed in seconds per Mbit or in seconds per Mbyte) be $bw_0 = 1$, $bw_1 = 2$, and $bw_2 = 3$, respectively. Thus, with our definition, in this example, bw_0 is the fastest channel, bw_1 is the next fastest and so on. Hence, sending a unit load on bw_0 takes less time to reach S than from others. We assume that when the server S receives the document from another server, it starts the playback simultaneously at the user terminal. A discussion on this aspect is presented in 2.5. After locating the respective servers having the requested multimedia document (in this case servers 0 to 2), server S adopts the following strategy. From each server a portion of the entire document is retrieved and the parts are collected by S in a particular order. Upon receiving a portion from S_0, the playback is started at the user terminal. Let the inverse of the playback rate(expressed in the same units as the bw_i's), denoted as R_p be 5.3333 units (MPEG I stream). Now, the retrieval strategy is such that before the playback of this portion comes to an end, the next portion of the requested multimedia document is collected from S_1. This process is repeated for all the servers participating in the retrieval process.

This example describes one of the possible MSR strategies, namely the *single installment* retrieval strategy. Later, we will see an example for the case of the multi-installment strategy. This strategy has some inherent advantages. Firstly, it retrieves disjoint portions from different servers and thus, minimizes the retrieval time. Secondly, the strategy inherently takes care of continuity requirements, which are crucial when implementing such a strategy on network based environments. Thus, the continuity in the presentation is one of the aspects that a MSR strategy guarantees in the retrieval process apart from access time minimization. A fundamental assumption of all the analysis that follows is that the playback of a portion of the document is (or can be) initiated after this portion is completely received from the corresponding server. For the reason we also call this approach as a Play-After-Retrieval (PAR) strategy. In Chapter 4, we relax this assumption and examine the effect on performance when playback and retrieval are concurrent.

The *access time* or the *wait time* is defined as the time between the start of the downloading and the start of the playback [13]. The access time is directly proportional to the size of the portion retrieved from server S_0, i.e., starting from the time at which the downloading starts to the time at which the playback starts. An immediate naive choice would be to make this size as small as possible to minimize the access time of the entire multimedia document. However, in that case we will later show that the *presentation continuity* cannot be guaranteed by choosing the first retrieved portion arbitrarily as small as we desire. Hence, using this strategy, the problem now is to decide on the optimal sizes of the portions of the multimedia document to be retrieved from each of the servers, satisfying the presentation continuity, using the bandwidth constraints, and the playback rate constraints to minimize the access time.

In the above example, we see that the following size distribution $m_0 = 85.997$ MBytes, $m_1 = 272.3235$ MBytes, $m_2 = 665.679$ MBytes satisfies the constraints, where m_i is the size of the data retrieved from server S_i, $i = 0, 1, 2$, respectively. For this distribution, the access time (following the definition) is given by $m_0 bw_0 = 85.997$ secs.

An elegant representation of this retrieval strategy is by means of *directed flow graphs* (DFGs)[4]. Figure 2.2 shows the directed flow graph for this example. The communication nodes at the first level are assigned a weight equal to the total communication time of the portions of the multimedia document they are transferring to S. The dots on these communication nodes indicate that all these servers start their downloading simultaneously at time t units. Without loss of generality, we assume $t = 0$. If m_0 is the portion of the multimedia document communicated by S_0, then the total communication delay (which is the weight of the node 0 (see figure 2.2) is given by $m_0 bw_0$. The second level nodes are referred to as *playback* nodes. The weight of the playback node 0 is propor-

[4]Directed graphs are usually used to capture any precedence relationships between the nodes in the graph. The nodes of the graph may represent program modules, events or states in general.

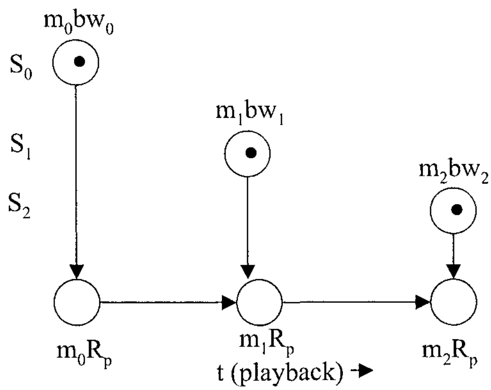

Figure 2.2. Directed Flow Graph representation for the case of 3 servers.

tional to the total time of playback of that portion of video, given by $m_0 R_p$, where R_p is the inverse of the rate of playback expressed in seconds per unit load of display. The directed arrows depict the *causal precedence* relationships between the node events. Thus, at S, the display of the portion from S_2 starts only after the display of the portion from S_1 is completed and also the portion from S_2 must be completely available.

2.2.2 Some definitions

Throughout the chapter we will use the following definitions.

1. *Retrieval schedule distribution*: This is defined as an N ordered tuple m given by
$$m = (m_0, m_1, ..., m_{N-1}) \tag{2.1}$$
where, m_i is the portion of the multimedia document downloaded from server S_i, $i = 0, 1, 2, ..., N-1$. Further,
$$\sum_{k=0}^{N-1} m_k = L \tag{2.2}$$
and
$$0 \leq m_i \leq L, \ \ i = 0, 1, ...N-1, \tag{2.3}$$
The set of all such retrieval schedule distributions is denoted as Γ.

2. The *Access Time* or the *wait time* is defined as the time between the instant at which the servers start uploading their portions to the time at which the

presentation starts. This is denoted as, $AT(m)$.

Typically, this is the time to access the first portion of the downloaded data, given by $m_0 bw_0$, where bw_0 is the bandwidth of the established communication path from S_0 to S. Hereafter, we shall use the term "access time" throughout the chapter.

3. *Minimum access time* is defined as,

$$AT^* = min_{m \in \Gamma} AT(m) \tag{2.4}$$

Thus, from the above set of definitions and the strategy illustrated in the above example, the objective is to minimize the access time by determining the optimal sizes of the portions of the video to be retrieved from different servers involved in the retrieval process.

2.3. Single Installment Retrieval Policy

In this section, following the retrieval strategy mentioned in the previous section, we shall analytically determine the sizes of the various portions retrieved from all the N servers. We then derive a closed-form solution for the minimum access time. It will be shown that the continuity and the bandwidth constraints are implicitly considered, stemming naturally as a property of the optimal solution in the analysis.

Figure 2.3 shows a generalized version of the example illustrated in the previous section, with N servers. From figure 2.3, we can derive a relationship between the communication nodes i and $i+1$ and the playback time of the portion m_i with the use of causal precedence relations and continuity constraint as,

$$m_{i+1} bw_{i+1} \leq m_i bw_i + m_i R_p, \quad i = 0, 1, ..., N-2. \tag{2.5}$$

Let us denote $(bw_i + R_p)/bw_{i+1} = \rho_i$. Using in (2.5), we have,

$$m_{i+1} \leq m_i \rho_i, , \quad i = 0, ..., N-2. \tag{2.6}$$

One can solve these set of recursive relations with equality conditions. Using equality relations in (2.5) and (2.6) produces the *maximum* size of all document portions other than m_0. So under the constraint of (2.2), this yields the *minimum* m_0 or equivalently the minimum access time. Thus, we have a recursive set of $(N-1)$ equations with equality relations from (2.6). Each m_j in (2.6) can be expressed in terms m_0 as,

$$m_j = m_0 \prod_{k=0}^{j-1} \rho_k, \quad j = 1, ..., N-1. \tag{2.7}$$

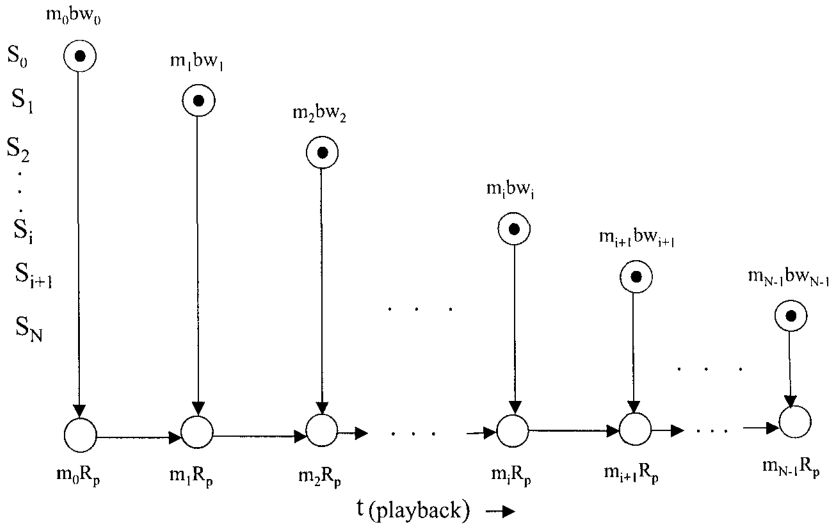

Figure 2.3. Directed Flow Graph representation using single-installment strategy for multimedia document retrieval from N servers

In general, the continuity relationship[5] states that *the playback duration of a portion of a video must be greater than or equal to the total retrieval time of the immediate successive portion of the video.* Otherwise, the continuity of the presentation will be lost and leads to a considerable performance degradation. In the above set of recursive equations, it may be noted that we have taken care of this continuity constraint by using the equality relationships rather than the inequality relationships. That is, the next playback will start immediately after the current playback comes to an end. Thus, when these set of linear recursive equations are solved with equality relationships, we obtain the minimum access time, as the solution set (m_0 to m_{N-1} through this procedure) gives the minimum value of m_0. Equation (2.7) generates $N-1$ equations, but we have N media portions. However, along with (2.2), we have N equations which can be solved to obtain the individual disjoint portions of the requested multimedia document. Substituting each m_i from (2.7) into (2.2), we obtain,

$$m_0 = \frac{L}{\left(1 + \sum_{p=1}^{N-1} \prod_{k=0}^{p-1} \rho_k\right)} \tag{2.8}$$

[5]See [89] for further reading on continuity constraints.

Substituting (2.8) in (2.7), we obtain the individual sizes of the portions as,

$$m_j = \frac{L \prod_{k=0}^{j-1} \rho_k}{\left(1 + \sum_{p=1}^{N-1} \prod_{k=0}^{p-1} \rho_k\right)} \tag{2.9}$$

for all $j = 1, ..., N-1$. Note that the access time (see figure 2.3) is given by,

$$AT(m) = m_0 bw_0 = \frac{L\ bw_0}{\left(1 + \sum_{p=1}^{N-1} \prod_{k=0}^{p-1} \rho_k\right)} \tag{2.10}$$

where we have used (2.8) for m_0. The retrieval schedule distribution demonstrated in the motivating example in the previous section is derived through these set of formulas.

2.3.1 Homogeneous channels

We consider a network with identical channel bandwidths or a channel that is shared by several servers. In other words, we have, $bw_i = bw$, for all $i = 0, ..., N-1$. The individual sizes of the portions retrieved from the servers S_0 till S_{N-1} are given by,

$$m_0 = \frac{L(\rho - 1)}{\rho^N - 1} \tag{2.11}$$

$$m_j = \frac{L(\rho - 1)\rho^j}{\rho^N - 1}, \quad j = 1, ..., N-1, \tag{2.12}$$

where we have used $bw_i = bw$ for all $i = 0, 1, ..., N-1$ in (2.8) and (2.9) to obtain these expressions. Hence, the access time is given by,

$$AT(m) = m_0 bw_0 = \frac{L\ bw\ (\rho - 1)}{\rho^N - 1} \tag{2.13}$$

We shall later present a detailed discussion on the behavior of this homogeneous system.

2.3.2 Effect of sequencing on the access time

It is worth noting at this juncture that throughout the above analysis we have assumed that retrieval follows a particular order, referred to as *fixed sequence*, S_0 to S_{N-1}. Given a set of N servers, we have $N!$ retrieval sequences possible. However, one may wonder if it could be possible to gain performance by varying the sequence in which the multimedia document is retrieved from the servers. The following lemma and theorem prove that such a behavior is not possible when a single installment strategy is employed.

Lemma 1. Let the access time of a requested multimedia document by the server S be denoted as $AT(m, \sigma(k, k+1))$, where

$$\sigma(k, k+1) = (S_0, ..., S_{k-1}, S_k, S_{k+1}, S_{k+2}, ..., S_{N-1}),$$

denotes the sequence in which the requested multimedia document is retrieved from the servers. Then, for a sequence

$$\sigma'(k, k+1) = (S_0, ..., S_{k-1}, S_{k+1}, S_k, S_{k+2}, ..., S_{N-1})$$

, the access time $AT(m', \sigma'(k, k+1))$ is equal to $AT(m, \sigma(k, k+1))$ where, $\sigma'(k, k+1)$ denotes a retrieval sequence in which the adjacent channels k and $k+1$ are swapped, i.e., portion from server S_{k+1} is retrieved first and then from server S_k.

Proof. The denominator of (2.10) can be written as:

$$denom(m) = 1 + \rho_0 + \rho_0\rho_1 + \rho_0\rho_1\rho_2 + \ldots = 1 + \rho_0(1 + \rho_1(1 + \rho_2(\ldots)))) \tag{2.14}$$

We can distinguish two cases depending on whether the first server S_0 is involved or not. If S_0 is not involved, when a switch is made between two successive servers, the new denominator is different from the original one in three ρterms. This difference can be written as

$$\begin{aligned} denom(m') - denom(m) = \\ &\frac{R_p + bw_{i-1}}{bw_{i+1}}\left(1 + \frac{R_p + bw_{i+1}}{bw_i}\left(1 + \frac{R_p + bw_i}{bw_{i+2}}\right)\right) \\ &- \frac{R_p + bw_{i-1}}{bw_i}\left(1 + \frac{R_p + bw_i}{bw_{i+1}}\left(1 + \frac{R_p + bw_{i+1}}{bw_{i+2}}\right)\right) \end{aligned} \tag{2.15}$$

With little algebraic manipulation, the above equation returns zero.

In the second case, where the first two servers S_0 and S_1 are switched, the difference in access time is,

$$\begin{aligned} AT(m) - AT(m') &= \frac{L\, bw_0}{denom(m)} - \frac{L\, bw_1}{denom(m')} \\ &= L\frac{(bw_0 denom(m') - bw_1 denom(m))}{denom(m)\, denom(m')} \end{aligned} \tag{2.16}$$

By using (2.14) ($denom(m)$ and $denom(m')$ differ in two ρ-terms), the numerator of the above fraction becomes equal to

$$bw_0\left(1 + \frac{R_p + bw_1}{bw_0}\left(1 + \frac{R_p + bw_0}{bw_2}\right)\right)$$

$$-bw_1\left(1+\frac{R_p+bw_0}{bw_1}\left(1+\frac{R_p+bw_1}{bw_2}\right)\right) \tag{2.17}$$

which in turn can be easily proven to be equal to zero. □

The significance of the lemma is that the order in which the portions are downloaded affects only the respective size distribution, but not the access time when adjacent servers are swapped. We prove this claim in general for the case of N servers, as follows.

Theorem 1. Given a pool of N multimedia servers capable of rendering the requested multimedia document, using the single installment strategy, the access time is independent of the retrieval sequence used.

Proof. Direct application of Lemma 1 proves the theorem. Any valid sequence of servers can be derived from a single sequence by iteratively switching the positions between adjacent servers. Lemma 1 guarantees that these operations do not affect access time. □

2.4. Multi-installment Servicing Policy

In this section, we present a generalized servicing policy, which provides a tuning control mechanism for the access time. This tuning provides a better control on the retrieval process and allows the system to adapt to the variations in the bandwidth of the network. This policy constitutes of retrieving the multimedia data from each of the servers S_0 to S_{N-1} in more than one installment. This means that every server participates more than once in the process of uploading disjoint portions of the requested document one after other, in a particular order. Since this facilitates the accessing of the data by S at a much earlier time, the access time decreases. As in the case of the single installment policy, the continuity of the presentation must be guaranteed. The necessary and sufficient conditions for the multi-installment policy, can be derived by extending the directed graph representation of the previous section. By assuming that each server uploads a disjoint portion of the multimedia data in n installments, we can use the extended graph to derive a set of recursive equations.

2.4.1 Recursive equations and solution methodology

Figure 2.4 shows the directed graph for this policy. Note that we have extended the communication and playback processes from single installment to multi-installment using causal precedence relations (time orderliness). Let $m_{i,j}$ represents the part of the L in total multimedia data, that is downloaded from the S_i server during the j-th installment, where $j = 0, \ldots, n-1$. Thus, there is a total of Nn portions of the multimedia document that are retrieved from servers S_0 to S_{N-1} in n installments. These are $m_{0,0}, m_{1,0}, \ldots, m_{N-1,0}, m_{0,1}, \ldots,$

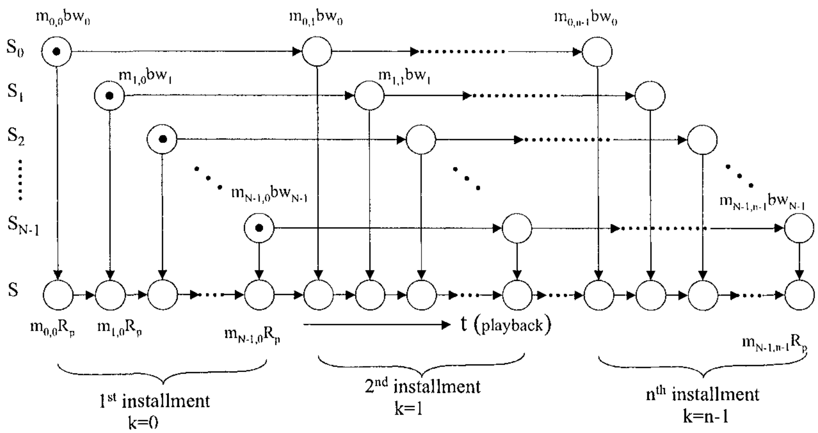

Figure 2.4. Directed Flow Graph representation using Multi-installment strategy for multimedia document retrieval from N servers

$m_{N-1,1}, \ldots, m_{0,n-1}, \ldots, m_{N-1,n-1}$. It can be easily deducted from (see figure 2.4) that the causal precedence relations and the continuity relationships impose the following inequalities:

$$m_{k,0}bw_k \leq m_{k-1,0}bw_{k-1} + m_{k-1,0}R_p, \tag{2.18}$$

for all $k = 1, 2, ..., N - 1$. For $i = 1, 2, ..., n - 1$, we have,

$$m_{k,i}bw_k \leq \left(\sum_{p=k}^{N-1} m_{p,i-1} + \sum_{p=1}^{k-1} m_{p,i} \right) R_p \tag{2.19}$$

for all $k = 0, 1, ..., N - 1$.

The $m_{i,j}$ parts are also connected by the normalizing equation:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{n-1} m_{i,j} = L \tag{2.20}$$

The *minimum* size of $m_{0,0}$, which determines the minimum access time, can be obtained by seeking the *maximization* of all other $m_{i,j}$. This goal can be achieved by using the equality relations in Equations (2.19) and (2.20). Deriving a closed-form solution from the above set of recursive equations is too tedious. However, since

$$m_{1,0} = m_{0,0} \cdot \frac{bw_0 + R_p}{bw_1} \tag{2.21}$$

$$m_{k,0} = m_{0,0} \cdot \prod_{j=1}^{k} \frac{bw_{j-1} + R_p}{bw_j} \tag{2.22}$$

$$m_{k,i} = \left(\sum_{j=k}^{N-1} m_{j,i-1} + \sum_{j=0}^{k-1} m_{j,i} \right) \frac{R_p}{bw_k} \tag{2.23}$$

we can reach a solution by using the following procedure. If we assume that $m_{0,0} = 1$ we can obtain from (2.21) $m_{1,0}$, and from Eq. (2.22) and (2.23) successively all $m_{k,i}$. The original assumption is the equivalent of multiplying $m_{0,0}$ by a constant K so that it equals unity. Because of the way $m_{i,j}$ are related through Eq. (2.21), (2.22) and (2.23), this process returns all $m_{k,i}$ also multiplied by K. K can in turn be estimated from the normalizing Equation (2.20), which becomes:

$$\sum_{j=0}^{N-1} \sum_{l=0}^{n-1} m_{j,l} = K \cdot L \tag{2.24}$$

and this completes the solution. We denote the access time, using multi-installment strategy, as $AT(N, n)$, as we have now two parameters N an n to control the access time. The access time is given by,

$$AT(N, n) = m_{0,0} bw_0, \tag{2.25}$$

where $m_{0,0}$ is obtained by solving the recursive equations a mentioned above. Following example demonstrates this strategy.

Example 2. Suppose that the client at S requests a 2GB multimedia document from servers S_0, S_1, S_2 and S_3, which are linked with S with 128KByte/sec connections. If we assume a playback rate of 4Mbits/sec, typical of a MPEG II document, then a single installment, i.e. $n = 1$, results in an access time of 2841.67 seconds. However, doubling the number of installments results in an access time of 1362.0 seconds. Thus, we gain a significant decrease (of 52%) in the access time.

2.4.2 Homogeneous channels

Here too, we shall consider analysis for homogeneous channels. Thus, the above set of recursive relations ((2.18) and (2.19)) reduces to,

$$m_{k,0} bw \leq m_{k-1,0} bw + m_{k-1,0} R_p, \tag{2.26}$$

for all $k = 1, 2, ..., N - 1$. Then, for $i = 1, 2, ..., n - 1$, we have,

$$m_{k,i} bw \leq \left(\sum_{p=k}^{N} m_{p,i-1} + \sum_{p=1}^{k-1} m_{p,i} \right) R_p \tag{2.27}$$

for all $k = 0, 1, ..., N-1$. Denoting, R_p/bw as σ, we have,

$$m_{k,0} \leq m_{k-1,0}(1+\sigma) \tag{2.28}$$

for all $k = 1, 2, ..., N-1$. Then, for $i = 1, 2, ..., n-1$, we have,

$$m_{k,i} \leq \left(\sum_{p=k}^{N} m_{p,i-1} + \sum_{p=1}^{k-1} m_{p,i} \right) \sigma \tag{2.29}$$

for all $k = 0, 1, ..., N-1$. Using the equality relationship, we write,

$$m_{k,0} = m_{k-1,0}(1+\sigma) \tag{2.30}$$

for all $k = 1, 2, ..., N-1$. Then, for $i = 1, 2, ..., n-1$, we have,

$$m_{k,i} = \left(\sum_{p=k}^{N} m_{p,i-1} + \sum_{p=1}^{k-1} m_{p,i} \right) \sigma \tag{2.31}$$

for all $k = 0, 1, ..., N-1$. Each of the $m_{k,0},\ \ k = 1, ..., N-1$ from (2.30) can be expressed as a function of $m_{0,0}$ as,

$$m_{k,0} = m_{0,0} P(\sigma, k), \tag{2.32}$$

where, $P(\sigma, k) = (1+\sigma)^{k-1}$. It is worth noting that the polynomial $P(\sigma, k)$ contains binomial coefficients when expanded. Also, it may be observed from (2.31) that to obtain $m_{k,i}$ for any $i > 0$, we need to simply add N of its preceding terms and multiply by a σ.

We define a transformation $k = i(n-1) + jN$, and denote the portions of the multimedia document retrieved from $S_0, ..., S_{N-1}$ in n installments as, $Q_k,\ k = 0, 1, ..., Nn-1$, where k is as defined above. For instance, when $N = 4$ and $n = 2$, $m_{1,1}$ which is actually the second installment for server S_1 is denoted by Q_5, as $k = 1.(1) + 1.(4) = 5$. Thus, with this notation and transformation, and with (2.31) and (2.32), we can easily generate the following table. We have shown the table for $N = 4$ and $n = 2$ case. The entries in each row of the table are the coefficients of the respective powers of σ. Thus, the maximum number of columns will be $Nn - 1$. As an example, $m_{1,1}$ corresponds to the row Q_5, given by (2.31) as $3\sigma + 10\sigma^2 + 10\sigma^3 + 5\sigma^4 + \sigma^5$, and the entries are precisely these coefficients of the various powers of σ.

$m_{i,j}$	$j \rightarrow$ / k	0	1	2	3	4	5	6	7
0,0	0	1	0	0	0	0	0	0	0
1,0	1	1	1	0	0	0	0	0	0
2,0	2	1	2	1	0	0	0	0	0
3,0	3	1	3	3	1	0	0	0	0
0,1	4	0	4	6	4	1	0	0	0
1,1	5	0	3	10	10	5	1	0	0
2,1	6	0	2	12	20	15	6	1	0
3,1	7	0	1	12	31	35	21	7	1

Thus, generalizing this idea, we have the following boundary conditions and a recursive definition to generate a particular entry E(i,j) in the table for arbitrary N and n.

The boundary conditions, which generate entries for the first installment $n = 0$ are given by,

$$E(k,0) = 1, \forall k = 0, 1, ..., N-1, \quad (2.33)$$

$$E(k,0) = 0, \forall k = N, ..., Nn-1, \quad (2.34)$$

$$E(k,j) = 0, \forall j > k, \text{ and} k, j = 0, 1, ..., Nn-1, \quad (2.35)$$

$$E(k,j) = E(k-1, j-1) + E(k-1, j), \forall k = 1, 2, ..., N-1 \quad (2.36)$$

Note that the first entry $E(0,0)$ is assumed to be equal to 1. This is just for the purpose of generating the table entries as a function of $m_{0,0}$ which is assumed to be unity. However, when actually solving the recursive equations, we can express each of the $m_{i,j}$ as a function of $m_{0,0}$. Now, for the remaining rows, $Q_N, ..., Q_{Nn-1}$, we have,

$$E(k,j) = \sum_{p=k-N}^{k-1} E(p, j-1), \; \forall k = N, ..., Nn-1, \; j = 1, 2, ..., Nn-1, \quad (2.37)$$

Thus, we have for $Q_5 = m_{1,1} = E(5,0) + E(5,1)\sigma + ... + E(5,4)\sigma^4 + E(5,5)\sigma^5$. This is nothing but the polynomial shown above. Following this notion, we can write $m_{i,j}$ as,

$$m_{i,j} = Q_k = \sum_{j=0}^{i} E(i,j)\sigma^j, \forall i = 0, 1, ..., Nn-1, \quad (2.38)$$

We have a total of Nn unknowns with $Nn - 1$ equations. As in the previous section, we use the normalizing equation,

$$\sum_{i=0}^{Nn-1} \sum_{j=0}^{i} E(i,j)\sigma^j = L \tag{2.39}$$

to have a total of Nn equations to solve for all the unknowns. Note that each of the $m_{i,j}$ can be expressed in terms of $m_{0,0}$, by using the recursive definition of (2.38) and using (2.39), we obtain,

$$m_{0,0} = \frac{L}{\sum_{i=0}^{Nn-1} \sum_{j=0}^{i} E(i,j)\sigma^j}, \tag{2.40}$$

where, $E(i,j)$ is generated by using equations (2.34) to (2.37).

Thus, given a set of N multimedia servers having identical channel bandwidth connections, it is easier to obtain the optimal sizes of the portions of the multimedia document to be retrieved from each server by immediately using equations (2.34) to (2.40).

2.5. Discussions on MSR Strategy

The problem tackled in this chapter presents a generalized approach to the theory of minimizing the access time for network based multimedia document retrieval/distribution. Researchers in this field have addressed the problem from different perspective, however, on different application specific requirements. For instance, the results equally apply to a single server multiple client multimedia video-On demand system by replacing the bandwidths specified in this chapter by the disk bandwidths. On a network based environment, one of the crucial bottlenecks being the available bandwidth, the treatment we presented suggests an elegant solution to minimize the access time. Future network based multimedia applications demanding a trade-off between the available network bandwidth and the service from geographically well separated servers become the natural candidates of the problem addressed in this chapter. We shall now present some interesting observations made during the analysis.

We have derived the closed-form solution for the partitioning problem in a single-installment strategy. By using the recursive procedure, obtaining the optimal sizes of the media portions (solution set) retrieved from the servers, can be achieved with $O(N)$ time complexity. An attractive and interesting feature of this strategy is that the optimal solution evolves naturally as we solve the set of recursive equations (2.7) with equality constraints (capturing the continuity relationships) together with (2.2). As it is evident from the closed-form solution, the access time monotonically decreases as we tend to utilize more and more servers. Figure 2.5 shows the behavior of the access time with respect to the number of servers utilized for the case of homogeneous channels, i.e., when $bw_i = bw$, for all the channels. As expected, as the requested multimedia document is available on more servers, the access time decreases, as we tend to utilize all the probable servers. In the figure, we have shown the plots for typical MPEG-I and MPEG-II video streams with 1.5MB/s and 4MB/s playback rates, respectively. Another important contribution in this research is the proof of *access time invariance property on sequencing*. The non-triviality lies in identifying such a behavior of the access time. The result of Lemma 1, though not astonishing, clears the fact that the access time is independent of the order in which the multimedia document is retrieved from the server pool, at least as long as communication costs do not suffer from start-up latencies. If the latter apply, a different outcome is produced as discussed in Chapter 5.

Alternative to this strategy and as a future extension to the problem considered in this chapter is the following. It may be possible with the current day technology that a client may be able to download multiple streams at the same time depending on its available buffer and the bandwidth. This scheme is attractive as it is adaptable to varying system loads. However, the implementation of this scheme involves the design of efficient resource management and scheduling policies at the client site. In the case of multi-installment strategy, obtaining a closed-form as in single-installment strategy, is tedious as equations are complex to solve. However, the recursive procedure proposed in Section 2.4.1 generates the optimal sizes of the media portions from the servers in $O(Nn)$ time complexity. Also, the recursive expressions presented in Section 2.4.2 for the case of homogeneous channels are easier to implement as the entries of the table (shown for $N = 4$ and $n = 2$ case) can be automatically generated and can be reused when the number of installments or the servers varies. Another attractive feature of this methodology is that the speed parameters (channel bandwidth and the playback rates) do not influence or affect the entries of the table, and hence these entries $(E(i, j), \forall i, j$ can be generated a-priori and can be stored using the best possible data structure for computation purposes.

As far as the performance of this strategy is concerned, a set of questions that might arise are the following:

- What is the relationship between the number of servers used and the number of installments?

- Can we expect that ever-increasing the number of installments will always result in gains with respect to access time?

Answers to the above questions are attempted by means of an experiment. An iterative procedure was employed in order to compute the access time under

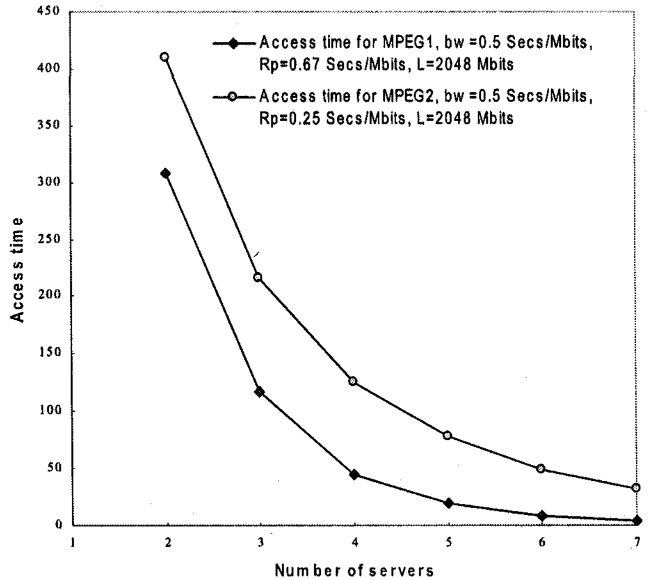

Figure 2.5. Access Time vs Number of multimedia servers using MPEG-I and MPEG-II Streams

ever-increasing number of installments. The document size was set to 2GBytes and the playback rate R_p to 2sec/MByte, typical of a MPEG II stream. The number of servers varied from 2 to 30 and their bandwidth from 1sec/MByte to 32sec/MByte. All connections were considered identical, i.e. $bw_i \equiv bw$ The iteration stopped, either when increasing the number of installments by 1 resulted in less than 5% gain in access time, or when the access time fell below 1 second. The number of installments at which the process ended is plotted in figure 2.6(a). Figure 2.6(b) shows the corresponding access times. The y-axis is labeled as $L\ bw$ in order to give a measure of the time necessary to download the multimedia document from a single server. The usage of the $L\ bw$ product is not a way for normalizing things. Actually, using different bandwidths with loads that have identical $L\ bw$ products results in different (quantitative) behaviors.

A striking observation (see figure 2.6) is that there is a barrier beyond which increasing the number of installments and/or the number of servers cannot benefit the access time. This barrier is network capacity related and it can result in very poor performance. Just before this barrier is reached, the number of installments that are suitable for near optimum performance grows rapidly and falls the same way after the barrier is exceeded. Before the barrier is reached,

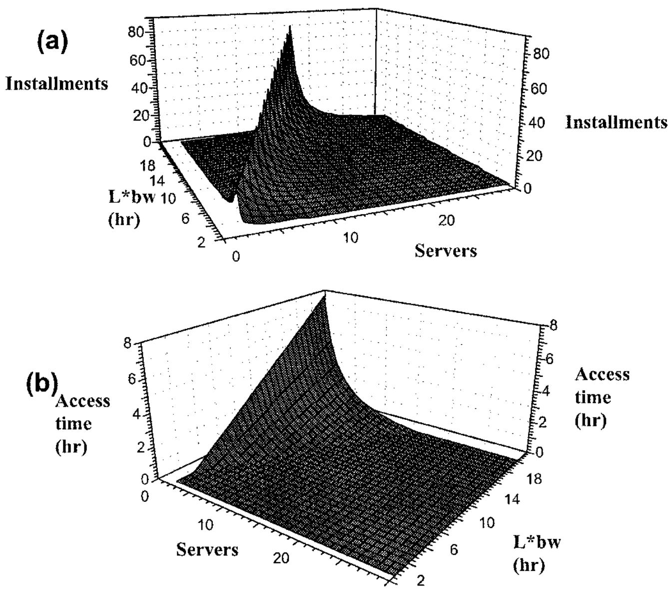

Figure 2.6. (a) Number of installments where the access time falls below 1 sec, or where an additional installment would improve the access time less than 5%, against the number of servers and connection speed. (b) The corresponding access times for the installments shown in (a).

the network connections are fast enough to guarantee that even with a small $m_{0,0}$ the portions of the document will reach the client prior to their turn for playback. Increasing the number of installments has just that effect, i.e. keeping $m_{0,0}$ small. After the barrier is exceeded, $m_{0,0}$ must be a large portion of the document in order to guarantee that the continuity constraints hold. As a result, the access time escalates while increasing the number of installments does not serve the purpose.

The relationship between the number of servers and number of installments is more clearly seen in Figures 2.7(a) and (b) which depict an experiment identical to the one described above, with the exceptions of

1 Adding more servers in the previous case, actually scales the client's connection bandwidth, which is not the usual case -for Internet at least- connections. So in this case, there is a shared connection whose bandwidth is divided among all servers.

2 The shared bandwidth ranged from 0.5 sec/MB to 4 sec/MB .

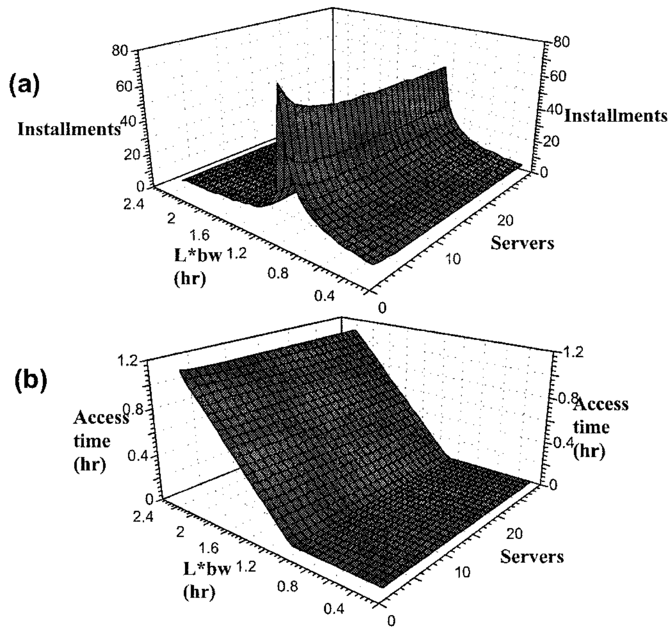

Figure 2.7. (a) Number of installments where the access time falls below 1 sec, or where an additional installment would improve the access time less than 5%, against the number of servers and connection speed. The difference from Figure 2.6(a) is that the servers share the same connection to the document requesting party. (b) The corresponding access times for the installments shown in (a).

The number of servers and installments are perpendicular ways for maximizing server utilization under the multi-installment strategy. This is clearly seen in figure 2.7(a), where as the number of servers increases, the near optimum number of installments decreases. This can work the other way around, i.e. instead of increasing the number of servers, the number of installments can be modified to achieve a target access time. Of course, there are limits to this approach especially if each connection is independent of the others (see figure 2.6). As a final remark, the multi-installment strategy has an inherent advantage in managing the available buffer at the client's site. If the buffer size available is limited, then the best possible way to optimize the performance is by the use of multi-installment strategy, as the sizes of the individual chunks retrieved are smaller than the individual sizes recommended by the single installment strategy. This in a way makes the scheme attractive, as the existing limited resources(buffer capacities) are cleverly utilized without any additional resource investment.

One may suggest that the multi-installment strategy fails to address the need to use the available multimedia servers for servicing multiple client requests. This, however, falls outside the scope of this chapter. Serving multiple clients, calls for employing other administration policies for allocating the bandwidth of each server. The strategies presented here are suited for analytical or otherwise systematical evaluation of such policies, which could very well be the subject of further research. For example, as Figures 2.6 and 2.7 indicate, MSRSs can be used to determine the minimum bandwidth that each server should allocate for a single request while keeping the access time to a minimum. In Chapter 3 we discuss on strategies that can handle multiple clients using this MSR strategy.

2.6. Concluding Remarks

In this chapter, we have introduced the MSR technology and the key idea behind it. We presented a generalized approach to the theory of minimizing the access time of retrieving a multimedia document requested by a client. On a network based environment, we have shown the impact of non-zero communication delays on the performance. We have designed and analyzed two different retrieval strategies that minimize the access time of the multimedia document. The single and multi-installment strategies proposed, can provide a basis for improved quality services in the emerging markets of VoD or MoD. Given the limiting technological aspects of the current network -mainly- infrastructure, these strategies form not just the basis for improved services, but can be the only way to realize them. In general, these strategies are designed to suit most of the network based multimedia applications in which a large volume of data needs to be transferred between sites incurring a considerable amount of communication delays. Approximations to the strategies and the model presented in this chapter precisely address the problems in the domain of multimedia applications in which the impact of network delays and bandwidth are significant.

In the case of the single-installment strategy, we have derived closed-form solution for the individual sizes of the portions retrieved from various servers and the corresponding access time. We have extended the analysis to the case of homogeneous or shared channels. In the case of the multi-installment strategy, we have presented a recursive procedure to obtain the individual sizes of the document portions from various servers. It has been shown that the multi-installment strategy has the added advantage of fine tuning the access time, which is crucial to time and delay sensitive applications. Rigorous performance tests were conducted to study the trade-off relationship between using several servers and using multiple installments. It was observed that the gain achieved by increasing the number of installments is in effect similar to employing more servers for the job. Of course, the former approach is far more attractive as a means towards minimization of the access time. This also means that the available servers can be appropriately partitioned to attend to different service requests, while at the same time keeping the access time as low as possible. This observation is not only non-trivial, but also allows an increase of the overall servicing capability of the network and makes such network based multimedia applications viable and more attractive. In the case of the single installment strategy, we have proved an important theorem which eliminates the need for seeking a particular server sequence. Although, the access time is independent of the sequence chosen, a deployed system may elaborate on the sequence in order to 'release' some servers before others. Of course, as the number of installments increases such actions become pointless.

The analysis presented in the Section 2.5 can be a stimulus for developing more fine-tuned approaches, that take into account the ever-changing nature of the network environment parameters. Firstly, an interesting question is to see if this MSR dogma can support multiple clients! We deal with this in Chapter 3. Secondly, constant bandwidths are rarely the case for the heavily loaded Internet connections that are shared by thousands of users. The volatility of the network connections is the key factor for future research. Although, the multi-installment strategy is the way to go, insuring QoS requires that a deployed system assigns and possibly modifies during a session, the document downloading responsibilities, subject to server availability and bandwidth irregularities. We address this in Chapters 5 and 6. Since multiple servers are used, it is obvious to investigate on fault-tolerance, especially under network and/or server failures. These are dealt in Chapter 7. Thus the idea of MSR approach opens up several possibilities to seek best quality of services for high-bandwidth applications via network based service infrastructure.

Bibliographic Notes

There has been an extensive study on many allied problems related to this MSR design. A Video on-Demand(VoD) or Movie on-Demand(MoD) service which is usually offered on hi-speed network based environments is one of the natural candidates [2,10] for the problem presented in this chapter. There has been a continuous effort to optimize the performance of multimedia servers designed for VoD applications [98]. In [19], a multimedia distribution network has been presented. Here, the authors introduce a three tier network architecture to the video distribution problem on networks. Applications such as collaborative video editing and synthesis of multimedia objects and other network based distributed applications can be found in [24, 37, 21, 108]. These essentially focus in optimizing the storage and retrieval of video blocks on the disks [11] and also in the design of servers to maximize the number of customers. Depending on the popularity of the movies in a networking domain, physical placement of movies on particular server sites is important in optimizing the monetary cost incurred for viewing that movie [18]. In order to optimize the monetary costs, clever placement of movies at strategic locations on the network is carried out [19]. Discussions on appropriate admission control and some scheduling algorithms can be found in [89, 108, 79].

A novel technique, referred to as pyramid broadcasting, is proposed in [109], as a means of serving a large pool of customers in Metropolitan Area Networks(MANs). This technique typically supports services like VoD or MoD on MANs. Another closely related theme can be found in a study by Candan et al. [24], in which retrieval schedules are generated based on the availability of the network and other necessary resources with flexible presentation requirements. The influence of the network service providers and on the buffer resources at the client site are also addressed in this study. A slight variant of this service architecture was proposed in the study on multimedia presentation planning in [53]. The study presented in [105] is one of the first attempts to introduce MSR to the multimedia literature.

Springer

http://www.springer.com/978-0-387-28873-4

Distributed Multimedia Retrieval Strategies for Large
Scale Networked Systems
Veeravalli, B.; Barlas, G.
2006, XXVIII, 268 p. 92 illus., Hardcover
ISBN: 978-0-387-28873-4