Modeling, Analysis, and Implementation of Universal Acceleration Platform Across Online Video Sharing Sites

Ke Xu, *Senior Member, IEEE*, Tong Li, Haiyang Wang, Haitao Li, Wei Zhu, Jiangchuan Liu, *Senior Member, IEEE*, and Song Lin

Abstract—User-generated video sharing service has attracted a vast number of users over the Internet. The most successful sites, such as YouTube and Youku, now enjoy millions of videos being watched every day. Yet, given limited network and server resources, the user experience of existing video sharing sites (VSSes) is still far from being satisfactory. To mitigate such a problem, peer-to-peer (P2P) based video accelerators have been widely suggested to enhance the video delivery on VSSes. In this paper, we find that the interference of multiple accelerators will lead to a severe bottleneck across the VSSes. Our model analysis shows that a universal video accelerator can naturally achieve better performance with lower deployment cost. Based on this observation, we further present the detailed design of *Peer-to-Peer Video Accelerator* (PPVA), a real-world system for universal and transparent P2P accelerating. Such a system has already attracted over 180 million users, with 48 million video transactions every day. We carefully examine the PPVA performance from extensive measurements. Our trace analysis indicates that it can significantly reduce server bandwidth cost and accelerate the video download speed by 80%.

Index Terms—Video Sharing, P2P, Acceleration, Replication.

1 Introduction

The recent years have witnessed the explosion of video sharing as an emerging killer application. These user generated video sharing sites (VSSes), unlike traditional TV/movie services, are greatly enriched by constantly updated contents from users worldwide. It is known that over 300 hours of videos will be uploaded to YouTube every minute [1]. The most successful VSS in China, Youku [2], also enjoys more than 100 million videos being watched every day. The success of their local counterparts, such as Ku6 [3] and Sina Video [4], further indicates an elevating market interest in video sharing. However, given limited network and server resources, user experience in existing VSSes is still far from being satisfactory. Recent surveys reveal that the average service delay of YouTube is more than $15s$ [5] [6], which is much longer than some earlier measurements (nearly $6.5s$) [7] [8]. This increasing latency can greatly affect the development of the VSSes [9]. In particular, Sitaraman *et al.* [10] indicated that per second delay results in a 5.8% increase in the number of viewers abandoning slow-loading videos.

It is known that the latency issue is mainly due to the explosion of VSS user population and the limited server capacity. To alleviate this service bottleneck, peer-to-peer (P2P) assisted video accelerators have been wildly adopted by many VSSes. In particular, Youku [11], Tudou [12] and iQIYI [13] provide different types of video accelerators[1], aiming to help their users to obtain better watching experience. As a side effect, a user browsing multiple VSSes has to install different accelerators for each site. For users, the installation of different accelerators is both time and resource consuming. While for service providers, the redundant development of customized accelerators is not cost-effective, either.

In this paper, we for the first time investigate the potential of providing a universal video acceleration platform. This platform is designed to serve multiple P2SP networks, fully exploring the aggregated video and client resources across VSSes, especially for identical videos replicated in diverse sites. Our model analysis indicates that a universal video accelerator can obtain better download performance with lower deployment costs. In this system, users will have enough incentive to use the universal accelerator because it can achieve 80% improvement in terms of the video download speed. Based on this analysis foundation, we further present the implementation of our real-world system: *Peer-to-Peer Video Accelerator* (PPVA) [14]. This commercial system has already been used by over 180 million users over the Internet. To better explain its design principals, we closely examine the performance of PPVA and highlight the unique challenges during the system implementation. This universal video acceleration platform is then evaluated

- *Ke Xu, Tong Li and Song Lin were with the Dept. of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: (xuke@, litong12@mails.) tsinghua.edu.cn, linsong1984@gmail.com*
- *Haiyang Wang was with the Dept. of Computer Science and Technology, University of Minnesota at Duluth, the United States. E-mail: haiyang@d.umn.edu*
- *Haitao Li was with UCAR Inc, Beijing, China. E-mail: lhtao0607@gmail.com*
- *Wei Zhu was with PPLive Inc., Shanghai, China. E-mail: 16867960@qq.com*
- *Jiangchuan Liu was with the School of Computer Science and Technology, Simon Fraser University, British Columbia, Canada. E-mail: jcliu@cs.sfu.ca*

1. These dedicated softwares are also called stand-alone video accelerators.

by detailed trace analysis, providing valuable guidelines for future enhancements. To the best of our knowledge, this is the first research that presents the detailed design of a real-world VSS acceleration platform.

The rest of this paper is organized as follows: Section 2 describes the related work. Before designing the real-world system PPVA, we firstly present a model-based analysis to show why a universal accelerator can achieve better performance in Section 3. Section 4 further models the competition between different video acceleration approaches. Based on these analyses, we present the framework as well as the design of our real-world system, PPVA, in Section 5. Section 6 explores the PPVA performance through real-world measurements and Section 7 further discusses the characteristics of video contents and the random seeking support of PPVA. Finally, Section 8 concludes the paper.

2 Related Work

P2P delivery has been used to accelerate diverse content distribution systems, e.g., file sharing [15], software updates [16], live streaming [17] [18] [19], and on-demand streaming (P2P-VoD) [20] [21] [22] [23]. With each peer contributing its bandwidth to serve others, a P2P overlay scales extremely well with larger user bases.

Particularly, Paiet *et al.* [17] proposed Chainsaw and Zhang *et al.* [18] proposed CoolSreaming where a peer maintains a partial view of others as its neighbors, and schedules the video segment transmission by sending outgoing requests to its neighbors. Since 2006, there have been several start-up companies in China developing VoD systems, such as PPStream [24], which has attracted more than 10 million users and more than 300 programs of living channels. In terms of system deployment, Cheng *et al.* deployed GridCast [20] on CERNET (China Education Network). They discussed viewing session characteristics, popularity distribution of videos, as well as implications on designing efficient distribution architectures. Another large-scale P2P-VoD system has been presented in [21], which extends PPLive, one of the most successful P2P live streaming systems. Liu *et al.* [22] presented the first production deployment of random network coding as a core technology in the UUSee P2P-VoD system operated by UUSee Inc., one of the leading peer-assisted media content providers in China. The use of network coding [25] has emerged as a potential remedy to overhead challenges in P2P video streaming systems [26] [27]. A follow-up study by Liu *et al.* [23] presented Novasky, a real-world Video-on-Demand (VoD) system capable of delivering cinematic quality video streams to end users.

However, due to bandwidth instability in P2P system, the current video transmission tends to use a peer-assisted pattern, which is called peer to server and to peer (P2SP) [28] system. P2SP system enables users to simultaneously download data from both servers and peers. For example, as a leading CDN services provider, Akamai [29] has proposed its peer-assisted content delivery to provide faster and more stable mass data transmission services. Besides, P2SP networks also provide users with accelerated services for video delivery. A typical case in China is that each P2SP network develops a corresponding accelerator [11] [12] [13] for their own websites. The PPVA implementation is closely related to P2SP but quite different. Whereas the existing VSS accelerators in P2SP systems are designed for dedicated sites, a user browsing multiple VSSes has to install different accelerators for each site. Neither coordination nor resource sharing exists across different video sites.

Different from existing studies that are generally confined to a particular or small collection of sites, we present a universal acceleration platform fully exploring the aggregated video and client resources across diverse VSSes. Our investigation indicates the efficiency of this universal video acceleration platform and identifies the implementation challenges therein.

The essence of PPVA is that by utilizing peering flexibility, P2P applications can cooperate to improve network efficiency. Xie *et al.* [30] considered that the same data may be available from multiple sources and P2P may have tremendous flexibility in rewiring their traffic patterns to improve network efficiency. They further suggested P4P, a new cross-torrent collaboration architecture that cooperate P2P applications with ISPs. However, their design only considers a single ISP, and asks for P2P applications to consult for ISP-biased network information, which is non-transparent. Our PPVA, however, will provide transparent services that do not need to change client-server protocols across VSSes, which can be smoothly and incrementally deployed on existing video clients.

3 Performance Analysis of Universal Video Acceleration Platform

Before designing and deploying the real-world system PP-VA, we start from theoretical analysis to show whether a universal accelerator can achieve better performance. In this section, we will clarify the performance gain of a universal video accelerator through model-based analysis. In particular, considering z VSSes with dedicated video accelerators, we will compare PPVA performance to existing stand-alone acceleration platforms. Table 1 summarizes the notations in our modeling.

TABLE 1: Summary of notations in the performance model

Notation	Description
z	The total number of VSSes
n	The total number of nodes in the system
m	The total number of unique chunks in the system
k	The storage capacity on the node (number of chunks)
μ_r	The peer's expected value of download bandwidth
μ_u	The peer's expected value of upload bandwidth
p_r	The probability of peer's chunk requirement
$\mathrm{Peer}_{\mathrm{sep}}$	The total available download bandwidth of peers across z stand-alone video accelerators
$\mathrm{Peer}_{\mathrm{union}}$	The total available download bandwidth of peers with a universal video accelerator
λ_i	The probability that a user visits the i-th VSS
P_r	The probability that a user requests chunk r
B	The set of all nodes in the system

3.1 Modeling of Video Acceleration Platforms

For a given peer in P2P-based video systems, we use $\mathrm{Peer}_{\mathrm{sep}}$ to denote the total amount of download bandwidth that can be obtained by using z stand-alone video accelerators, that is, $\mathrm{Peer}_{\mathrm{sep}}$ is obtained without the interaction of multiple VSSes. We use $\mathrm{Peer}_{\mathrm{union}}$ to refer to the download bandwidth using a universal video accelerator. It is easy to see that the ratio $\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}}$ can be used to quantify the relative performance of stand-alone and universal video acceleration platforms.

To clarify such a ratio, we first estimate the maximum available bandwidth provided by different acceleration platforms as follow. (The derivation of this equation can be found in Appendix I.)

$$\mathrm{Peer}_{\mathrm{union}} \approx \min(\mu_r n p_r, n\mu_u(1 - e^{-\frac{knp_r}{m}})) \tag{1}$$

where n is the total number of peers in the system, m is the total number of unique chunks in the system, k is the node storage size (number of chunks), μ_r is the peer's expected value of download bandwidth, μ_u is the peer's expected value of upload bandwidth, and p_r is the probability of requesting a chunk.

Without the loss of generality, we assume that there are z VSSes and users have a probability of λ_i to visit the i-th VSS. Thus, $\sum_{i=1}^{z} \lambda_i = 1$. In VSS i, its peer population, total number of chunks and cache size can be referred as

$$n_i = \lambda_i n, m_i = \lambda_i m, k_i = \lambda_i k \tag{2}$$

The total amount of upload bandwidths provided by the peers in VSS i can be obtained as follow:

$$\min(\lambda_i \mu_r n p_r, \lambda_i n \mu_u(1 - e^{-\frac{knp_r\lambda_i}{m}})) \tag{3}$$

Based on this definition, the total amount of upload bandwidths provided by peers in all the VSSes can be approximated as

$$\mathrm{Peer}_{\mathrm{sep}} \approx \sum_{i=1}^{z} \lambda_i \min(\mu_r n p_r, n\mu_u(1 - e^{-\frac{knp_r\lambda_i}{m}})) \tag{4}$$

Therefore, the ratio between $\mathrm{Peer}_{\mathrm{sep}}$ and $\mathrm{Peer}_{\mathrm{union}}$ can be defined as

$$\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \approx \frac{\sum_{i=1}^{z} \lambda_i \min(\mu_r p_r, \mu_u(1 - e^{-\frac{knp_r\lambda_i}{m}}))}{\min(\mu_r p_r, \mu_u(1 - e^{-\frac{knp_r}{m}}))} \tag{5}$$

It is easy to see that when $\mu_r p_r \leq \mu_u(1 - e^{-\frac{knp_r}{m}})$, i.e.,

$$\frac{knp_r}{m} \geq -\ln(1 - \frac{\mu_r p_r}{\mu_u}) \tag{6}$$

We can approximate $\min(\mu_r n p_r, n\mu_u(1 - e^{-\frac{knp_r\lambda_i}{m}})) \approx \mu_r n p_r$. Therefore, we have $\mathrm{Peer}_{\mathrm{sep}} \approx \mu_r n p_r$. The ratio can thus be obtained as

$$\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \approx 1, \text{ if } \mu_r p_r \leq \mu_u(1 - e^{-\frac{knp_r}{m}}) \tag{7}$$

Otherwise, when $\mu_r p_r > \mu_u(1 - e^{-\frac{knp_r}{m}})$, the ratio can be obtained as

$$\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \approx \frac{\sum_{i=1}^{z} \lambda_i \mu_u(1 - e^{-\frac{knp_r\lambda_i}{m}})}{\mu_u(1 - e^{-\frac{knp_r}{m}})} \tag{8}$$

where we define

$$g(\lambda) = \frac{1 - e^{-\lambda x}}{1 - e^{-x}}, 0 < \lambda < 1 \tag{9}$$

Using Calculus and the Arithmetic Mean, i.e., Geometric Mean Inequality, we have

$$\lambda < g(\lambda) < \min(1, \min(x + 1, e^{x/2})\lambda) \tag{10}$$

We thus denote

$$\psi = \sum_{i=1}^{z} \lambda_i^2 \tag{11}$$

Based on Formula 8, we can obtain the ratio as

$$\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \in (\psi, \min(1, \psi \min(1 + \frac{knp_r}{m}, e^{\frac{knp_r}{2m}}))] \tag{12}$$

Based on the above analysis, we summarize the conclusions as Theorem 1.

Theorem 1. *Under the same circumstances, we have the following results of bandwidth ratios provided by other peers in multiple P2P networks to those in the video acceleration platform:*

$$\begin{cases} \frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \in (\psi, \min(1, \psi(1 + \frac{knp_r}{m}), \psi e^{\frac{knp_r}{2m}})], \\ \quad \text{if } \mu_r p_r > \mu_u(1 - e^{-\frac{knp_r}{m}}) \\ \frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \leq 1, \frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \approx 1, \\ \quad \text{if } \mu_r p_r \leq \mu_u(1 - e^{-\frac{knp_r}{m}}) \end{cases} \tag{13}$$

where ψ is given by Formula 11.

We denote $\chi = \frac{knp_r}{m}$. In the case that $\mu_r p_r > \mu_u(1 - e^{-\frac{knp_r}{m}})$, when χ is small, $\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}}$ will be close to ψ. For example, when $\frac{knp_r}{m} < 0.1$, $\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}} \in (\psi, 1.052\psi)$.

In the case that $\mu_r p_r \leq \mu_u(1 - e^{-\frac{knp_r}{m}})$, we can find that when χ is small (for example, smaller cache sizes), the universal video acceleration platform will significantly outperform that of stand-alone video acceleration platforms; when χ is large, the performance of the two systems will be quite similar. Unfortunately, large local resources for caching can hardly be obtained in the real-world implementation. The limited number of peers will further enlarge such a performance gap especially in the unpopular channels. Note that we will further clarify this performance gap in the next subsection.

3.2 Numerical Evaluation

In this part, we will calculate the ratio $\frac{\mathrm{Peer}_{\mathrm{sep}}}{\mathrm{Peer}_{\mathrm{union}}}$ with numerical values. It is known that there are over 30 major VSSes in Asia,[2] most of which have provided their own video accelerators. Based on the existing measurements, the top three VSSes are Youku, Tudou[3] and iQIYI [31], which occupied 39.1%, 20.3% and 15.3% of the market shares, respectively.[4] We therefore apply the following real-world parameters for our evaluation.

$$\begin{aligned} &n = 3 \times 10^8, p_r = 0.1, k = 250, m = 10^{10}, \\ &z = 1000, \mu_u = 300, \mu_r = 500. \end{aligned} \tag{14}$$

2. http://www.reelseo.com/list-video-sharing-websites/.

3. Two of the most popular VSSes in China, contracted a merger in March 2012. However, our analysis regards them as two different VSSes according to their reserved platform independence. http://www.globaltimes.cn/content/699963.shtml.

4. http://news.iresearch.cn/Zt/140136.shtml.

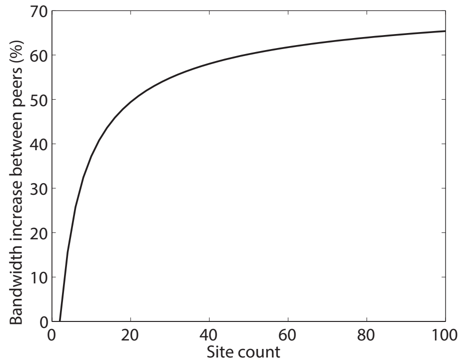


Fig. 1: Performance gain for different site count z

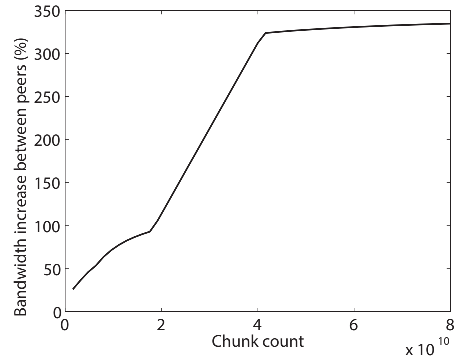


Fig. 2: Performance gain for different chunk count m

where λ_i follows a Zipf [32] distribution. Based on the popularity of the top three VSSes, the maximum likelihood estimation (MLE) of α is 1.5887. The universal video acceleration platform can therefore increase the bandwidths provided by other peers by 73.2% compared with those in the stand-alone video platforms.

Based on this model, we further analyze the performance of the universal video acceleration platform. As we can see from Figure 1, the performance gain will increase with more VSSes. This is because the universal platform can increase the sharing opportunities among different VSSes, which potentially increases the utilization of peer's upload capacity. Note that the total number of peers is fixed in this experiment. The performance gain will be bounded when peer's upload capacities are fully utilized. Figure 2 shows the performance gain with different chunk population. We can find that the universal acceleration platform can achieve better performance with more chunks. This is because chunk population can increase the sharing efficiency among peers.

In conclusion, this section statically conducts the comparison of P2P transmission bandwidth between the universal video acceleration approach and the stand-alone video acceleration approach, we finally obtain Theorem 1. Theorem 1 indicates that transmission bandwidth of stand-alone video acceleration approach will not be more than that of the universal video acceleration approach in P2P networks, and smaller χ contributes to the performance gain of the universal acceleration platform. As to accurate numerical calculation of the transmission bandwidth of the two approaches, we can directly use Formula 5. Model analysis and numerical evaluation demonstrate that the universal video acceleration approach outperforms the traditional stand-alone video acceleration approach in performance. However, this universal video acceleration approach also brings about extra overhead, which may hinder the universal video acceleration platform deployment. We will discuss the competition between the universal video acceleration platform and traditional ones in Section 4.

4 Competition Analysis of Universal Video Acceleration Platform

In this section, we will discuss the competition of three representative approaches in this section: the Universal Video Acceleration (UVA) approach, the Stand-Alone Video Acceleration (SAVA) approach and the C/S Video Acceleration (CSVA) approach[5]. Table 2 summarizes the notations in the competition modeling.

TABLE 2: Summary of notations in the competition model

Notation	Description
U	The set of all users
x	The set of users using different acceleration approaches
F	The total capacity of servers
D	The user's expected value of download bandwidth
C	The cost of users
Util_1	The utility of a universal video acceleration platform
Util_2	The utility of a stand-alone video acceleration platform
Util_3	The utility of a C/S acceleration platform
θ_{user}	The user-related variable to characterize different users in valuing the objective benefit [33]
η	The chunk sharing rate among users
H	The set of users who have positive utility while using a given acceleration approach

4.1 Competition Considerations

Internet delivery has gone through three stages: the C/S pattern, cache pattern represented by CDN, and P2P (P2SP). Section 3 shows that a universal video acceleration platform can largely improve user's download experience. However, its deployment also introduces more overheads on users, such as larger local cache and higher upload bandwidth. These overheads may prevent users from using the universal acceleration platform.

Before deploying PPVA, it is strategic to evaluate whether UVA is likely to be widely deployed, competing with the current SAVA and CSVA. If it is, we eventually reveal that video delivery can evolve to a new (the fourth) stage: UVA, i.e., fully exploring the aggregated video and client resources across multiple VSSes. More importantly, based on the competition analysis, we can find out the preconditions for our design to outperform the others. This conditions show potential in guiding the deployment of our real-world system, PPVA.

4.2 Modeling of User Utility

To further clarify the trade-off and understand user's incentive, we apply a classic competition model [33] to analyze

5. The client-server acceleration refers to the conventional approach where service providers deploy more servers to enhance download performance.

the utility of different acceleration approaches. The general user utility function is defined as follow:

$$\mathrm{Util}_{\mathrm{net}}(t) = \theta_{\mathrm{user}} D_{\mathrm{net}}(t) - C_{\mathrm{net}} \tag{15}$$

where $D_{\mathrm{net}}(t)$ is expected as the quality of service the platform can provide, here we simply use the user's expected value of download bandwidth instead. θ_{user} is a user-related variable uniformly distributed in $[0,1]$, characterizing different users in valuing the objective benefit. Further definition of this variable can be seen in [33]. C_{net} is the cost of the user, such as the contribution of storage and upload bandwidth. Note that CSVA has no additional cost because users only need to download the videos from servers.

Since PPVA is mainly used in China (account for over 97% downloads [14]) and recently many VSSes have developed their corresponding accelerators [11] [12] [13], these dedicated accelerators have already reached certain proportion. But they are far from being popular among users. According to the real scene, without loss of generality, we use $x_1(t)$ to refer the set of users in the universal video acceleration platform at time t. $x_2^i(t)$ and $x_3^i(t)$ refer to the set of the users who choose the stand-alone and C/S acceleration platform of the i-th VSS at time t, respectively.

Based on Formula 1 and 15, the utility of using the UVA can be obtained as follows:

$$\mathrm{Util}_1 \approx \theta_{\mathrm{user}} \min(\mu_r, \frac{\mu_u}{p_r}(1-e^{-\beta_2 x_1(t)}) + \frac{F}{Up_r}) - C_1 \tag{16}$$

where U is the set of users and $\beta_2 = \frac{kUp_r}{m} > 0$. F is the total capacity of servers.

For the remaining users choosing non-UVA (not using the universal video acceleration platform), their utility is as follows:

$$\widehat{Util_1} = \sum_{i=1}^{z} \max(\mathrm{Util}_2^i, \mathrm{Util}_3^i) \tag{17}$$

where Util_2^i is the utility of using stand-alone acceleration of the i-th VSS. Util_3^i is the utility of using C/S acceleration of the i-th VSS.

Next, we introduce the calculation of Util_2^i. In particular, the users have a probability of λ_i to view videos in the i-th VSS, install its video accelerator, gain benefits and contribute to others. Meanwhile, the users also need to provide local cache with the size of $k\lambda_i$ chunks. The local cache is considered as consistent costs (i.e., $C_2^i = \lambda_i C_1$). Based on Formula 2, the popularity parameter of the VSS is λ_i, the number of users using the VSS with the stand-alone video accelerator is $U\lambda_i x_2^i$. We therefore have

$$n_i = U\lambda_i x_2^i(t), m_i = m\lambda_i, k_i = k\lambda_i, C_2^i = \lambda_i C_1 \tag{18}$$

Based on Formula 1 and 15, we can approximate Util_2^i as

$$\mathrm{Util}_2^i \approx \lambda_i \theta_{\mathrm{user}} \min(\mu_r, \frac{\mu_u}{p_r}(1-e^{-\beta_2\lambda_i x_2^i(t)}) + \frac{F}{Up_r}) - \lambda_i C_1 \tag{19}$$

where p_r is the probability of requesting a chunk.

On the other hand, Util_3^i is the utility of using C/S acceleration of the i-th VSS. In this scenario, the users using CSVA do not need to serve other peers and therefore have no additional cost. We have

$$\mathrm{Util}_3^i \approx \lambda_i \theta_{\mathrm{user}} \min(\mu_r, \frac{F}{Up_r}). \tag{20}$$

4.3 Modeling of Competition

Based on the utility functions in Formulas 16 and 17, we can now analyze the competition among different acceleration approaches.

4.3.1 Competition Between UVA and Non-UVA

In particular, the users will choose a better approach with larger utility (between Util_1 and $\widehat{Util_1}$). To better understand the competition, we will focus on the situation with plenty of user requirements. Util_1 can therefore be approximated as

$$\mathrm{Util}_1 \approx \frac{\theta_{\mathrm{user}} F}{Up_r} + \frac{\mu_u}{p_r}(\theta_{\mathrm{user}}(1-e^{-\beta_2 x_1(t)}) - \beta_1) \tag{21}$$

where $\beta_1 = \frac{C_1 p_r}{\mu_u} > 0$ refers to the normalized time cost of uploading local chunks to other peers. Therefore, $\widehat{Util_1}$ can be obtained as

$$\begin{aligned}\widehat{Util_1} &\approx \frac{\theta_{\mathrm{user}} F}{Up_r} \\ &+ \frac{\mu_u}{p_r}\sum_{i=1}^{z}\lambda_i \max(\theta_{\mathrm{user}}(1-e^{-\beta_2\lambda_i x_2^i(t)}) - \beta_1, 0)\end{aligned} \tag{22}$$

Based on Formulas 21 and 22, the difference of utility can be approximated as

$$\begin{aligned}\mathrm{Util}_1 - \widehat{Util_1} &\approx \frac{\mu_u}{p_r}(\theta_{\mathrm{user}}(1-e^{-\beta_2 x_1(t)}) - \beta_1) \\ &- \frac{\mu_u}{p_r}\sum_{i=1}^{z}\lambda_i \max(\theta_{\mathrm{user}}(1-e^{-\beta_2\lambda_i x_2^i(t)}) - \beta_1, 0)\end{aligned} \tag{23}$$

We denote the right side of Formula 23 as $Y(\theta_{\mathrm{user}})$. It is easy to see that when $Y(\theta_{\mathrm{user}}) \geq 0$, their θ_{user} will satisfy $\theta_{\mathrm{user}} \in [1-H_1(t), 1]$. We therefore have the chunk sharing rate among users as follow:

$$\eta_1(t) = \sum_{i=1}^{z}\lambda_i(1-e^{-\beta_2\lambda_i x_2^i(t)}) \tag{24}$$

We further discuss two cases as follows:

Case 1: $1-e^{-\beta_2 x_1(t)} \leq \eta_1(t)$. Since we have $\sum_{i=1}^{z}\lambda_i \max(\theta_{\mathrm{user}}(1-e^{-\beta_2\lambda_i x_2^i(t)}) - \beta_1, 0) \geq \theta_{\mathrm{user}}\eta_1 - \beta_1$, in this case, $Y(\theta_{\mathrm{user}})$ will be less than zero.

Case 2: $1-e^{-\beta_2 x_1(t)} > \eta_1(t)$. When $\theta_{\mathrm{user}} = 1$, we have $Y(\theta_{\mathrm{user}}) > 0$. We can see that if $\theta_{\mathrm{user}} = 0$, $Y(\theta_{\mathrm{user}}) \leq 0$. Therefore, $Y(\theta_{\mathrm{user}}) = 0$ has at least one solution θ_0 in $[0,1]$. The value of $Y(\theta_{\mathrm{user}}) - Y(\theta_0)$ has both non-negative coefficient for θ_{user} and non-negative constant term. Thus for $\theta_{\mathrm{user}} \in [\theta_0, 1]$, we have $Y(\theta_{\mathrm{user}}) \geq 0$. In other words, if some users have $Y(\theta_{\mathrm{user}}) \geq 0$, then their θ_{user} satisfies $\theta_{\mathrm{user}} \in [1-H_1(t), 1]$.

For both cases, the users who choose the universal video accelerator have $\theta_{\mathrm{user}} \in [1-H_1(t), 1]$. As the migration always happens near the boundary of different categories, at any moment, users who choose UVA have $\theta_{\mathrm{user}} \in [1-x(t), 1]$; while users with $\theta_{\mathrm{user}} \in [0, 1-x(t))$ will choose between SAVA and CSVA for each VSS.

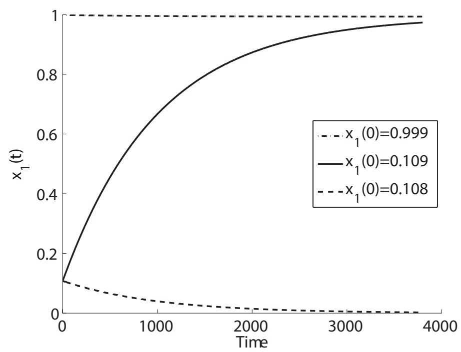


Fig. 3: Evolution of the proportion of the universal video acceleration platform

4.3.2 Equilibrium Between SAVA and CSVA

For the i-th VSS, we will now further consider user's choice between SAVA and CSVA. The difference of their utility functions is

$$\mathrm{Util}_2^i - \mathrm{Util}_3^i \approx \frac{\lambda_i \mu_u}{p_r}(\theta_{\mathrm{user}}(1 - e^{-\beta_2 \lambda_i x_2^i(t)}) - \beta_1) \tag{25}$$

Note that both SAVA and CSVA have been deployed for a long period of time, and the choices between them can always move to the equilibrium point [33]. We therefore have

$$\kappa_{2,i} = \beta_2 \lambda_i = \frac{kUp_r\lambda_i}{m} > 0 \tag{26}$$

where $\kappa_{3,i}$ is the only real number satisfying the following factors:

$$\begin{aligned}&(\kappa_{2,i}(1 - x_1(t) - \kappa_{3,i}(t)) + 1)e^{-\kappa_{2,i}\kappa_{3,i}(t)} = 1,\\ &\kappa_{3,i}(t) \in (0, 1 - x_1(t))\end{aligned} \tag{27}$$

we define

$$\kappa_{6,i}(t) = (1 - x_1(t) - \kappa_{3,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{3,i}(t)}) \tag{28}$$

when $\beta_1 \le \kappa_{6,i}(t)$, let $\kappa_{4,i}(t)$ and $\kappa_{5,i}(t)$ be the unique real numbers satisfying the following equations:

$$\begin{aligned}&(1 - x_1(t) - \kappa_{4,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{4,i}(t)}) - \beta_1 = 0,\\ &\kappa_{4,i}(t) \in (0, \kappa_{3,i}]\end{aligned} \tag{29}$$

$$\begin{aligned}&(1 - x_1(t) - \kappa_{5,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{5,i}(t)}) - \beta_1 = 0,\\ &\kappa_{5,i}(t) \in [\kappa_{3,i}(t), 1 - x_1(t))\end{aligned} \tag{30}$$

In the case that SAVA has been deployed for a long period of time, when $\beta_{1,i} > \kappa_{6,i}(t)$, we have $x_2^i = 0$ (i.e., the users who have not adopted UVA will choose CSVA). When $\beta_1 \le \kappa_{6,i}(t)$, the initial proportion will exceed $\kappa_{4,i}$. This makes the user proportion of that network convergent to $x_2^i = \kappa_{5,i}$ instead of 0.

4.3.3 Further Analysis of Competition

Based on the analysis of the equilibrium between SAVA and CSVA in Section 4.3.2, the utility difference can be further approximated as follows. (The details are shown in Appendix.II.)

$$\begin{aligned}&\mathrm{Util}_1 - \widehat{Util_1} \approx (\theta_{\mathrm{user}}(1 - e^{-\beta_2 x_1(t)}) - \beta_1)\\ &- \sum_{\lambda_i \ge \eta_2} \lambda_i \max(\theta_{\mathrm{user}} \frac{\beta_1}{1 - x_1(t) - \kappa_{5,i}(t)} - \beta_1, 0)\end{aligned} \tag{31}$$

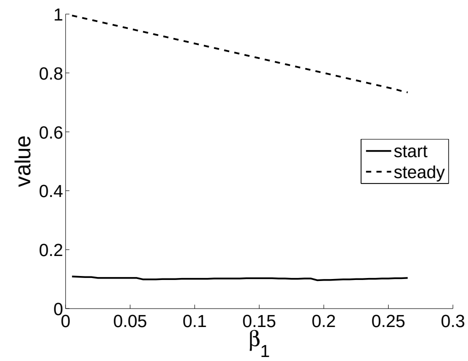


Fig. 4: The deployment of the universal video acceleration platform for different β_1

Based on Formula 31, it is easy to see that $H_1(t)$ can be calculated with a binary search algorithm. Since $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_z$, we can prove that $\kappa_{5,1} \ge \kappa_{5,2} \ge \cdots \ge \kappa_{5,z}$. Let $\eta_3(t)$ be the value of Formula 31 when $\theta_{\mathrm{user}} = 1 - x_1(t)$, we have

$$\begin{aligned}\eta_3(t) =& (1 - x_1(t))(1 - e^{-\beta_2 x_1(t)}) - \beta_1\\ &- \sum_{\lambda_i \ge \eta_2(t)} \lambda_i \beta_1 \frac{\kappa_{5,i}(t)}{1 - x_1(t) - \kappa_{5,i}(t)}\end{aligned} \tag{32}$$

Based on Formula 31, we have

$$\eta_3(t) > 0 \iff H_1(t) > x_1(t) \tag{33}$$

Based on our competition model, we can obtain

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = \gamma(H_1(t) - x_1(t)) \tag{34}$$

The conclusion in this subsection can be summarized in Theorem 2. This theorem shows that users will move to the universal video acceleration platform when their chunk sharing rate is larger than zero.

Theorem 2. *For competitions among the universal video acceleration platform, the stand-alone video acceleration platform and the C/S video platform, if $\eta_3(t) > 0$, the proportion of users who choose the universal video platform $x_1(t)$ will increase; if $\eta_3(t) < 0$, $x_1(t)$ will decrease; if $\eta_3(t) = 0$, the system is at an equilibrium point.*

4.4 Numerical Evaluation

In this subsection, we will further evaluate the proposed competition model. In particular, we will use the configurations in Formula 14 with two new parameters as follows:

$$U = 3 \times 10^8, C_1 = 20$$

Note that we have $\beta_1 = 0.00667$ and $\beta_2 = 7.5$. As shown in Figure 3, when the initial proportion is larger than 10.9%, UVA will be applied; its final proportion will be 97.3%.

Figure 3 shows that when the initial proportion $x_1(0)$ is larger than a certain value *start*, UVA will be applied. Its final proportion will be another certain value *steady*. Figure 4 shows the change of start and steady with different β_1 (normalized time cost of uploading local chunks). As we can see, steady is dramatically decreasing when β_1 is increasing. We can see that when β_1 is very large (e.g., users need to provide very large local cache), UVA has no obvious advantage in user utility. Also when $\beta_1 < 0.2$, steady will be larger than 80%. This means the final market share of UVA should be at least 80% with a reasonable cost. Note that we have estimated that $\beta_1 < 0.0067$ from real-world measurements. This means UVA is the final winner of this competition.

In conclusion, this section gives a popular competition model and shows that most users will prefer a universal video accelerator under a bounded overhead. We also use the real-world trace to indicate that such an overhead is smaller than such a bound. On the other hand, through the competition analysis of UVA, SAVA and CSVA, we find out the preconditions for our design to outperform the dedicated ones. These conditions show potential in guiding the deployment of our real-world system applying UVA, PPVA. For example, it is easy to obtain that smaller β_1 contributes to the deployment of the universal acceleration platform. According to the definition of β_1 , in order to decrease β_1, we should take following measures during the PPVA deployment: a) to decrease the user cost, or b) to encourage user to increase the upload bandwidth limit without bringing out heavier overhead.

5 PPVA: Universal and Transparent Video Accelerator

Our model analysis indicates that a universal accelerator can efficiently improve user performance with an acceptable level of overhead. According to our performance analysis, we find that the sharing efficiency will be significantly enhanced with more VSSes and larger contents. Based on the competition analysis, we further indicate that the majority of users will have enough incentive to use the universal acceleration platform when β_1 (normalized time of uploading local chunks) is smaller than 0.2. These model-based analysis results greatly motivate the development of our commercial system, PPVA. In this section, we will present the framework as well as the design of this real-world universal acceleration platform. Except for the general issues that should be addressed in P2P accelerators for individual sites, there are still many unique design challenges for a universal transparent platform. And we will elaborate our solutions in the PPVA implementation.

5.1 Design Principles

Different from the classic CDN/P2P based video/content delivery [29], the design of PPVA highlights the full exploration that aggregating video and client resources across multiple P2SP networks, especially for identical videos replicated in diverse VSSes.

Universality. We aim to provide universal P2P accelerating services and shadow site heterogeneity. In detail, such services are independent to different site architectures, video formats, etc. It can also make effective use of various users and video resources across stand-alone sites to achieve better performance.

Transparency. Our system should provide transparent services that do not need to change client-server protocols across existing VSSes. In this way, our service can be smoothly and incrementally deployed on existing video clients.

Scalability. The existing VSSes have already attracted a great number of users. The system scalability will be an important concern when we integrate these users together for global optimization. To satisfy such a great number of users, our universal accelerating platform should be more scalable especially comparing to the stand-alone accelerators.

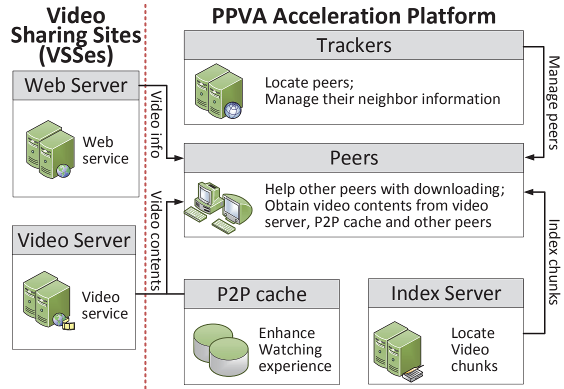


Fig. 5: The PPVA architecture

5.2 Objectives

Server Bandwidth Cost Alleviation. Although video sharing has become an immensely popular service in the recent years, revenues may be hard to achieve due to the enormous deployment costs. Therefore, bandwidth cost alleviation is urgently required for content providers.

Acceleration Effectiveness. Given limited network and server resources, user experience with existing VSSes are far from being satisfactory. With regard to acceleration effectiveness, we focuse on two aspects: reduction of the videos that cannot be viewed smoothly, and download speed acceleration. Note that although higher average download speed does not necessarily mean better user experience, it can reflect viewing experience to some extent.

5.3 Framework Design

To achieve these design goals, as shown in Figure 5, we design the PPVA framework which consists of the following key components:

Video Application: Video applications are the so-called VSSes, including video repository servers and their related web portals. Note that we neither impose any specific design guidelines on these servers, nor limit their video file formats, bitrates, or sizes. We only record indexed information of their video chunks on index servers.

Tracker: Trackers are used to manage peer dynamics (e.g., arrival and departure). A peer will be registered on a tracker when joining the system. Related information such as viewing preference and online duration will be also recorded on trackers.

Index server: Index servers are used to perceive peer's watching/sharing behaviors. For example, selecting a video to watch, implementing a seeking interaction and seeking for a watching position. It also provides indexed information of video chunks.

P2P cache: PPVA performance can be enhanced by optional P2P caches. These caches are dedicated servers which store replicas of the video contents and share these replicas upon requests.

Peers: They refer to the clients who run PPVA client software to fetch video data. These clients can access VSSes with conventional operations. PPVA will intercept the requests and transparently provide accelerated streaming services through P2P or a combination of peers and server downloads.

It is worth noting that PPVA provides three options while processing download acceleration: server only, P2P only, and a hybrid of both. By default, hybrid download is applied in our system. This is because such an approach can achieve a more stable download performance and accommodate accesses to both popular and non-popular videos.

5.4 System Operations

In this part, we will present the basic operations of PPVA.

Request Interception: In general, PPVA serves as a proxy between the client and its web browser. It will redirect and optimize user's video download requests.

Join the System: Once the P2P engine is invoked, a newly arrived client will first register its ID, IP address, shared resource list on trackers and update the resource list at preset intervals. It will also obtain a list of potential neighbors to fetch video data.

Play: Once the accelerator is invoked, three download options are available: from the server only, P2P only, and a hybrid of both. By default, the hybrid download is used by PPVA, which achieves the best download performance and accommodates access to all videos.

Note that for P2P operations, PPVA adopts the video engines in PPLive [34] and incorporates necessary extensions to achieve universality and transparency. The details will be given in the next subsections.

5.5 Peer Registration

For a particular video, all the peers that have previously downloaded this video serve as potential suppliers, forming an overlay for this video. PPVA utilizes distributed trackers to increase scalability and reduce lookup latency. First, all trackers are grouped accordingly. For one certain video, each group uses some trackers to manage this video. Thus, the peers who own this video are managed by multiple trackers in a tracker group. This can also prevent random failures of trackers.

When a peer joins the PPVA platform, it will register each of its video resources on a tracker. In particular, the peer sends Commit/KeepAlive to trackers it has registered in. A Commit message will be sent to trackers when a peer has downloaded new videos or deleted any watched videos. A KeepAlive message tells trackers that a peer is still online and its video resources are still available. To reduce overhead of Commit/KeepAlive messages, a peer chooses only one tracker from each tracker group to register in.

5.6 Video Identification and Indexing

VSSes have their own local video identification rules. PPVA, on the other hand, will assign a global video identifier to each accelerated video. A straightforward solution is to use the video URL for identification. Unfortunately, popular videos may have many different URLs, i.e., video replicas stored in different locations. To address this problem, PPVA adopts the hash value (e.g., MD5) of the video content, which is unique to each individual video. Note that a PPVA client cannot directly calculate such a video identifier before actually downloading it. Therefore, the video ID (VID) can only be indexed by index servers.

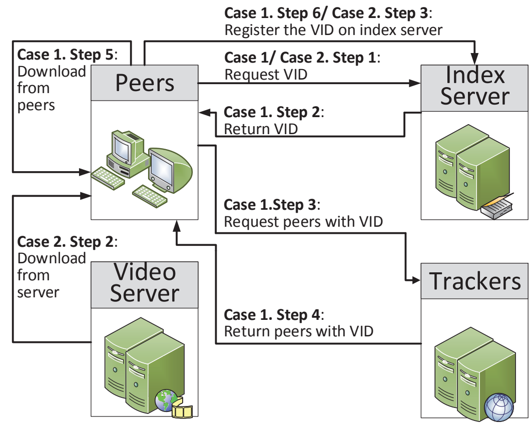


Fig. 6: Protocol of video indexing

As shown in Figure 6, there are two indexing scenarios: the non-first viewer (Case 1) and the first viewer (Case 2). The major difference is whether a global VID exists on index servers. This also indicates whether this video has been watched before (by other peers) in this system. Note that in Case 1, the trackers will return servers or peers with a given VID (Case 1. Step 4). This is to enable a hybrid download using both server and peers, which is similar to the very popular BitTorrent (BT) system. In particular, the PPVA servers are like the BT seeders and the PPVA peers are like the leechers in the BT system. To highlight the most important protocols in this design, we ignore some detailed steps in this figure. For example, the returned peer list could be empty. This means some peers have watched this video before though they are not currently online. The user will switch to Case 2 as a first time viewer. In Case 2, a peer sends a URL request to the index server. As the video has never been watched before, this peer can only download the video from the video server (Case 2. Step 2). After actually downloading the whole video, the peer registers the VID on the index server (Case 2. Step 3).

5.7 Caching and Replication

Unlike the streaming of live videos, the PPVA peers will not synchronize with each other while watching a video. In this case, the acceleration efficiency will be quite low, if the peers only cache what they are watching temporarily in their memories. For example, in YouTube, even when peers share their watched videos for a longer period of time (e.g., 1 day), the P2P-based approach will only assists 60% of the videos in the condition where there are at least 10 peers sharing videos continuously [35].

To mitigate this problem, the PPVA peers are required to contribute a fixed amount of hard disk storage (e.g., $1GB$).

The peers will cache their watched video files in the local storage when there is free hard disk space. As a result, for a client interested in a particular video, all the peers that have previously downloaded this video serve as potential suppliers, forming an overlay with the peers that are downloading this video. Obviously, a PPVA peer may appear in multiple overlays, and the server is by default in every overlay, ensuring at least one supplier exists. The entire viewer population as well as the video servers thus forms a hybrid P2P sharing system with much higher efficiency. It is easy to see that how to regulate this storage system is one of the most important design issues. To address this problem, PPVA applies the modified least frequency used strategy (LRU) that is applied in the existing PPLive system [36].

5.8 Pollution Prevention

In PPVA, all the video information is calculated and reported by peers. Errors may occur during data download and upload, and there could also be malicious attacks [37]. PPVA prevents them on two levels, namely, chunk level and piece level. We divide a video into different granularities for different purposes. A piece (e.g., with the size of 1KB) is the basic unit of a data packet. A chunk (e.g., with the size of $2MB$) is the basic unit of a disk. Piece level prevention is described as follow:

A piece is the minimum unit for data transmission in PPVA. A typical value of a piece is 1KB. When receiving a data piece, a peer should check the certificate to make sure the piece is unpolluted. An example is shown as follows when *Peer 1* is to send a data piece to *Peer 2*:

1. *Peer 1* uses an encryption algorithm to get a key, with input parameters of *Peer 1* ID, *Peer 2* ID and VID;

2. *Peer 1* uses the key and the data piece as parameters to calculate the value of MD5 as *Certificate 1*;

3. *Peer 1* sends the data piece and *Certificate 1* to *Peer 2*;

4. *Peer 2* calculates *Certificate 2* in the same procedure and compares the two certificates. If they are identical, *Peer 2* will accept the data piece. Otherwise, it will discard the data piece and request again.

On the other hand, we also provide chunk level prevention. It is known that chunk is the minimum unit for a peer to store video contents. Before storing, peers need to check the MD5 value of the chunk to prevent potential pollution:

1. The first peer that downloads the chunk calculates the MD5 value of the chunk and reports it to the index server along with the chunk ID;

2. Each subsequent peer downloading the chunk will acquire its MD5 value from the index server with the chunk ID. If the value matches the locally calculated MD5 value, the peer will accept the data; otherwise, the data will be discarded;

3. If the MD5 value uploaded by the first peer is wrong, many unmatched cases will happen after subsequent peers download the chunk. The index server can then examine whether the first uploaded MD5 value is wrong; in particular, it can compare the first uploaded MD5 value with the value downloaded directly from the original server.

It is worth noting that PPVA is a complex system. This section only reveals some key components in an acceleration platform. We believe that the design of these components can facilitate the research as well as the development of similar systems.

6 Performance Evaluation

PPVA is a real-world system for universal and transparent video acceleration. In this section, we will discuss the PPVA performance from the log and trace analysis. Particularly, we will estimate the system performance from three aspects[6]: server bandwidth cost, acceleration effectiveness and client overhead. Besides, the large-scale deployment of PPVA also enables us to systematically examine the similarity and differences of diverse VSSes, which we will further discuss it in Section 7.

6.1 Measurement Methodology

To monitor our system performance and user behavior, we have deployed a number of log servers to collect reports from peers. In particular, peers will send reports to log servers at two time points: 1) when peers completed the download of a video; 2) when peers terminate their PPVA client software. Such reports include peer ID, video ID, file size, bitrate, etc. IP addresses as well as content-level details (e.g., video titles) are not collected to ensure user privacy. Based on our tracing, users can generate 3.5 million reports every day. We therefore apply this internal data from these log servers to our analysis. Note that such internal data cannot be obtained from passive measurements.

6.2 Server Bandwidth Cost

To accurately estimate the bandwidth cost that PPVA has reduced, we introduce a metric Bandwidth Saving Ratio (BSR). The higher the BSR is, the more bandwidth is saved for servers. For a particular video m, its BSR is given by

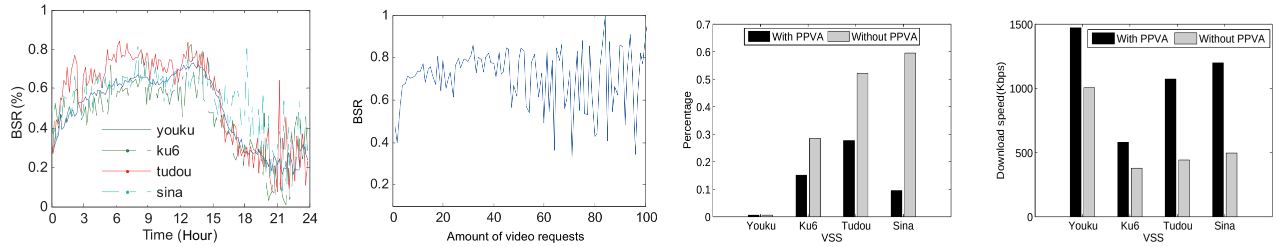
$$BSR_m = \frac{Upload_m}{Download_m} \tag{35}$$

where $Download_m$ is the bytes downloaded by peers who watch video m, and $Upload_m$ is the bytes uploaded by peers who watch video m. Therefore, the difference between $Download_m$ and $Upload_m$ is caused by servers. For example, if a peer watches a video of $100MB$, with $20MB$ from resource servers and $80MB$ from other peers, then the BSR for this download is 80%. Let M and N be the number of videos and the number of peers in the system, respectively, and N_m be the number of peers who watch video m. Then the BSR of the system becomes

$$BSR = \frac{1}{N} \sum\nolimits_{m=1}^{M} (N_m * BSR_m) \tag{36}$$

In Figure 7(a), we calculate the average BSR every 10 minutes within a 24-hour time frame across all VSSes. As we can see from the figure, PPVA can achieve up to 80% traffic saving on the servers even for the most popular VSSes. It is also worth noting that the dynamic of this ratio follows a very clear daily pattern. Specifically, the lower BSR during night is mainly because the disposal ability of the index server is limited. During the peak time (around

6. Note that we are focusing on the performance issues of PPVA. The detailed peer dynamics are not discussed due to the privacy issues in this commercial system.



(a) BSR evolution (b) BSR vs. video popularity (c) Rough video proportion (d) Download speed

Fig. 7: Performance evaluation of PPVA

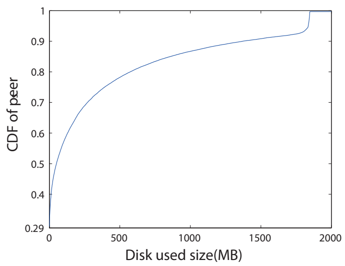


Fig. 8: Overheads of local disk

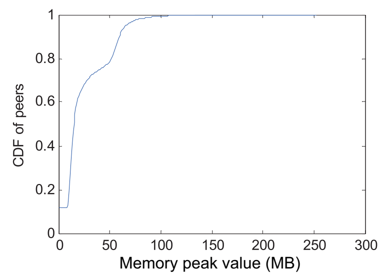


Fig. 9: Overheads of local memory

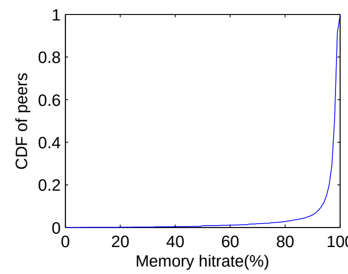


Fig. 10: Memory hit ratio

21:00), as the index server cannot dispose so many requests, some are abandoned, resulting in a relatively high abandon ratio. The bandwidth saving rate therefore depends on user demands on different VSSes. Also, we think another reason to the lower BSR is that the proportion of PPVA users may decrease even when the total number of users increases during the peak time. To mitigate these issues, providers are suggested to deploy higher-performance index servers in PPVA systems and encourage more users to use PPVA system.

Figure 7(b) shows BSR of videos with different popularities; X-axis represents the amount of video requests in a one-day period. We can see that BSR increases with video popularity when the number of video requests in one period is less than 10. This is because the peers watching unpopular videos can hardly find enough replicas to accelerate downloads.

6.3 Acceleration Effectiveness

As the download speed and bitrate may vary, videos whose average download speeds are less than the video bitrate usually cannot be viewed smoothly. We call them *rough videos* in this paper. On the other hand, although higher average download speed does not necessarily mean better user experience, it can reflect viewing experience to some extent. With regard to acceleration effectiveness, this section focuses on two aspects: the rough video reduction and the download speed acceleration.

It is easy to measure average download speed using PPVA, since PPVA client software records and reports this data. However, it is impractical to measure the average download speed without using PPVA. We make an approximate measurement here. Many viewers cannot find peer resources, possibly because they are watching unpopular videos or interacting with each other. Since the download is totally from servers, we define the speed as *speed without PPVA*. Accordingly, we define the speed of downloads from both peers and servers as *speed with PPVA*.

First, Figure 7(c) shows the rough video proportion with and without using PPVA, respectively. It illustrates that Youku provides the best viewing experience, while the other three video sites provide bad viewing experience. Particularly, nearly 60% of the Sina videos cannot be viewed smoothly. The acceleration effectiveness is obvious in Ku6, Tudou, and Sina.

Second, Figure 7(d) shows the average download speed of a peer with and without PPVA, respectively. We find the download speed increases after applying PPVA to Ku6, Tudou and Sina. This result is also applicable to Youku, though Youku's download speed without PPVA is already higher than that of the others.

6.4 Client Overhead

Although PPVA can improve user experience by accelerating download speed and reduce server bandwidth costs, it introduces additional costs to peers participating in a P2P overlay. We now examine the major client costs, including disk, memory and CPU.

Figure 8 shows the disk cost distribution. We can see that around 29% of the peers contribute zero spaces, about 80% of the peers contribute less than $500MB$, and all peers contribute less than $2000MB$, which is reasonable to current personal computers.

Figure 9 shows the memory cost distribution. We can see that every peer uses less than $100MB$, and nearly 80% of the peers use memory less than $20MB$. This is because peers store much more memory than their capacity. If the request data cannot be found in upload memory, peers will get the request data from the disk, which involves an I/O operation. Frequent I/O operations will result in bad user experience. Figure 10 shows memory hit ratio. Although a small memory is used, the memory hit ratio is still very high, which is also because PPVA employs an improved replacement algorithm rather than a naive FIFO. For active peers, the hit rates above 90% account for 44.54%.

Table 3 shows the peak CPU cost distribution. Again, 97.38% of the peers use only less than 5% of the CPU time. This indicates that the overheads are controlled to an acceptable level on PPVA clients.[7]

TABLE 3: Peak CPU cost distribution

CPU utility (%)	0-5	5-10	10-20	20-100
Percentage (%)	97.38	0.84	0.96	0.62

The log and trace analysis of the real-word system shows that PPVA achieves large amount of bandwidth saving on servers and obvious acceleration effectiveness in most VSSes. This is because PPVA is a universal platform exploring the aggregated video and client resources across diverse VSSes with acceptable overheads. As a consequence, from theoretical analysis to practical deployment, PPVA shows great potential in efficient resource integration and user experience enhancement.

7 Further Discussions

The large-scale deployment of PPVA provides us an opportunity to examine the performance of such a universal and transparent P2P accelerating service in real-world. It also enables us to systematically examine the similarity and differences of diverse VSSes. In this section, we firstly discuss some practical issues that are also related to user's watching experience. In particular, we will examine the details of video contents on the PPVA platform. Such characteristics have the potential to facilitate our future system enhancements.

On the other hand, based on the evaluation results, we can see that PPVA can provide efficient video acceleration service to real-world users. PPVA performance is largely benefited from the design of cross-VSS video acceleration. However, existing VSSes do not provide public interface which tells random seeking information such as the seeking position. Without special handling, PPVA can only implement random seeking by treating them as new video requests, its replication efficiency would be low. To better understand PPVA deployment, we further introduce how we eliminate the effects of random seeking in PPVA.

7.1 Characteristics of Video Contents

We first explore the video site popularity distribution, which will help to design P2P caching and ISP caching strategies. Figure 11 shows the aggregate views against normalized video ranks in one month. We can see that the top 10% popular videos account for 82% views, and the top 20% popular videos account for 94% views. On the other hand, we can also find that nearly 74% videos are not viewed at all. An immediate implication of this skewed distribution is that caching can be very efficient since storing a small set of

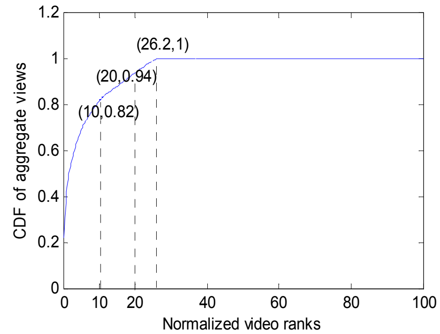


Fig. 11: Skewness of user interests

objects can enable high hit ratios. For example, by storing only 10% of the popular videos, a carefully designed cache system should be able to optimize 80% of the total requests.

We next examine such content-level characteristics as video sizes (MB), video bitrates (Kbps), and video lengths (Minute). Figures 12 presents the size distribution by different sites. This indicates that these VSSes mainly serve small videos less than $100MB$. Figures 13 shows the video bitrate distribution by different sites. We find that the bitrates in most VSSes are basically around $250Kbps$ with some minor differences. This indicates that low-bitrate videos are popular in these UGC websites. Figures 14 shows the lengths of videos with different sites. We can see the video length are quite different across different VSSes. For example, more than 99.8% of the Youku videos are less than 8 minutes. On the other hand, more than 50% of the Sina videos are more than 30 minutes. This is because some VSSes, such as Youku and ku6, have video length constraints.

Table 4 summarizes the median values of file characteristics. This statistics shows that the current VSSes can hardly support high-quality videos, due to limited server capacity. Based on the exiting analysis, we can see that the low-bitrate short videos are still dominating most VSSes. In addition to memory, storage and CPU usage, there are other factors that affect user experience, for example, increasing storage limits may impact the acceleration performance. To provide better watching experience, the growing trend of high-quality/extra-large video sharing will increase the complexity of video acceleration systems. One of our ongoing work is to design a cloud-based video acceleration system for high-quality/extra-large video contents. We are currently testing our prototypes and aiming to make it a build-in function in the PPVA platform.

TABLE 4: Median files at different sites

	Bitrate(Kbps)	Size(MB)	Length(Minute)
Youku	214	9	6.3
Ku6	231	11	7.1
Tudou	274	45	22.5
Sina	319	28	13.5

7.2 Random Seeking in PPVA

Our measurement shows that the random seeking interactions account for at least 18% requests [14]. Unfortunately, existing VSSes do not provide public interface which tells random seeking information such as the seeking position.

7. Note that we did not provide detailed trace/log analysis of the stand-alone video acceleration platforms. This is because the trace/log information of these commercial systems is not enclosed to the general public.

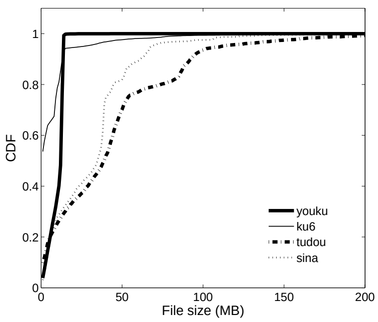


Fig. 12: Video size distribution

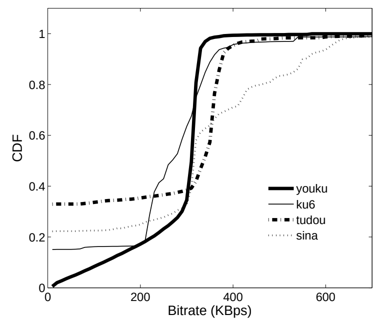


Fig. 13: Video bitrate distribution

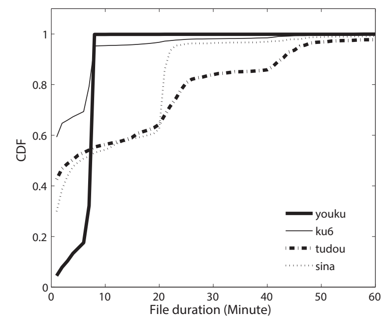


Fig. 14: Video length distribution

For example, a peer skips half of file A and then downloads the other half. This download request will be treated as a new file B rather than the half of file A. It can neither download data from peers that already have file A, nor upload data to peers that are watching file A. This problem could be best addressed by giving a uniform and public interface which discloses the information of user behaviors. However, this method may put some demands on existent video sites and there is no such interface between PPVA and these sites so far.

Figure 15 summarizes the percentage of the invoked P2P delivery upon two types of requests. *No seeking* means viewing requests that do not contain seeking interactions. In this measurement, a request with seeking interactions is regarded as a new file request that takes the server-only download option. We can see that the percentage of P2P delivery being invoked is much less if seeking requests are included, this shows that seeking interactions can significantly reduce VoD P2P delivery efficiency in PPVA. Moreover, Youku as the most popular site, has the highest value in *no seeking*, because it has a larger user base for P2P acceleration than others. In terms of Ku6 service, we find the proportion remains 40% upon the two types of requests. This is because Ku6 has the smallest user base and does not enable random seeking interactions.

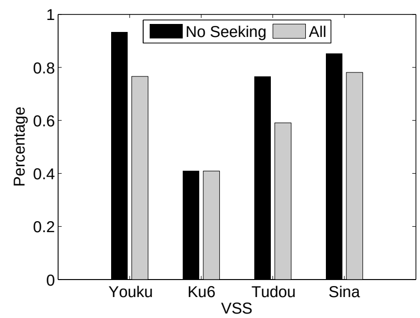


Fig. 15: Percentage of invoked P2P delivery

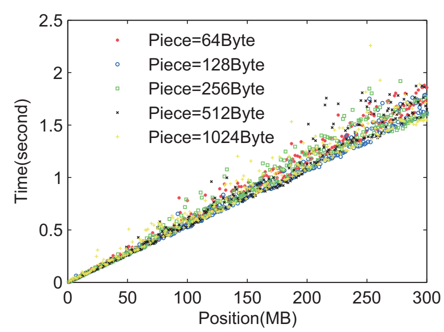


Fig. 16: Matching cost

7.2.1 Distributed Seeking Identification

To eliminate the effects of random seeking in PPVA, we propose a distributed solution as follows:

- First, PPVA client parses whether a request is a seeking interaction. For example, it captures and parses the resource's URL to check whether it contains string "$?start =$ ". If so, this is a seeking request of the current watching videos.
- Second, PPVA client will download a small portion of data (e.g., $2KB$) directly from servers.
- Third, it sends the current VID and the downloaded $2KB$ data to the neighbor peers it is downloading from.
- Fourth, its neighbors will match the $2KB$ data with its local file that has the same VID.
- Fifth, the neighbors will then send the offset back, or send "$null$" if it does not match or matches more than one position.

If a peer receives "$null$" from its neighbors, it will download directly from servers. If a peer receives an offset feedback, it will use this offset and VID to request more neighbors. Note that there could be multiple matches if the data size for the search is too small. Our experiments will show that $2KB$ size is good enough to guarantee a unique match in most cases.

7.2.2 Deployment Overhead

Matching cost is the main overhead of the distributed seeking identification method. Figure 16 illustrates the matching cost with different matching data lengths and matching positions. Here we use KMP [38] fast pattern matching in strings algorithm to match data. The complexity of this algorithm is $O(M + K)$, where M is the size of file and K is the size of matching data. We find that matching cost is nearly linear to the file size. In particular, the cost is less than $2s$ when the file is $300MB$. As our measurement reveals that the average file size is $20MB$ [14], the average cost would be less than $200ms$.

For PPVA configuration, one important parameter is the matching data size. Table 5 shows the percentage of more than one match with different file sizes and matching data sizes. It shows that $2KB$ is large enough to uniquely identify the seeking position.

TABLE 5: Percentage of more than one match

Piece=	64B	128B	256B	512B	1KB	2KB
File=5MB	0.17%	0.11%	0.08%	0.06%	0	0
File=10MB	0.23%	0.21%	0.16%	0.09%	0.01%	0
File=20MB	0.25%	0.22%	0.21%	0.11%	0	0
File=300MB	0.11%	0.08%	0.04%	0.03%	0	0

In conclusion, we propose a distributed seeking identification method to handle the random seeking in PPVA. Our measurement reveals that its overhead is acceptable. Unlike regarding the seeking as new file requests, with the development of applying the distributed seeking identification method, PPVA will be capable of fully utilizing the replicas in peers.

8 Conclusion and Future Work

In this paper, we investigate the modeling as well as the implementation of PPVA platform. It is easy to see that our framework design can transparently bridge users together across multi-thousand sites, enabling enhanced and fully compatible viewing experiences. The analysis of real-world traces also enables us to thoroughly investigate the effectiveness and potential loopholes, providing valuable guidelines for the future enhancements.

As a future work, we are particularly interested in the cloud deployment of PPVA. Especially, to deploy a cloud-based video acceleration service, we have to carefully examine the performance analysis of TCP/UDP flows on different types of VMs on public cloud platforms such as Amazon EC2 [39]. We find that VMs' *hypervisors* (also known as *virtual machine managers* such as Xen, KVM and VMware) and the total capacity play important roles during the video traffic dispatching. Moreover, some unique features of the cloud platforms such as task interference [40] and performance variation [41] [42] will significantly affect the performance of our video accelerator and thus need to be very carefully considered in our enhancement design. We believe that our system will not only improve the overall performance of video sharing services, but also facilitate the development of many other content delivery systems with similar design issues.

Acknowledgments

This work was supported by National Natural Foundation of China (61472212), National Science and Technology Major Project of China (2015ZX03003004), National High Technology Research and Development Program of China (863 Program) (2013AA013302, 2015AA015601), and EU Marie Curie Actions CROWN (FP7-PEOPLE-2013-IRSES-610524). J. Liu's research is supported by a Canada NSERC Discovery Grant, an NSERC Strategic Project Grant, and an NSERC E.W.R. Steacie Memorial Fellowship.

References

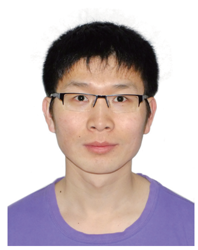
[1] "300+ hours of video uploaded to youtube every minute," http://www.reelseo.com/youtube-300-hours/.
[2] "Youku," http://www.youku.com/.
[3] "Ku6," http://www.ku6.com/.
[4] "Sina video," http://video.sina.com.cn/.
[5] S. Khemmarat, R. Zhou, D. K. Krishnappa, L. Gao, and M. Zink, "Watching user generated videos with prefetching," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 343–359, 2012.
[6] D. K. Krishnappa, S. Khemmarat, and M. Zink, "Planet youtube: Global, measurement-based performance analysis of viewer's experience watching user generated videos," in *IEEE Local Computer Networks (LCN)*, 2011, pp. 948 – 956.
[7] M. Saxena, U. Sharan, and S. Fahmy, "Analyzing video services in web 2.0: A global perspective," in *Proc. of NOSSDAV*, 2008.
[8] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. of IWQoS*, 2008, pp. 229 – 238.
[9] "Report on user generated content by tubular labs," http://www.reelseo.com/user-generated-content-brand-videos/.
[10] "Building faster youtube," https://www.bostonglobe.com/business/2012/09/23/building-faster-youtube/JqbVsEFUJfa5tpQmgbujkL/story.html.
[11] "Youku accelerator," http://desktop.youku.com/iku2/iku2.1_setup.exe.
[12] "Tudou accelerator," http://download.tudou.com/speedup/download/TudouVa_1.61.3.4_Release_Build.exe.
[13] "iqiyi accelerator," http://app.iqiyi.com/pc/acc/.
[14] K. Xu, H. Li, J. Liu, W. Zhu, and W. Wang, "Ppva: A universal and transparent peer-to-peer accelerator for interactive online video sharing," in *Proc. of IWQoS*, 2010, pp. 1–9.
[15] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *Proc. of SIGCOMM*, 2004.
[16] C. Gkantsidis, T. Karagiannis, P. Rodriguez, and M. Vojnovic, "Planet scale software updates," in *Proc. of SIGCOMM*, 2006.
[17] V. Pai, K. Kumar, K. Tamilmani, V. Sambamurthy, and A. Mohr, "Chainsaw: Eliminating trees from overlay multicast," *P2P systems IV*, pp. 127–140, 2005.
[18] X. Zhang, J. Liu, B. Li, and T. Yum, "Coolstreaming /donet: A data-driven overlay network for efficient live media streaming," in *Proc. of INFOCOM*, 2005.
[19] F. Liu, B. Li, L. Zhong, B. Li, and H. Jin, "Flash crowd in p2p live streaming systems: Fundamental characteristics and design implications," *IEEE TPDS*, vol. 23, no. 7, pp. 1227–1239, 2012.
[20] B. Cheng, L. Stein, H. Jin, and Z. Zhang, "Towards cinematic internet video-on-demand," in *Proc. of EuroSys*, 2008.
[21] Y. Huang, T. Z. Fu, D.-M. Chiu, J. Lui, and C. Huang, "Challenges, design and analysis of a large-scale p2p-vod system," in *Proc. of SIGCOMM*, 2008.
[22] Z. Liu, C. Wu, B. Li, and S. Zhao, "Uusee: Large-scale operational on-demand streaming with random network coding," in *Proc. of INFOCOM*, 2010.
[23] F. Liu, S. Shen, B. Li, B. Li, H. Yin, and S. Li, "Novasky: Cinematic-quality vod in a p2p storage cloud," in *Proc. of INFOCOM*, 2011.
[24] PPstream, http://www.pps.tv/.
[25] C. Jing, L. Tong, D. Ruiying, F. Jianming, and L. Jianwei, "Efficient reliable opportunistic network coding based on hybrid flow in wireless network," *China Communications*, vol. 8, no. 4, pp. 125–131, 2011.
[26] B. Li, J. Liu, and W. Zhu, "Two decades of multimedia streaming: the past reveals the future," *Multimedia Com.*, vol. 2, 2012.
[27] F. Liu, S. Shen, B. Li, B. Li, and H. Jin, "Cinematic-quality vod in a p2p storage cloud: Design, implementation and measurements," *JSAC*, vol. 31, no. 9, pp. 214 – 226, 2013.
[28] F. Liu, B. Li, B. Li, and H. Jin, "Peer-assisted on-demand streaming: Characterizing demands and optimizing supplies," *Transactions on Computers*, vol. 62, no. 2, pp. 351–361, 2013.
[29] "Akamai," http://www.akamai.com/.
[30] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4p: Provider portal for applications," in *Proc. of SIGCOMM*, vol. 38, no. 4, 2008, pp. 351–362.
[31] iQIYI, www.iqiyi.com/.
[32] G. K. Zipf, *Human Behavior and the Principle of Least Effort*. Boston, MA, USA: Addison-Wesley, 1949.
[33] S. Sen, Y. Jin, R. Guérin, and K. Hosanagar, "Modeling the dynamics of network technology adoption and the role of converters," *Networking*, vol. 18, no. 6, pp. 1793–1805, 2010.

[34] PPVA, http://codecpack.co/download/PPLive-Video-Accelerator.html.
[35] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, “I tube, you tube, everybody tubes: Analyzing the world’s largest user generated content video system,” in *Proc. of IMC*, 2007.
[36] H. Li, K. Xu, J. Seng, and P. H, “Towards health of replication in large-scale p2p-vod systems,” in *Proc. of IPCCC*, 2009.
[37] T. Li, F. Yu, Y. Lin, X. Kong, and Y. Yu, “Trusted computing dynamic attestation using a static analysis based behaviour model,” 2011.
[38] D. E. Knuth, J. H. Morris, and V. R. Pratt, “Fast pattern matching,” *Siam Journal on Computing*, vol. 31, no. 1, p. 2006, 1977.
[39] “Amazon ec2,” http://aws.amazon.com/ec2/.
[40] H. Wang, R. Shea, F. Wang, and J. Liu, “On the impact of virtualization on dropbox-like cloud file storage/synchronization service,” in *Proc. of IWQOS*, 2012.
[41] M. Schwarzkopf, D. G. Murray, and S. Hand, “The seven deadly sins of cloud computing research,” in *Proc. of HOTCLOUD*, 2012.
[42] J. Dai, F. Liu, and B. Li, “The disparity between p2p overlays and isp underlays: issues, existing solutions, and challenges,” *Network*, vol. 24, no. 6, pp. 36–41, 2011.

Ke Xu (M’02-SM’09) received his Ph.D. from the Department of Computer Science & Technology of Tsinghua University, where he serves as full professor. He is a co-inventor of three US patents and received the Best Paper Award of GLOBECOM 2015. He serves as Associate Editor of IEEE Internet of Things Journal and has guest edited several special issues in IEEE and Springer Journals. His research interests include next generation Internet, P2P systems, Internet of Things, network virtualization, and network economics. He is a member of ACM.

Tong Li received the B.S. degree from the School of Computer Science of Wuhan University, Hubei, China in 2012. Currently he is a Ph.D. student in the Department of Computer Science & Technology of Tsinghua University. His research interests include network virtualization and resource management, network science, and P2P system.

Haiyang Wang received the Ph.D. degree from Simon Fraser University, Burnaby (Metro-Vancouver), British Columbia, Canada, in 2013. He is currently an assistant professor in the Department of Computer Science at the University of Minnesota Duluth. His research interests include cloud computing, peer-to-peer networking, social networking, big data and multimedia communications.



Haitao Li is currently a research scientist at UCAR Inc, Beijing, China. He received Master Degree from Tsinghua University and PhD from Simon Fraser University. His research interests are in networking, social network and multimedia communication.

Wei Zhu received the B.S. degree in Huazhong University of Science and Technology, China in 2007. He is now an engineer in PPTV, Shanghai, China. His research interests include peer-to-peer system, cloud computing, social network and mobile Internet.

Jiangchuan Liu (S’01-M’03-SM’08) is a University Professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada, and an NSERC E.W.R. Steacie Memorial Fellow. He is an EMC- Endowed Visiting Chair Professor of Tsinghua University, Beijing, China (2013-2016). From 2003 to 2004, he was an Assistant Professor at The Chinese University of Hong Kong.

He received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a co-recipient of the inaugural Test of Time Paper Award of IEEE INFOCOM (2015), ACM SIGMM TOMCCAP Nicolas D. Georganas Best Paper Award (2013), ACM Multimedia Best Paper Award (2012), IEEE Globecom Best Paper Award (2011), and IEEE Communications Society Best Paper Award on Multimedia Communications (2009). His students received the Best Student Paper Award of IEEE/ACM IWQoS twice (2008 and 2012).

His research interests include multimedia systems and networks, cloud computing, social networking, online gaming, big data computing, wireless sensor networks, and peer-to-peer networks. He has served on the editorial boards of IEEE Transactions on Big Data, IEEE Transactions on Multimedia, IEEE Communications Surveys and Tutorials, IEEE Access, IEEE Internet of Things Journal, Computer Communications, and Wiley Wireless Communications and Mobile Computing. He is Steering Committee Chair of IEEE/ACM IWQoS (2015-2017) and TPC co-chair of IEEE IC2E’2017 and IEEE/ACM IWQoS’2014. He serves as Area Chair of IEEE INFOCOM, ACM Multimedia, IEEE ICME, and etc. According to Google Scholar, the citations of his papers are over 10,000, and his H-index is 46.

Song Lin received the B.S. and Ph.D. degrees in computer science from Tsinghua University, China in 2007 and 2012, respectively. His research interests include next generation Internet and P2P.

APPENDIX. I

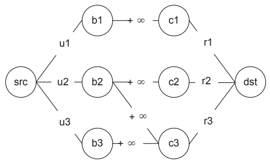


Fig. 1: Network flow

In Section 3, we use $\mathrm{Peer}_{\mathrm{union}}$ to refer to the download bandwidth using a universal video accelerator. This appendix elaborates the derivation of $\mathrm{Peer}_{\mathrm{union}}$.

To understand the maximum available bandwidth on peers, we convert the problem into a maximum flow problem[1] in a flow network[2]. As shown in Figure 1, node bandwidth constraints are transferred into edge capacities. Without loss of generality, we also add a virtual source and a virtual sink in the flow network $G(V, E)$.

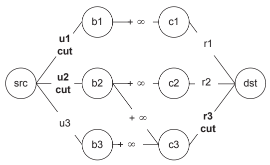


Fig. 2: An example for cut

According to the max-flow min-cut theorem[3], the maximum network flow for $s \to t$ equals to the min-cut of the graph. As to one cut $(\hat{S}, \hat{T})$, we let

$$L = \hat{T} \cap C$$

$$R = B \backslash \hat{S}$$

$$R^\star = \bigcup_{c \in L} O(t, c)$$

where $B(b_i \in B)$ is the set of all peers, $C(c_i \in C)$ is the set of all chunks, L is set of chunks required by all requests, $L \subseteq C$. R is the set of peers who have download requirements. $R^\star$ is the set of peers who already downloaded at least one chunk in L. $O(t, c)$ denotes the set of peers owning chunk c at time t. u is the upload bandwidth of peers and r is the expected download bitrate of the chunk. As infinite path does not exist in the cut, peers in $\hat{S}$ do not own the chunks in $\hat{T}$. We therefore have

$$R^\star \cap \hat{S} = \phi$$

$$R^\star \subseteq R$$

We use u to denote the total upload capacity of peers and use d to denote the total download bitrate of requests. Therefore, the edges across $\hat{S}$ and $\hat{T}$ have the total capacity of

$$\sum_{c_i \in C \backslash L} d(c_i) + \sum_{b \in R} u(b) = d(C \backslash L) + \sum_{b \in R} u(b)$$

[1] http://en.wikipedia.org/wiki/Maximum_flow_problem
[2] http://en.wikipedia.org/wiki/Flow_network
[3] http://en.wikipedia.org/wiki/Max-flow_min-cut_theorem

For a fixed L, the minimum value of $d(C \backslash L) + u(L)$ can be achieved when

$$R = R^\star$$

i.e.,

$$\hat{T} = \{\mathrm{dst}\} \cup \mathrm{L} \cup \mathrm{R}^\star$$

If we traverse over all the subsets L of C, we have the capacity of the min-cut as follow:

$$\min_{L \subseteq C}(d(C \backslash L) + u(L)) \tag{1}$$

Figure 2 shows an example of a cut $(\hat{S}, \hat{T})$, where $\hat{S} = \{src, b_3, c_3\}$, $\hat{T} = \{p_1, c_1, p_2, c_2, dst\}$, $L = \{c_1, c_2\}$, $R = \{p_1, p_2\}$,$R^\star = \{p_1, p_2\}$.

Theorem 1. *When the download requirements are uniformly distributed among all the chunks, the maximum bandwidth that all peers can obtain (from all the other peers) can be approximated as:*

$$\mathrm{Peer}_{\mathrm{union}} \approx \min(\mu_r n p_r, n\mu_u(1 - e^{-\frac{knp_r}{m}}))$$

where n is the total number of peers in the system, m is the total number of unique chunks in the system, k is the node storage size (number of chunks), μ_r is the peer's expected value of download bandwidth, μ_u is the peer's expected value of upload bandwidth, and p_r is the probability of requesting a chunk.

Proof. According to the definition of n and p_r, the number of peers having download requirements is

$$|R| \approx np_r \tag{2}$$

Based on Formula 1, we have

$$\mathrm{Peer}_{\mathrm{union}} = \min_{L \subseteq R}(d(R \backslash L) + u(L)) \tag{3}$$

We define $h = |L|$, which denotes the number of chunk requests in L. The intuition of this equation is based on the *Max-flow min-cut theorem* when we put the peer-chunk relationship into a flow network. According to the Including Excluding Principle, we can therefore obtain $\mathrm{cc}(h, m)$ as follow, which indicates the expectation of taking h repeatable items (from m items).

$$\mathrm{cc}(h, m) = \sum_{j=1}^{\min(m,h)} \left(jC_m^j \sum_{i=1}^{j} (-1)^{j-i} C_j^i (\frac{i}{m})^h \right)$$

Here we simplify our calculation based on the approximation as follow:

$$\mathrm{cc}(h, m) \approx m - me^{-\frac{h}{m-0.5}} \approx m(1 - e^{-\frac{h}{m}})$$

Further, for a given L, the probability that a peer contains at least one of the chunks in L is

$$1 - (1 - \frac{\mathrm{cc}(h, m)}{m})^k \approx 1 - e^{-\frac{kh}{m}}$$

The total upload capacity $u(L)$ is the product of μ_u and the number of peers that contains at least one chunk in L, i.e.,

$$u(L) = n\mu_u(1 - e^{-\frac{kh}{m}})$$

$$d(R\backslash L) = \mu_r(|R| - h)$$

For a fixed h, $d(R(t)\backslash L)$ and $u(L)$ are similar among L with $|L| = h$. According to Formula 3, we have

$$\text{Peer}_{\text{union}} \approx \min_{0 \le h \le |R(t)|} (\mu_r(|R(t)| - h) + n\mu_u(1 - e^{-\frac{kh}{m}}))$$

The minimal function value for all the real numbers in an interval can be approximated by the minimal function value for integers in the interval. From Formula 2, we get

$$\text{Peer}_{\text{union}} \approx \min_{0 \le x \le np_r} (\mu_r(np_r - x) + n\mu_u(1 - e^{-\frac{kx}{m}}))$$

i.e.,

$$\text{Peer}_{\text{union}} \approx \min(\mu_r np_r, n\mu_u(1 - e^{-\frac{knp_r}{m}}))$$

the theorem is proved. □

Appendix. II

In this part, we give an approximation to better obtain the differences between user utility values (when using different acceleration approaches). In particular, we elaborate the derivation of $\text{Util}_1 - \widehat{Util_1}$, where Util_1 is the utility of using the universal video acceleration platform and $\widehat{Util_1}$ is the utility of not using the universal video acceleration platform.

To improve readability, here we still enumerate some essential formulas which have been stated in Section 4. As mentioned before, the general user utility function is defined as follow:

$$\text{Util}_{\text{net}}(t) = \theta_{\text{user}} D_{\text{net}}(t) - C_{\text{net}} \tag{4}$$

where $D_{\text{net}}(t)$ is expected as the quality of service the platform can provide. Here we simply use user's expected value of download bandwidth instead. θ_{user} is a user-related variable uniformly distributed in $[0, 1]$, characterizing different users in valuing the objective benefit. C_{net} is the user cost, such as the contribution of storage and upload bandwidth.

Based on Theorem 1 and Formula 4, the utility of using the universal video acceleration platform can be approximated as

$$\text{Util}_1 \approx \frac{\theta_{\text{user}} F}{Up_r} + \frac{\mu_u}{p_r}(\theta_{\text{user}}(1 - e^{-\beta_2 x_1(t)}) - \beta_1) \tag{5}$$

where $\beta_1 = \frac{C_1 p_r}{\mu_u} > 0$ refers to the normalized time cost of uploading local chunks to other peers. U is the set of users, F is the total capacity of servers, μ_u is the peer's expected value of upload bandwidth, and p_r is the probability of requesting a chunk.

For the remaining users (not using the universal video acceleration platform), their utility is as follows:

$$\begin{aligned} \widehat{Util_1} \approx & \frac{\theta_{\text{user}} F}{Up_r} \\ & + \frac{\mu_u}{p_r} \sum_{i=1}^{z} \lambda_i \max(\theta_{\text{user}}(1 - e^{-\beta_2 \lambda_i x_2^i(t)}) - \beta_1, 0) \end{aligned} \tag{6}$$

Firstly, we denote

$$\kappa_{2,i} = \beta_2 \lambda_i = \frac{kUp_r\lambda_i}{m} > 0 \tag{7}$$

where $\beta_2 = \frac{kUp_r}{m} > 0$. k is the node storage size (number of chunks), λ_i is the probability that a user visits the i-th VSS, and m is the total number of unique chunks in the system. $\kappa_{3,i}$ is the only real number satisfying the following factors:

$$\begin{aligned} & (\kappa_{2,i}(1 - x_1(t) - \kappa_{3,i}(t)) + 1)e^{-\kappa_{2,i}\kappa_{3,i}(t)} = 1, \\ & \kappa_{3,i}(t) \in (0, 1 - x_1(t)) \end{aligned} \tag{8}$$

$$\kappa_{6,i}(t) = (1 - x_1(t) - \kappa_{3,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{3,i}(t)}) \tag{9}$$

When $\beta_1 \le \kappa_{6,i}(t)$, let $\kappa_{4,i}(t)$ and $\kappa_{5,i}(t)$ be the unique real numbers satisfying the following equations:

$$\begin{aligned} & (1 - x_1(t) - \kappa_{4,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{4,i}(t)}) - \beta_1 = 0, \\ & \kappa_{4,i}(t) \in (0, \kappa_{3,i}] \end{aligned} \tag{10}$$

$$\begin{aligned} & (1 - x_1(t) - \kappa_{5,i}(t))(1 - e^{-\kappa_{2,i}\kappa_{5,i}(t)}) - \beta_1 = 0, \\ & \kappa_{5,i}(t) \in [\kappa_{3,i}(t), 1 - x_1(t)) \end{aligned} \tag{11}$$

It is easy to see that based on Formulas 8 and 9, we can have

$$\beta_1 \le \kappa_{6,i}(t) \iff \beta_1 \kappa_{2,i} \le e^{\kappa_{2,i}\kappa_{3,i}} + e^{-\kappa_{2,i}\kappa_{3,i}} - 2$$

Based on Formula 8, we have

$$\kappa_{2,i}(1 - x_1(t)) + 1 = e^{\kappa_{2,i}\kappa_{3,i}(t)} + \kappa_{2,i}\kappa_{3,i}(t)$$

We can therefore get

$$g_1(x) = x + \ln x, x > 0$$

where $g_1(x)$ is strictly increasing. Assume $g_1^{-1}(x)$ is the inverse function of $g_1(x)$. $g_2(x)$ and $g_3(x)$ can therefore be defined as

$$g_2(x) = x - 2 + \frac{1}{x}, x > 0$$

$$g_3(x, a) = \frac{g_2(g_1^{-1}(ax + 1))}{x}, x > 0, 0 < a \le 1$$

We can see that $g_3(x, a)$ is a strictly increasing function of x, where $g_3^{-1}(x, a)$ is the inverse function. We therefore have

$$\eta_2(t) = \frac{g_3^{-1}(\beta_1, 1 - x_1(t))}{\beta_2}$$

Based on Formula 8, we have

$$g_1^{-1}(\kappa_{2,i}(1 - x_1(t)) + 1) = e^{\kappa_{2,i}\kappa_{3,i}(t)}$$

$$\beta_1 \le \kappa_{6,i}(t) \iff \beta_1 \le g_3(\kappa_{2,i}, 1 - x_1(t))$$

Based on Formula 7, we have

$$\beta_1 \le \kappa_{6,i}(t) \iff \eta_2(t) \le \lambda_i$$

Based on Formulas 5, 6 and 11, we have

$$\begin{aligned} \text{Util}_1 - \widehat{Util_1} \approx & (\theta_{\text{user}}(1 - e^{-\beta_2 x_1(t)}) - \beta_1) \\ & - \sum_{\lambda_i \ge \eta_2} \lambda_i \max(\theta_{\text{user}} \frac{\beta_1}{1 - x_1(t) - \kappa_{5,i}(t)} - \beta_1, 0) \end{aligned}$$