Peer-Assisted Content Distribution Aided by Video Popularity Evolution Model

Mohamad Dikshie Fauzie* Achmad Husni Thamrin* Jun Murai† *Graduate School of Media and Governance †Faculty of Environment and Information Studies
Keio University, 252-0882 Kanagawa, Japan
dikshie@sfc.wide.ad.jp husni@ai3.net jun@wide.ad.jp

***Abstract*—In this paper, we present peer-assisted CDN model, the peer-to-peer networks that can be used to help CDN to deliver streaming media. Our peer-assisted CDN model is aided by Youtube VoD views popularity model called CPPro. We have comparatively evaluated CPPro through trace-driven simulations with synthetic workloads. We use three scenarios in our simulation which are (a) the video popularity in the CPPro system follows the global popularity of the video. (b) the video popularity in the CPPro system is lagged behind the global popularity by several weeks. (c) the video popularity in the CPPro is unrelated to the global popularity. Our results show that CPPro gives lower number of replicas while maintaining same number of peers contribution compare to counterpart work. We also do the significance to the number of replicas using the Kolmogorov-Smirnov statistic on two samples and we find our results are significant ($p - values < 0.1$).**

***Index Terms*—Internet Video on Demand (VoD), P2P, CDN, and Caching.**

I. INTRODUCTION

Streaming content, especially video, represents a significant fraction of the traffic volume on the Internet, and it has become a standard practice to deliver this type of content using Content Delivery Networks (CDNs) such as Akamai and Limelight for better scaling and quality of experience for the end users. For example, YouTube uses Google cache and MTV uses Akamai in their operations.

With the spread of broadband Internet access at a reasonable flat monthly rate, users are connected to the Internet 24 hours a day and they can download and share multimedia content. P2P (peer to peer) applications are also widely deployed. In China, P2P is very popular; we see many P2P applications from China such as PPLive, PPStream, UUSe, Xunlei, etc. [1]. Some news broadcasters also rely on P2P technology to deliver popular live events. For example, CNN uses the Octoshape [2] solution that enables their broadcast to scale and offer good video quality as the number of users increases.

From the Internet provider point of view, the presence of so many always-on users suggests that it is possible to delegate a portion of computing, storage and networking tasks to the users, thus creating P2P networks where users can share files and multimedia content. Starting from file sharing protocols, P2P architectures have evolved toward video on demand and support for live events.

Alternatively, video contents can be efficiently distributed on services offered by managed network architectures and CDN companies. The major issues of CDN are high deployment cost and good but not unlimited scalability in the long term. Given the complementary features of P2P and CDN, in recent years some hybrid solutions have been proposed and applied to the operational of CDN [3]–[5] to take the best of both approaches. In Peer assisted CDN, users can download content from CDN nodes from or other users or peers. A user may cache the content after download to serve requests from other users. Due to the complexity of the behavior of peers, the process should be done in the home gateway user where the ISP can control it.

In this work, we will revisit Guo et al.'s, [6] PROP as a basis to evaluate peer-assisted CDN and propose an improvement to the model for the PROP called CPPro. This system is called CPPro abbreviated from out technical term "CDN-P2P Project". We will take Youtube as an example of an Internet VoD service model. In the Youtube service model, we can get data such as (1) the time when a video is uploaded and (2) number of access or number of view. We can get such data using Youtube's API. In seminal work, Borghol et al., [7] used the above information to estimate when a video will become very popular. They divide a video's popularity into three phases: before-peak phase, at-peak phase, and after-peak phase. We will use an estimate of a video's popularity phases for helping PROP. We will explain video popularity in Sec.III. Our contribution is as follows: (1) We use the idea of VoD view popularity model to aid the PROP model. To the best of our knowledge, the combination of the PROP model and the VoD view popularity model is new. (2) From simulation-based experiments, we find that peer contributions in CPPro are almost as good as PROP while the numbers of replicas are lower than PROP resulting in a reduction of resources required.

Our paper presentation as follows: (1) we describe related work in sect.II; (2) we explain detail of Youtube popularity evolution model in sect.III; (3) we explain the caching strategy for CDN and peer in sect.IV; (4) we explain our simulation design, simulator, and its evaluation in sect.V. Finally, we present our conclusions in section VI.

II. RELATED WORK

Content Distribution Networks with peer assist have been successfully deployed on the Internet, such as Akamai [8], [3] and LiveSky [9]. The authors of [8] examine the risks and benefits of peer-assisted content distribution in Akamai and measure the effectiveness of its peer-assisted approach. The authors of [3] conclude from two real world traces that hybrid CDN-P2P can significantly reduce the cost of content distribution and can scale to cope with the exponential growth of Internet video content. Yin et al. [9] described commercial operation of a peer-assisted CDN in China. LiveSky solved several challenges in the system design, such as dynamic resource scaling of P2P, low startup latency, ease of P2P integration with the existing CDN infrastructure, and network friendliness and upload fairness in the P2P operation. Xu et al. [10] used game-theory to show the right cooperative profit distribution of P2P can help the ISP to maximize the utility. Their model can easily be implemented in the context of current Internet economic settlements. Misra et al. [11] also mentioned the importance of P2P architecture to support content delivery networks. The authors use cooperative game theory to formulate simple compensation rules for users who run P2P to support content delivery networks.

The idea of telco- or ISP-managed CDN has been proposed in recent years. The complexity of the CDN business encourages telcos and ISPs to manage their own CDN, rather than allow others to run CDNs on their networks. It has been shown that it is cost effective [12] [13]. Kamiyama et al. [14] proposed optimally ISP operated CDN. Kamiyama et al. mentioned that, in order to deliver large and rich Internet content to users, ISPs need to put their CDNs in data centers. The locations are limited while the storage is large, making this solution effective; using optimum placement algorithm based on real ISP network topologies. The authors found that inserting a CDN into an ISP's ladder-type network is effective in reducing the hop count, thus reduce total link cost. Based on the author definition: Ladder-type network is a network with a maximum degree under 10. Cisco has initiated an effort to connect telco- or ISP-managed CDNs to each other, to form a CDN federation [12] using open standards [15]. They argue that the current CDN architecture is not close enough to the users and ISPs can fill this position.

The idea of utilizing the user's computation power to support ISP operation is not new. The Figaro project [16] proposed the residential gateway as an integrator of different networks and services, becoming an Internet-wide distributed content management for a proposed future Internet architecture [16]. Cha et al., [17] performed trace analysis and found that an IPTV architecture powered by P2P can handle a much larger number of channels, with lower demand for infrastructure compared to IP multicast. Jiang et al. [18] proposed scalable and adaptive content replication and request routing for CDN servers located in users' home gateways. Maki et al., [19] propose traffic engineering for peer-assisted CDN to control the behavior of clients, and present a solution for optimizing the selection of content files. Mathieu et al., [20] are using data gathered from France telecom network to calculate reduction of network load if customers are employed as peer-assisted content delivery.

Guo et al., [6] work's PROP is closest with our work. PROP uses local system (local counter) to calculate the segment popularity in peer-assisted proxy system. PROP uses popularity for proxy cache replacement strategy. In peer side, the author use utility function for cache replacement strategy. A utility function assigns numerical value to outcomes, in such a way that outcomes with higher utility are always preferred of outcomes with lower utilities. In PROP's utility function, the difference between very popular videos and unpopular video is very difficult to differentiate. The utility function is also function from popularity. While the authors successfully show that the results are very good, the peer-assisted system behavior over time is not explain because the author focus on properties such as proxy cache size variations and peer cache size variations. The explanation of the optimal number of replicas is not also clear because unavailable information when the snapshot is taken. In our work, we complement Guo et al., [6] work with VoD viewing popularity evolution model and describe the behavior of the peer-assisted CDN over the time.

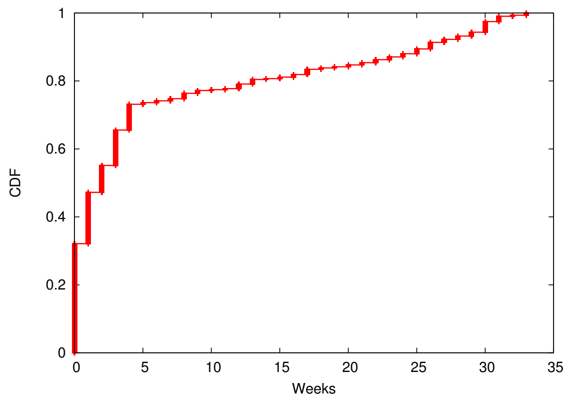


Fig. 1: Time to peak empirical distribution data from [7].

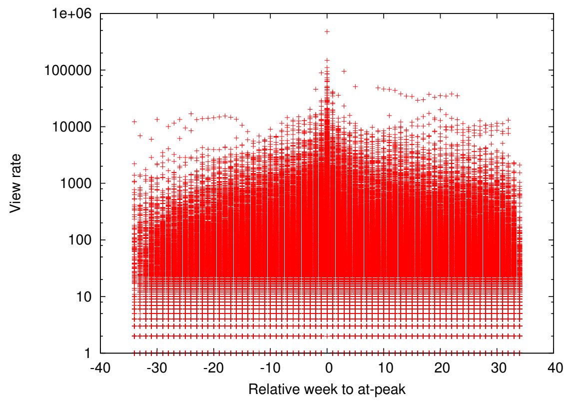


Fig. 2: View rate distribution versus week relative to at-peak phase week for every video, where y-axis in log scale. Every point lies in negative x-axis mean view rate of every video in before-peak phase. Every point lies in x-axis$=0$ mean view rate of every video at-peak phase. Every point lies in positive x-axis mean view rate of every video in after-peak phase. As we see in this graph, while fig.1 mentioned that 75% of videos reach at-peak within six weeks, we also see that some vides reach at-peak after six weeks. Data from [7].

III. Determining Internet VoD Popularity Phase

The objective of determining the Internet VoD popularity phase is to determine whether a video is at before-peak, at-peak, or after-peak phase, to be used by peers in their caching strategy. For this purpose we use the Youtube content popularity dataset from Borghol et. al., [7] which contains the data of 29791 videos, including the view count and upload time, during 36 weeks of measurements. Figure 1 is cummulative distribution function (CDF) the time-to-peak distribution from Borghol et. al., [7] which shows that around three-quarters of the videos peak within the first six weeks after upload. The time-to-peak is exponentially distributed up-to the sixth week, and it is uniform beyond the sixth week. Borghol et al., [7] define time-to-peak for a video as its age (time since upload) at which its weekly viewing rate is the highest during measurement (from the first week until end of measurement). Because we know the peak time (at-peak phase) of every video, we can also find the before-peak phase and after-phase of every video. For detail we refer the readers to [7].

Suppose a video v in Borghol dataset has a viewing rate $r_v(t)$, $0 <= t < t_f$, and $r_v(t)$ peaks at t_{vp}. The data is transformed by including the relative time-to-peak, such that each data point is a 2-tuple: video rate and relative time to peak, i.e., $rp_v(t) = (r_v(t), tp(t)), tp(t) = t - t_{vp}$. Figure 2 shows the Borghol's dataset with time axis for each video is translated by t_{vp} to the left, such that each video peaks at time 0.

Algorithm 1 Averaging relative weeks from the nearest neighbor points

Require: dataset that consist of weeknumber, viewrate, and relative week to at peak.
$t \leftarrow$ read(weeknumber) {read week number from dataset}
$r_v \leftarrow$ read(viewrate) {read view rate from dataset}
$tp \leftarrow$ read(relativeweeksatpeak) {read relative week at peak from dataset}
t_e {week number of a requested video}
r_e {view rate of a requested video}
$t_e^{before} \leftarrow (t_e - 1)$ {at one week before}
$tp_{before} \leftarrow find\_tp(t_e^{before}, r_e, t, r_v, tp)$
$t_e^{at} \leftarrow (t_e)$ {at same week}
$tp_{at} \leftarrow find\_tp(t_e^{at}, r_e, t, r_v, tp)$
$t_e^{after} \leftarrow (t_e + 1)$ {at one week after}
$tp_{after} \leftarrow find\_tp(t_e^{after}, r_e, t, r_v, tp)$
$tp\_final \leftarrow average(tp_{before}, tp_{at}, tp_{after})$
if $tp\_final < 0$ **then**
  $phase \leftarrow before$
else if $tp\_final == 0$ **then**
  $phase \leftarrow at$
else
  $phase \leftarrow after$
end if

In determining the phase of a requested video e with known age t_e and view rate r_e at t_e, we find the three r_v data points whose rates are closest to r_e at t_e, $(t_e - 1)$, and $(t_e + 1)$, and then average the tp of the three data points. The phase of the requested video e is estimated to be before-peak, at-peak, or after-peak based on whether the average is negative, 0, or positive. The view rate r_e of a video is calculated by substracting the view counts at the time of the current and the previous video requests. The pseudo code for averaging the tp is shown in algorithm 1. But when a video is being requested for the first time, the phase can only be estimated using the age of the video. In this case, we draw 36 random integer numbers s_i, $0 <= i <= 35$, using the time-to-peak distribution in fig. 1 then calculate the count of each integers between 0 and t_e from the drawn numbers then divide the result by 36, i.e., $estphase = \sum_0^{t_e} \frac{count(i,s)}{36}$. The number 36 come from the duration of measurement and each week has its own probability as we shown in fig. 1. This result represents the estimated phase. From time to peak distribution, 50% of video reach peak within four weeks. At that level, we expect that half of videos may reach at-peak and half of videos are not yet reach at-peak. Therefore we put 0.5 as low threshold. Still from the same time to peak distribution 75% of video reach peak within six weeks, and beyond six week the distribution is considered, it means there are not much additional view count. In other words, beyond six weeks we consider videos reach after-peak phase. Therefore we put 0.75 as high threshold. The pseudo code for this purpose is shown in algorithm 2.

Algorithm 2 Determine phase for the first access a requested video

Require: t_e and time-to-peak distribution
$len \leftarrow 35$
for $i = 0$ to len **do**
  draw integer random number between 0 and 35 respect to time-to-peak distribution: $d \leftarrow draw\_integer\_random\_number()$
end for
$total \leftarrow 0$
for $i = 0$ to t_e **do**
  $total \leftarrow total + count(d, i)$ {counting how many each integer random number and sum those}
end for
$estphase \leftarrow total/36$
if $estphase > 0.75$ **then**
  phase $\leftarrow$ after-peak
else if $estphase \leq 0.75$ and $estphase > 0.5$ **then**
  phase $\leftarrow$ at-peak
else
  phase $\leftarrow$ before-peak
end if

IV. System Description

A. System Overview

The main components of the system are: (1) CDN and (2) peers which are self organized into a P2P overlay network. Each peer in the system has two functionalities. First, a peer is a client that requests a video. Second, a peer is a contributor or share the cached video with other peers in the system.

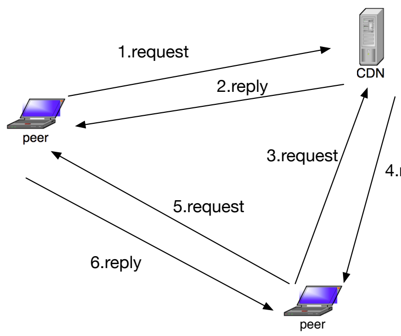


Fig. 3: Peer assisted CDN works as follows: when a peer requests a video, it always goes to a CDN server (step 1). The CDN provides the videos to the peer (step 2). If there is another peer request same video, that request will go to CDN (step 3). A CDN will check its record to see if there is some peers cache that requested video. If there is some peers cache that requested video, a CDN will reply with redirect message that asking a peer to download requested video from other peer (step 4). If there s no peers have requested video, a CDN will serve the video. A peer then can request the video to other peer and get the video (step 5 and step 6).

Peers control the number and utilization of their connection based on current resources availability. In fig.3, we describe the process of a peer that requests a video which derived from PROP. When a video is requested for the first time, the CDN is resposible to deliver the requested video. When a CDN receives a query for a same video, a CDN will find suitable peers that currently have a copy of a requested video. The CDN then returns information about these peers to the querying peer.

B. *Peer caching strategy*

A peer stores the requested videos into its cache based on the value of a utility function, i.e., in a peer with a full cache a newly requested video will replace one or more videos in the cache whose utility values are smaller than that of the new video. The utility function of a video is :

$$u = \frac{(f(p) - f(p_{min}))(f(p_{max}) - f(p))}{r^{\alpha+\beta}} + z(t) \qquad (1)$$

where the first term is the utility function of PROP, and the second term, $z(t)$, is the phase factor that we introduced in CPPro, where $z(t)$ is large if the video phase is at-peak. p, p_{min}, and p_{max} are the popularity of the video, the minimum popularity, and the maximum popularity in the system. The popularity of a video is the determined by the view rate of the video. r is the number of replicas. $alpha$ and $beta$ are adjustment factor [6].

Following [6], we can calculate p as follows:

$$p = min\left(\frac{n_i^r}{t_i^r - t_a^i}, \frac{1}{t_{cur} - t_i^r}\right) \qquad (2)$$

Where n_i^r is number of requested video, t_i^r is last time the video is requested, t_a^i is the uploaded time of the video, and t_{cur} is the current time. To able to track the simulation, we use default value from PROP thus we refer the readers to [6] for the details.

The utility function reflects the popularity of a video in the system that considering number of copy of its video or replica. u value itself lies in interval $[0, 2]$ Guo et al., [6]. We choose video with the smallest utility value as the candidate to be replaced when a peer's cache is full. The $z(t)$ values are set as follows:

$$z(t) = \begin{cases} 0.15 & \text{if phase estimation is before-peak} \\ 0.47 & \text{if phase estimation is at-peak} \\ 0.38 & \text{if phase estimation is after-peak} \end{cases} \qquad (3)$$

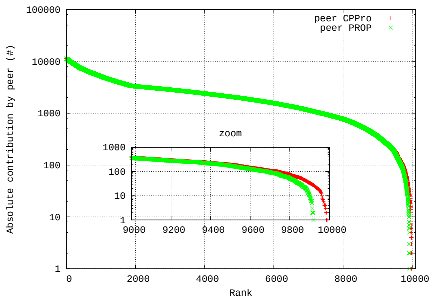
where the numbers are based on the proportion of video counts in the YouTube dataset before the peaks, around the peaks, and after the peaks. $z(t)$ for at-peak is determined by the total proportion of video counts from 1-week-before until 1-week-after the peak, which accounts to almost half of the total. The value of $z(t)$ is assigned after we finish to determine a video popularity phase. For example: if we determine a video popularity phase is at-peak, then we assign $z(t) = 0.47$. In PROP's utility function, the difference between very popular videos and unpopular video is very difficult to differentiate. For an unpopular video, $f(p)$ will be very close to $f(p_{min})$ thus $f(p) - f(p_{min})$ will be very close to 0 then the utility function becomes very small. For a very popular video, $f(p)$ will be very close to $f(p_{max})$, thus $f(p_{max}) - f(p)$ will be very close to 0 and the utility function becomes very small. Linear addition of $z(t)$ factor can help to differentiate the value of utility function.

V. Evaluation

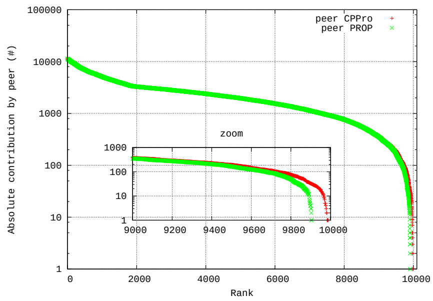
We performed simulations to compare the performance of CPPro against PROP in two main metrics: (1) peer contributions in video delivery, and (2) number of video replicas in peers. The peer contribution is the number of videos served by a peer instead of by the CDN server. The number of video replicas shows how much storage is used in peers to contribute in video delivery. The peer contribution metric is related to the byte-hit-ratio. The byte-hit-ratio is defined as the total bytes of content served by peers normalized by the total bytes of video all peers and the CDN consume. With more peer contributions, we will have higher byte-hit-ratio because peer can get content from other peers. However, because we only interested in peer performance, we compare peer contribution between PROP and CPPro. Contribution ratio of peer to total contribution (comparing to CDN contribution) becomes irrelevant in this case.

A. *Simulation Design*

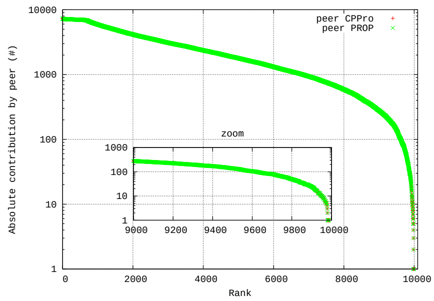
This peer-assisted CDN is simulated using an event driven simulation implemented in Python. Peers request videos from a video catalog where the peer request as well as the videos in the catalog are generated using certain distributions.



(a) Scenario A.

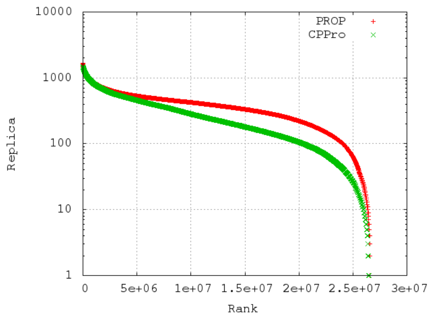


(b) Scenario B.

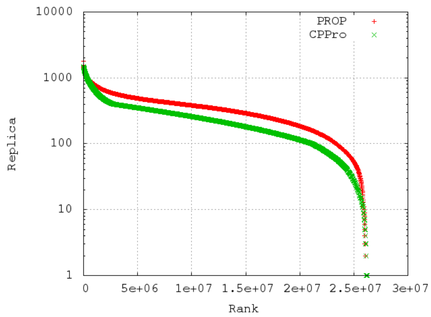


(c) Scenario C.

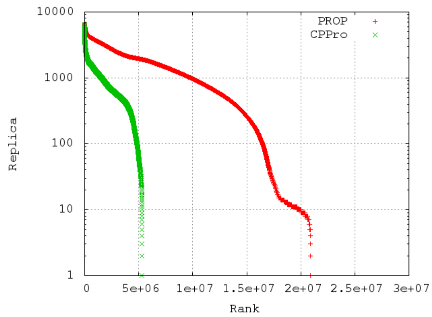
Fig. 4: Absolute peer contributions compared between CPPro and PROP (y-axis in log-scale and y-axis unit is times).



(a) Scenario A.



(b) Scenario B.



(c) Scenario C.

Fig. 5: Comparison of available replicas between model and prop when a peer requests a video (y-axis in log-scale). In scenario C, we found many zero replica when a peer requests a video for CPPro. Because we use log-scale in this figure, the zero numbers can not be viewed

TABLE I: Percentage of Cached events and Not-Cached events in CPPro and PROP.

Scenario	Type	Cached (times)	Not-Cached (times)	Cached (PetaByte)	Not-Cached (PetaByte)
A	CPPro	33.5%	66.5%	814.4	1616.6
	PROP	52%	48%	1264.1	1166.9
B	CPPro	34.8%	65.2%	845.9	1585.1
	PROP	52.6%	47.4%	1278.7	1152.3
C	CPPro	32.4%	67.6%	829.1	1729.9
	PROP	67.7%	32.3%	1732.4	826.6

TABLE II: Percentage cached and not-cached events for each video popularity phase in CPPro.

Scenario	Type/Events	Before-Peak	At-Peak	After-Peak
A	CPPro Cached	8.2%	1.2%	24.1%
	CPPro Not-Cached	11.2%	0.8%	54.5%
B	CPPro Cached	6.2%	1.2%	29.8%
	CPPro Not-Cached	5.2%	0.8%	56.8%
C	CPPro Cached	8.0%	1.8%	22.7%
	CPPro Not-Cached	15.1%	0.8%	51.6%

1) Video Catalog: Each video in tha catalog has the following properties: video-id,size,upload time, final view count, view count function parameters. View count parameters are the distribution parameters. The final view count is the total number of views of a video at the end of simulation and it is generated using uniform distribution. Upload time interval is a Poisson process with $\lambda = 1$. Video size is generated using a uniform distribution. Because of the very weak relationship betwen video size and popularity [21] and because our work focuses on the impact of the popularity aspect on the utility function rather than storage optimization we believe that the choice to assign a random uniform video size from the Youtube dataset does not have an effect to our results. The view rate progression from the upload time until the end of simulation time is modelled using a Beta distribution [7]. As Borghol et al., [7] showed that view rate of a video can be modelled using beta distribution we can calculate α and β parameters. Since we have view rate at peak, we can use Beta distribution mode

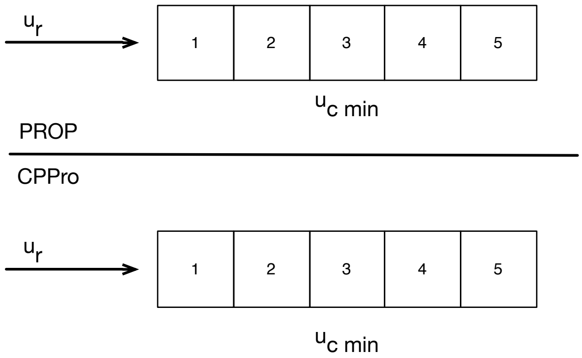


Fig. 6: Process how'ss a video will be cached or not by a peer. Top part for PROP and bottom part for CPPro. u_r is utility function for a requested video. $u_c$min is utility function for a video inside peer's storage.

formula to calculate α or β. In this case, we choose α random uniform between 1 and 3, thus β parameter can be calculated using mode formula.

2) Peer Request Generator: Peers request videos from the catalog using a Poisson process with $\lambda = 1$ [22] for the inter-arrival time. For the requested videos there are three scenarios namely A,B,and C. Scenario A is where the video popularity in the peer-assisted CDN system follows the global popularity of the video. Scenario B is where the video popularity in the peer-assisted CDN system lagging four weeks behind the global popularity of the video. We choose four weeks based on probability from time-to-peak distribution that half of videos are already reach peak within four weeks. Scenario C is where the video popularity in the peer-assisted CDN system does not follow the global popularity of the video. We use Zipf distribution with rate= 0.9 for this purpose [23].

3) Simulation Parameters: The simulation parameters are follows:

- Length: 360 days.
- Video size: uniform random between 1MB and 200MB.
- Peer storage capacity: 500MB.
- CDN storage capacity: 10000MB.
- Number of peers: 100000.
- Number of videos: 10000.
- Peer's caching strategy: CPPro, PROP.

Finally, we compare our results to PROP [6] implementation.

B. Result and Discussion

Figure 4 shows the peer contributions in each scenario ranked by the number of videos served by each peer. They exhibit a similar pattern and only differ in the tails (see insets) for Scenarios A and B, where CPPro gives higher, albeit insignificant, results. However, in the scenario C the peer contribution is almost identical. These results show that the introduction of $z(t)$ component to the utility function does not affect the performance in terms of peer contributions.

The advantage of CPPro to PROP is evident in fig. 5 which shows the number of replicas of the requested videos at each request event.

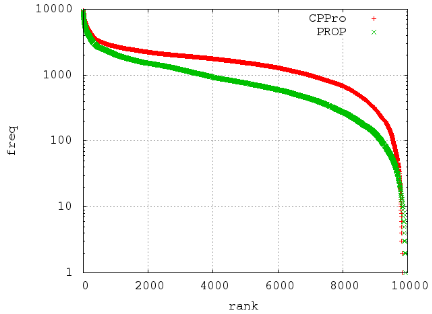
When a peer requests a video from another peer, a peer has to decide whether a peer wants to cache the video or not to cache the video based on the utility function value. If a peer decided to cache a video, we called it cached event. If a peer did not decide to cache a video, we called it not-cached event. We breakdown how many cache events occur in our simulation as shown in table.I. Furthermore, we breakdown again by video phase popularity for CPROP's cached and not-cached events as shown in table. II.

From table.I, we can see that CPPro cached less videos compared to PROP for all scenarios. In scenario A, CPPro are cached 33% of requested video compared to 52% in PROP. In scenario B, we also have similar number where CPPro are cached 34.8% of requested video compared to 52.6%. In scenario C, CPPro are cached 32.4% of requested video compared to 67.7% in PROP. However, by summing data volume for cached events and no-cached events, we see that CPPro deliver or contribute of videos delivery more than PROP. These number quantifies our previous analysis.

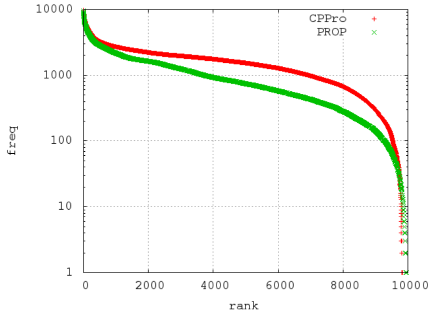
In table II, we can see that most of cached events and not-cached events for CPPro occured when a requested video was estimated as after-peak phase. In scenario A, cached events occured in after-peak phase around 24.1% from total events (cached events and not-cached events). Similarly in not-cached events, events of a peer was not cache a requested video occured in after-peak phase around 54.5% from total events. Scenario B also shows similar result where percentage of cached events occured in after-peak phase is 29.8% from total events and percentage of not-cached events in after-peak phase is 56% from total events. Scenario C shows similar result where percentage of cached events occured in after-peak phase is 22.7% from total events and percentage of not-cached events in after-peak phase is 51.6% from total events. Intuitively , We learn from these numbers that in at-peak phase, many replicas available thus a peer in CPPro system is decided to cache less video.

As we mentioned before that a peer decision to cache to not-cached a video is based on utility function value. We want to know what makes replica in CPPro is lower than PROP. Assume a peer requests a video with utility function u_r and the minimum utility function inside peer's cache is u_c-min as shown in fig. 6 where we have five videos inside peer's cache. Intuitively, When a requested video in after-peak phase, while videos inside peer's storage are in at-peak phase then in CPPro case high probability that a requested video will not be cached. In PROP case, since PROP does not has video popularity phase the probability of a requested video will not be cached is lower than CPPro. We also calculate the Kolmogorov-Smirnov statistic on two samples and we find that for all scenarios the p-values are less than 0.1 thus we consider the result are significant.

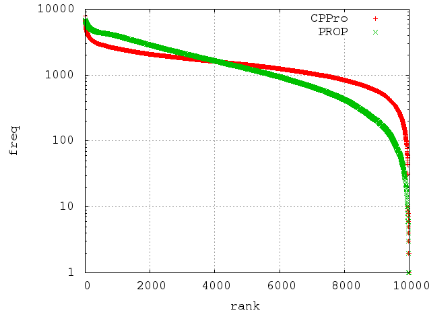
Figure 7a, 7b, and 7c show the frequency of a video stay in peers compared between CPPro and PROP. As all figure show CPPro has higher frequency than PROP to stay in peers except for the beginning rank of data where CPPro has same frequency with PROP in first and second scenario. In the third scenario, in the beginning rank of data CPPro has lower frequency than PROP, then around rank 1000 CPPro has higher



(a) Frequency a video stays in peers for scenario A.

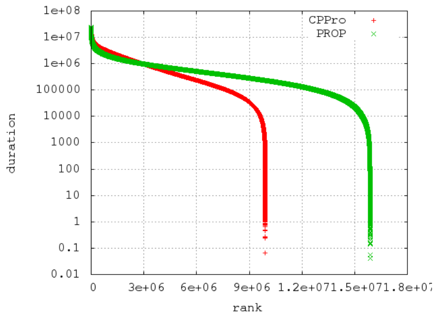


(b) Frequency a video stays in peers for scenario B.

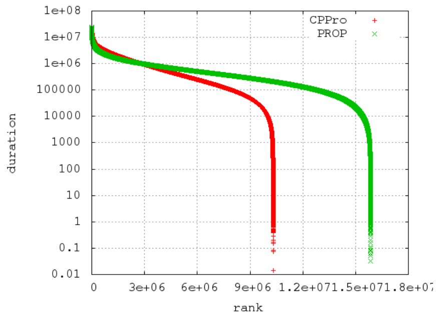


(c) Frequency a video stays in peers for scenario C.

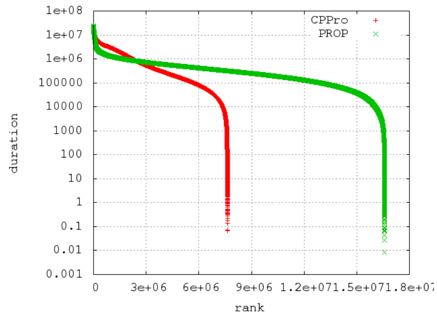
Fig. 7: Frequency a video stays in peers compared between model and prop.



(a) Cache duration of a video in peers for scenario A.



(b) Cache duration of a video in peers for scenario B.



(c) Cache duration of a video in peers for scenario C.

Fig. 8: Cache duration compared between model and prop.

frequency than PROP until the end of data. The frequency a video stay in a video can also be viewed in fig 8a, 8b, and 8c, where in CPPro some videos have longer cache duration than PROP, while others have shorter cache duration than PROP. Thus, we can see the relationship between cache duration and frequency a video stays in peers.

VI. Conclusion and Future Work

This paper presents a scheme for peer-to-peer network can help CDN to deliver the content over the Internet. We show that by introducing z factor to utility function CPPro can maintain same peer contribution while reducing number of replicas. We found that there are no much different all scenario in term of peer contribution to deliver a video. However, we found that in the all scenarios, CPPro gives lower replicas than PROP. This is because in CPPro, we found intuitively that probability not-cached events occur are higher than cached events, futhermore compared to PROP, probability a peer is not cache a requested video is higher in CPPro compared to PROP. Therefore, in CPPro the numbers of available replicas are lower than PROP. We also did the significance test to the number of replicas using the Kolmogorov-Smirnov statistic on two samples and we find that for all scenarios the p-values are less than 1% thus the results are significant.

Some areas of improvement that we have identified for future are: (1) since we know the CPPro offer a little bit peer contribution with less replica, we are interested to know the energy trade-off of this peer-assisted CDN architecture in order to know how much energy saving by ISP and how much increase of energy at users home gateway side in this architecture since we have higher peer contribution. (2) Involving the different popularity model from different geographically VoD service. This is very important to see the system behavior when receiving request from different location with different popularity while still in the same global CDN service. Futhermore, we can exploit this behavior for commercial ads in VoD service.

Acknowledgment

The authors would like to thank Internet research laboratory member at Keio University and anonymous reviewers.

References

[1] L. Vu, I. Gupta, K. Nahrstedt, and J. Liang, "Understanding overlay characteristics of a large-scale peer-to-peer iptv system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 4, pp. 31:1–31:24, Nov. 2010. [Online]. Available: http://doi.acm.org/10.1145/1865106.1865115

[2] Octoshape, "Octoshape," http://www.octoshape.com/cnn-com-using-octoshapes-p2p-for-live-feed/.

[3] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid cdn-p2p: why limelight needs its own red swoosh," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '08. New York, NY, USA: ACM, 2008, pp. 75–80. [Online]. Available: http://doi.acm.org/10.1145/1496046.1496064

[4] H. Jiang, J. Li, Z. Li, and J. Liu, "Efficient hierarchical content distribution using p2p technology," in *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, dec. 2008, pp. 1 –6.
[5] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Design and deployment of a hybrid cdn-p2p system for live video streaming: experiences with livesky," in *Proceedings of the 17th ACM international conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 25–34. [Online]. Available: http://doi.acm.org/10.1145/1631272.1631279
[6] L. Guo, S. Chen, and X. Zhang, "Design and evaluation of a scalable and reliable p2p assisted proxy for on-demand streaming media delivery," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 5, pp. 669–682, May 2006.
[7] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti, "Characterizing and modelling popularity of user-generated videos," *Perform. Eval.*, vol. 68, no. 11, pp. 1037–1055, Nov. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.peva.2011.07.008
[8] M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec, "Peer-assisted content distribution in akamai netsession," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 31–42. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504752
[9] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Livesky: Enhancing cdn with p2p," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, pp. 16:1–16:19, Aug. 2010. [Online]. Available: http://doi.acm.org/10.1145/1823746.1823750
[10] K. Xu, Y. Zhong, and H. He, "Can p2p technology benefit eyeball isps? a cooperative profit distribution answer," *CoRR*, vol. abs/1212.4915, 2012.
[11] V. Misra, S. Ioannidis, A. Chaintreau, and L. Massoulié, "Incentivizing peer-assisted services: a fluid shapley value approach," *SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 215–226, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1811099.1811064
[12] Cisco, "Cdn federation," 10 2012, available on http://www.cisco.com/web/about/ac79/docs/sp/CDN-Federation_Phase-2-Pilot.pdf.
[13] W. B. Norton, *The Internet Peering Playbook: Connecting to the Core of the Internet*, 2nd ed. DrPeering Press, December 2012.
[14] N. KAMIYAMA, T. MORI, R. KAWAHARA, and H. HASEGAWA, "Optimally designing isp-operated cdn," *IEICE Transactions on Communications*, vol. E96.B, no. 3, pp. 790–801, March 2013.
[15] IETF, "Cdn interconnect ietf working group," 2013, available on https://datatracker.ietf.org/wg/cdni/. [Online]. Available: https://datatracker.ietf.org/wg/cdni/
[16] "Figaro project," http://www.ict-figaro.eu/, 2012.
[17] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On next-generation telco-managed p2p tv architectures," in *Proceedings of the 7th international conference on Peer-to-peer systems*, ser. IPTPS'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855641.1855646
[18] W. Jiang, S. Ioannidis, L. Massoulié, and F. Picconi, "Orchestrating massively distributed cdns," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 133–144. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413193
[19] N. MAKI, T. NISHIO, R. SHINKUMA, T. MORI, N. KAMIYAMA, R. KAWAHARA, and T. TAKAHASHI, "Traffic engineering of peer-assisted content delivery network with content-oriented incentive mechanism," *IEICE Transactions on Information and Systems*, vol. E95.D, no. 12, pp. 2860–2869, December 2012.
[20] B. Mathieu and Y. Levene, "Impact of ftth deployment on live streaming delivery systems," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*, july 2012, pp. 000 259 –000 264.
[21] A. Abhari and M. Soraya, "Workload generation for youtube," *Multimedia Tools and Applications*, vol. 46, no. 1, pp. 91–118, 2010.
[22] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network - measurements, models, and implications," *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2008.09.022
[23] F. Guillemin, T. Houdoin, and S. Moteau, "Volatility of youtube content in orange networks and consequences," in *Communications (ICC), 2013 IEEE International Conference on*, June 2013, pp. 2381–2385.