A Dynamic Distributed Video on Demand Service

Christos Bouras[1,2] , Vaggelis Kapoulas[1,2] , Agisilaos Konidaris[1,2] , Afroditi Sevasti[1,2]

[1] *Computer Technology Institute-CTI, Kolokotroni 3, 26221 Patras, Greece*

[2] *Department of Computer Engineering and Informatics, University of Patras, 26500 Rion, Patras, Greece*

e-mail : bouras@cti.gr

Abstract

In our days, the use of Internet services has become a part of every day life. The need for more advanced applications is increasing and solutions have to be provided in order to meet it. Multimedia information is constantly getting more popular. The connections' bandwidth though is not ready yet to cope with the rising expectations of applications such as Video on Demand (VoD), which are very consuming in terms of bandwidth. In this paper we propose a way to implement such a service over a limited bandwidth/best effort Internet based network. Our proposal consists of the implementation of algorithms and the attempt to introduce Quality of Service features on a network such as the Internet that inherently does not support such features. In this paper we propose two techniques that can serve in the implementation of an Internet based Video on Demand service. The first deals with the distribution of video titles on the several Video Servers and is called Disk Manipulation Algorithm (DMA). The second gives a dynamic nature to the service, imposes a virtual routing procedure to the system and is called Virtual Routing Algorithm (VRA).

Keywords: Video on demand, distributed systems, routing, data stripping, video server, network utilization

Relevant technical areas: Distributed Algorithms, Multimedia and Digital Libraries, Internet Computing, Web-Based Applications

Introduction

A limited but thorough research in the current situation of projects and scientific approaches on the subject of VoD services has resulted in the general remark that the contemporary existent or research systems are mainly based on either high-speed networks (ATM, Ethernet, wireless networks etc.) or over cable-TV networks. The requisition for bandwidth and quality does not allow the implementation of such demanding services as VoD over the Internet.

Moreover, current research and statistics show that Internet is heading its way towards congestion and the advanced services emerging require more bandwidth than can be offered. It has also been observed that certain areas in the Internet display high and constant levels of service, while others appear to have congestion and low quality. All the above, have discouraged all efforts for advanced services over the Internet, including VoD.

It is interesting, however, to make a short reference to those implementations of VoD services over high-speed networks or cable TV networks. In [1] the authors report their progress in developing an advanced video-on-demand (VoD) testbed, which will be used to accommodate various multimedia research and applications. The testbed supports delivery of MPEG-2 audio/video stored as transport streams over various types of high-speed networks, e.g. ATM, Ethernet, and wireless. Their current research focus covers video transmission with heterogeneous Quality-of-Service (QoS) provision, variable bit rate (VBR) traffic modeling, VBR server scheduling, video over Internet, and video transmission over IP-ATM hybrid networks. In [2], a project where a video-on-demand system was designed and implemented, is described. The system uses TCP for control messages and either TCP or UDP for the video data and is applicable over ATM or switched Ethernet networks.

To the second case of VoD systems, where VoD systems over cable-TV networks are developed, belong several implementations. Among these is the “Distributed IBC Applications for MultiMedia On Demand (DIAMOND)” project which was completed successfully in December 1995. The project developed a fully functional prototype Video on Demand system, which was tested in Helsinki, Finland and Sligo, Ireland. The project dealt with several aspects including the overall system specification and architecture, the specification of components of the cable TV network, the VoD application user interface design and the service viability. Another VoD trial system implemented over a cable-TV infrastructure that will serve 200 households has been conducted in the Science-Based Industrial Park, Hsinchu, Taiwan. It is an interactive broadband large-scale technical trial conducted by the Computer & Communication Research Laboratories, ITRI. The VoD trial system is composed of a video server, network equipment, and 200 set-top-boxes. The video server and network equipment are housed in the National Center for High Performance Computing (NCHC, Hsinchu, Taiwan) where analog cable TV equipment is also available. The set-top-boxes connect the household TV to the network and provide interactive video services to the users. Daewoo Electronics Co. Ltd. has contributed to the field of VoD services with it's newly developed interactive multimedia system capable of offering apart from VoD, telemedicine, home shopping, distance learning, videoconferencing and many other services. The system is based on DAVIC (Digital Audio-Video Council) international standards for multimedia systems. The set-top boxes produced are compatible with VoD systems based on DAVIC standards.

An interesting approach to the problem of utilization of video servers and load balancing is presented in [3]. The authors propose a resource manager of a distributed VoD server in order to take advantage of the large set of resources (disks and CPUs) to enhance the server's reliability. The proposed available-resource manager implements a load balancing strategy, which provides reliability on demand and guarantees the server's accessibility. This paper details the corresponding algorithm and presents results proving that the load balancing strategy reduces the cost of reliability in terms of resource consumption.

Our approach attempts to combine many of the features of the papers that were mentioned here. We propose two algorithmic techniques that deal with video server resource manipulation and routing of video content. These techniques address, as well as possible, the problems of load balancing and network link congestion.

The VoD Service

Brief description

The proposed VoD service can be implemented over a network the participating nodes of which, are known in advance. This network can be a part of the Internet, or some other Internet-like network. The bandwidth provided for the communication of the network nodes is considered to be specific and limited. The network servers that participate in the service can also run other services (as all Internet servers). The servers are only requested to have all the service’s software modules installed as well as the players for the video titles.

The proposed service is based on the principles of dynamic adjustment to the network features, and the distribution of the available video titles on all the servers of the network. This service has the ability to adjust itself to the changes occurring to the network that it utilizes. Such changes may be bandwidth shortages or server configuration changes. The other major characteristic of the service, which is the distribution of the video titles, are introduced as the service begins to work providing even greater speed and flexibility to the service.

Service architecture

The implementation of the proposed service requires an interface for user and administration interaction as well as access to the video titles available on the network. This interface consists of two basic modules. The first is a full access module, with which the user is able to find and watch the available video titles (user interface) and the second is a limited access module to which only the administrators of the service can have access (see Figure 1).

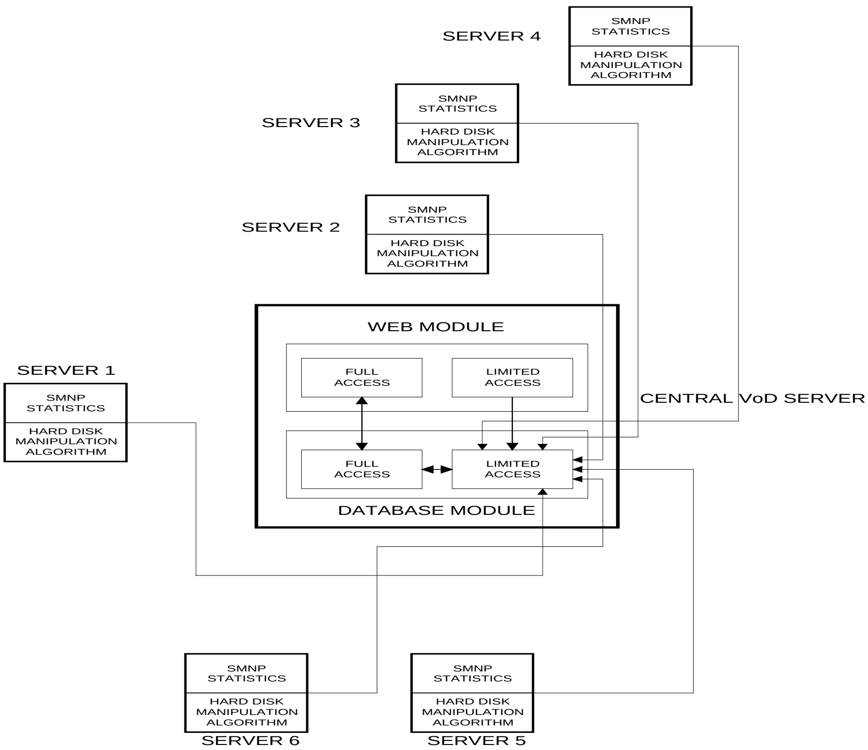

Figure 1. The system's modules

The implementation of the service also requires the maintenance and management of a respective amount of data, thus the introduction of an appropriate database is necessary. Both of the interface modules interact with the database that holds all of the information managed by the service. The database itself is conceptually divided into two similar modules: the full-access one and the limited access one. It is structured in such a way that each server can have its own, unique entry in it, however different attributes of this entry are accessible from each one of the two interface modules.

The Web Module

In the full access web module, the user is able to view all of the available video titles and search for a certain video title. The user cannot choose the server used to deliver to him each video title, as this will be determined by the proposed routing algorithm.

In order for the proposed service's principles to be imposed, each time the user places a request, an application implementing the DMA and VRA algorithms runs. The application requires certain information as input in order to produce reliable results. This input information includes the requesting user's IP address and some of the attribute values of the entries stored in the limited access module of the database. Taking into consideration this information, the application constantly determines the optimum server from where the user's request should be satisfied. This information is returned to the interface for the video title to start being displayed to the user.

The database module

The database module consists of entries concerning each one of the servers and network links participating in the VoD service. The full access sub-module of each entry can be accessed directly by making a request to the database. This module contains all the titles of the videos available on the particular server, and some information on them.

The limited access sub-module of each server and link's entries hold network and configuration information. Network information can be inserted by the administrators and local scripts manipulating SMNP statistics at each server and link. Configuration information can be entered during the initialization phase of the service-described in more detail further down and can be changed by the administrator each time a configuration feature changes. The limited access sub-module can be accessed only by the VoD network administrators. It is constantly updated with all the latest information on the network such as bandwidth utilization of connection lines or configuration changes. The limited access sub-module and the full access sub-module are accessed by the application running the service's VRA to provide the results.

The SMNP statistics module

This module provides the limited access database sub-module with updated information on the network traffic at any moment. The information is inserted into the database in the form of line utilization in percent. Every time a predefined time limit expires (1-2 minutes, which seems a reasonable interval compromising between the mutation rate of network characteristics and the imposed overhead) the SMNP statistics module on every server is responsible for inserting the line utilization of all the adjacent to the node links used by the VoD network. The information provided by this module plays a very crucial role in the decision-making procedure of the VRA run by the proposed application.

Service initialization

Initially all participants of the service are asked to contribute the information that the service requires, by using the limited access web module mentioned above. This information is provided by the use of forms in web pages and is immediately inserted into the limited access database module.

The information that the participants in the VoD service are required to contribute is:

1. Network links' bandwidth (The bandwidth connection of the server to the network)
2. The video titles available on each VoD the server.

Every time a feature included in the requested information changes, the administrators of the VoD servers must update the corresponding database entry with the latest configuration. Of course all of the above information could be inserted automatically into the system, either for the system initialization or in cases of configuration changes.

The VoD Service algorithms

The Disk storage and Manipulation Algorithm (DMA)

The DMA is the technique that we propose for the distribution of the video titles on the several video servers. It is a technique that permits redundancy, meaning that a video can be stored on more than one video servers, and this is done according to the "most popular" concept. Every video is stored locally on several disks, using a data-striping method that is described later in this section.

The “most popular” concept

The basic concept of all network-caching techniques is that anything downloaded from a local location instead of a distant location is downloaded much faster. Using the same concept we propose the implementation of a local cache on every VoD network server in order to keep many video titles locally. This way the users can have faster access to them.

In order to facilitate this concept, the video titles that are kept in the local caches can not be any video title downloaded by any user using the server, as is the concept of a proxy server. The video titles that are held in the local caches are the most popular ones, in terms of number of requests by the users using a certain server. The video titles that are requested the most from a video server, are the ones that are held locally in the server’s cache.

The algorithm

In order to implement the caching of the most popular video titles on every server, which improves the overall system performance, we introduce a disk manipulation algorithm. The algorithm allocates a predefined disk space for use by the VoD service. It counts the requests that are made for every video title and when a video title is requested for over a certain number of times, it is held in the local cache. When the cache is full, and another video title is requested for more than the predefined number of times, it replaces the least popular video title from those already in cache. By using this technique we meet the requests of the users that are utilizing a certain server and may have different orientations than other users. This concept is the idea that implements the distributed feature of the VoD service. The pseudocode of the disk manipulation algorithm is shown in Figure 2.

We have determined when a video is going to be stored on a local server disk but we have not determined how it will be stored. We propose that the videos should be stored on several disks with the use of data striping (see [4]) and the storage should be capacity oriented.

More specifically, we propose the determination of a, fixed and common for all disks, cluster size of *c* Mbytes/cluster, in such a way that each video will be divided into *p= (Video size in Mbytes)/c* parts. These parts will then be distributed for storage with a cyclic manner to the available disks. Thus, assuming a number of n available disks, if n>p then one video part is stored in each one of the first p hard disks. Otherwise, if n<p the first n video parts are stored in the n available disks and the rest p-n parts are distributed to the same disks starting from disk 1 and reusing as many of them as needed.

```
DO WHILE Video Service is Online
    IF (Server has begun downloading a video) THEN
        Get name and size of Video
        IF (Video is already on disk) THEN
            Give a point to the Video
        END IF
        IF (Video is not already on disk) THEN
            IF (Disks can tolerate the Video) THEN
                Write Video to Disks
            ELSE
                Give a point to video
                IF (Video's points>Least popular on disk Video's points) THEN
                    Delete Least Popular Video
                    IF (Disks can tolerate the Video) THEN
                        Write Video to Disks
                    END IF
                END IF
            END IF
        END IF
    END IF
LOOP
```

Figure 2. The Disk manipulation algorithm

This data striping technique is very useful for the implementation of our proposal because it helps, as we will see below, to avoid any possible network congestion. We must also point out that we propose the use of as many disks as possible. The Disk architecture is shown in Figure 3.

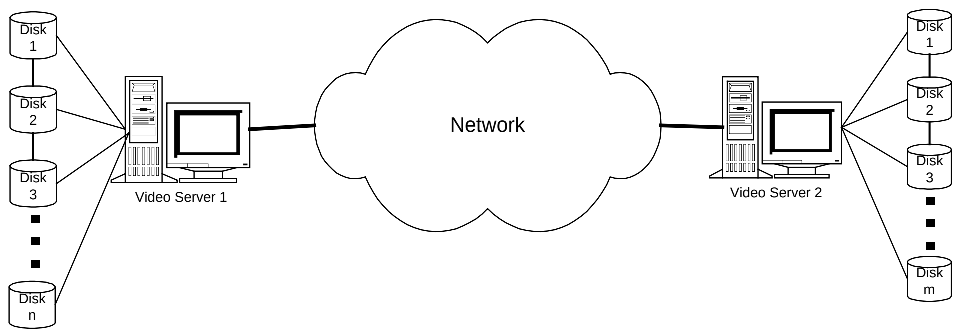

Figure 3. The disk storage architecture

The Virtual Routing Algorithm (VRA)

The basic feature of the proposed VoD service is the algorithm that eventually chooses the server from which a video title will be downloaded. The algorithm takes into consideration three parameters. These are shown in Table 1 below.

As mentioned before, most of this information is entered periodically into the limited access database module by the VoD server administrators or automatically. The routing algorithm must take all the input given to it and provide an output that is delivering to the user the requested video title, faster, at every moment. In order to provide the anticipated output the algorithm must evaluate all of this information and according to how crucial it is, make a decision. For the algorithm to do so we must provide it with a way to make optimum choices.

When a user chooses a video title that he wants to see, the routing algorithm starts running.

The Virtual Routing Algorithm parameters	
Parameter	Source
The SMNP statistics (network links' used bandwidth in Mbps and usability in percent)	The SMNP module
The total available network links' bandwidth	The administrators (the information exists in the limited access database module)
The available video titles on every server	The administrators (the information exists in the limited access database module)

Table 1. The parameters taken into consideration by the Virtual Routing Algorithm

Every Internet based network is a best effort network. This means that Quality of Service (QoS) issues are really out of the question. What we want to achieve by enforcing our routing algorithm is to provide a minimum QoS, which should be equal to the minimum video frame rate for which a video can be considered decent. Because our network is predefined we can actually enforce routing rather that wait for a best effort algorithm to deal with it. We propose the use of the Dijkstra's routing algorithm by our VRA in the process of determining the best route for the video to be downloaded.

After a video is requested by a client connected to a Video Server of the network, the VRA first determines the video servers that have the video stored. If the client's adjacent server has the video stored, the VRA chooses this server for the transmission of the video and terminates. However, the set of video servers that have the video stored may not include the client's adjacent server and are bound to be more than one because of the most popular concept already explained above. The VRA at this point has more that one candidate sources from where it can request the video. In this case, Dijkstra's routing algorithm ([5], [6]) is utilized by the VRA. The Dijkstra algorithm runs at the server with which the client is directly connected. It determines, for each server that has the video stored, the best route until the client's adjacent server. At this point the VRA takes control again and chooses among the already determined best routes to the candidate servers for downloading the video, the one with the smallest total cost. In order for the VRA and hence the Dijkstra algorithm to work as described, each network connection that is involved in every alternative route, from candidate servers to offer the requested video to the client's adjacent server, must be validated, thus assigned with a numeric value. In our case and for the Dijkstra algorithm, the network connections are assigned with a numeric weight of negative value, in other words the larger this validation number (referred to as LVN below) is the worst is the network link's performance.

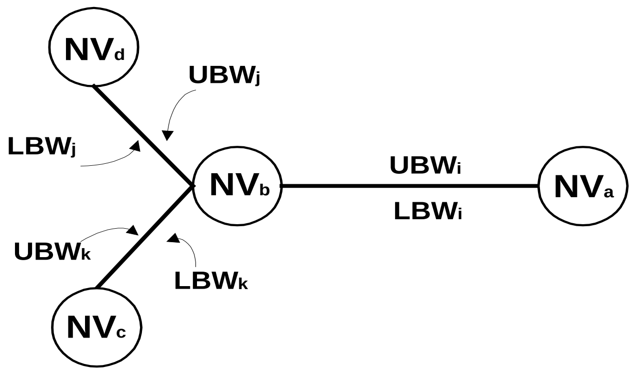

Figure 4. A network link that participates in an alternative route

In Figure 4 we can see a network link (between NV_b and NV_a) that participates in one of the alternative routes that we want to validate.

The validation procedure will be made according to the following equation:

$$\mathbf{LVN_i = max\ \{NV_a,NV_b\} + LU_i} \qquad \mathbf{(1)}$$

where:

LVN: Link Validation Number (the numeric weight assigned to each link)

NV_a: Validation of Node a, adjacent to the link of interest (a numeric value assigned to each network node that represents the node's current workload)

NV_b: Validation of Node b, adjacent to the link of interest

LU: Link Utilization (the parameter that accounts for the network link's workload

In (1), the first term of the sum represents the performance burden imposed by the adjacent to the link nodes, while the second term represents the link's traffic aggravation.

NV_a and NV_b are calculated according to the following equation:

$$\mathbf{NV_a = (\sum UBW_m) \ / \ (\sum LBW_m), \ \forall m \in \{set\ of\ links\ adjacent\ to\ node\ a\}} \quad \mathbf{(2)}$$

where:

UBWm: Used bandwidth of Link m in Mbps

LBWm: Total bandwidth of Link m in Mbps

In the example of Figure 4 the NV_b is calculated as follows:

$$NV_b = (UBW_i + UBW_j + UBW_k) / (LBW_i + LBW_j + LBW_k).$$

LU_i is calculated according to the following equation:

$$\mathbf{LU_i = LT_i * LV_i} \quad \mathbf{(3)}$$

where:

LT_i: Traffic of link i (percentage of used bandwidth over total available bandwidth)

LV_i: Value of link i (see below)

The last variable represents the points granted to each network link according to its total available bandwidth. For reasons of normalization, we ended up with the following equation that calculates the value of LV_i with respect to the link bandwidth size in Mpbs:

$$\mathbf{LV_i = \frac{Link\ Bandwidth\ (in\ Mbps)}{Normalization\ Constant}} \quad \mathbf{(4)}$$

The Normalization Constant suggested is an integer with a value approaching 10.

Now that we have determined all the details, it is time to sum up the VRA steps:

Get the IP address of the client placing the video request

Determine the server to whom the requesting user is directly connected (referred to as home server) by this IP

IF (the adjacent to the client video server can provide the requested video) THEN

    Authorize the server to start transferring the video

    QUIT

ELSE

    Make a list of all the servers on the network that have the requested video title

    Poll all of those servers to find out which ones can provide the video

    Calculate the Link Validation Number for each network link

Run the Dijkstra's routing algorithm to calculate the least expensive paths from the client's adjacent server to all other network nodes

Select those least expensive paths that start from the client's adjacent server and end at the servers that can provide the video

From those alternative least cost paths choose the one with the smallest cost

Notify the video server at the end of the selected path to start transferring the video

QUIT

END IF

Figure 5. Virtual Routing Algorithm

After determining the optimal server, the video will start being transferred. The routing algorithm also continues to run at the connecting server. The problem that might appear is that even though the optimal server for downloading can be determined at a certain point of time, this server might not be the optimal server after some time. If we continue to download the video from the same server, we compromise the system's attempts to impose some kind of QoS to the system because we are not downloading from the optimal server. We make an attempt to moderate this effect in the following section.

We have already described the data striping method that is used to store the videos in the available set of hard disks in each video server. When the system determines the optimal server for the first time, the first data strip of the video starts downloading. As mentioned before, the routing algorithm does not stop after the first determination of the optimal server. It continues to validate the network routes constantly. If the optimal server remains the same for as long as the first cluster of the video is downloaded and played, then the second cluster is requested from the same server. If the optimal server changes due to the change of certain network features during the downloading of a certain cluster, then the next cluster will be requested by the new optimal server. This scheme assures the downloading of the video cluster from the optimal server at every moment. Of course this requires certain synchronization techniques that will assure the constant playback of the video between cluster requests. It is obvious that the size of the cluster c as determined in the description of the DMA, plays a decisive part in dealing with network congestion according to this latest technique.

Case Study

The proposed path optimizing technique has been simulated for the Greek Universities Network Backbone. The topology of the network is shown in Figure 6.

Figure 6. The Greek Research and Technology Network Backbone

SNMP gave us the following network statistics (see Table 2) for a specific day. The percentages represent total link utilization, namely:

$$\frac{\text{Traffic\_in + Traffic\_out (in Mbps)}}{\text{Total bandwidth (in Mbps)}} \quad (5)$$

Link	Time			
	8am	**10am**	**4pm**	**6pm**
Patra-Athens (2Mbits link)	200 kb	1820 kb	1820 kb	1820 kb
	10%	91%	91%	91%
Patra - Ioannina (2Mbits link)	100 bits	170 bits	200 kb	240 kb
	0.005%	0.0085%	10%	12%
Thessaloniki - Athens (18Mb link)	1700 kb	7 Mb	9.8 Mb	9.6 Mb
	9.4%	38.8%	54.4%	53.3%
Thessaloniki – Xanthi (2Mb link)	480 kb	520 kb	750 kb	600kb
	24%	26%	37.5%	30%
Thessaloniki – Ioannina (2Mb link)	300 kb	1480 kb	1860 kb	1300 kb
	15%	74%	93%	65%
Athens – Heraklio (18Mb link)	500 kb	2.5 Mb	5.5 Mb	6 Mb
	2.7%	13.8 %	30.5%	33.3%
Xanthi - Heraklio (2Mb link)	100 bits	150 bits	200 bits	150 bits
	0.005%	0.005%	0.01%	0.0075%

Table 2. The Network status

Our simulation, calculated the LVN values for each link in the network, at the same times of the day when the statistics were retrieved. Table 3 contains the results:

Link	Time			
	8am	10am	4pm	6pm
Patra-Athens (2Mbits link)	0.083	0.632	0.687	0.697
Patra - Ioannina (2Mbits link)	0.07501	0.450017	0.535	0.539
Thessaloniki - Athens (18Mb link)	0.2819	1.1075	1.5433	1.4824
Thessaloniki – Xanthi (2Mb link)	0.168	0.4611	0.6391	0.583
Thessaloniki – Ioannina (2Mb link)	0.1427	0.5571	0.7501	0.653
Athens – Heraklio (18Mb link)	0.1116	0.5462	0.999	1.0574
Xanthi - Heraklio (2Mb link)	0.1201	0.13001	0.275015	0.3

Table 3. The Link Validation Numbers

According to the data above four experiments were made:

Experiment A:

Suppose that at 8:00 am a client connected to the Patra’s Video Server (node U2) requires a video title that can only be provided by the Video Servers of Thessaloniki (node U4) and Xanthi (U5). The VRA, at this point, uses the Dijkstra’s routing algorithm and the values of Table 4 to determine the least-cost paths from U4 to U2 and U5 to U2. In fact, all shortest paths from all the network nodes to U2 are determined, however the VRA is only interested in the most efficient paths from U4 to U2 and from U5 to U2. Below is given the table of path values occurring as the Dijkstra’s algorithm is running, according to [7]:

Step	Nodes	D3	Path	D1	Path	D4	Path	D5	Path	D6	Path
1	{U2}	0.075	U2,U3	0.083	U2,U1	$\Re$	-	$\Re$	-	$\Re$	-
2	{U2,U3}	0.075	U2,U3	0.083	U2,U1	0.365	U2,U1,U4	$\Re$	-	0.195	U2,U1,U6
3	{U2,U3,U1}	0.075	U2,U3	0.083	U2,U1	0.365	U2,U1,U4	0.315	U2,U1,U6,U5	0.195	U2,U1,U6
4	{U2,U3,U1,U6}	0.075	U2,U3	0.083	U2,U1	0.365	U2,U1,U4	0.315	U2,U1,U6,U5	0.195	U2,U1,U6
5	{U2,U3,U1,U6,U5}	0.075	U2,U3	0.083	U2,U1	0.365	U2,U1,U4	0.315	U2,U1,U6,U5	0.195	U2,U1,U6
6	{U2,U3,U1,U6,U5,U4}	0.075	U2,U3	0.083	U2,U1	0.365	U2,U1,U4	0.315	U2,U1,U6,U5	0.195	U2,U1,U6

Table 4. The Dijkstra’s algorithm table for experiment A

From the above, it is derived that:

Best path from U4 to U2 (or U2 to U4): U2,U1,U4 with total cost 0.365

Best path from U5 to U2 (or U2 to U5): U2,U1,U6,U5 with total cost 0.315

At this point the VRA decides that the requested video should be downloaded from node U5 (Video Server of Xanthi) following the route U5,U6,U1,U2 which has the minimum cost.

Experiment B:

Suppose that at 10:00 am the same as above situation occurs: a client connected to the Patra's Video Server (node U2) requires a video title that can only be provided by the Video Servers of Thessaloniki (node U4) and Xanthi (U5). Below is given the table of path values occurring as the Dijkstra's algorithm is running:

Step	Nodes	D3	Path	D1	Path	D4	Path	D5	Path	D6	Path
1	{U2}	0.45	U2,U3	0.632	U2,U1	ℜ	-	ℜ	-	ℜ	-
2	{U2,U3}	0.45	U2,U3	0.632	U2,U1	1,007	U2,U3,U4	ℜ	-	1,178	U2,U1,U6
3	{U2,U3,U1}	0.45	U2,U3	0.632	U2,U1	1,007	U2,U3,U4	1,308	U2,U1,U6,U5	1,178	U2,U1,U6
4	{U2,U3,U1,U4}	0.45	U2,U3	0.632	U2,U1	1,007	U2,U3,U4	1,308	U2,U1,U6,U5	1,178	U2,U1,U6
5	{U2,U3,U1,U4,U6}	0.45	U2,U3	0.632	U2,U1	1,007	U2,U3,U4	1,308	U2,U1,U6,U5	1,178	U2,U1,U6
6	{U2,U3,U1,U4,U6,U5}	0.45	U2,U3	0.632	U2,U1	1,007	U2,U3,U4	1,308	U2,U1,U6,U5	1,178	U2,U1,U6

Table 5. The Dijkstra's algorithm table for experiment B

From the above, it is derived that:

Best path from U4 to U2 (or U2 to U4): U2,U3,U4 with total cost 1,007

Best path from U5 to U2 (or U2 to U5): U2,U1,U6,U5 with total cost 1,308

At this point the VRA decides that the requested video should be downloaded from node U4 (Video Server of Thessaloniki) following the route U2,U3,U4 which has the minimum cost.

Experiment C:

Suppose that at 4:00 pm a client connected to the Athens' Video Server (node U1) requires a video title that can only be provided by the Video Servers of Thessaloniki (node U4), Xanthi (node U5) and Ioannina (node U3). The VRA runs, similarly to the above and it is calculated that:

Best path from U4 to U1 (or U1 to U4): U1,U4 with total cost 1.5433

Best path from U5 to U1 (or U1 to U5): U1,U6,U5 with total cost 1.274

Best path from U3 to U1 (or U1 to U3): U1,U2,U3 with total cost 1.222

At this point the VRA decides that the requested video should be downloaded from node U3 (Video Server of Ioannina) following the route U3,U2,U1 which has the minimum cost.

Experiment D:

Suppose that at 6:00 pm the same as above situation occurs: a client connected to the Athens' Video Server (node U1) requires a video title that can only be provided by the Video Servers of Thessaloniki (node U4), Xanthi (node U5) and Ioannina (node U3). The VRA runs and it is calculated that:

Best path from U4 to U1 (or U1 to U4): U1,U4 with total cost 1.4824

Best path from U5 to U1 (or U1 to U5): U1,U6,U5 with total cost 1.3574

Best path from U3 to U1 (or U1 to U3): U1,U2,U3 with total cost 1.236

At this point the VRA decides that the requested video should be downloaded from node U3 (Video Server of Ioannina) following the route U3,U2,U1 which has the minimum cost.

Advantages of our proposal

The basic advantage of our proposal is the implementation of a demanding service on a limited bandwidth network. Until now such applications were only implemented over cable networks that had the ability to provide much greater bandwidth than the bandwidth provided by Internet based networks.

Another advantage of our service is its inherent ability to adjust to the network changes without the need for reprogramming. The service grows with the network and has the ability to adjust to a large variety of diverse networks. High bandwidth or low bandwidth networks can use this service with many advantages. The use of wide spread Internet technologies on the other hand makes the service easy to implement and very popular. The other feature that can make this service popular is the very low costs at which it can be implemented. Due to the large variety of computers and connections that can be used, almost anyone can connect to the network with minimum costs.

The expandability of the network with very little effort of reprogramming is another feature that must be looked at as an advantage. New nodes can easily be connected to the network and the only thing that has to be changed is corresponding database entries. The user interface can also be designed to upgrade itself automatically in the procedure of inserting a new node to the service network.

Conclusions - Future work

As mentioned above, the needs of the growing Internet community for multimedia and especially video applications are constantly increasing. New solutions must be introduced to meet these needs. Our proposal is a step forward in this procedure. With the growth of the Internet and the higher bandwidth that will be available such applications will be easier and more cost effective to implement. The future is open for multimedia services such as VoD and in not so many years they will be considered basic services on the Internet.

Our future work aims at improving certain features of our proposal. First of all we are aiming at improving the QoS standards that we have imposed onto the network. Even if QoS is not really an easy feature to impose onto an Internet like network we can impose it onto a predefined network with the techniques that we have described. Another feature that we are aiming to improve is the data striping methods used to store the videos. We believe that even though the method that we have described is satisfactory we could have even better results if the various videos were stripped not on the hard disks of one server but of different servers according to the popularity. This means that the most popular technique that we have described will not be imposed on whole videos but on video strips. Another feature that we are aiming to improve is the validation technique that we are going to use. This means that we must make clear what the role of every Server configuration factor (CPU speed, available RAM etc.) is to our Video service.

References

[1] "Columbia's VoD and Multimedia Research Testbed with Heterogeneous Network Support", S.-F. Chang, A. Eleftheriadis, D. Anastassiou, S. Jacobs, H. Kalva, and J. Zamora, Journal on Multimedia Tools and Applications, Special Issue on Video on Demand, Kluwer Academic Publishers, 1997

[2] "Building Multimedia Systems for the Broadband Internet", Sami Tikka, Master's Thesis, Computer Science and Engineering Department, Helsinki University Of Technology, May 23, 1997

[3] "Improving Reliability of Distributed VoD Servers", Manuel Billot, Valérie Issarny, Isabelle Puaut and Michel Banâtre. In Proceedings of the 4th IEEE

International Conference on Multimedia Computing and Systems June 1997, Ottawa, Canada.

[4] "Parallel video servers: a tutorial", Lee, J.Y.B., Chinese Univ. of Hong Kong, Shatin, Hong Kong, appears in: IEEE Multimedia, April-June 1998 Vol. 5 Issue: 2, pages: 20 – 28

[5] Data Structures and Network Algorithms, R.E. Tarjan, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1983

[6] Introduction to algorithms, T. H. Cormen, C.E. Leiserson and R.L. Rivest, MIT Press, Cambridge, MA, 1990

[7] "Routing Algorithms,", R. Jain, Part of a 3-day course taught at Nortel, Ottawa in December 1998, http://www.cis.ohio-state.edu/~jain/courses/bnr/e_5pkt.htm

[8] The Berkeley Distributed Video-on-Demand (VOD) system: http://bmrc.berkeley.edu/frame/research/storage/

[9] Dynamic Service Aggregation for Interactive Information Delivery over Networks: http://hulk.bu.edu/projects/summary.html

[10] "Randomized adaptive video on demand", C. Bouras, V. Kapoulas, T. Pantziou, P. Spirakis, 15th ACM-PODC, 1996, Philadelphia PA, USA (short paper)

[11] "Internet Routing Architectures", Bassam Halabi, Cisco Press, ISBN 1-56205-652-2

[12] "IP Routing Primer", Robert Wright, Cisco Press, ISBN 1-57870-108-2

[13] "An Analysis of Dynamic Routing Protocols in Highly Distributed Data Networks", Andrew Steven Kessler, B.S., State University of New York, Interdisciplinary Telecommunications Department, 1992

[14] "Adaptive video on demand", Sudhanshu Aggarwal, Juan Garay, and Amir Herberg, in Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing, page 402, Los Angeles, CA, August 1994.

[15] "Prospects for interactive video-on-demand", T.D.C. Little and D. Venkatesh, IEEE Multimedia I(3), pp. 14-24, 1994.