P2P Assisted Streaming for Low Popularity VoD Contents

Sasho Gramatikov[1], Fernando Jaureguizar[1], Igor Mishkovski[2], Julián Cabrera[1] and Narciso García[1]

[1] Grupo de Tratamiento de Imágenes, ETSI Telecomunicación,
Universidad Politécnica de Madrid, Spain
{sgr,fjn,julian.cabrera,narciso}@gti.ssr.upm.es
[2] University "Ss Cyril and Methodius", Skopje, Macedonia
igor.mishkovski@finki.ukim.mk

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, due to its high traffic demand nature, the VoD streaming systems are faced with the problem of huge amounts of traffic generated in the core of the network, especially for serving the requests for content items that are not in the top popularity range. Therefore, we propose a peer assisted VoD model that takes advantage of the clients unused uplink and storage capacity to serve requests for less popular items with the objective to keep the traffic on the periphery of the network, reduce the transport cost in the core of the network and make the system more scalable.

Keywords: P2P, streaming, VoD, popularity, cost, scalability.

1 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 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 client's point of view, the VoD service is an issue from provider's 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 [2][3][4]. 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 [5][6]. Despite the P2P streaming on the Internet has given positive results, its main disadvantage is the reliability of the peers and the Internet. 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. Some solutions for the use of P2P in IPTV networks are presented in [7][8][9][10].

Although popular contents generate considerate amount of the overall traffic, there is a large number of contents that are not in the high popularity range, but still take significant part of the streaming traffic. Assuming that in the IPTV networks the content items are distributed according to their popularity, the requests for the less popular contents are a burden for the core of the Telecom-managed network since these contents are stored in servers which are placed further from the clients. Therefore, we propose a solution for a hierarchically structured network with popularity based distribution of contents that aims to reduce the traffic on the servers situated in the core of the network by providing peer assisted streaming of the low popularity contents. This would offload the backbone links of the network and would enable growth of the number of clients subscribed to VoD service without considerate changes and high costs in the core of the network. Despite the common practice to store the most popular contents in the peers for best traffic reduction [10], in our approach, we choose to store the low popularity contents in the peers, thus providing locally close availability of all the popularity range of the videos. We use this approach because the popular contents are already stored in the edge servers which are close to the clients. We tend to reduce the traffic in the core of the network by maintaining replicas on the clients' STBs of those items that are not popular enough to be stored in the servers at the edge of the network. Although the upload capacity of the clients 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. With this approach we achieve to concentrate large part of the overall traffic in the periphery of the network and thus, to reduce the transport cost of the traffic from the streaming servers to the clients. Unlike many P2P solutions where the peers self-organize themselves, in our proposed model, the peers have a role of passive contributors to the streaming process having no knowledge of the existence of other peers. 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 2, we describe the proposed model for peer assisted VoD streaming. In Section 3, we explain the division of the contents for better utilization of the storage capacity of the STBs. In Section 4, we describes the request process for VoD contents and in Section 5, we present the simulation scenarios and the obtained results. Finally we give our conclusions in Section 6.

2 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. It consists of hierarchically organized streaming servers, management servers and STBs. 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 alleviate 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 storage and streaming capacity can be reserved for the needs of the peer assisted streaming. The hierarchical server architecture is populated with content items in such a way, that the most popular contents are placed in the servers on the edge of the network and the less popular ones are placed in the higher layers of the hierarchy. Since most of the requests are for the popular contents, the traffic will be concentrated on the network periphery, but there will be still a considerable amount of traffic in the backbone of the network for serving the less popular contents. Therefore, we place replicas of these unpopular contents in the peers so that instead of streaming them from the servers in the core of the network, they can be streamed from the peers. In the cases when there are not enough available resources on the peers for streaming the entire content of the low popularity items, the streaming servers are available to serve the missing parts. With this approach, we provide maintaining most of the streaming traffic close to the edge of the network.

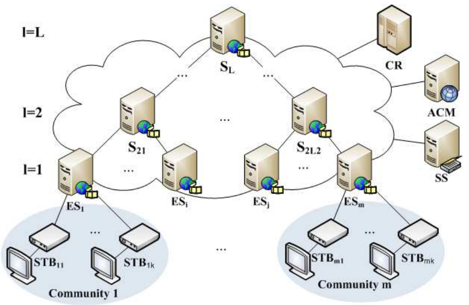

Fig. 1. Model architecture

The streaming servers are organized in a hierarchical structure according to the vicinity to the clients (Figure 1). 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 currently hosted content items. 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.The Central Repository (CR) 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. When necessary, the ACM server runs a redistribution algorithm which decides the number of replicas for each content item and its position in the servers according to the content's popularity and server's utilization data [4]. 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. 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 the low popularity 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.

3 Content Division

The limited up-link capacity of the last-mile links that inter-connect 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. When all the streaming portions are delivered to the receiving peer, they are assembled and the content item is played. 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 [10], where each strip contains equidistant portions with a predetermined size that will depend on the minimum allowed initial delay. The distance between the portions is k multiple of portions, 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.

4 Request Process Description

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, and the ES doesn't have the content but it knows which peers partially contains it. 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 and at the same time initiates streaming session for the missing strips with the server assigned by the SS. The ES keeps track of available streaming capacity of each peer.

5 Simulations and Results

We developed a simulation environment for testing the behaviour of the proposed model. In our experiments we consider a network of $S = 13$ streaming servers organized in a tree structure with $L = 3$ levels (Figure 1), such that the lowest level consists of 10 ES, the next level has 2 servers and the highest level has one server. Each of the ES forms a local community of $N = 400$ clients. The streaming capacities of the servers at each level are such that all the requests for the contents they contain can be immediately served. The links that interconnect the servers have enough capacity to support the maximum streaming load of all the servers. The streaming servers host $C = 1500$ Standard Definition (SD) quality contents with playback rate $r_s = 2$ Mbps and average duration of 90 min. For P2P streaming purposes, the contents are divided in $k = 10$ strips. The clients posses STB with capacity to store the entire length of 5 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 of three simultaneous strip streams which is $u = 600$ kbps.

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. In our simulation scenarios, the shifting coefficient $q = 10$ and $\alpha = 0.8$.

We divide the contents according to their popularity in two groups: popular and unpopular contents. We consider that the first 20% of the videos are popular. This division is based on the pareto distribution where 80% of all the requests are a aimed for the first 20% most popular contents and is a common practice for classifying the contents in many related works, although the requests process does not obey the pareto distribution. The process of generating requests for VoD contents is modelled as a Poisson process with average waiting time between two request of 20 min.

The contents are previously distributed on the servers and on the STBs in the hours of the day when there is very low activity of the clients and there is plenty of unused available link capacity that can be dedicated for distribution of the contents. The popular content items are stored in the ES, and the unpopular contents are stored in the higher levels of the hierarchy. The classification of the contents in popularity groups and the decisions on how to be distributed each of the contents in the servers are done by the ACM server.

In the simulations, we considered several different scenarios. The first scenario is the reference for comparison and consists of the simple case when the streaming process is completely done by the servers. In the second scenario, the clients are capable of serving part of the requests for the low popularity content items. In order to compare the results of the proposed model and a related work [10], we consider the case when only the most popular contents are distributed on the STBs. We also include the scenario when both the popular and unpopular contents are equally distributed on the STBs. We assume that the contents are distributed uniformly, i.e. the number of replicas of any content is the same, no matter its popularity. Since each STB contains different strips of the content items, in order to obtain equilibrium of the uplink utilization of each STB, we distribute the strips in such a way that the average popularity of all the strips stored in one STB is the same.

The overall server streaming traffic and the backbone traffic (the streaming traffic that is originated from the servers at the higher levels of the hierarchy) are shown in Figure 2. As expected, the introduction of any type of P2P streaming reduces the overall streaming traffic. However, there is a remarkable distinction in the reduction of the backbone traffic in each of the scenarios. We can see that the distribution of only the popular contents does not change the amount of generated backbone traffic compared to the case when there is no P2P at all, while the distribution of the unpopular contents considerably reduces this traffic. When the peers are serving the requests for unpopular contents, a significant part of the backbone traffic is redirected to the peers, keeping the ES equally busy as in the case of pure server streaming. Thus, a larger amount of the overall traffic is concentrated in the periphery of the network. When peers are serving the requests for the popular contents, a significant part of the ES traffic is reduced, which is not of a great importance, since the ES servers are not optimally used and there is no reduction of the backbone traffic. The only advantage of the later case is that a slightly better reduction of the overall server traffic is achieved. The reason for this is the fact that, although the popular contents are represented by only 20% of the total contents, they generate approximately 56% of the total traffic in the system (1) and therefore, serving the requests for these contents by the peers will reduce more the overall traffic compared to serving the unpopular contents. The values for the overall server and backbone traffic in the last case when all the contents are equally distributed on the STBs are in-between the values of the previously considered P2P distributions.

Since one of our goals is to concentrate most of the traffic within the local communities, we introduce a value called traffic locality that will note the level of localization of the traffic in the system. We define this value as percent of the overall streaming traffic in the system streamed by any member of a local community i.e. a peer or ES. This value is mostly dependant on the amount of the traffic that are able to serve the peers. Therefore, we show the traffic locality for the previously considered cases of distribution of the contents in the STBs for various streaming capacities of the peers. The values of the streaming capacity of each peer vary from bandwidth enough to stream only one strip to bandwidth for streaming $k = 10$ simultaneous strips, which is same as the play rate r_s of the video. The results are shown in Figure 3.

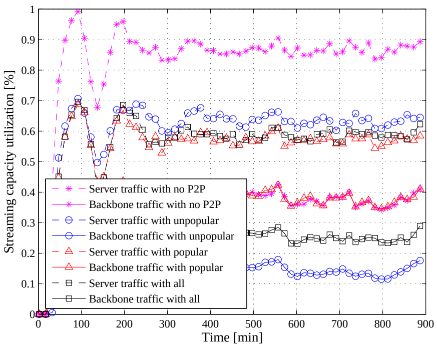

Fig. 2. Throughput Utilization

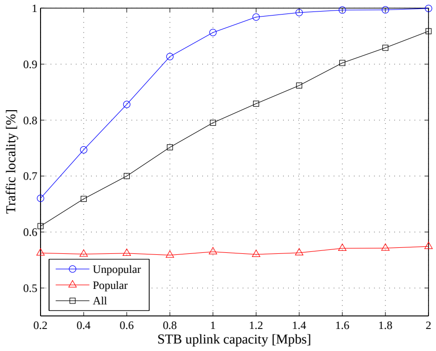

Fig. 3. Traffic locality

In the figure we can observe that when only the popular contents are distributed in the STBs, there is no change of the traffic locality, and moreover, it keeps the same value as the case when no P2P is implemented at all. This result is quite expected because although the peers serve a significant part of the traffic, they only relieve the ES servers, and the traffic generated for serving the popular contents by the ES is now passed to the peers. In that case, the traffic of the unpopular contents remains the same and there is no improvements in the locality. On the contrary, the distribution of only the unpopular contents considerably improves the traffic locality even for small values of the STBś streaming capacity. As the streaming capacity of the STBs grows, the traffic locality rapidly grows and finally reaches 100% local coverage of service. In the case of distribution of all the contents, the traffic locality increases linearly with the STBs uplink streaming capacity, but it has lower values than the case of distribution of unpopular contents and never reaches full local coverage of service.

In order to estimate the contribution of each of the considered distributions, we define a transport cost for delivering the content items to the clients. This measure is mainly based on the distance of the servers from the clients they are serving and their current load.

$$Cost = \sum_{s=1}^{S} d(s)u(s) \tag{2}$$

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

Figure 4 shows the average transport cost for P2P assisted streaming for various streaming rates of the STBs, relative to the case when all the streaming is done by the servers. We can see that both the curves of only popular and only unpopular contents have a similar behaviour. In the lower range of streaming rates, the traffic cost decreases as the rate increases, and than asymptotically converges towards a certain limit value. The difference between these two distributions is

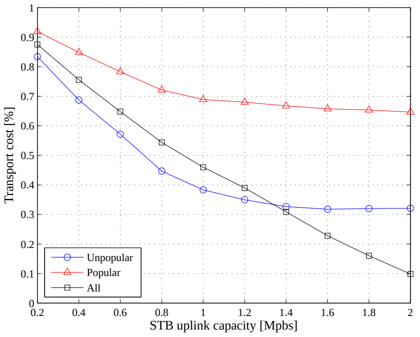

Fig. 4. Transport cost

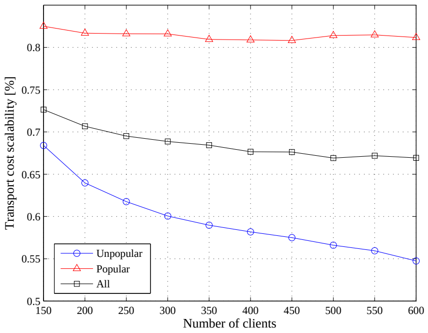

Fig. 5. Transport cost scalability

that distributing the unpopular contents contributes to a lower transport cost. Moreover, it gives lowest values of all the distributions for a smaller values of the streaming rates of the STBs. For higher values it is over-performed by the uniform distribution of all the contents because in this case, the traffic on each level of the hierarchy reduces as all the contents are both placed in the servers and in the STBs. In the cases of distribution of only certain range of popularity, there is only reduction of the traffic on the edge of the network or in the higher levels. We should also keep in mind that with the current ADSL technology most of the operators offer limited uplink capacity of its users which reaches values of up to 1 Mbps. This is actually the range in which the distribution of unpopular contents gives the best results.

One of the advantages of the P2P systems, in general, is their scalability. The more clients are present in the system, the more streaming and storage capacity is available. However, the increased number of clients also implies more requests, which in our model for P2P assisted streaming will also imply more streaming traffic for the servers. Our objective is to see how the different distributions will affect the scalability of the system since the higher request rate will also require increasing the number of streaming servers and the capacities of the links in the core of the network. In Figure 5 we present the dependence of the transport cost of the system relative to the case when no P2P is implemented on the number of clients in the system. In this simulation scenario, the STBs have streaming rate of $u = 600$ kbps. The distribution of the popular contents has no improvement in the scalability of the transport cost with the growth of the system size. This means that the cost relative to the system with no P2P distribution will not reduce although there are more available resources. The distribution of the unpopular contents appears to give far better results, since the higher number of clients will reduce the transport cost relative to the case when no P2P is implemented. As in the previous cases, the distribution of all the contents gives a moderate contribution that is between the contributions of a single range popularity distribution.

Apart from the considered transport cost, we should not forget that there is a certain cost for installation of new servers in any level of the network when the size of the system increases. According to the results discussed before, installation of new servers at any of the levels in any of the distributions would be inevitable. What makes the distribution of the unpopular contents favourable, is the fact that installation of new ES is the only price that has to be paid for increasing the number of the clients. This contributes to serve more clients with the same streaming capacity of the servers in the core of the network. In the case of distribution of popular contents, however, the growth of the number of clients will not require only new servers, but it will also require installation of new links in the core of the networks, which is quite an expensive process.

6 Conclusions

In this work we proposed a P2P assisted VoD streaming model for privately managed network that takes the advantage of the unused storage and uplink capacities of the STBs to serve the less popular content items. We made analysis for different distributions of the contents on the STBs and showed that by storing the unpopular contents it can be achieved to concentrate most of the streaming traffic within the service area of the edge servers. The localization of the streaming traffic close to the clients contributes to reduce the backbone traffic of the network, which consequently reduces the traffic transport cost. Other advantage that comes with the reduction of the backbone traffic is the increased scalability of the system since the same links of the core of the network are able to serve higher number of new clients with no additional upgrades. The only cost that has to be paid is the installation of new server on the edge of the network, which is an inevitable cost of any P2P distribution.

References

1. Simpson, W., Greenfield, H.: IPTV and Internet Video: Expanding the Reach of Television Broadcasting. Elsevier Science & Technology (2009)
2. De Vleeschauwer, D., Laevens, K.: Performance of caching algorithms for iptv on-demand services. Broadcasting, IEEE Transactions on **55**(2) (2009) 491 –501
3. Verhoeyen, M., De Vleeschauwer, D., Robinson, D.: Content storage architectures for boosted IPTV service. Bell Lab. Tech. J. **13**(3) (2008) 29–43
4. Gramatikov, S., Jaureguizar, F., Cabrera, J., Garcia, N.: Content delivery system for optimal vod streaming. In: Telecommunications (ConTEL), Proceedings of the 2011 11th International Conference on. (june 2011) 487 – 494
5. Carlsson, N., Eager, D.L., Mahanti, A.: Peer-assisted on-demand video streaming with selfish peers. In: Networking. Volume 5550 of Lecture Notes in Computer Science., Springer (2009) 586–599
6. Huang, C., Li, J., W. Ross, K.: Peer-Assisted VoD: Making Internet Video Distribution Cheap. IPTPS07 (2007)
7. Cha, M., Rodriguez, P., Moon, S., Crowcroft, J.: On next-generation telco-managed p2p tv architectures. In: Proceedings of the 7th international conference on Peer-to-peer systems. IPTPS'08, Berkeley, CA, USA, USENIX Association (2008) 5–5
8. Suh, K., Diot, C., Kurose, J., Massoulie, L., Neumann, C., Towsley, D.F., Varvello, M.: Push-to-peer video-on-demand system: Design and evaluation. IEEE Journal on Selected Areas in Communications **25**(9) (2007) 1706–1716
9. Chen, Y.F., Huang, Y., Jana, R., Jiang, H., Rabinovich, M., Rahe, J., Wei, B., Xiao, Z.: Towards capacity and profit optimization of video-on-demand services in a peer-assisted iptv platform. Multimedia Syst. **15**(1) (2009) 19–32
10. Chen, Y.F., Jana, R., Stern, D., Wei, B., Yang, M., Sun, H., Dyaberi, J.: Zebroid: using IPTV data to support STB-assisted VoD content delivery. Multimedia Systems (may 2010)