Volume is not Enough: SVC-aware Server Allocation for Peer-assisted Streaming

Julius Rückert*, Osama Abboud†, Martin Kluge*, and David Hausheer*
* Peer-to-Peer Systems Engineering, Technische Universität Darmstadt, Germany
Email: {rueckert|hausheer}@ps.tu-darmstadt.de, martin.kluge@stud.tu-darmstadt.de
† Multimedia Communications Lab, Technische Universität Darmstadt, Germany
Email: osama.abboud@kom.tu-darmstadt.de

***Abstract*—Peer-assisted delivery of video content has shown a great potential to reduce upload bandwidth requirements for content providers by exploiting idle client resources in the video dissemination process. As primary content sources, the servers run by content providers play a critical role in such systems, making their adequate provisioning a key part of the streaming mechanism. While dynamic resource provisioning has been studied before, little is known about resource allocation for streaming of scalable media content. Besides the pure amount of resources, here, the quality level of the delivered video content becomes relevant. The spreading of video blocks with the wrong quality can lead to situations where peers are forced to reduce their video qualities, despite them having enough download capacity. To address this problem, in this paper, a new SVC-based adaptation policy and a request-based extension to it are proposed, enabling content providers to manage their streaming services in a video quality-aware manner. Prototypical evaluations show that the mechanisms outperform existing quality-agnostic approaches in terms of delivered SVC video quality.**

I. INTRODUCTION

Recent studies show that delivery of video content is the most prominent contributor to the worldwide Internet traffic for several years now [2], [12]. Reasons can be seen in steadily increasing demands for high quality online video streaming services, the growing number of users and devices, as well as a trend towards more on-demand video consumption [3].

To handle the expected future demands for the delivery of video data over the Internet, a number of techniques and communication paradigms have been proposed. In particular, peer-to-peer (P2P) and peer-assisted techniques have shown to be promising alternatives to traditional centralized dissemination mechanisms for video-on-demand (VoD) services [5], [6]. By leveraging idle client resources, they enable a highly flexible and, for the content provider, cost-efficient scaling of the streaming systems. For a commercial scenario, usually, it is assume that video content is hosted by the content provider in a centralized manner. Its servers directly deliver the video content divided into blocks to a subset of clients that reassemble the video stream for playback and further distribute the blocks in a P2P manner. For this process, the adequate provisioning of the server upload capacities plays a key role for the system performance. Wrong provisioning decisions can either lead to serious content bottlenecks, where video data is not replicated fast enough, or a waste of precious server bandwidth as the server delivers video blocks that could have been served by other peers already. Therefore, server-side allocation mechanisms have shown to be important for the management of streaming services [7], [9].

While such mechanisms are already a challenge for non-adaptive video streaming, they are especially challenging in the context of adaptive streaming, e.g. using a scalable video codec such as H.264 SVC. Here, clients can switch between quality levels to adapt the video bit rate to their available network and device resources. Existing approaches both in the context of centralized [8] as well as P2P-based streaming systems [9] show to well support adaptations to fluctuating number of user. An additional scaling according the video quality requested by individual clients is not yet fully understood. In addition to the amount of provisioned upload capacities, here, the video quality of the video blocks delivered by the servers shows to play an important role. Provisioning, e.g., additional resources to deliver already well distributed quality layers easily leads to a waste of server resources. Peers that depend on poorly replicated quality layers might not be connected directly to the server, thus, having no influence on the content injection process. Consequently, they are forced to switch to a low-quality video layer or even experience playback stalling due to missing base layer data.

Goal and Contribution: The goal of this paper is to propose a new server allocation policy for SVC-based peer-assisted VoD streaming that takes into account the quality of the delivered content. Therefore, established video quality-agnostic allocation mechanisms as presented in [4], [9] are used as base for this work. To address their limited applicability to adaptive streaming scenarios, two new mechanisms are proposed: (1) A global speed allocation policy, respecting SVC characteristics in the resource management process, and (2) an extension managing requests for video blocks depending on their video quality, actively impacting the distribution process of individual quality levels. A more detailed description of these two mechanisms, a comprehensive discussion of further related work in the field, as well as an additional policy targeting the delivery of a minimum video quality to individual clients are presented in [10], an extended version of this work.

The remainder of the paper is structured as follows: Section II introduces the newly proposed adaptation policies. Subsequently, Section III presents the evaluation results and Section IV concludes the paper.

II. SVC-BASED GLOBAL SPEED POLICY

For the following description of the proposed server allocation mechanisms a state-of-the-art peer-assisted VoD streaming system with support for SVC-encoded video streams is assumed. While the authors presented such a system in earlier works [1], [11], the proposed mechanisms are designed to be applicable to other pull-/mesh-based streaming systems as well. Due to space limitations, the reader is referred to [10] for more details on the used streaming system.

The quality-agnostic server allocation mechanisms from [9], which are used as basis for this work, assume that a client's download capacity is higher than or equal to the video bit rate. For the streaming of scalable video content this assumption does not necessarily hold. Following the SVC-based quality adaption mechanisms, clients initially decide for a maximum video layer that might have a smaller bit rate than the maximum available bit rate of the video. During the streaming process, further progressive quality adaptations are issued to further switch between compatible quality layers. This breaks the above mentioned assumption and, as shown in Figure 1, leads to an undesired resource allocation behavior when applying the mechanisms from [9]. Relying on a common, fixed target bit rate of the video across all clients, these mechanisms result in a provisioning of much more resources than required.

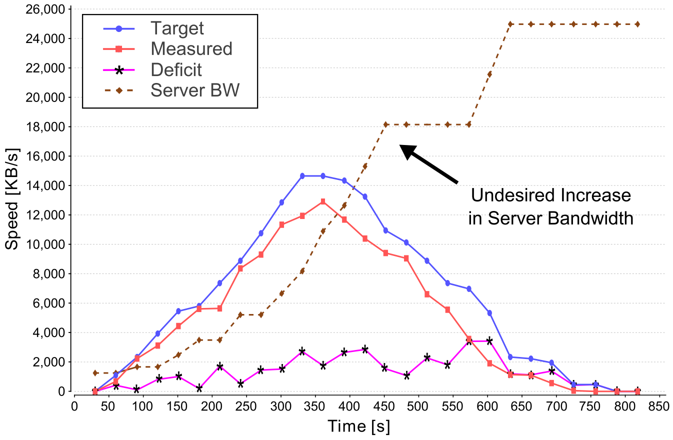


Fig. 1. Allocation progress of the quality-agnostic global speed policy.

Replacing the fixed target bit rate by the initially selected video bit rate (the IQA bit rate r_{iqa}) of the individual clients would be a straight-forward attempt to fix the undesired behavior. Unfortunately, it helps but does not solve the problem. The reason is that the actually streamed video bit rate, a result of the progressive quality adaptation (PQA), is per definition smaller than or equal to r_{iqa}. Therefore, a deficit of resources is indicated to the allocation mechanism as soon as a single peer reduces its video quality. As the newly provisioned resources are determined as sum of the previously provisioned resources and the deficit, this still results in a graph similar to Figure 1, with slightly smaller adaptation steps.

To overcome this problem and fundamentally change the allocation behavior, with (1) a new definition to calculate the deficit is proposed. It defines the overall required server bandwidth by considering the difference of the individual peers' desired maximum video bit rates and the amount of measured uploading bandwidth contributed by the clients themselves. This way, the allocation does not rely on the previously measured performance of former allocations but solely on the currently measured absolute demand:

$$Bandwidth_{server} = \sum_{p \in L} (f * r_{iqa}(p) - d_{peers}(p)), \qquad (1)$$

where f is the prefetching factor (with $f \geq 1$) as introduced by [9], $r_{iqa}(p)$ the IQA bit rate of a peer p, and $d_{peers}(p)$ the current incoming download throughput that peer p experiences from other peers he is connected to. The latter excludes connections to content delivery servers. L is the set of all active streaming peers, i.e. those peers that are actively retrieving a stream. This lightweight approach relies on a single report of the IQA layer of each client as well as a periodic report of their perceived download throughput. In contrast to the reporting every 5 seconds as proposed in [9], it was found that a less frequent reporting, i.e. every 10 seconds and for very resource limited peers every 20 seconds, is sufficient in targeted setup. For a large number of clients connected to a single server, the server still could become a bottleneck. Nevertheless, the assumed overhead is still substantially smaller than for reporting the overall content availability as proposed in a work by Mokhtarian and Hefeeda [7].

Active Request Management Extension

The previously presented allocation policy aims to decide on the amount of provisioned server resources in terms of upload bandwidth. This is the most intuitive dimension of the resource allocation process. In case of adaptive streaming, an additional dimension becomes relevant: the video quality. Due to the layered nature of the content, a server can decide on which requests for quality layers it serves first and what share of bandwidth it allocates to them. In a P2P scenario only a small subset of the peers is directly connected to the servers. They serve as the first hop for all delivered data pieces in the system and, thus, also define which video quality other peers can be served with. Due to the distributed nature of the system, the server does not have any influence on the dissemination process between peers. Nevertheless, it can very well influence the process in its first hop by biasing the decision on which data requests to answer or reject. It could, e.g., constantly serve a few connected peers with heavy data streams for high quality video layers or more evenly dedicate its available resources to serve many peers with lower quality videos layers.

The ACTIVE REQUEST MANAGEMENT (ARM) extension is proposed as a lightweight addition to any global speed policy. It does not influence the overall provisioned amount of server resources but helps to define the share of video quality layers that are injected to the system. The proposed approach can be adopted for a large range of scenarios, depending on the applied function to decide on which peer requests to serve at which time. In the context of this work, the aim was to improve the streaming process for different groups of peers with heterogeneous resources. As long as server resources are not scarce, the ARM extension is not required since all peer requests can be answered immediately. The situation changes as soon as several requests compete for the available resources. To decide whether and to what degree to apply ARM, an indicator was required for the load on the server that goes beyond the used upload throughput itself. Depending on the system implementation, several indicators could be used.

For the prototype used in this work, the server-side queue of waiting peer requests turned out to be the most reliable indicator for the current load of a server. Based on the length of this queue, or any other load indicator, a simple, yet powerful, prioritization of peer requests can be done. In [10] example functions to reject video block requests depending on the video quality as functions of the queue length are described. With increasing queue length, requests for the different quality layers are rejected different, increasing probabilities. While the base layer might not be rejected at, enhancement layer requests could be rejected with a steadily increasing probability, higher layers with a higher probability than lower layers. This way, available resources can be used to serve the more important requests for low quality layer to guarantee a stable system performance and avoid content bottlenecks for those important layers in cases of serious overloads. Adequate reject functions depend on the characteristics of the streamed content as well as its quality layers and, thus, have to be defined by the content provider itself. For the presented evaluation of ARM, the above mentioned example functions were used as they showed to be sufficient to show the impact of the mechanism.

III. EVALUATION

For the evaluation of the proposed mechanisms, the metrics as defined in [1] were used to judge the playback and video quality characteristics. To better reflect the absolute differences in the video quality layers, a slightly redefined version of the relative received quality metric was used (see [10] for details).

The evaluation was conducted using real-world testbed measurements in the German Lab testbed [13], a country-wide research testbed with over 170 dedicated machines. The client software was run on the testbed machines and controlled by a dedicated server. To reflect a heterogeneous scenario, exemplary groups of clients were defined. Their maximum bandwidths as well as the IQA layers chosen by the initial quality adaptation are listed in Table I. For the bandwidth limitations, client-side traffic shaping was used. For the servers, traffic shaping was used as well to control the upload bandwidth following the taken allocation decisions.

TABLE I
THE EXEMPLARY PEER GROUPS, THEIR CAPABILITIES AND IQA LAYERS.

Group	**Bandwidth Up-/Download**	**IQA Layer**		
		Bit Rate	**Index**	**Resolution**
P_{slow}	150/300 kbps	238 kbps	$(0,3,0)$	176×144
P_{medium}	1100/2210 kbps	1765 kbps	$(1,3,0)$	352×288
P_{fast}	2350/4700 kbps	3722 kbps	$(2,3,0)$	704×576

The default scenarios consist of 60 peers in total, 20 belonging to each of the groups. Peers join the system with an exponentially distributed start-up interval with a mean $\lambda = 5$ seconds. Other start-up intervals were evaluated but did not significantly influence the overall evaluation results. After joining, the peers automatically start streaming a pre-configured SVC test video with a length of 5 minutes. The video was encoded with three different spatial and four temporal layers, resulting in a total of 12 layers with the maximum bit rate and resolution as depicted in Table I for the fast peer group and a frame rate of 30 fps. After finishing the playback, peers leave after an additional, uniformly distributed interval with a maximum of 60 seconds. In this additional time they purely act as seeders for the video. To rule out random effects from single evaluations runs, all experiments were repeated ten times. Results are reported with $95\,\%$ confidence intervals.

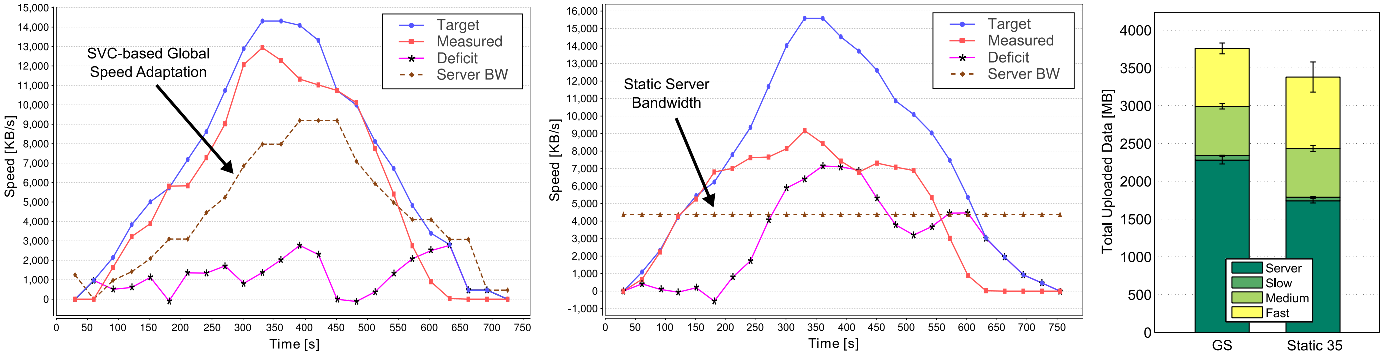
A. Evaluation of SVC-based Global Speed Policy

As presented in Section II, the global speed policy proposed in [9] showed an undesired behavior for the streaming of scalable content. The server bandwidth was steadily increased, even after most peers left the system. The reason was the dependency on former allocation steps, leading to an ever positive demand due to the quality adaptation characteristics. The newly proposed SVC-based global speed policy addresses this problem by relying on only the absolute deficit in resources at a given time. An example streaming progress using the new policy is depicted in Figure 2a. It shows an increase in server bandwidth as well as the desired decrease after peers start leaving the system again. Figures 2b and 2c, furthermore, show a comparison to a pre-dimensioned static server configuration. For a fair comparison, the server bandwidth was fixed to the average assigned bandwidth of the SVC-based allocation decisions. This results in a huge deficit in resources during the peak in online peers as missing resources could not be provisioned by the system. This is also visible in Figure 2c, showing the reduced overall contributed server resources. For the peer groups, no significant difference in the contribution can be observed, indicating that the peers themselves were not able to further contribute to the dissemination process.

B. Impact of the Active Request Management Extension

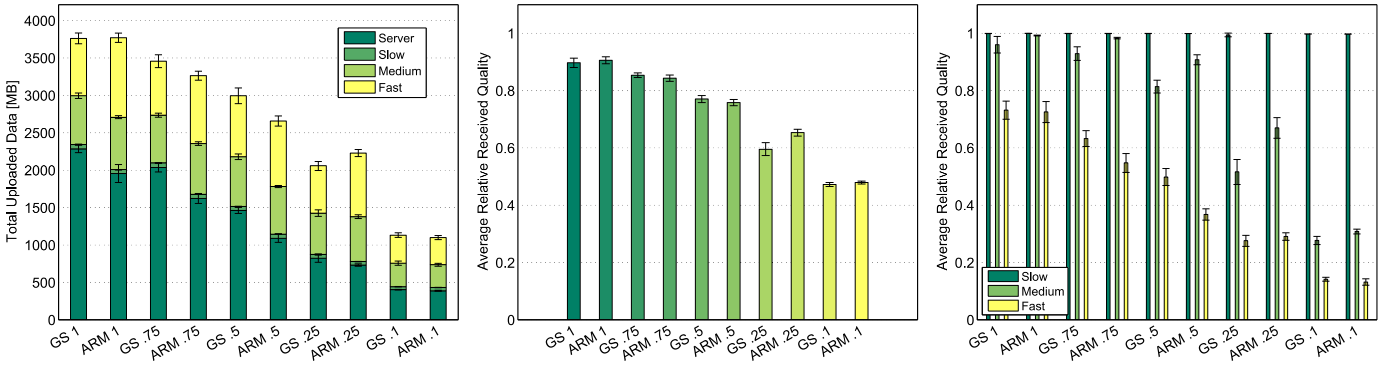
To evaluate the impact of the proposed ARM extension, the system was intentionally configured with a shortage of server capacities. This was done to show the behavior in case of overload scenarios where the ARM extension is intended to show its main benefits. The results are depicted in Figure 3, where the subfigures illustrate the different behavior using the plain SVC-based global speed mechanism and additionally using the ARM extension. Bars are intended to be compared in pairs of the same resource configurations.

Using the ARM extension, the server overall contributes less to the content dissemination process, whereas especially the fast peers contribute more resources. While for a decrease in server capacity the relative received quality also decreases, there is no significant difference between disabled and enabled



(a) SVC-based global speed allocation. (b) Static server resources with 35 Mbps. (c) Upload data volumes.

Fig. 2. Performance of the newly proposed SVC-based server allocation and a comparison to a static server configuration.



(a) Total uploaded data volume. (b) Relative received quality. (c) Relative received quality per peer group.

Fig. 3. Session and video quality for SVC-based global speed (GS) and ARM extension (bandwidth reduction depicted as floats, e.g. .25 = 25%).

ARM down to a level of 50% of the required bandwidth. For 25%, an increase in the average quality is observable. At a level of 10%, the resources are so scarce that, again, no difference is apparent. At this level, the system performance for most peers is very low as also detailed in the following.

The real strength of the ARM extension becomes apparent in Figure 3c where the relative received quality is shown per peer group. In all cases a slight reduction of the quality for the fast peer group is visible, while it is significantly improved for the medium peers. Here, the desired reallocation of resources between peer groups was achieved. For the slow peer group, no improvement could be achieved as this group already receives an almost 100% quality due to the base layer's low bit rate.

For the lowest capacity level, one third of the peers experience a very low and most probably unacceptable quality of 30% for the medium and only 10% for the fast peer group. At this stage, all peers switched to the base layer. The dissemination process still works but quality adaptations and ARM extension cannot show any effects anymore.

In summary, the ARM extension shows to be a simple but powerful tool to control the distribution of resources among the different peer groups without having to change the overall streaming protocol. The effects can be achieved by only altering the content delivery server behavior and adapting it to the system load level. As described before, the ARM extension can be freely adapted to specific needs of the content provider by changing the reject rate functions.

IV. CONCLUSION AND FUTURE WORK

In this paper, server allocation mechanisms were proposed to better manage peer-assisted SVC-based streaming services. By considering the specific characteristics of the adaptive streaming process, an improved global speed allocation policy could be defined to avoid a waste of content provider resources, by adequately adapting to fluctuating system loads. Furthermore, with the ARM extension a mechanism was proposed that allows controlling the resource provisioning in a video quality-aware manner. This makes the ARM extension a promising tool to better influence the system-wide video quality in heterogeneous scenarios. For future work it is planned to further study the video dissemination process on a block and video quality basis. Therefore, it is also planned to investigate the applicability of the proposed mechanisms to other streaming scenarios and systems.

ACKNOWLEDGMENT

This work has been funded in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCousin), the DFG as part of the CRC 1053 MAKI, and the Korean IT R&D program of MKE/KEIT (#10035587, Development of Social TV Service Enabler based on Next Generation IPTV Infrastructure).

REFERENCES

[1] O. Abboud, T. Zinner, K. Pussep, S. Al-Sabea, and R. Steinmetz, "On the Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems," in *ACM MMSys*, 2011.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2010–2015," Tech. Rep., 2011.

[3] Ericsson, "Ericsson ConsumerLab: TV and Video - An analysis of evolving consumer habits. Insight Summary Report." Tech. Rep., 2012.

[4] J. Flick, "Adaptive Resource Allocation Schemes for Peer-to-Peer Video-on-Demand (KOM-D-0408)," Master thesis, Technische Universitt Darmstadt, 2010.

[5] Y. Huang, T. Z. Fu, D.-M. Chiu *et al.*, "Challenges, Design and Analysis of a large-scale P2P-VoD System," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, 2008.

[6] Y. Liu, Y. Guo, and C. Liang, "A Survey on Peer-to-Peer Video Streaming Systems," *Peer-to-Peer Networking and Applications*, vol. 1, no. 1, pp. 18–28, 2008.

[7] K. Mokhtarian and M. Hefeeda, "Efficient Allocation of Seed Servers in Peer-to-Peer Streaming Systems with Scalable Videos," in *IEEE/ACM/IFIP International Workshop on Quality of Service (IWQoS)*, 2009.

[8] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2012.

[9] K. Pussep, O. Abboud, F. Gerlach, R. Steinmetz, and T. Strufe, "Adaptive server allocation for peer-assisted video-on-demand," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, 2010.

[10] J. Rückert, O. Abboud, M. Kluge, and D. Hausheer, "Quality-aware Server Allocation in Peer-assisted Video-on-Demand Streaming," Peer-to-Peer Systems Engineering, TU Darmstadt, Tech. Rep. PS-TR-2013-05, 2013, http://www.ps.tu-darmstadt.de/publications/PS-TR-2013-05.pdf.

[11] J. Rückert, O. Abboud, T. Zinner *et al.*, "Quality Adaptation in P2P Video Streaming Based on Objective QoE Metrics," in *IFIP NETWORKING*, 2012.

[12] Sandvine, "Spring 2013 Global Internet Phenomena Report," Tech. Rep., 2013.

[13] D. Schwerdel, D. Günther, R. Henjes *et al.*, "German-lab experimental facility," in *Future Internet (FIS)*, ser. Lecture Notes in Computer Science, A. Berre, A. Gmez-Prez, K. Tutschku, and D. Fensel, Eds. Springer, 2010, vol. 6369, pp. 1–10.