Peer-to-Peer VoD: Streaming or Progressive Downloading?

Attila Kőrösi and Balázs Székely

Abstract—**In this paper we compare the necessary server load of a generic Peer-to-Peer supported Video-on-Demand service in peak hours for two server delivery methods, *streaming* and *elastic progressive downloading*. We focus on the playback fault free scenarios. A simple user model is used for the VCR (Video Cassette Recording) actions behavior and for the behavior as an uploader peer. Based on analytical results we identify those settings in which streaming works better. This gives the opportunity for the operator to choose between the delivery methods.**

Index Terms—**Peer-to-Peer, Video-on-Demand, VCR, Progressive downloading, Streaming**

I. INTRODUCTION

Recently, wide range of video services have been introduced generating substantional network traffic. A potential killer video service is True Video-on-Demand (TVoD) endowed with VCR (Video Cassette Recording) options.[1] While, VoD services gain foothold one can observe improving performance close to the subscribers such as increasing bandwidth and storage capacity. These improving capacities are exploited by Peer-to-Peer supported VoD services, by existing solutions [1]–[3] and by models investigated in the literature e.g. [4]–[7].

There are two possibilities for delivering TVoD content from the server to the users: streaming and progressive downloading. In P2P-VoD system with streaming solutions the clients download the video content with a speed close to the bitrate of the video. This speed is the sum of the download speed from the server and from other clients. With progressive downloading the clients download the video with the highest speed possible between the server and the receiver. The speed achieved by progressive download is typically determined by one of these bottlenecks: the sending capacity of the server and the last mile link.

In this paper, we are investigating the circumstances that make streaming or progressive downloading more bandwidth efficient for transmitting the videos if Peer-to-peer like uploads support the system. Note that this problem has been investigates without Peer-to-Peer support in [8]. As far as we know, exect this paper and [8] there is no such a comparision available in the literature. If VCR actions are not allowed and there is a significant peak in the daily popularity, then progressive downloading is the better choice because the downloads of the clients that started the playback before the peak finish earlier because of the higher download speed before the peak. However, in this paper we compare the two delivery methods from the bandwidth consumption point of view if VCR actions are allowed. In Section II assumptions of the model are discussed and the analytical formulas are derived helping to compare the delivery methods. In Section III we present the practical consequences of the analysis.

The work has been partially supported by HSNLab, Budapest University of Technology and Economics, http://www.hsnlab.hu. The research of Székely was partially supported by NKTH-OTKA Foundation #77778. This work of Kőrösi is connected to the scientific program of the "Development of quality-oriented and cooperative R+D+I strategy and functional model at BME" project, this project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

A. Kőrösi is with the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, e-mail: korosi@tmit.bme.hu.

B. Székely is with the Institute of Mathematics Budapest University of Technology and Economics, e-mail: szbalazs@math.bme.hu

[1]In True VoD services the users experiment very short time between the video request and the starting of the video. Common VCR options are pause, rewind or fast forward.

II. SYSTEM MODEL AND THE REQUIRED SERVER CAPACITIES

A. Assumptions

We suppose that the VoD service is a high quality service, that is, failure in the play-back is not allowed.

Let us consider a video with length L from the video library of the operator. We assume that on a given day N subscribers watch this video according to the daily starting intensity function $I(t)$. We suppose that $I(t)$ is periodic with a period of 24 hours, differentiable, and for any t $\int_t^{t+24} I(t)\,\mathrm{d}t = 1$. Starting intensity means that in the time interval $[a, b]$ $N \cdot \int_a^b I(t)\,\mathrm{d}t$ subscribers start watching the video. We also suppose that the client downloads the video until either the whole video is downloaded or the client stops the connection.

We suppose that the video is CBR coded with bitrate ρ. Note that several papers [9]–[11] have shown that VBR coded videos can also be considered as CBR coded.

We assume that the upload capacities of the clients are utilized maximally. This means that if the aggregated upload capacity of the clients at time t is $U(t)$ then the server is offloaded by $U(t)$. We also assume that the capacity $U(t)$ is shared equally among the current downloaders, that is, the bandwidth share that a downloader obtains from the uploader clients is $u(t) = U(t)/N(t)$, where $N(t)$ denotes the number of active downloaders at time t.

If the video is delivered via streaming, then the aggregated download speed (the download coming from the server and the client) of a video on the downloader downlink is considered to be ρ. Using proper sliding window algorithm [4], $\rho - u(t)$ is enough server speed for the vast majority of the download time for achieving fault-free playback with high probability [4].

If the video is delivered using elastic progressive downloading (EPD), then *three* factors limit the download speed from the server, (i) the download limit assigned to the service on the first mile d, (ii) the upload share $u(t)$ that the downloaders obtain from other clients at time t. Consequently, the download speed from the server on the downlink can not be larger than $d - u(t)$. Further, (iii) the capacity of the central server S, so the download speed from the server of a downloader cannot be larger than $\frac{S}{N(t)}$ assuming equally distributed server capacity.

B. User model

In this subsection, we describe those user characteristics that are relevant when comparing the bandwidth consumption of the delivery methods.

We use two user characteristics that have already been introduced in [8]. In that paper the authors compare the delivery methods based on the user behavior with respect to the VCR actions if client uploads do not support the system. Two parameters that describe the user behavior during watching videos are the average viewing time (l) and the average ratio of the time spent by playing new parts (b). l is the average time elapsed between the start action and the stop action called. b is the average ratio of the time spent by playing new parts from the video and the entire viewing time. b is called "slowing down" parameter because b is small if the users apply backward jumps or pauses. The connection between l and b is that $b \le L/l$ or even $b \le \min\{L/l, 1\}$.

If client uploads support the system, then we have to introduce two additional parameters for describing the behavior of the clients as uploaders. The first parameter is the average upload bandwidth capacity which is denoted by u. The second parameter is the ratio of the non-active clients (currently do not watch any video) who do not turn off their STBs [2] over all non-active clients. This parameter is denoted by α.

We note that the bandwidth share $u(t)$ that a downloader obtains from the uploaders, defined in the previous subsection, can be written as follows

$$u(t) = u\frac{N(t) + \alpha(N - N(t))}{N(t)}. \tag{1}$$

C. The minimal required server capacities

In this subsection, we will determine the minimal required server capacities in the case of both delivery methods. First of all, we summarize the notation:

- ρ and L: the bitrate and the length of the video,
- N: number of subscribers,
- d and u: users' download and upload bandwidth capacities on the last mile assigned to the video service,
- α: the rate of those clients who share its upload capacity but are not active in watching video among the clients who are not active,
- $I(t)$: the daily *starting* intensity function,
- l: average viewing time,
- b: average ratio of the time spent by playing new parts,,
- $N(t)$: the number of active users at time t,
- $u(t)$: the average upload capacity of one client at time t,
- S: the capacity of the central server in the elastic case,
- $a \wedge b$: the minimum of the real numbers a and b.

Concerning the streaming case, let $A_s(t)$ denote the server load at time t. The minimal required server capacity in the streaming case is $S_s = \max_t A_s(t)$. For $A_s(t)$ we have the following simple expression:

$$A_s(t) = (\rho - u(t))b \cdot N \int_{t-l}^{t} I(\tau)\,\mathrm{d}\tau.$$

The integral along with multiplier N expresses the average number of downloads that started in interval $[t-l, t]$ (note that $I(t)$ is starting intensity). In streaming case the server uploads for those clients at time t who initiated the download at most l long time earlier. The first term $(\rho - u(t))b$ expresses the average speed of one video download from the server per user. The multiplier b decreases the required download speed since, if $b < 1$, then there are clients that pause the video or jump back and watch an already downloaded part.

The daily server load, that is, the number of bits downloaded from the server in a day can be written as

$$D_s = \int_t A_s(t)\,\mathrm{d}t.$$

We determine the minimal required server capacity S_{epd} in the elastic progressive download case. We have assumed that the server capacity S for arbitrary S is distributed equally among the clients hence the server speed per user is $S/N(t)$. The following system of equation has to be solved for obtaining $N(t)$:

$$\int_{t-y(t)}^{t} \left(\frac{S}{N(u)} + u(t)\right) \wedge d\,\mathrm{d}u = L\varrho, \tag{2}$$

$$N \cdot \int_{t-y(t)\wedge l}^{t} I(s)\,\mathrm{d}s = N(t). \tag{3}$$

In (2), $y(t)$ denotes the length of time in which the whole L long video can be downloaded by time t. From the point of the downloaded data the connection between the functions $N(t)$ and $y(t)$ is described in (2). Equation (3) expresses that the active clients at time t are those who started their download after $t - y(t)$ and are still watching the video (see the definition of l).

The system of equations (2) (3) can not be solved explicitly for $N(t)$. However, approximate solution with given accuracy can be reached, which converges to the exact solution.

Since the service does not allow play-back fault $S/N(t) \ge \rho - u(t)$ has to hold for each user. Therefore, the required minimal server capacity can be written as follows:

$$S_{epd} = \min\{S : \forall t,\, S/N(t) + u(t) \ge \rho\}.$$

Thus, the server load at time t with server having the minimally required capacity S_{epd} can be calculated as follows:

$$A_{epd}(t) = \max\left\{0, (d - u(t)) \wedge \frac{S_{epd}}{N(t)}\right\}.$$

[2]The videos shared in the system are stored in clients' equipments that can be a PC or a Set-Top-Box (STB).

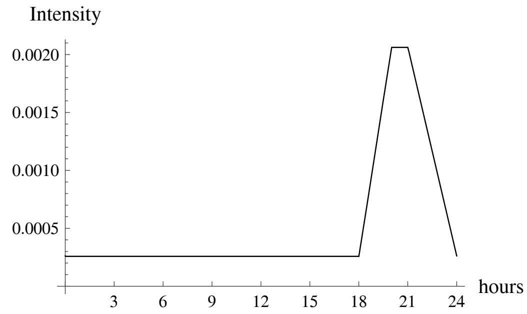

Figure 1. Starting intensity function. In this special case $I(t)$ is constant except in the interval $[18, 24]$ where it is linear and the peak is eight times higher than the constant part.

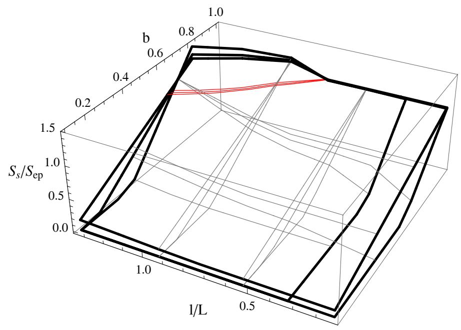

Figure 2. The capacity ratio $S_s/S_{ep}(u/, l/L, b)$ is depicted as a function of the first (l/L) and the second user parameter (b) for fixed upload capacities $u = 0$, $u = 0.5$, $u = 1$

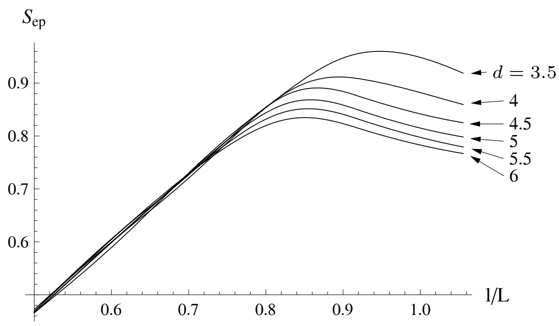

Figure 3. The server capacity S_{epd} is depicted as a function of the viewing time parameter (l/L) for fixed downlink capacity $d = 3.5$, $d = 4$, $d = 4.5$, $d = 5$, $d = 5.5$, $d = 6$

Further, similarly as in the streaming case the number of bits that are downloaded daily from the server D_{epd} can be written as follows

$$D_{epd} = \int_t A_{epd}(t)\,\mathrm{d}t.$$

III. PRACTICAL CONSEQUENCES

In this section, we make several observation of practical relevance, that is, we investigate which delivery method performs better for certain parameter set-up.

If it is not mentioned in the investigated scenarios below, then the following parameter set up will be used: $\rho = 3$ Mbps, $d = 2\rho$, $b = 0.9$; $L = 100$ minutes, $l = 0.8L$; $\alpha = 0.2$. We fix a starting intensity function $I(t)$ given in Figure 1 and we will use it to derive the following results. It is a typical intensity function in a local service provider network, similar to the one found in [12].

Observation 1: (On the effect of the viewing time parameter (l/L) and the slowing down parameter (b))
S_s/S_{epd} is a growing function of both the viewing time parameter (l/L) and the slowing down parameter (b), see Figure 2. Without any client upload support the same conclusion was drawn, see [8]. It can be seen in both cases (with and without client support) that streaming is the better choice if b and l/L is close to 1, that is, the majority of the users watch the entire video without any interaction. Practically, this usually holds for popular videos [13].

In Figure 2 it can be observed that the effect of the size of the client upload ($u = 0$, $u = 0.5$, $u = 1$ in the figure) is significant but the region in which streaming is better (the upper corner of the surfaces, above the thick curves) does not change significantly if the upload capacity increases.

Observation 2: (Finer study of S_{epd} as a function of l/L)
If the viewing time parameter (l/L) increases then the required server capacity (S_{epd}) increases until it takes its maximum, then S_{epd} decreases. See Figure 3 for the graphs of this function under several fixed downlink capacities. This phenomenon is reasonable since if the video has not been fully downloaded, then the minimal server capacity in the EPD case (S_{epd}) is a growing function of l/L until some point M_d where the maximum is taken. (We do not have explicit formula for M_d.) On the right hand side of M_d S_{epd} decrerases. This case, $l/L > M_d$ corresponds to that there are several clients having finished their downloads but they still in the system hence they contribute their upload capacities offloading the server. Before M_d the offloading property of the client is not stronger than their download bandwidth demand.

M_d is a decreasing function of d since the clients can download the video earlier with larger d and hence become "server offloader" sooner.

In this way, we have showed that if the operator choose EPD, then by increasing the downlink capacity the server capacity can be decreased.

Observation 3: (On the green-conscious subscribers) If the number of *ON* STBs (α) increases, the minimal required server capacity decreases for both delivery methods. Thus, the more user leave their STBs on the lower capacities are necessary. Indeed, the effect of the change α on $u(t)$ is similar to that of u since in equation (1) $u(t)$ is a linear function of u and α.

Observation 4: (*On the amount of downloaded data*) The consequence of Observation 1 is the same in qualitative, if we investigate the ratio D_s/D_{ep} of the amount of downloaded data from the server, see Figure 4. That is, D_s/D_{ep} is increasing in both b and l/L. The region where EPD is better is not sensitive to the size of the client upload capacity. Further, the region where EPD is better is smaller than in the case when the bandwidths were compared in Observation 1.

Observation 5: (*On almost constant starting intensity*) If the starting intensity function I of the video is constant then $N(t)$ and $y(t)$ are constant in Eq. (2) and (3). Computing

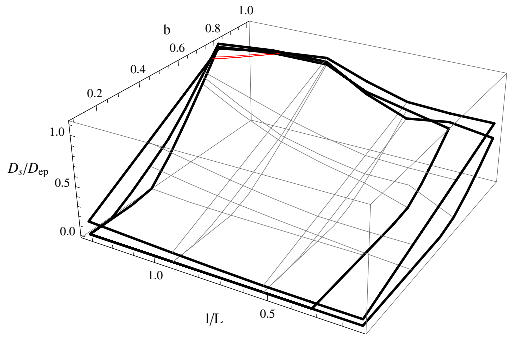

Figure 4. The capacity ratio $D_s/D_{ep}(u/, l/L, b)$ is depicted as a function of the first (l/L) and the second user parameter b for fixed upload $u = 0$, $u = 0.5$, $u = 1$

S_s/S_{epd} in this case, we have

$$S_s/S_{epd} = b \cdot \max\{1, l/L\} \leq \min\{L/l, 1\} \cdot \max\{1, l/L\} = 1.$$

This means that streaming is always the better choice independently of the other parameters including the upload capacity of the clients. This property has already been observed in [8] without any client upload.

IV. SUMMARY

The main conclusion of the paper is that comparing the systems with and without supporting client upload the results, which delivery method performs better, are the same in qualitative but different in quantitative. This is reason why the main property of a VoD system without supporting client upload remains valid: If the content is not educational and there is a significant peak in the daily popularity, then progressive downloading is a better choice than streaming. Otherwise, streaming performs better. Further, many in-depth observations have been made such as if the upload speeds of the clients increase, the provider should choose streaming instead of EPD, or if the operator choose EPD, then by increasing the downlink capacity the server capacity can be decreased.

REFERENCES

[1] "GridCast, http://www.gridcast.com."
[2] "PPLive, http://www.pplive.com."
[3] "UUSee, http://www.uusee.com."
[4] A. Kőrösi, B. Székely, and C. Lukovszki, "High quality P2P-Video-on-Demand with download bandwidth limitation," in *In Proc. of the 17th IEEE International Workshop on Quality of Service*, 2009.
[5] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Varvello, "Push-to-Peer Video-on-Demand System: Design and Evaluation," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 9, pp. 1706–1716, December 2007.
[6] C. Dana, D. Li, D. Harrison, and C.-N. Chuah, "BASSS: BitTorrent Assisted Streaming System for Video-on-Demand," in *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, 2005, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4014007
[7] C.-S. Lin and Y.-C. Cheng, "P2MCMD: A scalable approach to VoD service over peer-to-peer networks," *J. Parallel Distrib. Comput.*, vol. 67, no. 8, pp. 903–921, 2007.
[8] A. Kőrösi, B. Székely, and A. Császár, "TrueVoD: Streaming or Progressive Downloading?" *Communication Letters*, to appear, 2010.
[9] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Trans. Networking*, vol. 6, pp. 397–410, Aug. 1988.
[10] R. Chang, M. Chen, M. Ko, and J. Ho, "Optimizations of stored vbr video transmission on CBR channel," 1996. [Online]. Available: citeseer.ist.psu.edu/348654.html
[11] W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video," in *IS&T/SPIE Multimedia Networking and Computing*, Feb. 1997, pp. 316–327.
[12] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," *SIGOPS Oper. Syst. Rev.*, vol. 40, no. 4, pp. 333–344, 2006.
[13] B. Cheng, X. Liu, Z. Zhang, H. Jin, L. Stein, and X. Liao, "Evaluation and optimization of a peer-to-peer video-on-demand system," *Journal of Systems Architecture*, p. to appear, 2008.

Attila Kőrösi is an engineer at the Budapest University of Technology and Economics, Hungary working at the High Speed Network Laboratory. He received a M.S. degree in Mathematics from the Budapest University of Technology and Economics in 2007. His research interest is multimedia services and their stochastic models. He has published some papers on priority queuing systems.

Balázs Székely is an assistant professor at the Department of Stochastics at Budapest University of Technology and Economics, Hungary. He received his M.S. degree in Mathematics from Eötvös Loránd University, Hungary in 2000, and Ph.D. degree in Mathematics and Computer Science from Budapest University of Technology and Economics in 2005. His main research interests are queuing system analysis, random fractals, traffic and network modeling.