Title of Thesis:

“On the placement of video-clip content in the caches of mobile clients: An approach that combinely addresses client startup latency and system throughput”

Μανουσάκης Αντώνιος

Εξεταστική Επιτροπή

Καθηγητής	Μ. Πατεράκης (επιβλέπων)
Καθηγητής	Σ. Χριστοδουλάκης
Αν. Καθηγητής	Α. Ποταμιάνος

Technical University of Crete
Department of Electronic and Computer Engineering

Contents

Chapter 1

1.1. Introduction……. 3
1.2. Related work on Web and Video Caching…….. 4
1.3. Roadmap of the Thesis…… 6

Chapter 2

2.1. Problem Definition..... 8
2.2. System Models..... 9
2.2.1. First System Model..... 9
2.2.2. Second (Extended) System Model..... 11
2.2.3. Client Request Process and Video Clip Popularity Distr…. 13
2.3. Performance Metrics…….. 13
2.4. Video Content Placement in Caches……. 15
2.5. Overview……... 17

Chapter 3

3.1. Simulation Model and Examined Scenarios…… 19
3.2. First Case: Performance Results and Discussions…… 21
3.3. The Second Case: Performance Results and Discussions…… 29
3.4. Sensitivity of the Results to the Number of Video-Clips Stored in the Video Server……. 34
3.5. Overview…….. 37

Chapter 4

4.1. Conclusions……. 39
4.2. Ideas for Future Work……. 40

References..... 41

CHAPTER 1

1.1 Introduction
1.2 Related work on Web and Video Caching
1.3 Roadmap of the thesis

1.1 Introduction

The emerge of the Internet as a pervasive communication medium has fuelled a dramatic convergence of voice, video and data communications, resulting in the appearance of a broad range of Internet multimedia applications. Example of such applications include, live video and audio broadcast, distance learning, streaming of video and audio clips over the Web and transmission of time-varying information (e.g., stock prices, weather conditions, traffic information, e.t.c).

This explosive growth of the World Wide Web has led to significant increases in user latency and network congestion for Internet applications. One approach to reducing response time and network traffic is to deploy caches on the edge of the Internet close to the users. A proxy cache stores recently accessed objects or objects with high popularity, in the hope of satisfying future client requests without being necessary to invoke the server holding the corresponding object. As requests for, and delivery of streaming video over the Web becomes more popular, caching of media objects at the edge of the Internet has become increasingly important.

At the same time the rapidly expanding technology of wireless cellular communication networks and satellite communication systems makes possible for mobile users to access multimedia information anywhere at any time. It is envisioned that in the near future, millions of users will carry portable computing devices and will use them to do most of the work that today do with their fixed personal computer. From the above discussion, it is clear that there is an increasing need for advanced mobile technology that will enable the communication of time sensitive multimedia information to the mobile users.

There are, however many significant differences between the wireless communication network and the traditional wired network. Because of concerns of cost and of ease of portability, mobile users suffer from various resource limitations. For example, mobile computers have limited memory, storage and power supply. Equally important, the wireless communication channel has a much lower bandwidth than a communication channel in a today's wired network. This fact becomes the bottleneck of the whole wireless communication system if the wireless communication bandwidth is not carefully managed.

In this Thesis we look at the issue of caching of multimedia streams in order to improve the performance of data access and to reduce the requirements for wireless channel bandwidth.

We consider that the mobile devices are equipped with cache memory, and we develop efficient methods for video content placement in these caches. Subsequently, we consider that in addition to the clients' caches the system is equipped with another storing facility (with much higher capacity than a clients' cache) located at the fixed network very close to the wireless channel interface (access point, AP). We develop a technique for placing video content, not stored in the clients' caches, in the AP cache.

The content placement methods we develop are cost-based, and the cost function we use combinely addresses the startup latency experienced by a client and the throughput achieved by the system.

1.2 Related work on Web and Video Caching

In our work we try to combine useful approaches and results from two different technical research fields. The first area is associated with caching of multimedia streams and how the later can help the information delivery between a central server and the remote clients, and the second deals with how we can effectively use caching of multimedia streams over the wireless/mobile environment.

Multimedia proxy caching is a relatively new research area. During recent years, a few commercial multimedia proxy caches have been developed [1,2,3]. While there is no technical information available about these products, they apparently consist of a Web cache that is bundled with a media player. There are numerous works on proxy caching mechanisms for Web objects (e.g., [4, 5]). However, due to the larger size of multimedia streams compared to Web objects and the streaming nature of the multimedia delivery, existing Web proxy-caching schemes have been proven inefficient for multimedia streams.

A class of caching mechanisms for multimedia streams propose to cache only selected portions of multimedia streams to improve delivered quality. Clearly, these techniques do not greatly decrease the load on the server (or the network), but they can do so significantly.

Work in [6] presents a caching architecture for multimedia streams consisting of a cooperative group of proxies. Work in [7] investigates the layered video caching problem, using an analytical revenue model based on a stochastic approach. The authors develop several heuristics to decide which layers of which streams should be stored in the cache to maximize the accrued revenue.

Works in [8,9] refer to the improvement of the quality of the delivered to the client multimedia stream even if the connection between the remote server and the client is behind a bottleneck (e.g., a wireless link). The solution to this problem is multimedia proxy caching. A proxy cache resides close to the group of clients. Requested streams are always delivered from the original servers through the proxy to clients, thus the proxy is able to intercept and cache these streams in order to use them to satisfy future client requests.

Work in [12]-[14] is mainly interested in partial caching, as storing the whole content of a few multimedia streams would exhaust the capacity of a conventional proxy cache. Specifically, in [12] only an initial portion (referred to as prefix) of each video is cached in the proxy cache in order to improve the client-startup latency.

In [14], partial caching is used by segmenting the multimedia files in segments of variable size forming a pyramid and a portion of the cache capacity is dedicated to cache only the prefixes of each video, while the remaining part of the cache is used for the latter segments. Caching an appropriate size prefix of a video relieves clients from delays and delay jitter on the server-proxy path while proper cache admission and replacement policies can be used to achieve high cache performance even when video popularity changes occur.
In [13], the video segmentation approach is studied in general and the tradeoff between cache performance and responsiveness in popularity changes is examined for fixed and variable segmentation schemes. In contrast to [14], the authors in [13] do not adopt the idea of dedicating a portion of the cache capacity for caching prefixes of the various multimedia streams.

1.3 Roadmap of the thesis

The rest of the Thesis is organised as follows. In Section 2, we present an overview of system model and performance metrics that we have used in our experimental study. More precisely, Section 2.1 present the problem definition, Section 2.2 introduces our two system models that have been used in the experiments, Section 2.3 introduces the performance metrics and finally in Section 2.4 we present methods that we propose in order to store multimedia content in the client and the system caches.

In Chapter 3, we present and analyse the results from our experiments. More specifically, Section 3.1 presents the simulated models and the examined scenarios, Sections 3.2 and 3.3 contain the results of the performance study of the first and second method, respectively. Finally, in Section 3.4 we analyse the sensitivity of our schemes to the number of video-clips stored in video server. In Section 4, we outline our conclusions and provide some directions for future work.

CHAPTER 2

2.1 Problem Definition
2.2 System Models
2.3 Performance Metrics
2.4 Video Content Placement in Caches
2.5 Overview

2.1 Problem Definition

In this Thesis, we study the communication between a central server and a population of mobile clients via a high capacity wireless communication channel. We try to develop efficient ways so that the system can support the communications needs of the applications run by the mobile users. One demanding application (with regard to the volume of transmitted information and delay sensitivity) is the delivery of video-clips on demand from a central video-server located at the fixed network to the mobile users over the wireless channel.
We assume that video-clips have short duration (e.g., 2-3 minutes) because such clips are very popular today (they correspond, for example to music clips, film trailers, e.t.c).

We consider that mobile devices are equipped with cache memory, which is feasible today, in order to improve the Quality of Service (QoS) delivered to the mobile clients and the overall system performance. In order to achieve the above mentioned two goals, in our work we try to develop efficient methods for video-clips' placement in the client's cache. The scheduling we follow for delivering the requested information over the wireless channel depends on the way we decide to store video-clips in the client's cache. This scheduling takes into account the following: i) the client's cache size, ii) the way in which we decide to give preference for storing a whole video-clip versus a small initial part of it (prefix) in the cache in order to achieve high system performance and at the same time reduce the startup delays for playing the streaming-videos in the mobile devices, and iii) the popularity of the requested video-clip. We also examine an extension of the above system which is more scalable than the original and technically feasible in the next few years, as the various technologies related to mobile communications are developing. This extension assumes an increased video-clips size (up to 30 minutes, short duration movies) and from our study we can draw useful conclusions towards the further development of such systems. In this case, we introduce in our system a second storing facility (with much higher capacity than the clients cache). This storing facility is used for video clip prefixes which are not stored in the clients' caches. It is essential in achieving the desired system QoS because of the significant increase in the video-clip duration. This storing facility is assumed located at the fixed network, very close to the access point of the wireless channel.

2.2 System Models

In our study, we have examined two different models. The first refers to the case of short video-clip duration, in which we only assume the existence of clients' caches. In the second case, we consider an extended model in which we assume video-clips of much longer duration and we have added a storing facility located near the access point for storing video-clip content so that the system can achieve the following goals:

1) The information that is transmitted over the wireless channel can withstand at most a 0.01 % loss (information loss occurs whenever the delivery of a video packet exceeds its delay bound), without affecting the quality of displaying the video at the requesting mobile device [15].

2) The client startup-delay should be kept as low as possible (the startup delay is defined as the time from the instant a video-clip is requested by a user until the instant the playback of the particular video-clip starts at the requesting user).

2.2.1 First System Model

We consider a system consisting of a video server serving a population of mobile clients over a wireless channel that connects them with the server (see Figure 1).
The video server is assumed to be located close to the interface of the wireless channel to the fixed network. The video server stores a big database of video-clips (over one thousand). The clients send requests to the server informing it which video-clips they need to display in their monitors and the scheduler, located at the server, is responsible for transmitting this information to the clients over the wireless channel in an efficient manner.

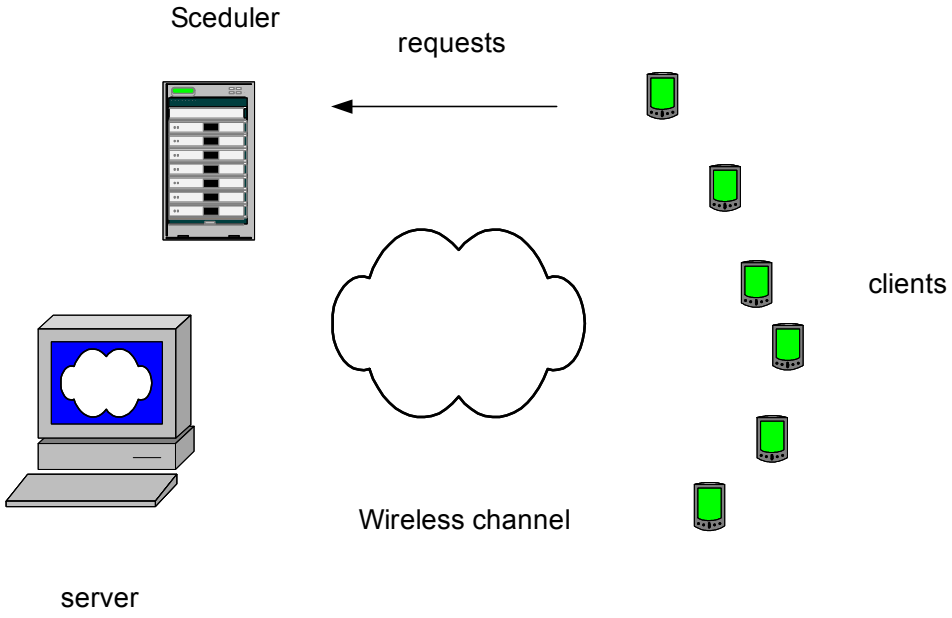

Figure 1

The broadcast channel connecting the server to the clients is assumed to be a high capacity wireless channel (e.g., of transmission speed equal to 9.045 Mbps, from [15]). All the information requested by the clients is transmitted by the server over this channel (downlink channel). In our study we focus on the downlink channel transmissions, and we do not explicitly consider the transmissions of the client requests to the server. For the latter, we assume that they arrive at the server without delays (i.e., as soon as they are generated by the corresponding clients). This is a reasonable assumption, since the information contained in a request is small compared to the video information transmitted on the downlink channel, and provided that the appropriate bandwidth is apportioned to the uplink channel the delays of the client requests will be low and they will not have a significant effect on the performance of the system.

The downlink channel, hereafter referred to as the channel, time is divided into frames (or slots) of fixed duration. The frame duration is selected to be equal to the duration of a video frame therefore, assuming a video encoding with 25 frames/sec the channel frame duration is equal to 40 msecs. Consequently, within each channel frame 45,225 Kbytes of information can be transmitted to the clients. If the aggregate amount of information that needs to be transmitted to the clients within a channel frame is larger than the channel frame size the

difference is assumed lost due to the time sensitive nature of the video streaming applications. An upper bound of 0.001 on the fraction of lost video traffic volume is assumed, in order to provide the required Quality of Service (QoS) to the clients.

Client mobile devices are equipped with a cache, which is used for storing prefixes (beginning portions) of, or entire video-clips. The size of the cache varies in our study; it can be as low as 140 Mbytes and as high as 5 Gbytes. If a client requests a video-clip that is contained in its entirety in the clients cache, then the client is served directly from its cache and no request is sent to the video server. Also if a prefix of the requested video-clip is contained into the clients' cache, the video-clip streaming starts without delay and a request is sent to the video server for the part of the video clip not contained in the client's cache. As a result of the above, the user start-up latency and the amount of information that is transmitted over the channel are reduced. These, help make the system more scalable with respect to the number of clients and the aggregate client request rate that can be supported.

2.2.2 Second (Extended) System Model

In the second case, we consider a system model similar to the model described in section 2.2.1 (Figure 2). Now, however the system has another cache. We refer to this cache, as the Access Point (AP) cache because it is located close to the point that the wireless channel is connected to the fixed network. The latter cache is considered to be of much larger capacity than that of the clients' cache, i.e., its capacity ranges from 1Gbyte up to 5Gbytes in our study. Furthermore, we now consider that the video server is located further away from the mobile clients than what it was assumed in the first system model and that it is connected to the wireless channel AP via a wide-area network (WAN, e.g. the Internet).

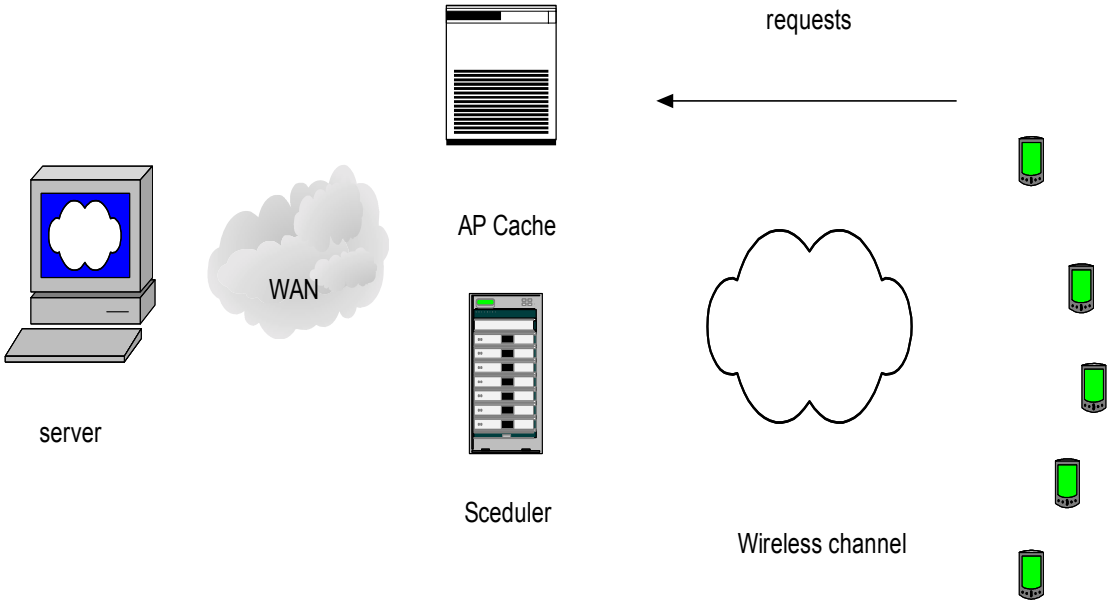

Figure 2

The extended model maintains all the other characteristics of the first system model (e.g., broadcast channel capacity, frame duration of downlink channel, video encoding, e.t.c).

The operation of the system under the second (extended) system model works as follows: if a client requests a video-clip that is contained in its entirety in the client's cache, then the client is served directly from its cache and no request is generated. If only a prefix of the requested video-clip is contained into the client's cache, the video streaming starts without delay and a request is sent to the Access Point (AP) cache for the part of the video-clip not contained in the client's cache. If the video-clip is contained in its entirety in the AP cache then it is sent over the wireless channel to the client, otherwise a new request is generated and sent to the video server for the part of the video-clip not contained in the AP cache. If the video-clip is not contained at all in the client's cache then a request is sent it to AP cache. If the video-clip is contained in its entirety in the AP cache then it is sent over the wireless channel to the mobile client, otherwise a new request is sent to the video-server for the part that is not contained in the AP cache.

2.2.3 Client Request Process and Video Clip Popularity Distribution

In our performance study we assume that the aggregate client request process is Poisson with mean λ requests/minute. Generated client requests are assumed to request the various video-clips according to a Zipf popularity distribution. The video-clip i is requested with probability Pi, where

$$P_i = C\ (1/i)^{\theta}, \qquad 1 \le i \le M$$

M is the total number of video-clips stored at video server,

$$c = 1 / \sum_{i=1}^{M} (1/i)^{\theta}$$ is a normalizing factor

and θ is a parameter referred to as the access skew coefficient. The distribution becomes increasingly "skewed" as θ increases. In our study we assume M = 1000 (for most of the experiments we have performed), and we consider three different values for θ; θ = 0.7 (less skewed access distribution), θ =1.17 (skewed access distribution) and θ = 1.3 (highly skewed access distribution).

2.3 Performance Metrics

The performance metrics we have used to evaluate the system behavior are:

- **System Throughput**

The first performance metric we use is the system throughput, i.e., the maximum number of requests per minute, denoted by λmax, the system can tolerate subject to the requirement that the information dropping probability does not exceed 0.001 or 0.1% in order to meet the constraints imposed by the QoS requirements of the clients. The dropping probability is defined as:

$$P_{DROP} = \frac{\text{number of dropped bits}}{\text{number of (transmitted + dropped) bits}}$$

- **Cost Function**

In order to decide which video-clips and video-clip prefixes to place in the client caches, we introduce a cost function. We assume that the system incurrs a cost to satisfy each client request and this cost is determined by the cost function. This cost consists of two different parts associated with the information transfer and the user startup latency. The first part refers to the cost incurred by the transfer of the requested information, not contained in the cache, from the video server over the wireless channel and is determined by the volume of the information to be transferred. The second part relates to the user start-up latency. Each time a client requests a video-clip, for which no prefix is contained in the corresponding cache, the system incurrs a cost, denoted by Cdelay.

To formally define the cost function we make the following assumptions:

1) Every video-clip is divided into equal size parts called segments. The cost of transferring a video segment over the wireless channel to the requesting client is denoted by Ctransfer, and in our study is assumed equal to 1.

2) For the first system model we introduce a parameter α, which is defined to be equal to the ratio:

$$a = \frac{Cdelay}{Ctransfer}$$

This parameter is assumed to be larger than one in our study, and its value shows how costly is delaying a customer's playback due to not having stored the corresponding video prefix in the client's cache, compared to the unit of cost in our system (i.e., the cost of transferring one video-clip segment from the video server, located close to the access point in the first system model, over the wireless channel).

3) For the second (extended) system model we additionally introduce a second parameter b, which is defined to be equal to the ratio:

$$b = \frac{C'_{transfer}}{C_{transfer}}$$

This parameter is assumed to be larger than one, and its value shows how costly (C'transfer) is transferring one video-clip segment, which is not placed either in the client's cache or in the access point cache, from the video server in the fixed network to the AP cache, compared to the unit of cost in our system. (i.e., Ctransfer, the cost of transferring one video-clip segment over the wireless channel to the requesting client).

2.4 Video Content Placement in Caches

We consider two approaches, for storing video-clips in the clients' caches, as follows:

1) We store entire video-clips starting from the most popular until the cache fills up. This approach is based on the fact that the video segments stored in the cache have higher popularity than the ones left out and it therefore minimizes the long term transfer part of the cost.

2) With this approach, some of the most popular video-clips are stored in their entirety after which prefixes of a number of less popular video-clips are also stored until the cache fills up. The condition based on which we decide whether a video clip will be placed in the cache in its entirety or only a prefix of it, is as follows. Suppose that we placed in the cache the prefix segment of video-clip j and that i> j.

If $(Ctransfer + Cdelay) * Pi > Ctransfer * Pj$ (1)

is true, where Pi, Pj are the popularities of video-clips i and j, respectively, then we put in the cache the prefix segment of video clip i, otherwise we put the entire video-clip j. This process is repeated by increasing the index i by one, if the above condition is true, or j by one if the above condition is false, until the cache fills up. The state of the cache at the end of the above described process is as shown in Figure 2.

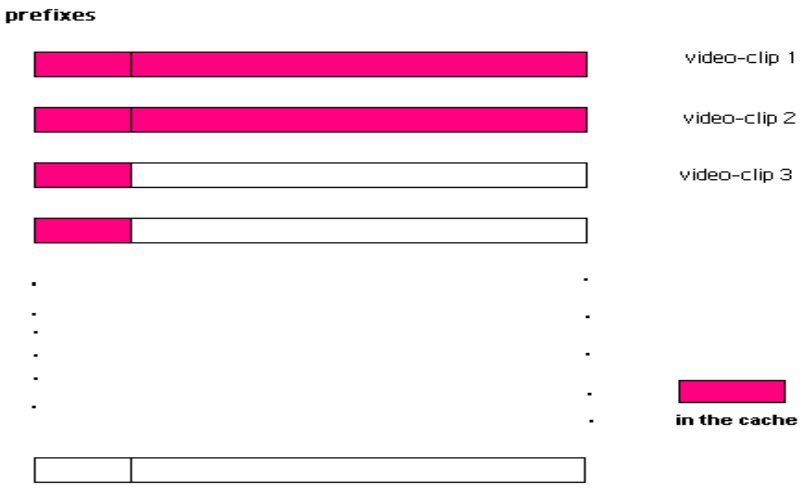

Figure 3

Some of the most popular video-clips are placed in their entirety, followed by some less popular video-clips for which only the prefix segments are placed while the remaining less popular video-clips are not placed at all in the cache.

For the second (extended) system model we have considered the user startup latency cost together with the information transfer cost from the remote video server to the client, and minimized the long-term average cost metric. The number of video-clips stored in their entirety in the cache versus the number of video-clips stored via their prefix-segment only, depends on the video-clips duration, the video segment duration, the cache size, the skeweness of the Zipf popularity distribution (parameter θ) and the parameters a and b of the cost function.

For placing video-clips in the access point cache we use an equation similar to (1). Specifically, suppose that we have placed in the access point cache the prefix of video-clip j and i > j:

$$\text{if } (Cdelay + Ctransfer + C'transfer) * Pi > (Ctransfer + C'transfer) * Pj \qquad (2)$$

is true, where Pi, Pj are the popularities of video-clips i and j, respectively, then we put in the cache the prefix segment of video clip i, otherwise we put the entire video-clip j. This process is repeated by increasing the index i by one, if the above condition is true or j by one if the above condition is false, until the cache fills up. The state of the AP cache at the end of the above described process is similar to the one shown in Figure 2.

2.5 Overview

In this section we have introduced two system models. In the first model, each mobile client is equipped with a small size cache, while in the second (extended) system model we additionally assume the existence of a larger cache located at the fixed network close to the interface with the wireless channel. We also introduce and discuss the system performance metrics we use to evaluate the performance of the system as well as, two methods for placing video-clip content in the client and system caches.

CHAPTER 3

3.1 Simulation Model and Examined Scenarios
3.2 First Case: Performance Results and Discussion
3.3 The Extended Case: Performance Results and Discussion
3.4 Sensitivity of the results to the number of video-clips stored in the video-server
3.5 Overview

3.1 Simulation Model and Examined Scenarios

The system performance is evaluated via simulations. Each computer simulation result we report is based on an average of ten (10) independent runs (Monte-Carlo simulation), each simulating 800,000 frames (approximately 9 hours of system operation). Video-clips are generated from actual video movie traces of longer duration, by randomly picking the clip's starting instant within the movie duration. The video-movies used are encoded according to the MPEG-4 standard, their characteristics are shown in Table 1, and the corresponding traces are taken from [11].

Table 1	
Characteristics of MPEG 4 movies	
MeanBitRate (bits/sec)	**963 866**
PeakBitRate (bits/sec)	**1 982 200**
Peak Bit Rate /Mean Bit Rate	**4. 99**
Variance of the Bit Rate	**4 701 126**
Cov	**1.092**
Compression	**25.5**

As mentioned in the previous chapter, clients request the various video-clips stored in the database according to a Zipf distribution with parameter θ. Table 2 shows the probabilities of the ten most popular clips (hot clips). As we can see from Table 2, as the parameter θ increases, the distribution becomes more "skewed". We focus on the popularity of the most popular video-clip for three different values of θ, and we get the following results: 4,22% when θ=0.7, 19,43% for θ = 1.13 and 28,27% for θ = 1.3 . Furthermore, the ten most popular video-clips have cumulative request probability equal to 16,71% for θ = 0.7, 50,74% for θ = 1.13 and 65,07% for θ = 1.3 .

Video Clip Index	θ = 0.7	θ = 1.17	θ = 1.3
1	0.0422	0.1943	0.2847
2	0.0259	0.0888	0.1156
3	0.0195	0.0561	0.0682
4	0.0159	0.0405	0.0469
5	0.0136	0.0315	0.0351
6	0.012	0.0256	0.0277
7	0.0108	0.0215	0.0226
8	0.0098	0.0185	0.019
9	0.0090	0.0162	0.0163
10	0.0084	0.0144	0.0146

Table 2: Popularities of requested clips

In our simulations we examine two different cases: in the first all video-clips have duration uniformly distributed between 1.5 and 3 minutes (first system model case), while in the second all video-clips have much longer constant duration equal to 30 minutes (in this case we use the second (extended) system model). In both of these cases we examine and compare the two different methods of storing video-clips in the caches as discussed in the previous chapter. We simulate the system under different sizes of client and AP caches, different numbers of video-clips stored in the database and different duration of video-clip segments.

3.2 First Case: Performance Results and Discussion

In this case, because of the small duration of video-clips we only use a cache located at each client. Subsequently, we examine the following two scenarios:

Scenario A: In this scenario, we examine the behavior of our system for several sizes of the client's cache in the range from 140Mbytes to 5 Gbytes. Client's cache stores the most popular video-clips in their entirety until it fills up.

Table 3 presents the various cache sizes we have considered together with the corresponding number of the most popular video-clips stored in the cache.
Furthermore, in Figure 3 we show the popularity mass of the video-clips stored in the cache versus the size of the clients' cache for the three different values of the access skew coefficient θ, considered in our study.

Table 3

Size of Client's Cache	Movies in Cache
Without cache	-
140 Mb	20
1Gb	152
2Gb	296
5Gb	751

The popularity mass is the probability that a client request can be served from the cache, and it increases with the cache size since more video-clips are then placed in the cache.

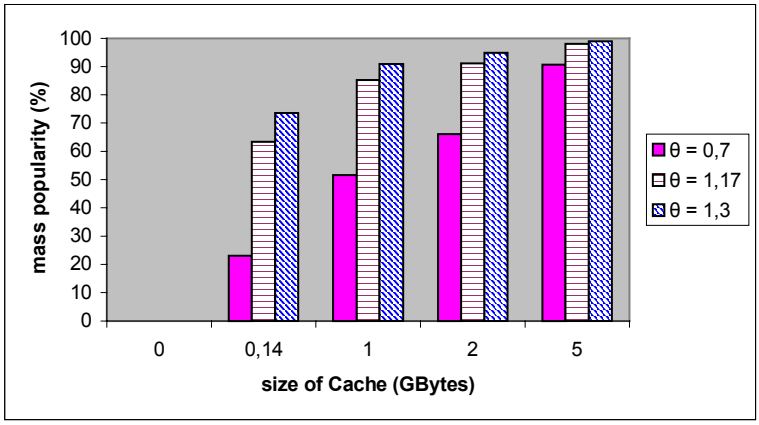

Figure 3

For a specific value of the cache size, the popularity mass of the video-clips stored in the cache increases with the skeweness of the access distribution (higher θ values) because the request probabilities for each of the stored video-clips become larger. In the case of the 5 Gbytes cache, the popularity mass is above 90% for all the values of the parameter θ we have examined, although only 75% of the entire database is placed in the cache.

Figure 4 presents the system throughput (i.e., the maximum request arrival rate, λmax, sustained by the system), versus the size of the clients' cache for the three values of the access skew coefficient examined.

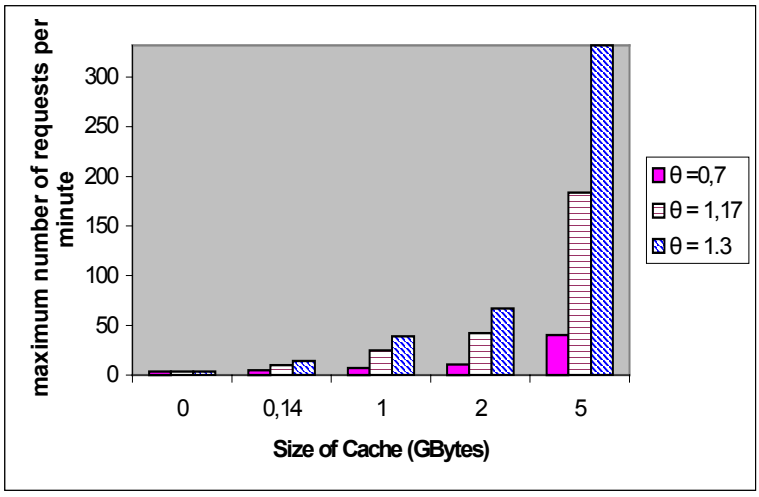

Figure 4

We notice that the system throughput increases with the cache size, and that for a given cache size the system throughput increases with the skewedness of the access distribution because of the increased popularity mass of the video-clips stored in the cache (as explained in the discussion of the results shown in Figure 3). The system throughput is equal to 3.3 requests/ min when there is no cache, irrespective of the access skew coefficient θ, 4.5 requests/min, when the cache size is 140 Mbytes and θ = 1.3, 42 requests/min when the cache size is 1 Gbyte and θ =1.3, and it increases up to 332 requests/min for a 5 Gbytes cache and θ = 1.3.

Scenario B: In this scenario, we assume that the size of the client's cache is fixed to a relatively small value (e.g., equal to 140 Mbytes). We further assume that video-clips are stored in client's cache following the second method introduced in the previous chapter (cost-based content placement). As a result of this method, in the client's cache some video-clips

are stored in their entirety, while for others only prefixes are stored. In our study we have considered two values for the duration of the video-clip segments namely, 5 secs and 12 secs.

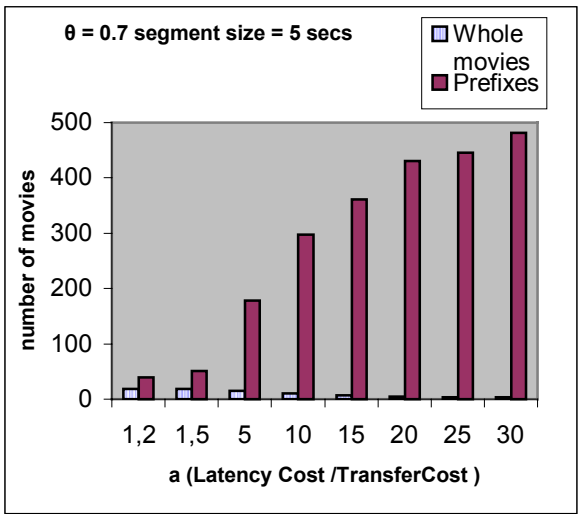

Figure 5

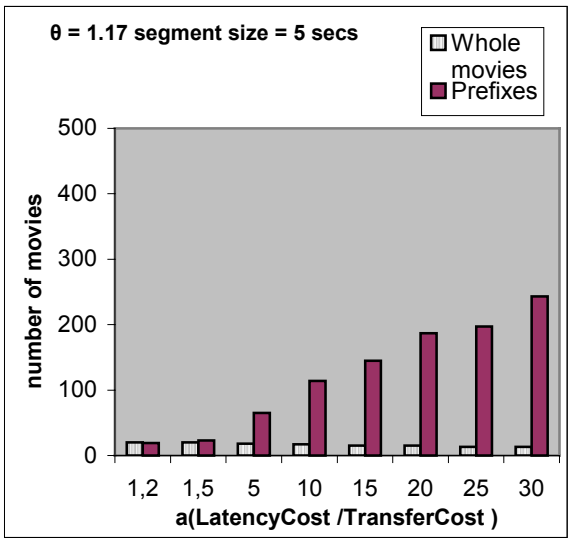

Figure 7

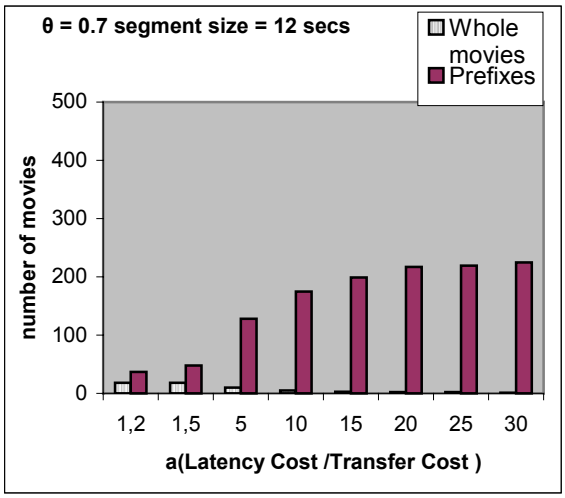

Figure 6

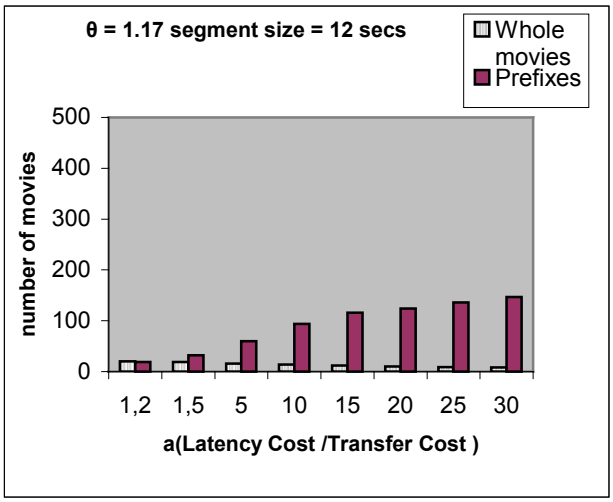

Figure 8

From Figures 5 to 8, we see that the number of video-clip prefixes stored in the client's cache increases as the parameter a increases. At the same time, the number of entire video-clips stored in the client's cache decreases. This is expected because of the form of equation (1) in Chapter 2, which is used in order to decide whether an entire video-clip or a prefix of it will be stored in cache. As the parameter a increases, our method puts more weight to the delay cost, that's why the system stores as many video-clip prefixes as it can in order to reduce the clients' start-up latencies.

Comparing the results in Figures 5 and 6, we conclude that for a given value of the access skew coefficient θ as the duration of video-clip segment increases the number of entire video-clips stored in the cache doesn't change while the number of video-clips prefixes (stored in the cache) decreases by approximately 50%. Furthermore, for the same video-clip segment duration, as we increase the parameter θ its clear from the results in Figures 5 and 7 and in Figures 6 and 8, that the number of entire video-clips stored in the cache decreases while the number of video-clip prefixes stored in the cache decreases dramatically (as a result of the impact of the more skewed distribution of video-clip popularities). This reduction becomes more rapid as the parameter a increases too.

The system throughput versus the parameter a, for the three values of the parameter θ (θ = 0.7, 1.17, and 1.3) and for the two values of the video-clip segment duration (5 secs and 12 secs, respectively) is shown in Figures 9 through 12. In these figures we show the throughput results for both methods of video-clip placement (cost-based and entire video-clips only).

Comparing the results in Figures 9 and 10, we observe that for θ = 0.7, the system achieves the same throughput when we use entire video-clips only placement method. This is expected, since in this case we store only entire video-clips and because the size of the client's cache is constant and equal to 140 Mbytes, the number of entire video-clips stored in the cache remains the same in both cases. If we follow the cost-based placement, we observe that for a specific value of the parameter α the system throughput is slightly increased and for small values of the parameter α in case we use a longer video-clip segment duration (i.e., a= 1.2 and 1.5). As the parameter a takes higher values (i.e., a >= 5), no significant difference in throughputs is observed.

Comparing the results in Figures 9, 11 and 12 we observe that for a fixed value of segment duration (e.g., equal to 5 sec), as the parameter θ increases the system throughput increases too for both content-placement methods. When a = 1.2, the system throughput takes the values of 4.6 requests/min for θ = 0.7, 10 requests/min for θ = 1.17 and 13.8 requests/min for θ = 1.3. The same behavior is observed as the parameter α increases for all values of parameter α, as parameter θ increases too.

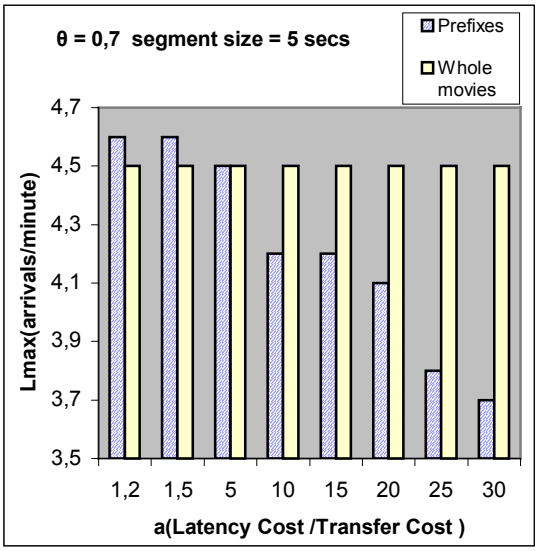

Figure 9

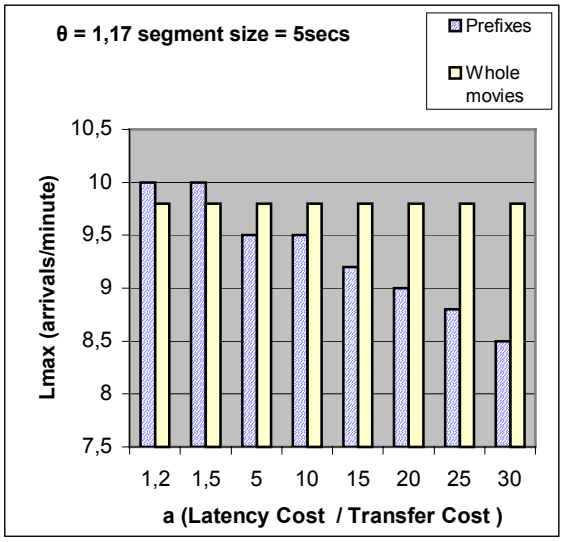

Figure 41

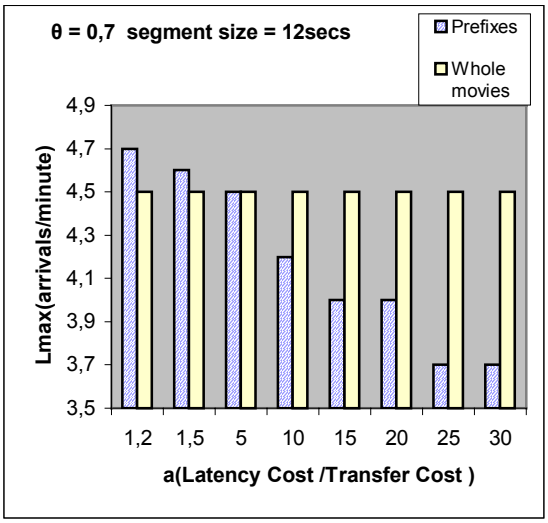

Figure 10

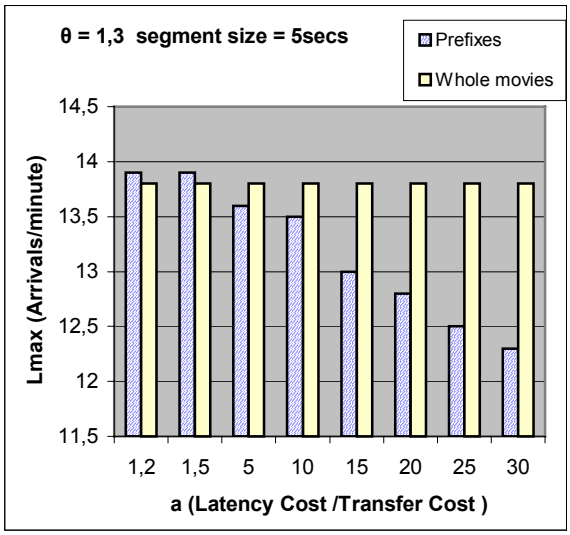

Figure 15

From the results in Figures 9 through 12, we can see a demonstrated trend. When the parameter a is large, the startup delay part of the cost function dominates. As a result, the cost-based content placement puts many video-clip prefixes in the cache, so that the overall cost is minimized. However, this content placement makes necessary the information transfer over the wireless channel for each requested video-clip (either in its entirety or for all of its segments except the first one) and therefore leads, to lower channel throughputs.

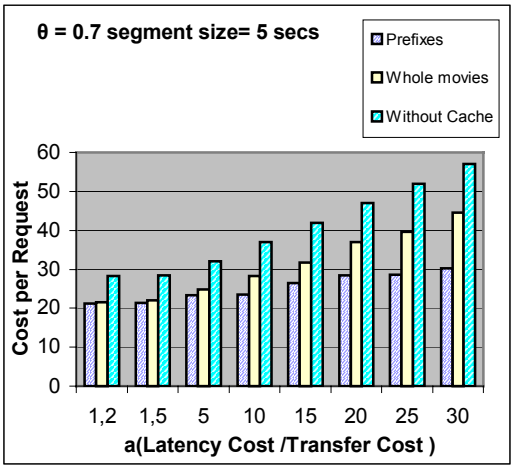

Figure 16

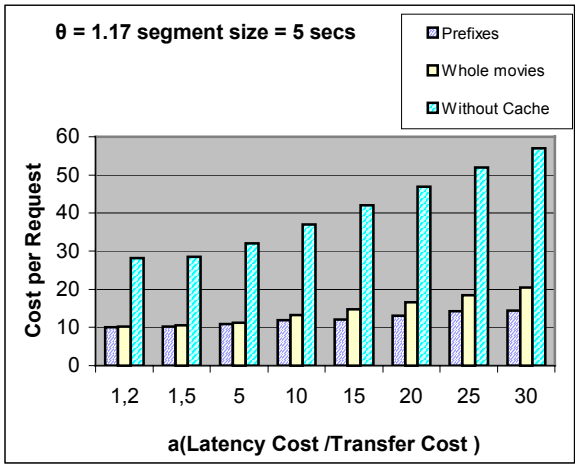

Figure 14

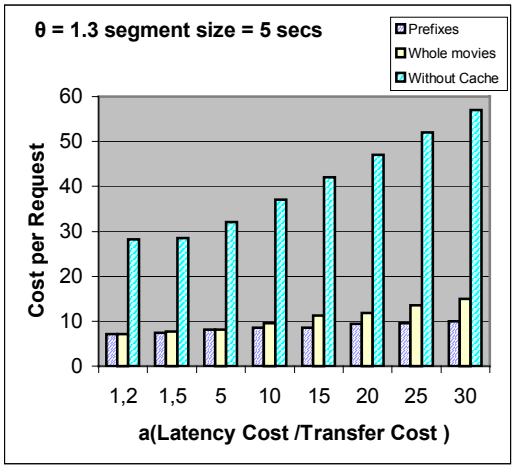

Figure 15

Finally, Figures 13 through 15, show the total incurred cost per request versus the parameter a for various values of the parameter θ. We show results of the system behavior for a fixed value of video-clip segment duration equal to 5 sec and fixed size of the client's cache equal to 140 Mbytes, for the two different ways of content placement in the cache. We also show the corresponding results, when no client cache is available. As expected, the best performance is achieved with the cost-based placement method and the worst when there is no cache available at the clients. Furthermore, as the parameter a increases, we observe that the cost-based content placement method incurrs lower costs and that the difference between the costs incurred by the entire video-clip and the cost-based placement methods becomes larger. Finally, as θ increases, the cost per request declines. This is expected because the access popularity distribution becomes more "skewed" and the cost-based placement method is designed to take advantage of this fact.

3.3 The Second Case: Performance Results and Discussion

In this case we assume that all the video-clips have longer duration than in the first case. Specifically, we assume that the video-clips duration is fixed equal to 30 minutes. In addition, to the client's cache, we use another cache of larger capacity located at the Access Point (i.e., the point at which the wireless subnetwork is connected to the fixed network). In the sequel, we examine two scenarios as follows.

Scenario A: This scenario assumes the existence of only the client's cache with a fixed relatively small value equal to 140 Mbytes. The content placement in the client's cache follows the cost-based method. We examine this scenario in order to evaluate the caching scheme performance when the duration of the video-clips is significantly increased.

Table 3 presents the system throughput (i.e., the maximum request arrival rate, λmax, sustained by the system) versus the skew coefficient θ of the request distribution.

	θ = 0.7	**θ = 1.17**	**θ = 1.3**
λ max (req/min)	**0.25**	**0.27**	**0.3**

Table 4

We observe that the system throughput doesn't change dramatically with the skewness of the access distribution (the only difference is a small increase versus the increment of skewness), because of the fixed placement of the content in the client's cache (1 whole video-clip and 206 prefixes) for all values of the access skew coefficient. The fixed placement occurs because of the small size of the client's cache (140 Mbytes) and the small values of the parameter a that we have considered with maximum examined value equal to 30, as it will be seen in the next section. The throughput behavior shown in Table 4 can be explained because with the increment of the access skew coefficient and as the content placement remains the same, the mass popularity of the video-clips which are contained in the client's cache increases significantly as we can conclude from Table 2 in Section 3.1.

Table 5 shows the minimum value of parameter *a*, for each case of parameter θ values beyond which the content placement in the cache contains only video-clips prefixes. (notice that the client's cache cannot store two entire video-clips because of the size of the video-clips). As we can see the minimum value of parameter *a* when θ = 0.7 is close to but larger than the maximum examined value of *a* in our study, however when θ takes higher values (e.g., 1.17 and 1.3, respectively) the minimum value of parameter *a* differs a lot from the examined values, and this explains why the content placement in the clients' caches does not change.

θ = 0.7	**θ = 1.17**	**θ = 1.3**	
42	**509**	**1018**	**a**

Table 5

Figure 16 shows the average total incurred cost per request versus the parameter a for various values of the access skew coefficient θ. As expected, the placement of content in the client's cache results in lower cost than that incurred when there is no cache available to the clients. Furthermore, as the value of the parameter a increases the difference between the incurred costs between the cases with and without cache becomes larger. The difference between the incurred costs for the different values of the access skew coefficient examined here, also becomes larger as the parameter a increases. This can be explained because given the static content placement, as the parameter a increases the total incurred cost increases as well, since the cost of each video-clip not contained in the cache increases accordingly.

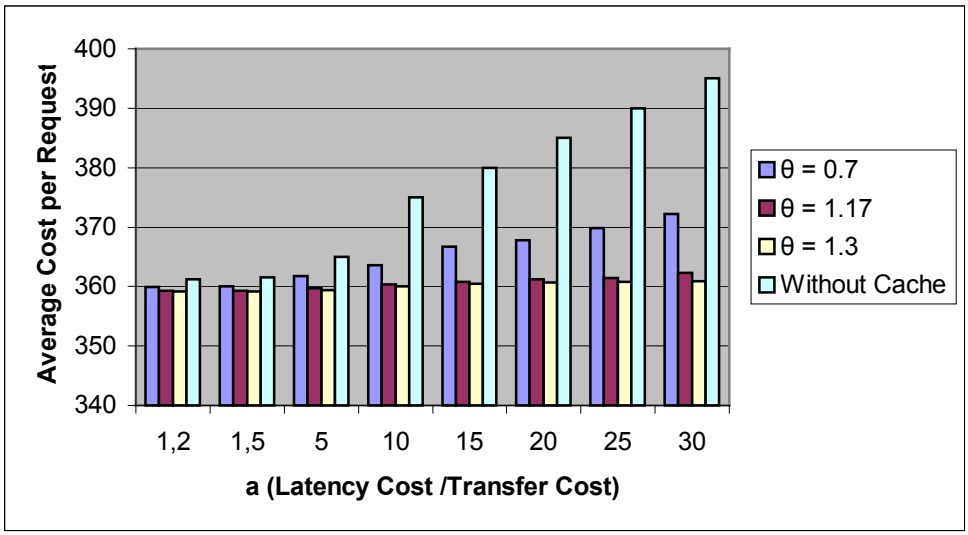

Figure 16

Scenario B: In this scenario, the size of the client's cache and the method of content placement in it remain the same with those in scenario A. Furthermore, we assume a cache at the Access Point (AP) with a size varying between 1Gbyte and 5Gbytes and we use the content placement method described in section 2.4 (see equation (2)) for storing video-clips in that cache. In addition to the parameter *a* we now also have the parameter *b*, which denotes the cost paid by the system in order to transfer a video-clip segment not contained in the AP cache from the corresponding video server to the Access Point cache.

AP cache Size = 1Gbytes	θ = 0.7	θ = 1.17	
	11w. and 310 p.	11w. and 310 p.	For all cases

AP cache Size = 2Gbytes	θ = 0.7	θ = 1.17	
	22 w. and 422 p.	22 w. and 422 p.	a = 1.2 and b = 1.5
	22 w. and 422 p.	22 w. and 422 p.	a = 1.2 and b = 2
	21 w. and 725 p.	22 w. and 422 p.	a = 20 and b = 1.5
	22 w. and 422 p.	22 w. and 422 p.	a = 20 and b = 5

AP cache Size = 5Gbytes	θ = 0.7	θ = 1.17	
	56 w. and 384 p.	56 w. and 384 p.	a = 1.2 and b = 1.5
	56 w. and 384 p.	56 w. and 384 p.	a = 1.2 and b = 2
	53 w. and all p.	55 w. and 598 p.	a = 20 and b = 1.5
	55 w. and 598 p.	56 w. and 384 p.	a = 20 and b = 5

Table 5

The system throughput results are the same with those is given in Table 3, as we have used the same size and the same placement method for the clients' caches, which determine the system throughput. The content placement in the client's cache is the same with the one in the previous scenario (e.g., 1 whole movie and 206 prefixes). The content placement at the access point cache depends on the cache size and the access skew coefficient θ, and the corresponding results are given in Table 5, where w. stands for entire clips and p. stands for clip prefixes.

Figures 17,18 and 19 show the percentage of decrease in the average incurred system cost per request compare to the case without cache as a function of the parameters *a* and *b* and for various values of the AP cache size. We assume that the client's cache size is equal to 140 Mbytes.

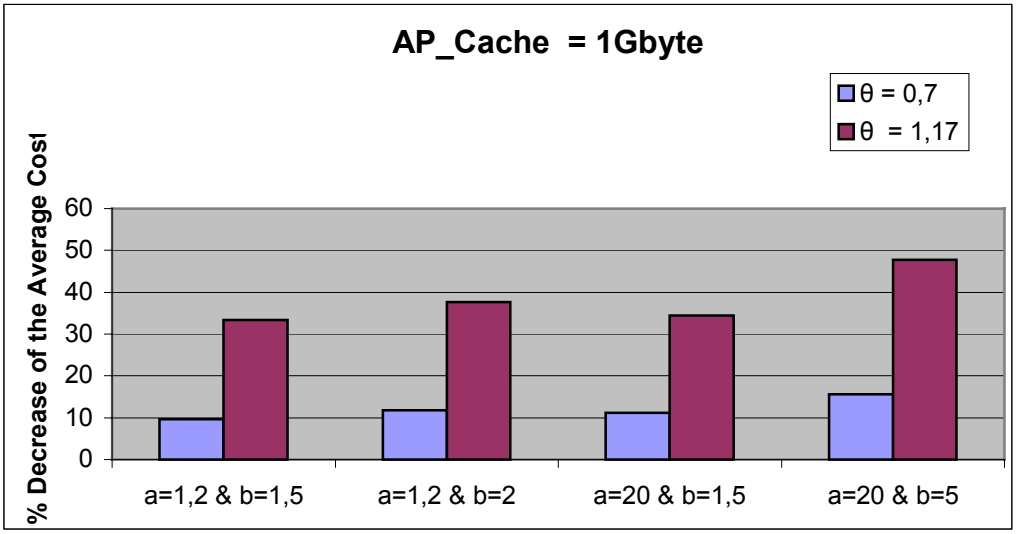

Figure 17

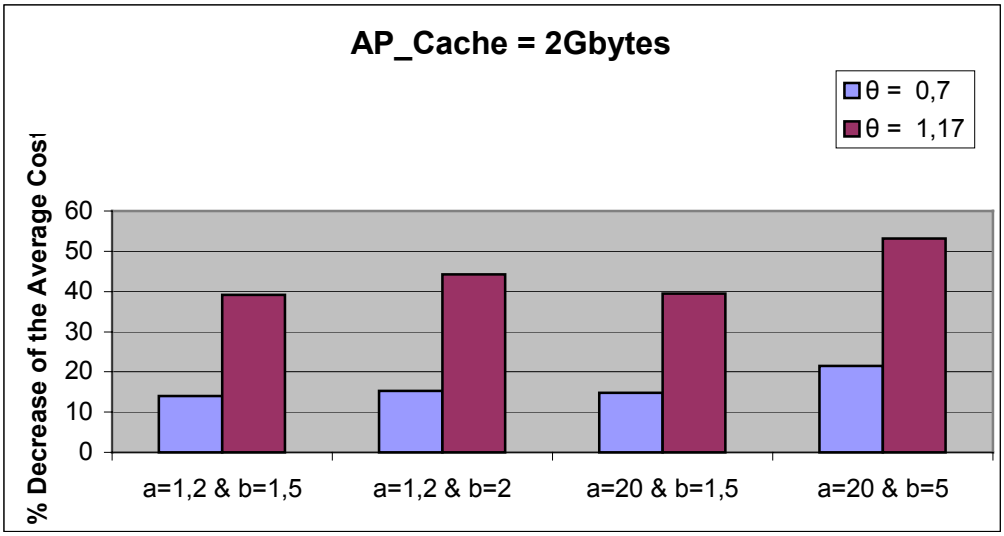

Figure 18

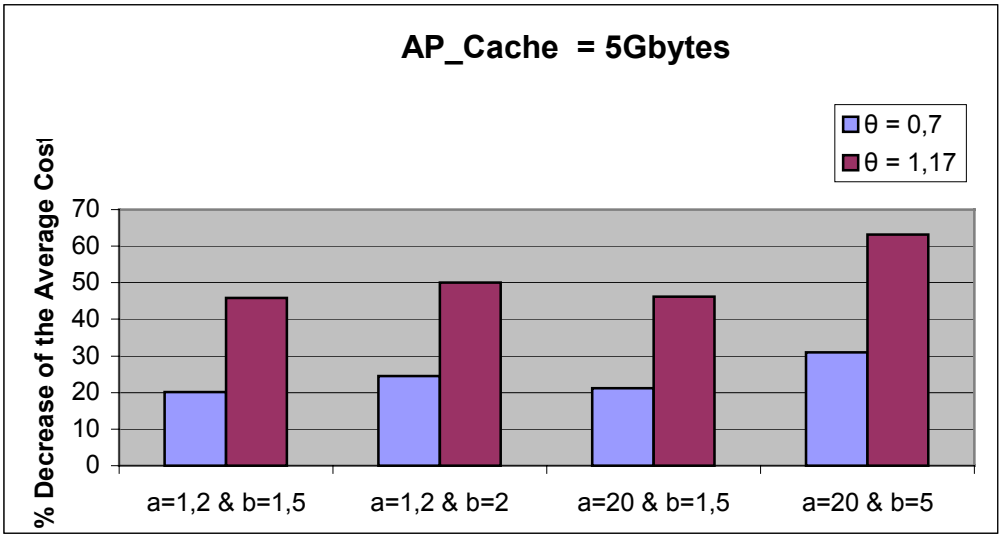

Figure 19

From the results in these figures we observe that the average cost decrease for each case is considerable. Especially, this gain becomes larger as the access skew coefficient θ, and the parameters *a* and *b* increase. Furthermore, this gain becomes larger as the size of the AP cache increases. When the AP cache size is 1 Gbyte and the values of the parameters *a* and *b* are 1.2 and 1.5, respectively, the percentage decrease of the average cost is 9.6% for θ = 0.7, it becomes 14.03% when the AP cache size is 2 Gbytes and 20.2 % when the AP cache size is 5 Gbytes.

If we look carefully into the results in the preceding three figures, we find that as the values of the parameters a and b increase, significant cost improvement is observed in the system. This improvement depends mostly on the value of the parameter b, this becomes clear if we compare the cases with values a = 1.2 & b = 1.5 and a = 1.2 & b = 2, and a = 1.2 & b = 1.5 and a = 20 & b = 1.5 for all the different cases of access skew coefficient and size of AP cache values that we have examined, in our study.

Finally, Table 6 shows the values of the average total incurred cost for each of the four different parameter value cases considered above when no cache is available at the Access Point and when the client's cache size is equal to 140 Mbytes.

Values of parameters a and b	Average Cost per Request without Cache at the AP
a = 1,2 & b = 1,5	901,2
a = 1,2 & b = 2	1081,2
a = 20 & b = 1,5	920
a = 20 & b = 5	2161

Table 6

3.4 Sensitivity of the Results to the Number of Video-Clips Stored in the Video Server

First Case: Figure 20 shows the system throughput versus the number of video clips in the video database using only the client's cache, and assuming that all video clips have duration uniformly distributed between 1.5 and 3 minutes. From the results in the Figure, we observe that the system throughput decreases as the number of clips in the video server increases something that is expected since the fixed size of client's cache causes a smaller mass popularity of video-clips to be stored in the client's cache. Of course, the system throughput is higher when we use a cache than without one.

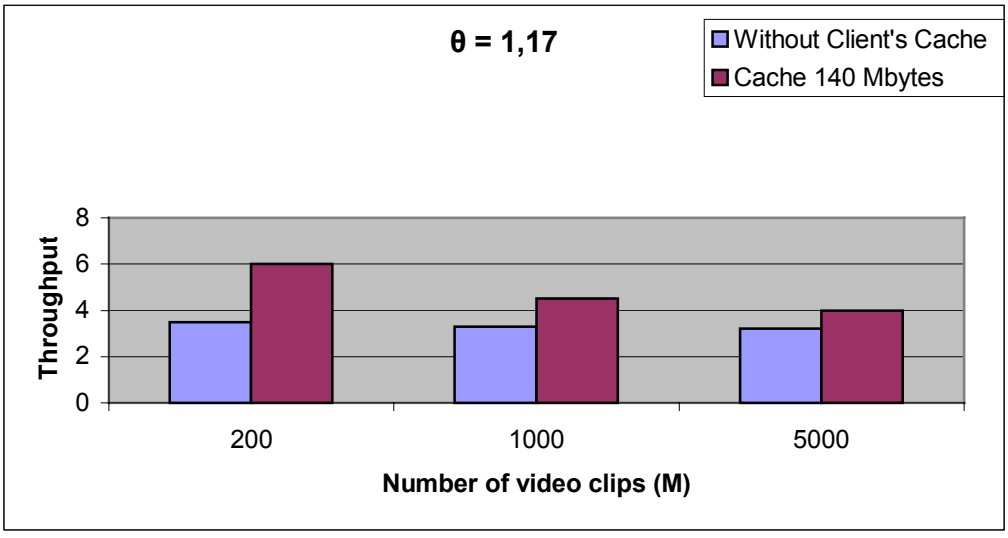

Figure 20

Figure 21 presents results of the average total incurred cost per request versus the number of video clips for two specific values of the parameter *a* (namely, the minimum and the maximum values we have used). As expected, the total average cost per request for a specific value of the parameter *a*, increases as the number of video clips in the video server increases.

This behavior can be again explained as before. Given the fixed size of the client cache as the number of video-clips increases the popularity mass of the content stored in the cache decreases and causes this behavior.

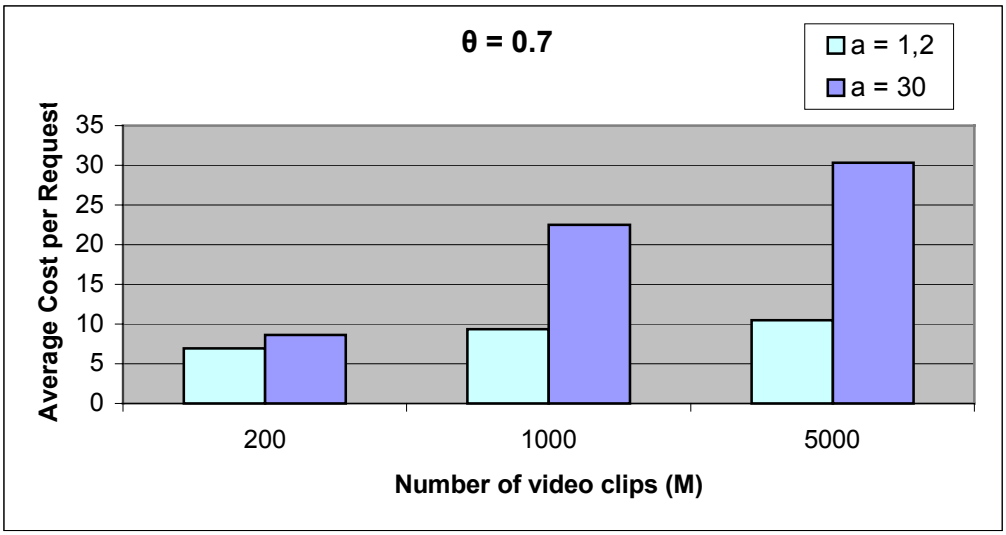

Figure 21

Second Case: Figure 22 shows the system throughput versus the number of video clips stored in the video server using an AP cache of size equal to 2 Gbytes in addition to the client's cache with size equal to 140 Mbytes. The system throughput decreases with the number of video clips. This behavior can be explained in a similar way with the results in Figure 20. The duration of the video clips is assumed constant, equal to 30 minutes.

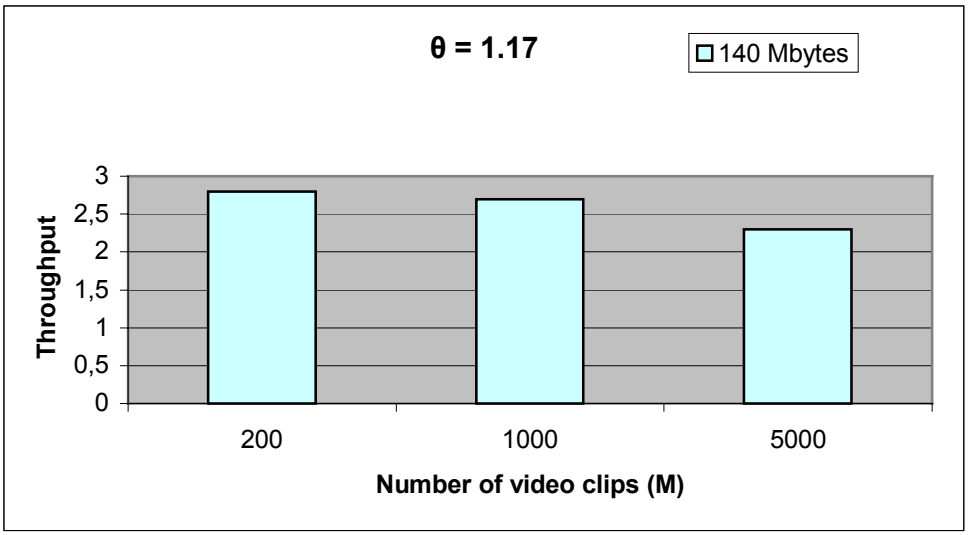

Figure 22

Finally, Figure 23 shows the average incurred cost per request versus specific values of the parameters a and b and for various number of video clips using an AP cache of 2 Gbytes and client's cache of 140 Mbytes. For each pair of parameter a and b values we observe that the average incurred cost per request increases with the number of video clips. This trend is more profound as the value of the parameter b increases.

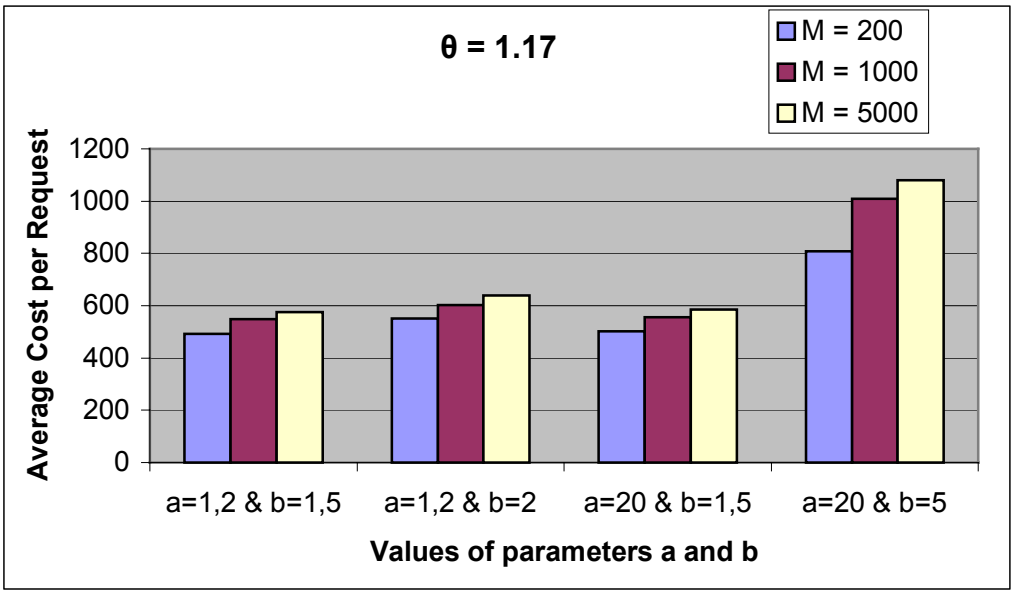

Figure 23

3.5 Overview

In this Chapter we present and discuss the results of our simulation study. In the case of the first system model, where caches are only available at the client mobile devices and video durations are short (e.g., between 1.5 and 3 minutes), we have studied the achieved system throughput for various client cache sizes ranging from 140 Mbytes to 5 Gbytes and, for various values of the skew coefficient of the popularity distribution. Subsequently, we fixed the client cache size to a relatively small value (140 Mbytes) and we studied the content placement that results from the two methods we have introduced (whole videos and cost-based placement) and the achieved system throughput as a function of the parameter α, which denotes how costly is delaying the playback of the requested multimedia stream to a client (because no prefix of the stream is cached) compared to the cost of transferring one segment of the video from the multimedia server to the client.

In the second part of this Chapter, we first examined again the first system model with a client cache capacity of 140 Mbytes and video duration of 30 minutes, in order to evaluate the system performance when the duration of the multimedia streams is significantly increased. Subsequently, we introduced the proxy cache located close to the AP of the wireless subnetwork with varying size between 1 Gbyte and 5 Gbytes, and we evaluated the system throughput and incurred average cost per client request for various values of the parameters α and b (where the parameter α has been defined earlier, and the parameter b denotes the cost paid by the system in order to transfer a video segment not contained in the AP cache from the corresponding remote video server to the AP cache).

Finally, in the third part of this chapter we present results of a sensitivity study of the system throughput and incurred average cost per client request to number of videos in the multimedia database. This number varies between 200 and 5000, while it's default value in the all the results presented in the first two parts of this Chapter was 1000.

CHAPTER 4

4.1 Conclusions
4.2 Ideas for Future Work

4.1 Conclusions

Our performance study of the first system model has shown that the existence of a cache in the clients' mobile devices can significantly improve the system performance and the startup delays experienced by the clients requesting the playback of video-clips from a video server over a high capacity wireless channel. As the size of the cache increases the system throughput increases as well. Furthermore, for a given cache size the system throughput increases as the video-clip popularity distribution becomes more skewed since then the clients' caches store content of higher popularity mass compared to the case when the video-clip popularity distribution is less skewed. In such case a client request has higher probability to be serviced directly from client's cache than from the video server.

Comparing the two examined cache video-clip placement methods we conclude that the cost-based content placement method we have introduced is very interesting and deserves further study, not only because it combinely addresses system performance (i.e., throughput) and client QoS (i.e., startup latency) but also because it achieves better average request cost for a wide range of values of the parameter a ($1 <= a <= 30$) and only slightly lower throughput compared to the corresponding results of the straightforward placement method which places only entire video clips in the clients' caches.

From the second case and especially in the first scenario we conclude that as the video-clip popularity distribution become more skew, the throughput increases because although the video-clips placement is fixed in the client's cache it now has bigger mass popularity. Moreover, the total incurred cost increases as the parameter a becomes larger however, the results are significantly better compared to case without cache at all. Also for a given value of parameter a, as the access skew coefficient θ increases we obtain lower average cost per request because of the operation of the cost-based content placement which puts emphasis on storing prefixes.

In the second scenario, in which we used two caches, the results have shown similar behavior with the ones in the previous scenario. Of interest is the fact that the total incurred cost improves a lot when we use two caches compared to case without cache. This gain becomes more profound as the access skew coefficient θ and AP cache size increase.

Moreover, when the parameter a increases we obtain a bigger reduction in average cost per request than in the other cases. From this we conclude that the cost is dominated by the value of parameter α. Furthermore, we have examined the sensitivity of the system performance

with respect to the number of video-clips, and we have concluded that the behavior of the system deteriorates according to the number of video clips stored in the video server. As the number of video-clips stored in video-server increases, the system throughput decreases and the average cost per request increases.

4.2 Ideas for Future Work

There are many ways in which the work presented in this Thesis can be extended. For example, one can investigate the impact that different distributions than the ones assumed in our study for the video access popularity and the video duration will have on the performance of the system. We can use empirical distributions fitted to observations from actual request traces collected at popular multimedia servers in the Internet to model the video popularities. Another important direction for future work, deals with the development of content placement schemes operating in the clients' and AP caches capable of somehow adapting to changes in the popularities of the videos when such changes occur.

REFERENCES

[1] Inktomi Inc., http://www.inktomi.com, “Streaming media caching brief”.

[2] “Infolibria MediaMall”, http://www.infolibria.com.

[3] RealSystem Proxy”, http://www.realnetworks.com.

[4] P. Cao and S. Irani, “Cost-aware WWW proxy caching algorithms,” in Proceedings of the USENIX Symposium on Internet Technologies and Systems, Dec. 1997, pp. 193-206.

[5] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, “Removal policies in network caches for world-wide web documents,” in Proceedings of the ACM SIGCOMM, Stanford, CA, 1996, pp. 293-305.

[6] M. Hofmann, E. Ng, K. Guo, S. Paul, and H. Zhang, “Caching techniques for streaming multimedia over the Internet,” Tech. Rep., Bell Laboratories, Apr. 1999.

[7] J. Kangasharju, F. Hartanto, M. Reisslein, and K. W. Ross, “Distributing layered encoded video through caches,” in Proceedings of IEEE INFOCOM, Anchorage, AK, Apr. 2001.

[8] R.Rejaie, M. Handley, H. Yu, and D. Estrin, “Proxy caching mechanism for multimedia playback streams in the Internet,” in fourth International WWW Caching Workshop, Mar 1999.

[9] R. Rejaie, H. Yu, M. Handely, D. Estrin, “Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet,“ Proceedings of IEEE Infocom'2000, Tel-Aviv, Israel, March 2000

[11] http://www-tkn.ee.tu-berlin.de/research/trace/trace.html.

[12] S. Sen, J. Rexford, and D. Towsley, “ Proxy prefix caching for multimedia streams” in Proc. IEEE Infocom, Mar. 1999

[13] Elias Balafoutis, Antonis Panagakis, Nikolaos Laoutaris, and Ioannis Stavrakakis, “The impact of replacement granularity on video caching“, in Proceeding of IFIP Networking 2002, Pisa, Italy, May 2002, pp. 214-225.

[14] Kun-Lung Wu, Philip S. Yu and Joel l. Wolf “ Segment-based proxy caching of multimedia streams.“ In Proc. of the 10th International WWW Conference, Hong Kong, 2001

[15] D. A. Dyson and Z. J. Haas, “A Dynamic Packet Reservation Multiple Access Scheme for Wireless ATM”, *ACM/Baltzer MONET Journal, 1999*.