Popularity Based Distribution Schemes for P2P Assisted Streaming of VoD Contents

Sasho Gramatikov, Fernando Jaureguizar, Julián Cabrera, and Narciso García
Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Madrid
Email: {sgr, fjn, julian.cabrera, narciso}@gti.ssr.upm.es

***Abstract*—The Video on Demand (VoD) service is becoming a dominant service in the telecommunication market due to the great convenience regarding the choice of content items and their independent viewing time. However, it comes with the downsides of high server storage and capacity demands because of the large variety of content items and the high amount of traffic generated for serving each request. Storing part of the popular contents on the peers brings certain advantages but, it still has issues regarding the overall traffic in the core of the network and the scalability. Therefore We propose a P2P assisted model for streaming VoD contents that takes advantage of the clients unused uplink and storage capacity to serve requests of other clients and present popularity based schemes for distribution of both the popular and unpopular contents on the peers with the objective to reduce the streaming traffic in the core of the network, improve the responsiveness of the system and increase its scalability.**

I. Introduction

The great expansion of the IPTV [**?**] has made a good ground for the Video on Demand (VoD) to become one of the most popular services. Although VoD is service that is also available on the Internet, it has attracted special attention in the Telecom-managed networks since they are already accustomed for implementation of a variety of TV services. Despite of its numerous advantages from clients point of view, the VoD service is an issue from providers point of view since it is very bandwidth demanding. Therefore the design of systems and algorithms that aim at optimal distribution of the content items has become a challenge for many providers. Some of the solutions include hierarchy of cache servers which contain replicas of the content items placed according to a variety of replica placement algorithms that depend on the users behaviour [**?**][**?**][**?**]. No matter how good these solutions might be, they all reach a point from where no further improvements can be done because of the resource limitations. One possibility to overcome this problem is the implementation of the classical P2P principles for exchange of files over the Internet for delivering video contents to a large community of users. There is a vast literature for various systems designed for streaming VoD both over the Internet and the Telecom-managed network. One such solution that implements BitTorrent-like protocols quite similar to the file sharing systems, but also proposes policies for peers selection for effective video exchange over the Internet is presented in [**?**]. Another P2P system for streaming VoD contents over the Internet is proposed in [**?**]. In this system each peer contributes in delivery of the video it is currently watching with the support of a streaming server in case of failures of missing contents on the peers.

Although the P2P streaming on the Internet has given positive results, its main disadvantage is the reliability of the peers and the Internet in general. The environment where the implementation of P2P streaming perfectly fits are the telecom-managed IPTV networks. Some of the reasons are that the set-top boxes (STBs) nowadays have considerable storage capacity and the operators have higher control over the devices on the clients premisses, avoiding the reliability issue of the classical P2P systems. The use of P2P in IPTV networks for live video contents and the contributions of various architectural designs are shown in [**?**]. In [**?**] a P2P assisted streaming system is proposed where the always present peers are supported by one server to provide the missing parts or make up for any failures. Another IPTV network architecture that takes advantage of the P2P is presented in [**?**] with the accent of the architectural aspects on the performance of the service. A solution that implements P2P streaming to reduce the load of hierarchically organized servers in busy hours is proposed in [**?**]. In this approach only the most popular content items are stored in the peers.

Assuming that in the IPTV networks the content items are distributed in a way that the most popular content items are stored in servers on the very edge of the networks so that they are closer to the clients, the the idea of storing copies of the popular contents in the STBs is quite a reasonable solution that could significantly reduce the traffic in the edge of the network, particularly in the busy hours when most of the traffic is dedicated for the popular contents. However there is a large number of contents that are not in the high popularity range, but still take significant part of the the overall traffic. Since they are stored in the servers that are in the core of the Telecom-managed network, the traffic generated for their streaming has to pass greater distance in order to reach the clients and therefore is a burden for the backbone of the network. The opposite case of distributing the unpopular contents on the STBs contributes to reducing traffic in the core of the network because it concentrates most of the traffic on the periphery of the network: the popular contents are streamed by the servers on the edge and a great part of the unpopular contents are streamed by the STBs. This is important when one of the objectives is reducing the transport cost in the network. Although both of the distributions bring improvements by reducing the overall traffic, they do not provide improved service in the cases of busy hours when the networks is taken to the limits. When the popular contents are stored on the STBs, the response time for service of unpopular contents is increased because the servers cannot serve all the incoming requests. The same happens when the unpopular contents are stored on the STBs with the difference that now not all the requests for popular contents can be immediately served.

Therefore we propose a solution for a network with popularity based distribution of contents, both on the streaming servers and STBs, that aims to reduce the traffic on the servers situated in the core of the network and at the same time tends to provide immediate service in the cases of high demand scenarios. One of the objectives targeted with the reduction of the traffic in the core are offloading the backbones from video traffic so that it can be used for other type of traffic and enabling growth of the number of clients subscribed to VoD service without additional changes and costs in the core of the network. Although the schemes that we propose consider all the contents, we put an accent on the low popularity contents by reserving more storage space on the STBs than the popular contents, thus providing locally close availability of most of the videos.

In our model we take advantage of the unused upload capacity of the clients and the storage capacity of the STBs to assist in the streaming of the VoD contents. Althought the uplink capacity of the links that connect the STBs to the network is far below the playback rate of the content items, uninterrupted viewing is achieved by combining a parallel streaming of various parts of the videos by as many peers as it is necessary for obtaining the required quality. Unlike many P2P solutions where the peers self-organize themselves, in our model the peers have a role of passive contributors to the streaming process having no knowledge of the existence of other peers and contents availability. They are only capable of serving those videos that they have already stored. All the decisions regarding redirection of the clients are taken by the servers on the edge of the network.

The rest of the paper is organized as follows. In Section II we describe the architecture of the proposed model for peer assisted VoD streaming, the division of the contents for better utilization of the storage of the STBs and the request process for VoD contents. In Section III we define the sizes of the streaming and storage capacity of the servers for their optimal utilization. In Section IV we present the simulation environment and analyse the obtained results for the proposed popularity based distribution schemes . Finally we give our conclusions in Section V.

II. Proposed model

A. Model arhcitecture

The model that we propose for optimal distribution of VoD contents is a hybrid solution that unites the advantages of both the IPTV and P2P architectures: the high reliability and scalability of the IPTV architecture and the storage space and unused up-link bandwidth of the P2P architecture. It consists of hierarchically organized streaming servers, management servers and STBs. The management servers are responsible for monitoring the system and taking decisions about redirection of the requests and the placement of the contents. We consider a managed network owned by a company which can be managed and configured according to the intensity of the requested traffic. The main streaming functionality is provided by the streaming servers, while the peers have the role to reduce the overall traffic in the network. Unlike the classical P2P solutions where the clients decide whether to share content or not, in an IPTV managed network the STBs are owned by the service provider and therefore part of their unused storage and streaming capacity can be reserved for the needs of the peer assisted streaming.

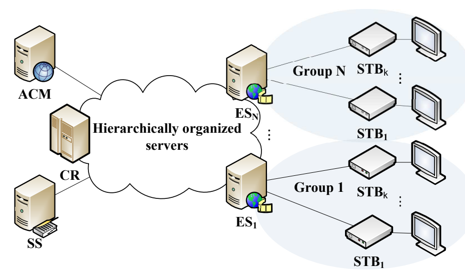

Fig. 1. Model architecture

The streaming servers are organized in a hierarchical structure according to the distance from the clients (Figure 1). These servers have limited storage and streaming capacity and therefore can host a limited number of contents and can serve a limited number of clients. The servers that are in the edge of the network, called Edge Servers (ES), serve only one group of locally connected clients. All the clients assigned to one ES form a local community. Each peer can serve only clients within the same local community. Each ES keeps track of the popularity of the entries it currently hosts and sends it to the Automatic Content Movement server (ACM) server for redistribution purposes. The ES also maintains availability data of the portions of the content items stored in its assigned peers. It uses this data to redirect the clients to other peers whenever there is request for contents that are already stored in the peers.

Another part of the system is the Central Repository (CR) which has the capacity to store all the contents. It is highest in the hierarchy and is entry point for new items. It doesn't directly serve the clients, but it supplies the streaming servers with the missing contents whenever it is necessary. The management servers are represented by the ACM and the Service Selection server (SS). The ACM server has the role to monitor the state of the network and to take decisions for a new replica distribution on the servers. When necessary, the ACM server runs a redistribution algorithm which decides the number of replicas for each content item and its position on the servers according to the contents popularity and servers utilization data. The main objective of the redistribution algorithm is to place the most popular contents starting from the edge servers to the servers higher in the hierarchy [**?**]. The ACM server periodically gathers information for the current state of the streaming servers in the system. Upon the execution of the redistribution algorithm, the ACM server issues commands which may include insertion or deletion of contents on particular servers.

The SS server is responsible for redirection of the requests to the right servers in a way that the transport cost is minimized and the load between the servers is equally distributed. In order to take the best redirection decisions, the SS server is frequently updated by the ACM server with the state of the system and the new position of the replicas.

The clients make requests to their assigned ES. If the ES is not able to serve the client, it addresses it to the SS server, which then redirects it to the most appropriate server. Clients can be served only by servers that are parents of their assigned ES. In the case when there are peers within the same community that contain parts of the requested content item, the ES takes the role of an index server. Additionally, the server redirects the client to the SS server for completing the streaming of the rest of the content. In case of failure of any peer, the missing parts are compensated from other peers or from the streaming servers.

The contents are distributed in the STBs in off-peak hours but we also use the volatile nature of popularity of the content items as an advantage for reduction of the distribution traffic. This property is due to users behaviour regarding repeating a request for a same content. Soon after a video is introduced in the system, it reaches high popularity, but as the time passes, the popularity decays because the clients who already saw the video are unlikely to request it again. Therefore a content item that is already viewed and stored in the STB of many clients is very likely to be later removed from the ESs as not popular. In such a way most of the contents with reduced popularity will be already stored in the STBs and available for peer assisted streaming. This saves a lot of additional traffic for distribution of contents from the streaming servers to the STBs. The decisions about the content placement on the peers are taken by the ES depending on the distribution determined by the ACM server.

B. Requesting process

The requesting process is initiated by the client which sends a request for a content item to its designated ES server. According to the content availability, there are the following cases: the ES already has the content; the server doesn't have the content nor any of the peers; the ES doesn't have the content but it knows which peers partially contain it; and the server is overloaded. In the first case, the ES sends acknowledgement to the client which is followed by a direct streaming session. In the second case the ES redirects the client to the SS server which then chooses the best server to serve it and sends it the address of the chosen server. Once the client has the address, the process is the same as the first case. In the case when some strips are stored in the peers, the ES looks up in its availability table and sends a strip-peer list of the available strips and their location. If there is not sufficient number of strips available on the peers, the ES redirects the client to the SS server. Just like in the previous case, the SS redirects the client to the best streaming server for the delivery of the missing strips. When the client receives the availability data of all the strips, it initiates streaming sessions with each peer of the list obtained by the ES server and at the same time initiates streaming session for the missing strips with the server assigned by the SS. The streaming sessions on the peers occupy the uplink capacity of the STBs and therefore once an ES sends the availability of the strips, it marks all the peers that contain those strips as unavailable until the end of the peer streaming session. When the streaming is over, the client updates the ES so that the strips of the content become available again. In the case when the server is overloaded, the request is rejected and the client retries requesting the content after determined time.

C. Content division

The limited up-link capacity of the last-mile links that interconnect the peers is several times smaller than the necessary playback rate of the content items. This capacity is insufficient for immediate and uninterrupted playing of the content items if they are streamed individually by the peers. With such a limitation, the peers cannot act as independent stream suppliers and therefore the content items are simultaneously streamed by as many peers as it is necessary for reaching its playback rate. Each peer streams a portion of the content item. When all the streaming portions are delivered to the receiving peer, they are assembled and the content item is played. The size of the streamed portion is important regarding for the initial delay since the content cannot be played until the entire length of the portion is received. Assuming that all the peers have the same up-link capacity u and that the maximum allowed initial delay is δ, then the size of each piece is $\Delta = \delta u$. The necessity of parallel streams requires storing many copies of the same content in many peers. This is quite an inconvenience considering the fact that in our model we store low popularity content items which represent the majority of the contents present in the system. Storing copies of such a big number of contents on the STBs would require huge storage capacity and also would generate significant amount of traffic for their distribution to the STBs. On the other side, the fact that each peer is streaming only a portion of the entire content makes it reasonable to store in the STB only those portions that the peer is capable to stream. This would contribute to increase the storage efficiency of the peers as well as the contents availability. Therefore we divide the content items into strips [**?**], where each strip contains equidistant portions with size Δ. The distance between the portions is $k\Delta$, where k is the number of required peers for uninterrupted streaming. Since the strips are k times smaller in volume than the original content, each peer can store k times more different content items, assuming that all the contents have on average the same size. All the contents that are stored in the STBs are entirely stored in the servers so that they can be delivered whenever the STBs are not able to provide any of the strips.

III. System dimensioning

The system we are considering consists of S streaming servers which belong to one of the L levels of the tree structure. Each server s, has a streaming capacity $U(s)$ and storage capacity $S(s)$ for storing a limited number of C content items. Each content item c has size $s(c)$ and playback rate $r_s(c)$. There are N clients in the system which are connected to one of the E edge servers.

One of the important issues for estimating the contributions of the proposed model is planning the streaming capacity $U(s)$ and storage capacity $S(s)$ of the servers so that they can comply to the requests of the N clients. Since the storage capacity of a server is more easily upgradeable than the streaming capacity and the capacities of the links that interconnect the servers, we will consider adjusting the storage space for a fixed streaming capacity. We assume that the servers at the edge of the network serve approximately the same number of clients and therefore have the same streaming and storage capacities. We also assume that all the content items have the same streaming rate r_s and average size $\overline{s}$.

We model the system size according to the popularity distribution of the content items and according to the way the servers are organized within the hierarchy. We consider that the popularity of the content items obeys the Zipf-Mandelbrot distribution and that they are previously ranked according to past request data and estimation of the recently inserted items. According to this distribution, the relative frequency (popularity) of the content item with *i*-th rank in the system is defined as:

$$f(i) = \frac{(i+q)^{-\alpha}}{\sum_{c=1}^{C}(c+q)^{-\alpha}} \tag{1}$$

where *q* is shifting constant and α is real number that typically takes values between 0.8 and 1.2. We consider that the distribution algorithm always places the most popular videos on the servers that are closest to the clients. The higher level a server has, less popular contents it will contain. Having this in mind, the condition that the streaming capacity $U(s)$ has to fulfil so that all the requests directed to server s can be served is

$$n(s) \cdot \sum_{c=a(s)}^{b(s)} f(c) r_s(c) \leq U(s) \tag{2}$$

where $n(s)$ is the maximum number of simultaneously served clients by server s. For the first level of the tree, $n(s)$ is the number of active peers in the local community of server s and in the rest of the levels it is the sum of all active clients in the communities that can be served by that server. The indexes $a(s)$ and $b(s)$ note the ranks of the first and the last most popular content items stored in server s. Considering the assumption that the edge servers serve the same number of clients, $n(s)$ can be expressed as

$$n(s) = \mu \frac{N}{E} T(s) \tag{3}$$

where $T(s)$ is number of served local communities and μ is the percent of active clients. The same assumptions let us define the initial rank of the contents on server s as one value above the rank of the least popular content stored in the servers in the level below. Thus, the problem is reduced to finding the rank $b(s)$ of the contents that will be placed on server s. If we substitute (1) and (3) in (2), than we get

$$\sum_{c=a(s)}^{b(s)} (c+q)^{-\alpha} \leq \frac{U(s)E}{\mu r_s T(s) N} \sum_{c=1}^{C} (c+q)^{-\alpha} \tag{4}$$

Once the indexes $a(s)$ and $b(s)$ are determined, the optimal storage capacity of the server is determined from the following condition

$$(b(s) - a(s) + 1)\overline{s} \leq S(s) \tag{5}$$

Since b(s) cannot be expressed in closed form, it is determined by using numerical methods.

IV. Simulations and results

We developed a simulation environment for testing the behaviour of the proposed model with various distributions of the content items on the STBs. In our experiments we consider a network of $S = 13$ streaming servers organized in a tree structure with $L = 3$ levels (Figure 2). The streaming capacities of the servers at level $l = 1, 2$ and 3 are $U(s) = 500, 1000$ and 1500 Mbps, respectively. The links that interconnect the servers have enough capacity to support the maximum streaming load of all the servers. The streaming servers host $C = 1700$ Standard Definition (SD) quality contents with playback rate $r_s = 2$ Mbps and average duration of 60 min.

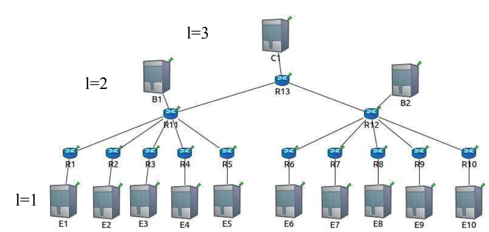

Fig. 2. Servers organization

The servers are serving $N = 5000$ clients divided into $E = 10$ groups, each group directly served by one ES. The maximum percentage of active clients in the system in the peak hours is $\mu = 0.9$. The clients posses STB with capacity to store the entire length of 3 content items. The STBs are connected to the network with links that have download capacity much higher than the playback rate of the SD video quality and uplink capacity $u = 200$ kbps, which is 1/10 of the SD playback rate ($k = 10$).

The popularity of the content items obeys the Zipf-Mandelbrot distribution with shifting coefficient $q = 10$ and $\alpha = 0.8$. The process of generating requests for VoD contents is modelled as a Poisson process. Taking into consideration these data, the storage and streaming capacities of the servers are dimensioned according to (5) and (4) in a way that they are optimally used. The contents are previously distributed on the servers.

In the simulations we considered several different scenarios. The first scenario is the reference for comparison and represents the simple case when the streaming process is completely done by the streaming servers. The number of clients that simultaneously request a content item is set to such a value that would keep the streaming servers constantly overloaded and the same request rate will be later used in all the simulation scenarios. In order to compare the contributions of the proposed distributions, we also consider the two simple cases when only the high popularity contents [**?**] and low popularity contents are uniformly distributed on the STBs. The low popularity content items are all those items that are not stored in the edge servers. In the simulations it is assumed that 20% of the total number of contents are popular.

In this paper we propose mixed schemes for distribution of the contents on the STBs which include both the popular and unpopular content items. Since our objective is to reduce the transport cost in the core of the network by P2P assisted streaming of the unpopular contents, but at the same time to keep the network highly responsive for popular contents in highly congested conditions, the key factor in definition of the distributions is the percentage of dedicated storage space for the unpopular and the popular contents. We define the distributions by assigning 10% of the storage capacity of the STBs for storing parts of the popular contents and the rest of the storage for the unpopular contents. The reservation of a small portion of the STBs storage space for the popular content items will provide sufficient alleviation of the edge servers in the busy hours and the rest of the storage will enable reduction of the backbone traffic. The distributions are based on the contents popularity and determine the number of strips of each content that will be distributed on the peers. Each distribution consists of two equal distributions applied to the popular and unpopular contents. Figure 3 shows some of the considered content distributions.

As the system is taken to its limits and the servers are constantly kept in a state of high utilization during the simulations, the requests directed to the overloaded servers are rejected and the clients are demanded to repeat the request latter. Figures 4 and 5 show the percentage of requests that are rejected for immediate service due to overloaded state of the servers and the time they have to wait until they are served. Since the network is taken above its limits in the scenario with no P2P assisted streaming, the high miss rate is rather expected. Figures 4 shows that the implementation of P2P introduces reduction in this rate, but it largely depends on the distribution of the contents on the STBs. The distribution of only popular contents significantly reduces the miss rate, which is not the case with the unpopular contents. The results are much better when the mixed distributions are used, because the miss rate decreases to values lower than 5%. The lowest miss rate is obtained in the case when the contents are distributed according to the Z-Zipf function, followed by the U-Uniform and L-Linear distribution.

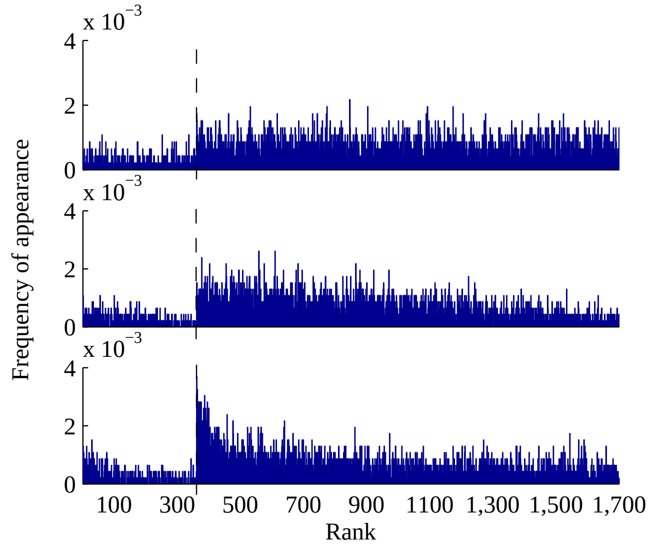

Fig. 3. P2P content distributions: U-Uniform, L-Linear and Z-Zipf

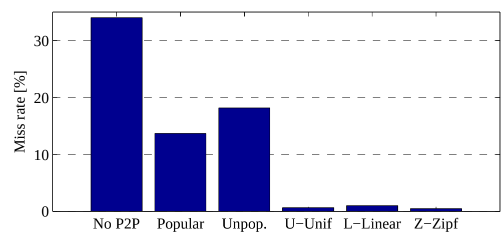

Fig. 4. Miss rate

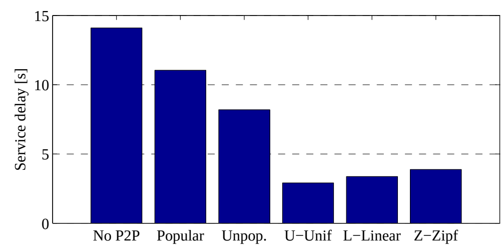

Fig. 5. Service Delay

The advantages of the mixed distribution schemes are also visible in the reduction of the service delay. Whenever a client is denied, it has to wait much shorter time when the contents are distributed according to the proposed mixed distribution rather than the cases of the other distributions. Another measure that we analyse in order to estimate the contribution of each of the considered distributions is the transport cost for delivering the content items to the clients. This measure is mainly based on the distance of the servers from the clients and their current load.

$$Cost = \sum_{s=1}^{S} d(s)u(s) \tag{6}$$

where $d(s)$ is the distance of server s from the local communities it is serving, counted as number of links and $u(s)$ is its current streaming rate. Since the P2P streaming is done over the unused uplink rate of the clients, we don't include it in the overall cost function.

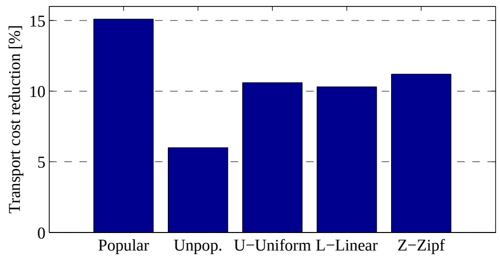

Fig. 6. Transport cost reduction

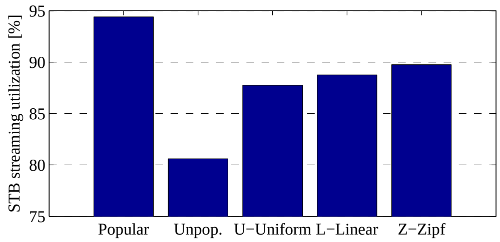

Fig. 7. STB capacity utilization

Figure 6 shows the average transport cost reduction that is obtained as a result of implementation of various distribution schemes for P2P assisted streaming compared to the case when only a server streaming is implemented. The P2P streaming of the most popular contents introduces lowest reduction because it only contributes to reducing the load on the edge servers. On the contrary, the distribution of the unpopular contents on the STBs reduces the traffic in the higher layers and therefore it reaches maximum value of the reduction of the transport cost. Although the difference is almost insignificant, the Z-Zipf distribution of on the STBs contributes the most for reduction of the transport cost, followed by the L-Linear and Z-Zipf distribution. On Figure 7 the utilization of the total streaming capacity of the STBs is shown. The peers capacity is almost entirely used when the popular contents are distributed. The lowest value of the utilization is in the case of the peer distribution of the unpopular contents.

These results show that although the mixed distribution schemes do not reach the maximum cost saving and peer utilization of the simple distributions, they are a good compensation for the weak points of each one of them. What is very important is that they significantly improve the number of immediately served clients and the average service delay, which under no condition can be reached by the simple distributions.

One important contribution of the reduction of the traffic in the network core is the possibility to serve more clients with the same streaming capacity of the servers in the core of the network. The advantage of the higher number of clients in the system is that it also implies higher storage and streaming capacities for serving the request for unpopular contents. The only price that has to be paid for the higher number of clients is the installation of new ES on the periphery of the network that would satisfy the demand of the most popular contents. In the case when the popular contents are stored in the STBs, a higher number of clients would require both installation of additional ES and increasing the capacity of the links and the streaming servers in the core of the network. Therefore, the proposed distribution schemes not only reduce the transport cost, miss rate and service delay, but also reduce the installation costs in case of increasing the number of clients in the system.

V. Conclusions

In this work we proposed a P2P assisted VoD streaming model that uses the unused storage and uplink capacities of the STBs. We also proposed popularity based distribution schemes of the contents on the STBs that help to reduce the transport cost in the core of the network and to better take advantage of the unused peer uplink capacity. In addition, the proposed schemes improve the responsiveness of the network by reducing the percentage of rejected clients for immediate service as well as the time they have to wait to be served. The reduced traffic in the core of the network gives the possibility to increase the number of clients without high costs and additional changes in the core of the network, making the system highly scalable.

Acknowledgement

This work has been partially supported by the Ministerio de Ciencia e Innovación of the Spanish Government under project TEC2010-20412 (Enhanced 3DTV).