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, ETSI Telecomunicación, Universidad Politécnica de Madrid, Spain
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 all requests. 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 we present popularity based schemes for distribution of both the popular and unpopular contents on the peers. The proposed model and the schemes prove to reduce the streaming traffic in the core of the network, improve the responsiveness of the system and increase its scalability.**

***Keywords*-P2P; VoD; streaming; popularity.**

I. Introduction

The great expansion of the IPTV [1] has made a good ground for the Video on Demand (VoD) to become one of the most popular services. Although VoD is a service that is also available on Internet, it has attracted special attention in the field of the Telecom-managed networks since they are already adapted to the implementation of a variety of TV services. Despite of its numerous advantages from client's point of view, the VoD service is a serious issue for the providers 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 a 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 [2][3][4]. No matter how good these solutions might be, they all reach a point from where no further improvements can be done due to resource limitations. One possibility to overcome this problem is the implementation of the classical P2P principles for exchange of files over Internet for delivering video contents to a large community of users. Some systems designed for streaming VoD over Internet are presented in [5][6]. Despite of its numerous advantages, the P2P streaming over Internet lacks reliability. The environment where the implementation of P2P streaming perfectly fits are the telecom-managed IPTV networks. Some of the reasons for that are the considerable storage capacity of the set-top boxes (STBs) nowadays and the higher control of the operators over the devices on the clients premisses, which avoids 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 [7]. In [8], a P2P assisted streaming system is proposed, where the 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 [9]. A solution that implements P2P streaming to reduce the load of hierarchically organized servers in busy hours is proposed in [10]. In this approach, only the most popular content items are stored in the peers.

Assuming that the content items in the IPTV networks are distributed in a way that the most popular content items are stored in servers that are closer to the clients, 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. However, there is a large number of contents that are not in the high popularity range, but still take significant part of the overall traffic. Since they are stored in more distant servers in the core of the network, the traffic generated for their streaming is a burden for the backbone of the network. The opposite case of distributing the unpopular contents in the STBs contributes to reducing the traffic in the core of the network because it concentrates most of the traffic in 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 for the entire set of contents in the cases of busy hours. When the popular contents are stored in 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 in 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 in the core of the network and, at the same time, tends to provide immediate

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-195-3

service in the cases of high demand scenarios. One of the objectives targeted with the reduction of the traffic in the core is 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 in 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 and storage capacity of the STBs to assist in the streaming of the VoD contents. The streaming is done by parallel streaming sessions of multiple STBs in order to compensate for their limited streaming capacity. 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. They are only capable of serving the 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 define the popularity based distributions and in Section V, we present the simulation environment and analyze the obtained results. Eventually, we give our conclusions in Section VI.

II. Proposed model

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.

A. Model arhcitecture

The proposed model's architecture 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 company owned network 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.

The streaming servers are organized in a hierarchical tree structure according to the distance from the clients (Figure 1).

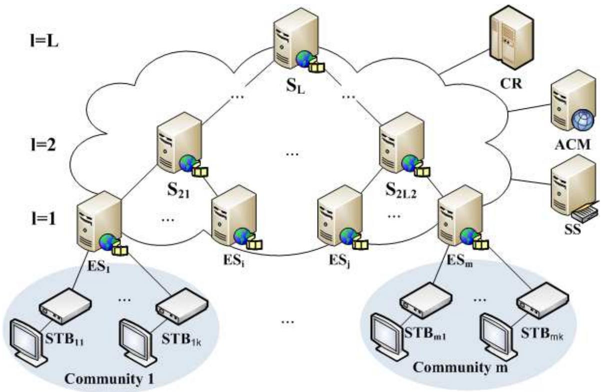

Fig. 1. Model architecture

These servers have limited storage and streaming capacity, so they 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. Clients assigned to one ES cannot be served by an other ES because the tree structure of the servers architecture implies longer distances between them. Each peer can serve only clients within the same local community. The clients from a community cannot be served by other communities because that would cause additional traffic burden in the core of the network. 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 these data to redirect the clients whenever there is request for contents that are already stored in the peers.

Another part of the system is the Central Repository (CR) which is a server with capacity to store all the contents. It is highest in the hierarchy and it is entry point for new items. It does not directly serve the clients, but it supplies the streaming servers with the missing contents when 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, using popularity and state data, decides the number and the position of the replicas for each content item within the network. The objective of the redistribution algorithm is to place the contents in a way that the most popular contents will be stored in the edge servers and the less popular contents in the servers higher in the hierarchy [4]. The ACM server periodically gathers information for the current state of the streaming servers. 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

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-195-3

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. In the proposed model, we consider that this functionality does not require additional hardware upgrades and delays of service. 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 comes as result of the behaviour of the users for not repeating a request for the 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 in the peers are taken by the ES depending on the distribution determined by the ACM server.

B. Content division

The division of the contents into smaller strips is inevitable in the implementation of the P2P assisted streaming. The main reason for that is the limited up-link capacity of the STBs, which is several times smaller than the necessary playback rate. For immediate and uninterrupted playing, a content item has to be streamed in parallel 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 portions reach the peer, they are assembled and the content is played. The size of the streamed portion Δ is determined as a product of the minimum STB's streaming capacity u and the maximum acceptable initial viewing delay, defined as the time necessary for the entire length of a portion to be received. Each strip consists of consequent streaming portions that are on distance $k\Delta$ between each other, where k is the ratio between the play rate r_s and the minimum up-link capacity u.

The division of the contents also contributes for increasing the storage efficiency of the peers and the contents availability. Considering that each peer is capable of streaming only a portion of the content makes it reasonable to store only those portions that it is capable to stream. Since the strips are k times smaller in size 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.

C. 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 does not have the content, nor any of the peers; the ES does not 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 in 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 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 obtained list 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, and the strips 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.

III. System dimensioning

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

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-195-3

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 the 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 is defined as:

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

where α is a real number that typically takes values between 0.8 and 1.2 and q is a shifting constant. This distribution is a generalized form of the Zipf distribution, which includes the shifting constant q in order to characterize the behaviour of the clients for not repeating requests of already seen content items [11]. We consider that the distribution algorithm always places the most popular videos in 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

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

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 in server s. If we substitute (1) and (3) in (2), we get

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

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. Distribution schemes

In this paper, we propose mixed schemes for distribution of the contents on the STBs which include both the popular and unpopular content items. By combining these simple distributions, we take advantage of the contributions of each one of them: the distribution of popular contents makes the network more responsive in highly congested conditions, and the distribution of the unpopular contents makes the streaming process locally closer to the clients for all the available contents and thus reduces the traffic in the core of the network. One of the key factors in the definition of the distributions is the percentage h of dedicated storage space for popular and unpopular contents. We should keep in mind that the STBs store only strips of the contents and, therefore, increasing the storage reserved for popular contents would keep more of the STBs busy and the strips of the unpopular contents could be rarely used.

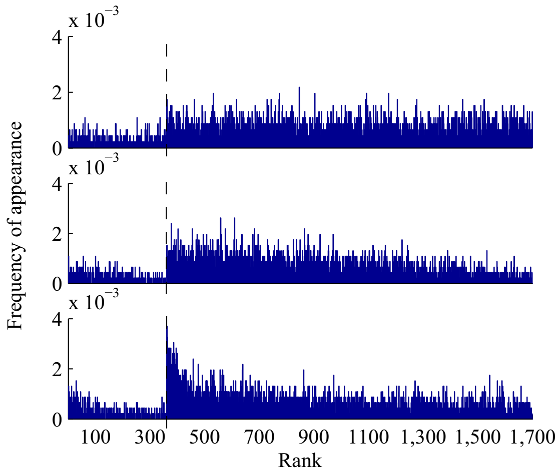

Fig. 2. P2P content distributions: U-Uniform, L-Linear and Z-Zipf

Since our main objective is to concentrate the traffic in the periphery of the network, we dedicate most of the storage capacity to 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. By means of simulation we obtained that our objectives are best fitting for values of h that belong to the interval between 10 and 15%. The distributions are based on the contents popularity and determine the number of strips of each content that will be distributed in the peers. Each distribution consists of two equal distributions applied to the popular and unpopular contents. The single distributions applied to both the popular and unpopular contents are Uniform, Linear and Zipf distribution. Another important issue is determining the border between popular and unpopular contents. Following the 80-20 rule of the Pareto distribution which states that 80% of the total number of requests is addressed for the first 20% most popular contents, we will consider 20% to

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-195-3

be the border which will distinguish the contents as popular or unpopular, although we consider different distribution of contents' popularity. Figure 2 shows some of the considered content distributions.

V. 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 1) where each level $l = 1, 2$ and 3, contains 10, 2 and 1 servers, respectively. The streaming capacities of the servers in the same order of levels are $U(s) = 500, 1000$ and 1500 Mbps. 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.

The servers are serving $N = 5000$ clients divided into $m = 10$ communities, each community directly served by one ES. The maximum percentage of active clients in the system in the peak hours is $\mu = 85\%$. The clients posses STBs with capacity to store the entire length of 3 content items. The portion of this storage reserved for strips of the popular contents is $h = 0.12$. 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 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 (no P2P). 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, proposed in [10], are uniformly distributed on the STBs and when the low popularity contents are distributed on the STBs.

Because in our simulations, the servers are kept in a state of high utilization, some of the requests directed to the overloaded servers are rejected and the clients are demanded to request the content latter. 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 are shown in Figures 3 and 4. The high miss rate in 3, in the scenario with no P2P assisted streaming, is

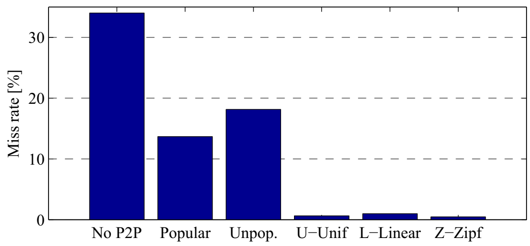

Fig. 3. Miss rate

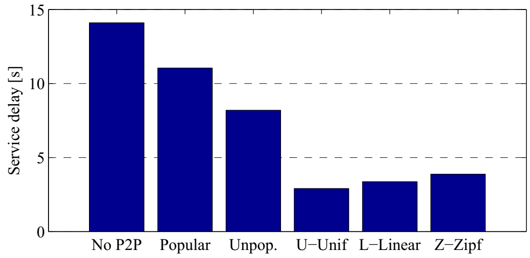

Fig. 4. Service Delay

quite expected result since the request rate of the clients is higher than the available resources. The figure shows that the implementation of P2P assisted streaming for any distribution of the contents on the STBs introduces reduction of the miss rate. The simple distribution of unpopular contents reduce the number of rejected clients to half. This effect is even more emphasized in the distribution of popular contents [10]. The proposed mixed distributions, however, introduce significantly lower miss rates with values around 1%. Although there is only slight difference, the lowest miss rate is obtained for the Z-Zipf distribution, followed by the U-Uniform and L-Linear distribution.

The advantages of the mixed distribution schemes are also visible in the reduction of the service delay (Figure 4). Whenever a client is denied, it has to wait much shorter time when the contents are distributed according to the proposed mixed distributions compared to the other cases.

Another measure that we analyse in order to estimate the contribution of the considered distributions is the transport cost for delivering the streaming traffic from the streaming servers to the clients. This measure is mainly based on the distance of the servers from the clients and their current load and it is expressed as

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

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 omit it in the calculation of the transport cost.

Figure 5 shows the average transport cost reduction obtained as a result of the implementation of the various distribution schemes for P2P assisted streaming relative to the case of pure server streaming. The P2P streaming of the most popular contents introduces lowest reduction because it only contributes

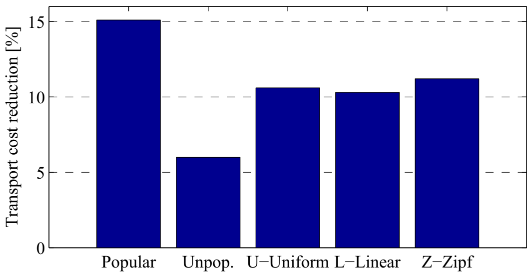

Fig. 5. Transport cost reduction

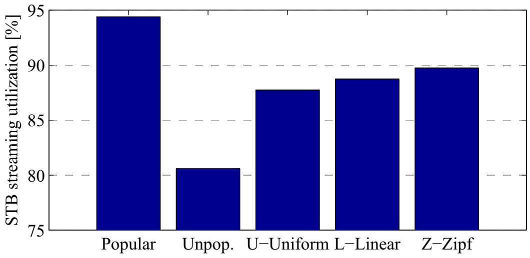

Fig. 6. STB capacity utilization

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 the maximum reduction of the transport cost. Although the difference is almost insignificant, the Z-Zipf distribution contributes the most for reduction of the transport cost, followed by the L-Linear and U-Uniform distribution.

The various distribution schemes also contribute to a different streaming capacity utilization of the STBs. This dependence is shown in Figure 6. The utilization of the proposed schemes lays between the maximum value obtained for the popular contents distribution and the minimum value obtained for the unpopular content distribution, which is a good compromise considering the improvements that the schemes introduce in the transport cost and the quality of service. The 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 both of them. In addition, 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 more requests. 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.

VI. 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 determined by assigning different portions of the available storage capacity for the popular and unpopular contents. These schemes prove to reduce the transport cost in the core of the network and to well utilize the uplink capacity of the STBs. In addition, the proposed schemes improve the quality of service that receive the clients by reducing the percentage of rejected request for immediate service as well as the time they have to wait to be served. The reduced traffic in the core of the network and the improved responsiveness give the possibility to increase the number of clients in the system without high installation 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).

References

[1] W. Simpson and H. Greenfield, *IPTV and Internet Video: Expanding the Reach of Television Broadcasting*. Elsevier Science & Technology, 2009.

[2] D. De Vleeschauwer and K. Laevens, “Performance of caching algorithms for iptv on-demand services,” *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 491 –501, June 2009.

[3] M. Verhoeyen, D. De Vleeschauwer, and D. Robinson, “Content storage architectures for boosted IPTV service,” *Bell Labs Technical Journal*, vol. 13, no. 3, pp. 29–43, 2008.

[4] S. Gramatikov, F. Jaureguizar, J. Cabrera, and N. Garcia, “Content delivery system for optimal vod streaming,” in *Proceedings of the 11th International Conference on Telecommunications*, june 2011, pp. 487 – 494.

[5] N. Carlsson, D. L. Eager, and A. Mahanti, “Peer-assisted on-demand video streaming with selfish peers.” in *Networking*, ser. Lecture Notes in Computer Science, vol. 5550. Springer, 2009, pp. 586–599.

[6] C. Huang, J. Li, and K. W. Ross, “Peer-Assisted VoD: Making Internet Video Distribution Cheap,” in *6th International Workshop on Peer-to-Peer Systems*, 2007.

[7] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, “On next-generation telco-managed p2p tv architectures,” in *Proceedings of the 7th international conference on Peer-to-peer systems*, 2008, pp. 5–5.

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

[9] Y.-F. 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.” *Multimedia Syst.*, vol. 15, no. 1, pp. 19–32, 2009.

[10] Y.-F. Chen, R. Jana, D. Stern, B. Wei, M. Yang, H. Sun, and J. Dyaberi, “Zebroid: using IPTV data to support STB-assisted VoD content delivery,” *Multimedia Systems*, vol. 16, no. 3, pp. 199–214, 2010.

[11] B. Krogfoss, L. Sofman, and A. Agrawal, “Caching architectures and optimization strategies for IPTV networks,” *Bell Labs Technical Journal*, vol. 13, no. 3, pp. 13–28, 2008.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-195-3