Resource Assignment in Box-Based P2P Video-on-Demand Systems

Juan Pedro Muñoz-Gea, Josemaria Malgosa-Sanahuja, Pilar Manzanares-Lopez, Pedro Jose Piñero-Escuer
Department of Information Technologies and Communications
Polytechnic University of Cartagena
Campus Muralla del Mar, 30202 Cartagena, Spain
Email: {juanp.gea, josem.malgosa, pilar.manzanares, pedrop.escuer}@upct.es

***Abstract*—Video-on-demand (VoD) systems that make use of the storage capacities at set-top boxes to assist the streaming servers have been introduced recently. In these systems, videos are separated into substreams and randomly stored in boxes, which are organized in a P2P network. By this way, this architecture combines the load balancing and fault tolerating features of P2P systems with the stability of set-top boxes, since they usually have much longer online time than traditional PC based peers. The VoD service provider controls two different resources: the allocation of substreams to the selected boxes, and the parameters associated to the streaming servers (number, bandwidth, storage, placement in the network). In this work, we present the Resource Assignment Problem (RAP) which tries to optimize the previous resources in order to reduce the associated costs. This problem is presented as a linear programming problem and it is solved using the MATLAB optimization toolbox. We have evaluated the influence of the bandwidth and the placement of the server in the Internet.**

***Keywords*-VoD; P2P; set-top box; linear programming.**

I. INTRODUCTION

Nowadays, there exist two main categories of video-on-demand (VoD) streaming over IP. The first category is the Internet Protocol Television (IPTV) that assumes video services delivery in a managed network, usually deployed and operated by broadband providers. On the other hand, there is the Internet video streaming which relies on 3rd party servers (usually Content Delivery Networks, CDNs) to stream multimedia content to end-users attached to Internet. This category of video streaming architectures is gaining unprecedented attraction, fuelled by the fact that the service delivery is well balanced and non-biased towards any main stakeholder (such ISP in the IPTV world) [1]. In addition, due to the open nature of Internet, the barrier to enter the Internet streaming market is much lower than in the IPTV case.

A key characteristic of CDNs is that they comprise of a large number of caches distributed throughout the network to facilitate more speedy access to content. Load balancing techniques are achieved by dynamically redirecting VoD request to appropriate caches based on the load and proximity of each cache to the end-user. However, this approach has multiple drawbacks, among which complexity of deploying data centers, power consumption, and lack of scalability are the most critical ones [2].

Peer-to-Peer (P2P) has recently emerged as the main approach to increase the scalability of streaming servers. However, one serious problem with using a conventional P2P solution, at least in the current technological landscape, is the limited upload bandwidth of each peer [3]. In addition, the available bandwidth has to be shared among different applications. The result is that a video request would require finding a great number of peers that have already downloaded that video, which can be highly unlikely, in order to get the uplink bandwidth related to the video bit rate.

Several works [3], [4], [5], [6] have suggested to making use of the storage capacities at set-top boxes (STBs) or residential gateways of clients, combined in a P2P approach, to assist the CDN of the video provider. In these systems, movies are broken into small substreams which are pre-cached through the P2P network during off-peak hours. Even though each peer can only afford to contribute limited upload bandwidth, the aggregate is sufficient to support high definition delivery. For example, ten peers with pre-cached content can serve substreams at a steady state rate of 200 Kbps to satisfy a 2 Mbps video request from a peer.

Therefore, in this scenario, the VoD service provider controls two different resources: First of all, the allocation of substreams to the selected boxes. A good allocation can represent a substantial saving when considering the network cost of the streaming sessions, i. e. the cost to transport the data from the boxes hosting the substreams to the client. And secondly, the parameters associated to the streaming servers, for example: number, bandwidth, storage capacity and placement in the network.

In this paper, we present the Resource Assignment Problem (RAP) which tries to optimize the allocation of substreams and the parameters associated to the streaming servers in order to reduce the associated costs: the number of traversed routers to transport the data, the use of the access networks of network operators and the use of the streaming servers.

The rest of the paper is organized as follows. Section II discusses several related works. Section III outlines the most remarkable issues of the system. Sections IV and V presents the Resource Assignment Problem and the results provided by the optimization tool. Finally, Section VI concludes the paper.

II. RELATED WORKS

The systems presented in [3], [4], [5] are designed to work in a managed network, deployed and operated by an IPTV provider. In all of them, the streaming server performs the main intelligence in the system because it allocates resources for each incoming VoD request. In order to perform this, it tracks two main resources: (i) the currently available uplink bandwidth at each peer, and (ii) the content stored in each peer. When a given peer requests a specific content, a VoD request is sent from the peer to the server. The server looks up its database to determine the most appropriate set of contributing STBs. If during the streaming session the available bandwidth of one of the peers changes, and the receiving peer is unable to properly recover the video, it sends a request to the server and it answers the identity of a new candidate.

In the previous proposals it is possible that the servers have a global knowledge about the system because they work in a managed network. However, controlling the state of all the peers on the Internet can be a very costly task. In our proposal, it is assumed that during the substream pre-caching process the peers also receive the identity of the STBs they have to contact with to download every video. By this way, the clients do not have to contact the server to start each streaming session. On the other hand, as some peers may fail, redundancy is necessary to guarantee smooth delivery. Before breaking the movie into substreams, we use an erasure code with a threshold (say 80 %). That is, the movie file is broken into segments, and each segment is encoded as, say, 12 blocks, and any 10 of which are sufficient to reconstruct the segment.

On the other hand, in [6] authors consider a global Internet scenario, and they study the allocation of substreams to the selected boxes in order to reduce the network cost. However, they only consider the network cost as the number of traversed routers. In our work we also want to take into account the cost associated to the use of the access networks and streaming servers. In addition, authors assume that when a new box joins the system, it iteratively explores the nearest boxes in the network until it discovers all the necessary substreams minus one of them, and then it receives the unassigned substream from the server. That is, the new box is the responsible to locate the complementary substreams that it needs using an iterative process. In our system, boxes do not have this extra cost, because they receive the identity of their complementary boxes from the streaming servers.

Finally, in [6] authors do not take into account that streaming servers can also provide substreams as STBs. In our system, we take into account that streaming servers can be considered as usual STBs during the substream pre-caching process. By this way, we can take advantage of the characteristics of streaming servers to reduce the cost associated to the streaming sessions.

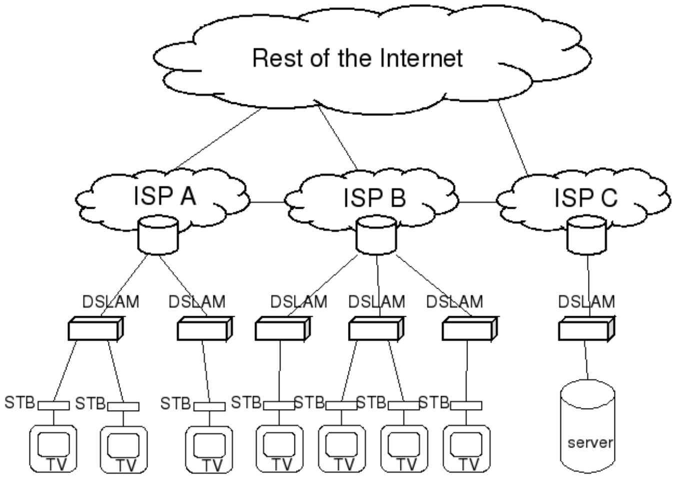

Figure 1. Internet scenario of the system.

III. SCENARIO AND ARCHITECTURE

A. Scenario

Research in P2P streaming typically considers Internet at a logical level: it represents the Internet as an abstract cloud and only considers the capacity of the content server and the characteristics of the access links to related hosts. This view of the Internet is referred as the ”cloud model”. In contrast to the cloud model, the physical model considers the network architecture and bandwidth constraints of the underlying links and network devices. A key insight of [3] is that using the ”cloud model” for P2P streaming is overly simplistic. More realistic results can be obtained by considering the network at the physical infrastructure level. Authors show that the cloud model of the Internet frequently used in simulation studies of peer-to-peer systems may drastically overstate the benefits of P2P video content delivery. Thus, one must consider physical network infrastructure to obtain more reliable results.

Figure 1 represents the scenario where our system is deployed. The clients are located in the networks managed by several ISPs. In addition, in order to simplify the proposed model, the streaming server of the video provider is located in a different network, where there are not any clients. We assume that the video provider knows the network of the box, i.e. the identifier of the first router that connects this box to the Internet, and the network cost between every two routers (see [7], [8] for some techniques that may be used).

B. Overall Architecture Description

We are going to consider separately one video of the top 5 % popular videos, in the same way that [3]. First of all the video provider breaks the video into segments, and then it applies erasure coding to every segment with a specific rate k/n. It means that each segment is encoded as n blocks $(b_1, b_2, ., b_n)$ and any k of them will be enough to reconstruct the segment. After that, it creates n substreams joining the b_i blocks of each segment and it allocates one substream to each box following a serial dispatching strategy. When the n substreams have been allocated, the process is repeated iteratively till all the boxes have a different substream. In addition, every peer also receives a neighbor table, which represents the identity of the n boxes it has to contact with to get the video (taking into account that it is only necessary to establish k connections). This neighbor table is created by the service provider taking into account the results of the optimization algorithm, which will be presented in next section.

The download of consecutive segments is managed by a sliding window algorithm. Counting from the moment of play-back, a segment may be downloaded from a STB if it is farther than a time parameter. That is, substreams are usually downloaded from other STBs in the network. However, nearer segments which have not been already downloaded from other peers can be downloaded from the streaming servers. Because segment reconstruction can occur only when k blocks are downloaded in their entirely, buffering is essential in our approach: a downloading peer delays movie rendering until the first segment is downloaded and reconstructed, and then downloads the blocks of the future segments while the current segment is being rendered. Because of buffering, when a downloading peer detects failure of a peer used for movie delivery, there is usually significant time left until the interrupted block will be needed for viewing.

IV. RESOURCE ASSIGNMENT PROBLEM

In this scenario, the VoD service provider controls two different resources: First of all, the allocation of substreams to the selected boxes. And secondly, the parameters associated to the streaming servers, for example: number, bandwidth, storage capacity and placement in the network. In this section, we present the Resource Assignment Problem (RAP) which tries to optimize the allocation of substreams and the parameters associated to the streaming servers in order to reduce the associated costs.

The RAP problem is presented as a linear programming problem. The decision variable $\overline{x}$ represents the number of substreams that the clients in every network have to get from each network. For example, in a scenario with 3 networks

$$\overline{x} = \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{bmatrix}$$

where x_{ij} represents the number of substreams that the clients in network i have to get from clients located in network j. The streaming server of the video provider is located in network 3, where there are not any clients. That is the reason why elements x_{3j} do not appear in $\overline{x}$.

The optimization problem can be characterized asfollows:

$$\min_{\overline{x}} \quad C_T(N, \overline{x}, \overline{v}) + C_S(N, \overline{x}) \tag{1}$$

$$s.t. \quad \sum_{j=1}^{N} x_{ij} = (k-1) \cdot p \cdot c_i \tag{2}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{N} x_{ij} \cdot r + x_{ii} \cdot 2 \cdot r + \sum_{\substack{j=1 \\ j\neq i}}^{N} x_{ji} \cdot r \leq BW_i \tag{3}$$

The objective function, Equation 1 seeks to minimize the cost of transport and the cost of streaming from the central server:

$$C_T(N, \overline{x}, \overline{v}) = \sum_{i=1}^{N-1} \sum_{j=1}^{N} \theta \cdot v_{ij} \cdot x_{ij} \tag{4}$$

$$C_S(N, \overline{x}) = \sum_{i=1}^{N-1} \sum_{j=1, j\in S}^{N} \beta \cdot x_{ij} \tag{5}$$

where θ is the cost of one hop in the transport network, v_{ij} represents the number of hops between the networks i and j and β is the cost of streaming one substream from the central server. They are called *cost of transport* and *cost of servers* respectively.

Constraint 2 is defined for every network i, and it represents the total number of substreams that are downloaded by the clients of a specific network in the peak hour. Constraint 3 is also defined for every network i, and it represents the total bandwidth consumed in network i. Every substream downloaded from a different network consumes a bandwidth similar to its streaming rate, whereas the substreams downloaded from the same network consume a double bandwidth. Finally, the substreams uploaded from a specific network with destination in other network also consume the corresponding upload bandwidth in the origin network. In the constrainsts k is the number of substreams, p is the percentage of active clients in the peak hour, c_i is the number of clients in the network i, N is the number of networks, r is the streaming rate of substreams and BW_i is the bandwidth constraint in the network i. They are called *number of substreams* constraint and *network bandwidth* constraint, respectively.

To solve the problem we use the MATLAB Optimization Toolbox. Specifically, we use the `linprog` program which solves linear programming problems.

V. PERFORMANCE EVALUATION

A. Introduction

In this section we present the results provided by the optimization tool in three small scenarios. First of all, we try to make readers understand that solving this problem is not as obvious as they could think. We are going to assume that there are three networks, with clients located in networks 1 and 2, and the server located in network 3. The number of hops between the 3 networks is represented in matrix H, where every element (H_{ij}) represents the number of hops between the networks i and j

$$H = \begin{pmatrix} 0 & 8 & 8 \\ 8 & 0 & 2 \\ 8 & 2 & 0 \end{pmatrix}$$

The erasure coding rate is assumed to be 10/12, therefore, the number of substreams that every VoD request needs is $k - 1 = 9$, because every STB has one of the necessary substreams. Every substream has a streaming rate of 200 Kbps and the bandwidth restriction of networks 1 and 2 is set to 18 Mbps. In addition, we assume that during the peak hour only a 75 % of the clients located in a network are using the VoD service. Finally, parameters β and θ are set to 1.

1) Scenario 1: The number of clients of both networks is set to 7. Therefore, the number of active clients during the peak hour is 5, and the total number of substreams that they need is $(10 - 1) \times 5 = 45$. The result provided by the optimization tool is the following

$$\overline{x} = \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{23} \end{bmatrix} = \begin{bmatrix} 45 \\ 0 \\ 0 \\ 0 \\ 45 \\ 0 \end{bmatrix}$$

This result tells us that all the substreams that the clients in both networks need must be obtained from their networks. On the other hand, we have to take into account that 45 substreams is the maximum number of substreams that the clients of a network can obtain from their own networks. These substreams consume a double quantity of bandwidth, therefore they correspond with a bandwidth of $45 \times 200 \times 2 = 18$ Mbps which corresponds with the bandwidth restriction.

2) Scenario 2: The number of clients in network 1 is set to 9, and the number of clients in network 2 is set to 5. Therefore, the number of active clients during the peak hour is 7 and 4, and the total number of substreams that they need is 63 and 36, respectively. The result provided by the optimization tool is the following

$$\overline{x} = \begin{bmatrix} 27 \\ 18 \\ 18 \\ 0 \\ 36 \\ 0 \end{bmatrix}$$

This result tells us that the 36 substreams that clients in network 2 need can be obtained from its own network. On the other hand, the 63 subtreams that clients in network 1 need must be distributed in the following way: 27 must be obtained from its own network, 18 from network 2, and 18 from the server. The algorithm tries to get the maximum number of substreams from clients located in network 2. Taking into account that this network is consuming $36 \times 200 \times 2 = 14400$ Kbps, there are 3600 Kbps free. It corresponds to $3600/200 = 18$ substreams. Finally, the algorithms gets other 18 substreams from the server. Explaining this last result is a bit more complicated, but it can be seen that with this configuration the bandwidth consumed in network 1 corresponds to the maximum. Let's see: the 27 substreams obtained from network 1 consume $27 \times 200 \times 2 = 10800$ Kbps. On the other hand, the 36 substreams obtained from network 2 and the server consume $36 \times 200 = 7200$ Kbps. The sum of both of them is 18000 Kbps which is the maximum bandwidth. Therefore, we deduce that the 18 substreams obtained from the server correspond to the minimum number of substreams to fulfill the bandwidth restriction.

3) Scenario 3: The number of clients in network 1 is set to 5, and the number of clients in network 2 is set to 9. Therefore, the number of active clients during the peak hour is 4 and 7, and the total number of substreams that they need is 36 and 63, respectively. The result provided by the optimization tool is the following

$$\overline{x} = \begin{bmatrix} 36 \\ 0 \\ 0 \\ 0 \\ 27 \\ 36 \end{bmatrix}$$

This result tells us that the 36 substreams that clients in network 1 need can be obtained from its own network. On the other hand, the 63 subtreams that clients in network 3 need must be distributed in the following way: 27 must be obtained from its own network and 36 from the server. Let's analyze this result. The first option would be to try to obtain the 63 substreams from network 2, however, it is impossible because these substreams consume a double bandwidth. Therefore, the algorithm tries to get the maximum number of substreams from the server, because the cost is less than obtaining them from the clients in network 1.

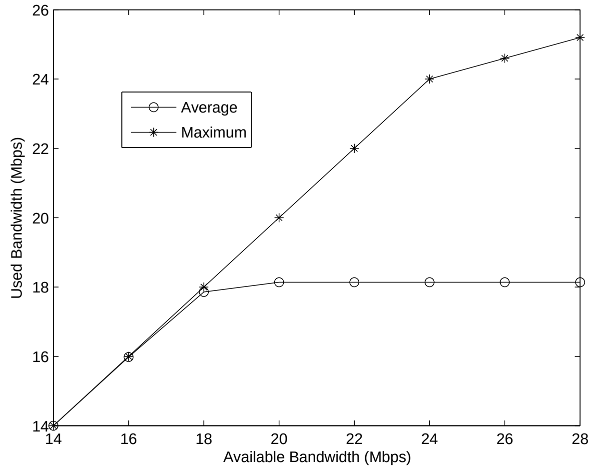

Figure 2. Bandwidth used in the networks.

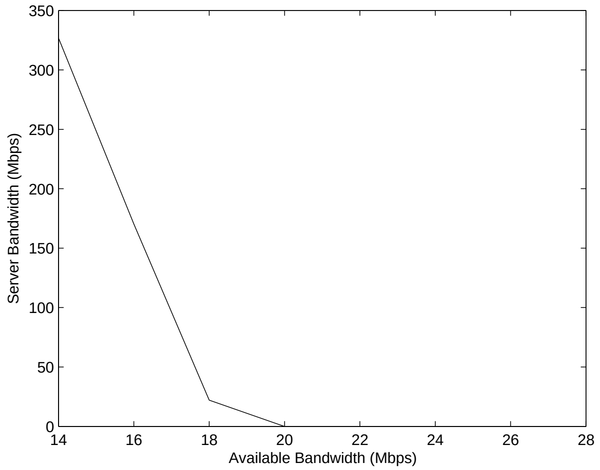

Figure 3. Bandwidth used by the server.

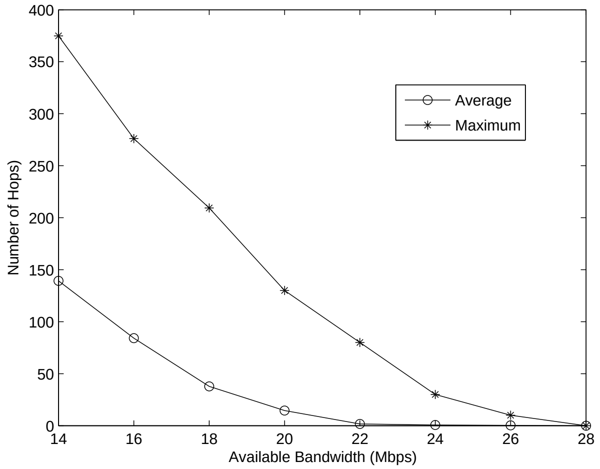

Figure 4. Number of necessary hops.

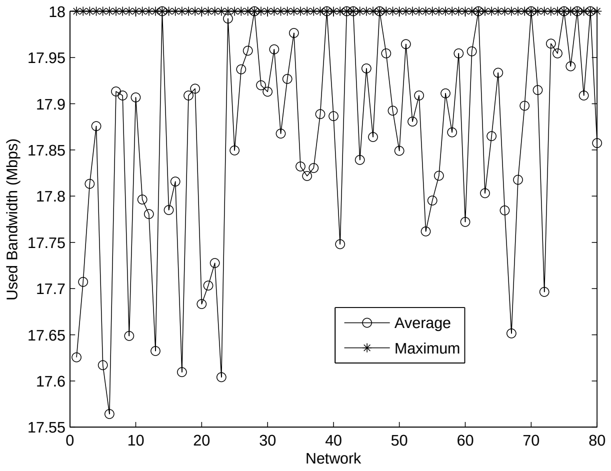

Figure 5. Bandwidth used in the networks.

B. Results

In this section, we show the influence of the *network bandwidth* constraint and the placement of the server in the Internet on the bandwidth used in the networks, the bandwidth used by the server and the number of necessary hops to satisfy the VoD request. A router-level topology resulted from the *Network Cartographer (nec)* [9] is used as our network topology. We randomly select 80 edge routers with degree equal to one, and we attach to each of them a number of elements following a normal distribution $N(7, 1.4)$. We assume that during the peak hour only a 75 % of the clients located in a network are using the VoD service. On the other hand, the erasure coding rate is assumed to be 10/12, and every substream has a streaming rate of 200 Kbps. Finally, parameters β and θ are set to 1.

1) Influence of the network bandwidth constraint:

Figures 2, 3 and 4 show the influence of the *network bandwidth* constraint. In these experiments, the bandwidth restriction of networks changes from 14 to 28 Mbps. As it can be seen in Fig. 2, the average bandwidth increases linearly till a bandwidth restriction of 18 Mbps, but from that point it is stabilized around 18 Mbps. On the other hand, the maximum bandwidth used increases linearly till a bandwidth restriction of 24 Mbps, and after that the increasing rate is reduced. The bandwidth used by the server (represented in Fig. 3) is reduced when the network bandwidth constraint increases, and it is almost 0 from a 20 Mbps local bandwidth restriction. Finally, Figure 4 shows the influence of this parameter on the number of necessary hops that every VoD requests needs to be satisfied.

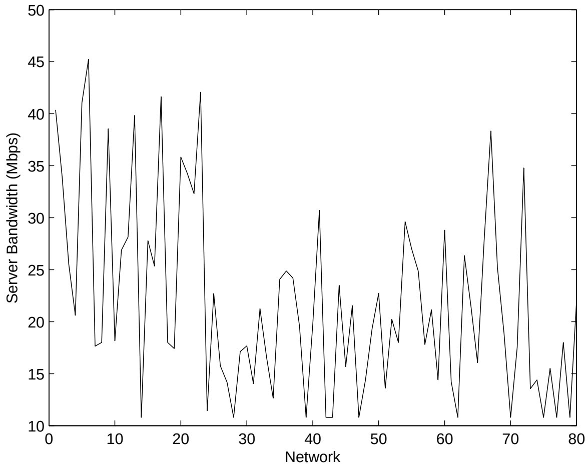

Figure 6. Bandwidth used by the server.

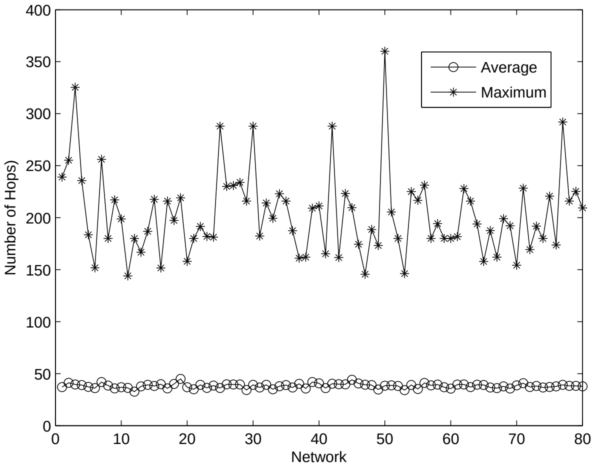

Figure 7. Number of necessary hops.

2) Influence of the placement of the server: Figures 5, 6 and 7 show the influence of the the placement of the server in the Internet. In these experiments, the bandwidth restriction of networks is fixed to 18 Mbps. The x-axis of the previous figures represents the identifier of the network in which the streaming server is located, and it changes from 1 to 80. These results can help us to find out what the best location of the server is in order to reduce the associated costs.

VI. CONCLUSION

Box-based P2P VoD systems make use of the storage capacities at STBs, combined in a P2P approach, to assist the CDN of the video provider. In these systems, movies are broken into small substreams which are pre-cached through the P2P network during off-peak hours. In this scenario, the VoD service provider controls two different resources: the allocation of substreams to the selected boxes, and the parameters associated to the streaming servers.

In this paper, we have presented the Resource Assignment Problem (RAP) which tries to optimize the allocation of substreams an the parameters associated to the streaming servers in order to reduce the associated costs: the number of traversed routers to transport the data, the use of the access networks of network operators and the use of the streaming servers.

ACKNOWLEDGMENT

This research has been supported by project grant TEC2007-67966-C03-01/TCM (CON-PARTE-1) and it is also developed in the framework of “Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM (Plan Regional de Ciencia y Tecnología 2007/2010)”.

REFERENCES

[1] F. Thouin and M. Coates, “Video-on-demand networks: Design approaches and future challenges,” *IEEE Network*, vol. 21, no. 2, pp. 42–48, 2007.

[2] N. Laoutaris, P. Rodriguez, and L. Massoulie, “Echos: edge capacity hosting overlays of nano data centers,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 1, pp. 51–54, 2008.

[3] Y. Chen, Y. Huang, R. Jana, H. Jiang, M. Rabinovich, J. Rahe, B. Wei, and Z. Xiao, “Towards capacity and profit optimization of video-on-demand services in a peer-assisted iptv platform,” *ACM Multimedia Systems*, vol. 15, no. 1, pp. 19–32, 2009.

[4] A. Nafaa, S. Murphy, and L. Murphy, “Analysis of a large-scale vod architecture for broadband operators: A p2p-based solution,” *IEEE Communications Magazine*, vol. 46, no. 12, pp. 47–55, 2008.

[5] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, “Push-to-peer video-on-demand system: Design and evaluation,” *IEEE Journal of Selected Areas in Communications*, vol. 25, no. 9, pp. 1706–1716, 2007.

[6] Y. Chen, B. Han, J. Leblet, G. Straub, and G. Simon, “Network-friendly box-powered video delivery system,” in *ITC 21 : 21st International Teletraffic Congress*, 2009, pp. 1–8.

[7] B. Wong, A. Slivkins, and E. G. Sirer, “Meridian: a lightweight network location service without virtual coordinates,” *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 85–96, 2005.

[8] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, “Avoiding traceroute anomalies with paris traceroute,” in *IMC ’06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 153–158.

[9] D. Magoni and M. Hoerdt, “Internet core topology mapping and analysis,” *Computer Communications*, vol. 28, no. 5, pp. 494 – 506, 2005.