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. We have comparatively evaluated our system through trace-driven simulations with synthetic workloads. We use three scenarios in our simulation which are (b) a peer requests a video that follow Youtube popularity, (b) a peer requests a video that follow Youtube popularity but we shift the request time four weeks, (c) a peer requests a video that based on zipf distribution popularity instead of Youtube. Our results show that our model 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. 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 our model 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,

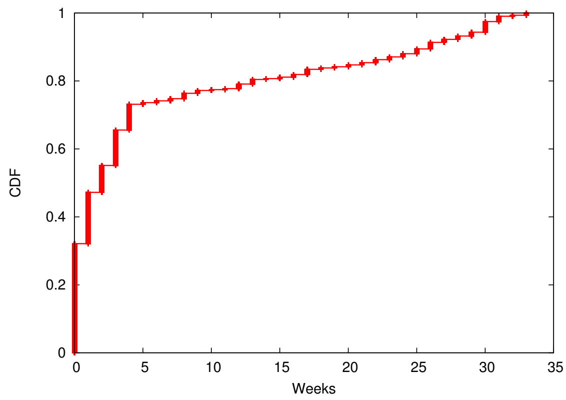


Fig. 1: Time to peak empirical distribution data from [7].

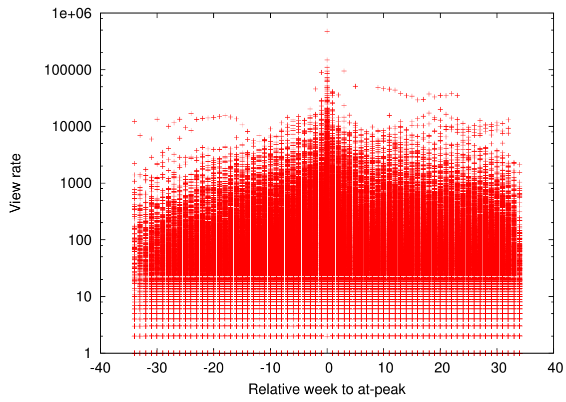


Fig. 2: View rate distribution versus week relative to at-peak phase week for every video, where y-axis in log scale. Every point lies in negative x-axis mean view rate of every video in before-peak phase. Every point lies in x-axis= 0 mean view rate of every video at-peak phase. Every point lies in 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].

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.

III. Determining Internet VoD Popularity

The objective of estimating the Internet VoD popularity phase is to determing whether a video is before-peak, at-peak, or after-peak. 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). We need those phases because we will use those in peer caching strategy that we will explain in Sec.IV-B. The value of those phase will effect the decision if a peer will cache or not cache the requested video.

We use Youtube as an example of VoD service where we get Youtube content popularity datasets from Borghol et al., [7]. The datasets consist of the measurement of 29000 videos, including view count and the time when videos are uploaded, during a period of 36 weeks. The time-to-peak distribution is shown in fig.1. Figure 1 shows that around three-quarters of the videos peak within the first six weeks since their upload. Beyond six weeks, we have uniform distribution thus the time-to-peak is exponentially distributed mix with uniform distribution. 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].

The steps to shift the week of the at-peak phase to zero for every video in dataset is shown as pseudo code in 1.

Algorithm 1 Shifting relative week to at-peak for every video in dataset

Require: $data \leftarrow viewrate$
length ← 36 {number of weeks}
for every video **do**
  viewratemax ← max(data) {find max view rate from a list}
  index ← findindex(viewratemax){find element index of maximum view rate}
  for $i = 0$ to $length$ **do**
    $j[i] \leftarrow 0$ {initialization with 0}
  end for
  for $i = 0$ to $length$ **do**
    $j[i] \leftarrow i - index$ {we got relative week to at-peak}
  end for
end for

As result from shifting the week of the at-peak phase to zero, we get relative week to at-peak phase. Next, we want to determine popularity phase of a requested video. A requested video has week number and view rate record that we can get from CDN. We can determine the popularity phase of a requested video by averaging the relative week to at-peak phase of the nearest points from a requested video. Pseudo code to determine the popularity phase is shown in 2 and 3. Pseudo code in 2 and 3 is only valid when we have view rate. Since for the first time a requested video does not has view rate, we have to guess the video popularity phase using time to peak probability distribution function as shown in 4. Pseudo code in 4 requires *weighted sampling* function which is shown in 5. *Weighted sampling* does sampling with replacement. The function *weighted sampling* is just algorithm fused with a walk of the items list to pick out the item selected by random numbers. This is work because the probability that n random numbers $0, \cdots, v$ will all happen to be less than z is $P = \frac{z}{v}^n$ and solving for z, we get $z = vP^{1/n}$. Subtituting

Algorithm 2 Averaging relative weeks from the nearest neighbor points

Require: dataset that consist of weeknumber, viewrate, and relative week to at peak.
x ← read(weeknumber) {read week number from dataset}
y ← read(viewrate) {read view rate from dataset}
z ← read(relativeweeksatpeak) {read relative week at peak from dataset}
xsearch ← weeknumber {week number of a requested video}
ysearch ← viewrate {view rate of a requested video}
$xs \leftarrow (xsearch-1)$ {search the nearest point at one week before}
$result1 \leftarrow findnearestpoints(xs, ysearch, x, y, z)$
$xs \leftarrow (xsearch)$ {search the nearest point at same week before}
$result2 \leftarrow findnearestpoints(xs, ysearch, x, y, z)$
$xs \leftarrow (xsearch+1)$ {search the nearest point at one week after}
$result3 \leftarrow findnearestpoints(xs, ysearch, x, y, z)$
$final \leftarrow average(result1, result2, result3)$
if $final < 0$ **then**
  $phase \leftarrow before$
else if $final == 0$ **then**
  $phase \leftarrow at$
else
  $phase \leftarrow after$
end if

Algorithm 3 Findnearestpoints function

Require: xs, ysearch, x, y, z.
$len_1 \leftarrow length(xs)$
for $i = 0$ to len_1 **do**
  if $x[i] == xs$ **then**
    $tempx[i] \leftarrow x[i]$
    $tempy[i] \leftarrow y[i]$
    $tempz[i] \leftarrow z[i]$
  end if
end for
$len_2 \leftarrow length(tempy[i])$
for $i = 0$ to len_2 **do**
  if $tempy[i] == ysearch$ **then**
    $result[i] \leftarrow tempz[i]$ {get relative week only for ysearch}
  end if
end for
return(result)

Algorithm 4 Determining popularity phase for the first time access

Require: Sorted PDF of time to peak in the form of list of tuple [(prob,week number),$\cdots$] and t which is week number of a requested video
$n \leftarrow 36$ {number of measurement weeks}
if $t > 36$ **then**
  $phase \leftarrow after$
else
  $result \leftarrow weightedsample(items, n)$
end if
for $i = 0$ to $length(result)$ **do**
  $temp[i] \leftarrow result[i])$ {add every element of result to a list temp}
end for
$temp \leftarrow sorted(temp)$ {sort temp}
$total \leftarrow 0$
for $i = 0$ to t **do**
  $numbercount \leftarrow count(temp, i)$ {count how many t's value inside temp}
  $total \leftarrow total + numbercount$ {add the number count to total}
end for
$total \leftarrow total/36$ {divide total by total number of week from dataset}
if $total \leq 0.5$ **then**
  $phase \leftarrow before$
else if $total > 0.5$ and $total \leq 0.75$ **then**
  $phase \leftarrow at$
else
  $phase \leftarrow after$
end if

Algorithm 5 Weighted sample (weighted probability sampling with replacement)

Require: Sorted PDF of time to peak in the form of list of tuple [(prob,week number), $\cdots$]
$total \leftarrow 1$
$i \leftarrow 0$
$w \leftarrow items[0][0]$ {the smallest probability}
$v \leftarrow items[0][1]$ {week number the pair of the smallest probability}
while n **do**
  $x \leftarrow total * (1 * random()^{(1.0/n)})$
  $total \leftarrow (total - x)$
  while $x > w$ **do**
    $x \leftarrow (x - w)$
    $i \leftarrow i + 1$
    $w \leftarrow items[i][0]$
    $v \leftarrow items[i][1]$
  end while
  $w \leftarrow (w - x)$
  $b \leftarrow append(v)$ {add v value to a list b}
  $n \leftarrow (n - 1)$
  return b
end while

a random number for P picks the largest number with the correct distribution. We repeat the process to select all the other numbers.

IV. System Description

In our work, we use the Youtube VoD view model to aid our work that based from PROP. The Youtube VoD view model will be used in our peer-caching strategy to exploit the video popularity while the caching strategy used in the CDN is out of scope for this project.

A. System Overview

The main components of the system are (1) CDN and (2) peers which are self organized into a P2P overlay network. CDN itself consist of edge servers that deliver the videos and control plane servers that coordinate or control between the peers and record peers activities. The recording function is basically maintain a database of which videos are currently available on which peers as well as details about the connectivity of these peers. Peers appear in control plane database when uploading a video and a peer has a video to share. In current peer-assisted CDN practice, the videos and their corresponding indices are decoupled. In other words, they are maintained by control plane servers. When a CDN receives a query for a video, a CDN will find suitable peers that currently have a copy os a requested video. The CDN then returns information about these peers to the querying peer. Peers control the number and utilization of their connection based on current resources availability. 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.

B. Peer caching strategy

We introduce the PROP's *utility function* of a video as:

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

where p represents the popularity of the video, p_{min} represents minimum popularity in the system, and p_{max} represents maximum popularity in the system. We choose video with the smallest utility value as the candidate to be replaced when a peer's cache is full. Since we can determine before-peak phase, at-peak phase, and after-peak phase of video, we modified the original utility function from PROP above by adding a $z(t)$ factor as follows:

$$u = \frac{(f(p) - f(p_{min}))(f(p_{max}) - f(p))}{r^{\alpha+\beta}} + z(t) \quad (2)$$

$\frac{(f(p)-f(p_{min}))(f(p_{max})-f(p))}{r^{\alpha+\beta}}$ is the utility function from PROP and $z(t)$ is the z factor from our model. $z(t)$ is a function from current time in simulation. $z(t)$ is used for every a requested video.

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

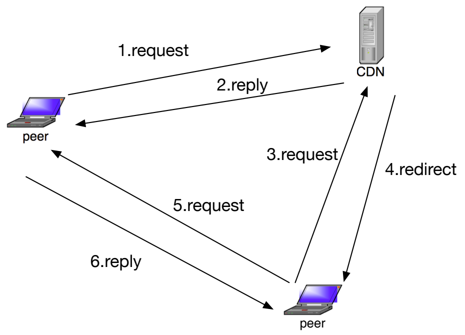


Fig. 3: Peer interaction in simulator. When a peer requests a video, it always goes to a CDN server (step 1). The CDN provides the videos to the peer (step 2). If there is another peer request same video, that request will go to CDN (step 3). A CDN will check its record to see if there is some peers 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).

Algorithm 6 Peer decision

Require: a requested video popularity phase

calculate $p_{request}$ for a requested video
if $phase == before$ **then**
    $z \leftarrow 0.15$
else if $phase == at$ **then**
    $z \leftarrow 0.47$
else
    $z \leftarrow 0.38$
end if
get p_{min} from CDN
get p_{max} from CDN
get r from CDN
$u_{request} \leftarrow \frac{(f(p_{request})-f(p_{min}))(f(p_{max})-f(p_{request}))}{r^{\alpha+\beta}} + z(t)$ {calculate u for a requested video}
for every video inside peer's cache **do**{calculate p and u for every video inside peer's cache}
    calculate p
    determine video popularity $phase$
    assign z value
    get r from CDN
    $u \leftarrow \frac{(f(p)-f(p_{min}))(f(p_{max})-f(p))}{r^{\alpha+\beta}} + z(t)$ {calculate u for a video inside peer's cache}
end for
$u_{min} \leftarrow min(u)$
if $u_{request} > u_{min}$ **then**
    if space not available **then**
        delete video with u_{min} inside peer's cache
        cache a requested video
    else
        cache a requested video
    end if
else
    do not cache a requested video
end if

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

$$p = min\left(\frac{n_i^r}{t_i^r - t_a^i}, \frac{1}{t_{cur} - t_i^r}\right) \tag{4}$$

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. $z(t)$ is proportion of view count that we get from Youtube datasets. The value of $z(t)$ decided after we finish to determine a video popopularity phase. For example: if we determine a video popularity phase is at-peak, then we assign $z(t) = 0.47$. To able to track the simulation, we use default value from PROP thus we refer the readers to [6]. 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. We summarize peer's decision for caching in pseudo code 6.

V. Evaluation

In order to evaluate the proposed peer-caching strategy using our algoritmic designation of the poplarity phase of before-peak, at-peak, and after-peak information from Youtube VoD view model, we have to compare our model to the PROP model. We evaluate three metrics, which are peer contribution to delivery contents during simulation, access frequency of cache during simulation, and number of replicas. 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. Access frequency of cache reflects the storage utilization. More access means more peer storage utilization. Number of replicas is also related to peer storage utilization. However, too many replicas will waste the storage resources. To evaluate these metrics, we developed a peer-assisted CDN simulator.

A. Simulation Design

An event driven simulator is developed using Python for this purpose. In fig.3, we describe the process of a peer that requests a video in simulator, which derived from PROP. A peer and a CDN are implemented in object oriented-model inside the simulator. In short, a peer always requests to CDN then CDN will decided if a requested video is available in other peers or not. If a requested video is available in other peers, CDN will redirect the request to other peers. If a requested video is not available CDN will serve the request.

1) Catalog Generator: The goal for the catalog generator is to create a catalog video that consist of the video-id, time when a video is uploaded, a video size, and final view count. We assume that a video is uploaded to the server following Poisson process with mean rate $\lambda = 1$. The final view count for every video is assigned randomly uniform from Youtube datasets and video size for every video is assigned randomly uniform between 1 and 200MB from Youtube datasets. Because of the very weak relationship betwen filesize 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 file size from the Youtube datasets does not have an effect to our results. Finally, we have a catalog that consists of: video-id, time when a video is uploaded, view count terminus, and video size.

2) Peer Request Generator: There are three scenarios for peer request (named as scenario A, B, and C): In scenario A, a peer chooses a video that has popularity following Youtube. The objective of the first scenario, we want to see the peer requests effect to peer-assisted CDN when the request following Youtube popularity. In scenario B, a peer chooses a video that has popularity following Youtube but we shift the request four weeks later. The objective of the second scenario, we want to see the peer requests effect to peer-assisted CDN when the request from peers are shifted four weeks after popular in Youtube. In scenario C, a peer chooses a video that has popularity following zipf distribution with rate$= 0.9$ [22] thus a peer choose a video that its popularity does not follow Youtube popularity. The objective of the third scenario, we want to see the peer request effect to peer-assisted CDN when the requests from peers are totally different from Youtube's videos popularity. In all scenarios we assume peer interarrival time request a video following Poisson process with a mean rate $\lambda = 1$ [23].

3) Simulation Parameters and Scenarios: 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: our model, PROP.

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

B. Result and Discussion

Figure 4a, 4b, and 4c show the absolute peer contribution to deliver videos comparing our model and PROP. Figure 4a and fig.4b exhibit the same pattern, with the peers giving more contribution in the tail. In the third scenario the peer contribution is almost identical in our model and PROP. A peers can contribute mode because a video has longer duration than other videos in a peer's cache thus other peer's requests are served by the peer. A video has longer duration than other videos in peer's cache because that a video has bigger utility function than other videos for example a video that will enter the cache.

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

Scenario	Type	Cached	Not-Cached
Scenario 1	Model	33.5%	66.5%
	PROP	52%	48%
Scenario 2	Model	34.8%	65.2%
	PROP	52.6%	47.4%
Scenario 3	Model	32.4%	67.6%
	PROP	67.7%	32.3%

Figure 5a, 5b, and 5a show the number of videos replicas available in system when a peer requests a video. As we can see from all figures, the model gives us lower number of replicas than PROP. The model gives lower number of replicas than PROP because when a peer requests a video, that peer is not cached the video. We can see the proportion of cached and not-cached event in table.I. We also present detail of the video phase breakdown in table.II. In model, not-cached events take around 65% from all events and majority of video phase is after-peak for both cached events and not-cached events. Because the majority of video phase is after-peak for both cached events and not-cached events, In PROP, cached events take around 52% from all events for the first scenario and the second scenario, while for the third scenario is 67.7%. In model not-cached events are bigger than PROP, means peers do not cached the videos thus we get lower replicas number than PROP.

Denote u_{dl} is the minimum utility function for a video inside the cache and u_{ms} is utility function for a video that will enter the cache, p_{dl} is the popularity for a video inside the cache and p_{ms} is the popularity for a video that will enter the cache. In order a requested video is cached by a peer, the utility function for u_{dl} must be lower than the utility function for u_{ms}.

$$u_{dl} < u_{ms} \tag{5}$$

$$\frac{(f(p_{dl}) - f(p_{min}))(f(p_{max}) - f(p_{dl}))}{r_{dl}^{\alpha+\beta}} + z_{dl} < \frac{(f(p_{ms}) - f(p_{min}))(f(p_{max}) - f(p_{ms}))}{r_{ms}^{\alpha+\beta}} + z_{ms} \tag{6}$$

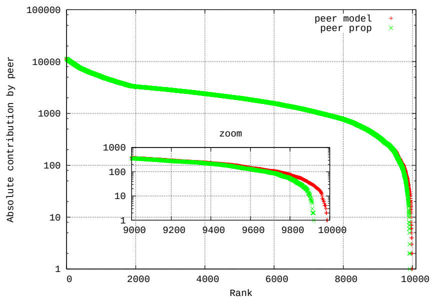
We assume that numbers of replicas are same, thus:

$$(f(p_{dl}) - f(p_{min}))(f(p_{max}) - f(p_{dl})) - (f(p_{ms}) - f(p_{min}))(f(p_{max}) - f(p_{ms})) < z_{ms} - z_{dl} \tag{7}$$

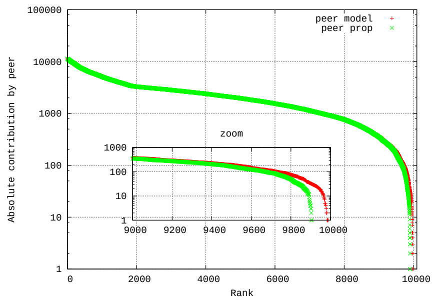
Since p_{min} and p_{max} are same for u_{dl} and u_{ms}, we can arrange the equation become:

$$f(p_{ms}) - f(p_{dl}) > z_{dl} - z_{ms} \tag{8}$$

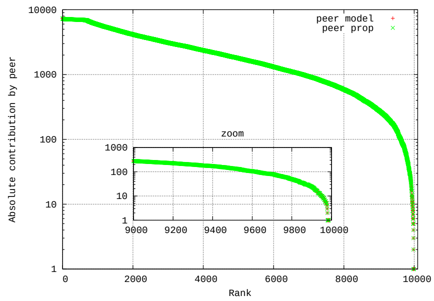
As we know from table.II that the majority of a requested video is after-peak phase and a requested video phase that



(a) Absolute of contribution of peer for scenario A (y-axis in log-scale).

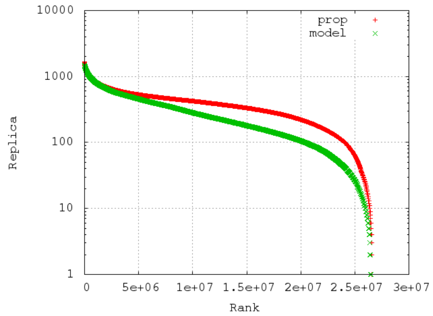


(b) Absolute of contribution of peer for scenario B (y-axis in log-scale).

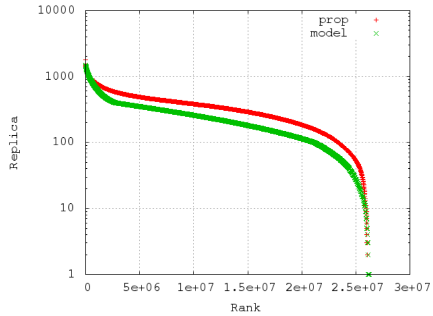


(c) Absolute contribution of peers for scenario C (y-axis in log-scale).

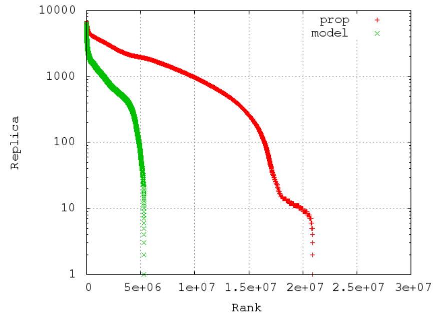
Fig. 4: Peer contributions compared between model and PROP.



(a) Number of a video replicas when a peer request a video for scenario A (y axis in log-scale).

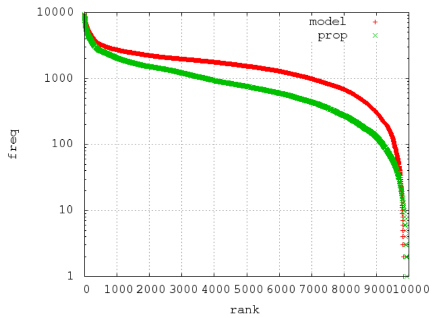


(b) Number of a video replicas when a peer request a video for scenario B (y axis in log-scale).

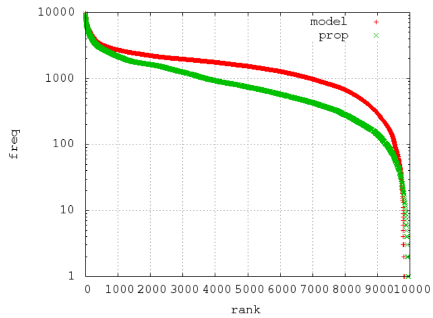


(c) Number of a video replicas when a peer request a video for scenario C (y axis in log-scale). In model we found many zero replica when a peer requests a video. Because we use log-scale in this figure, the zero numbers can not be viewed.

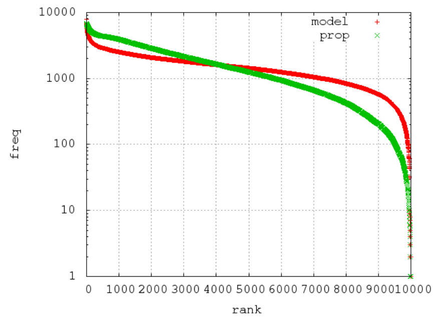
Fig. 5: Comparison of available replicas between model and prop when a peer requests a video.



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



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

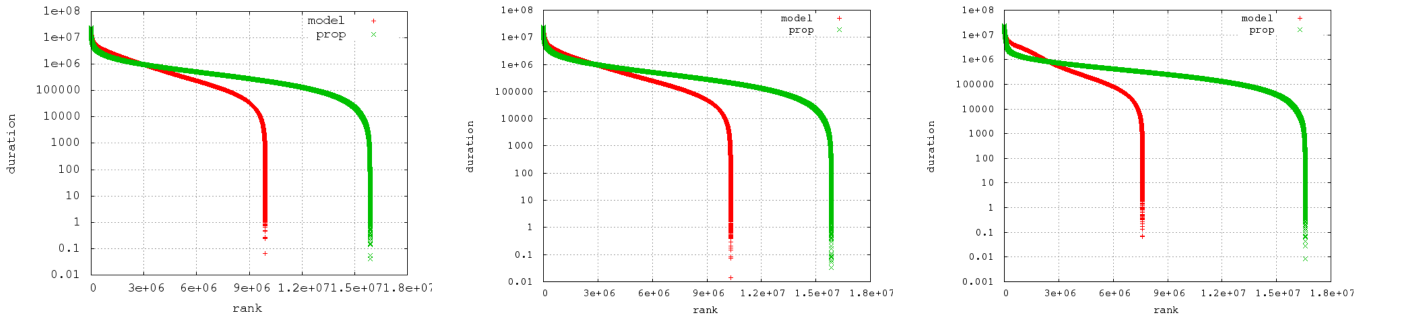


(c) Frequency a video stays in peers for scenario C.

Fig. 6: Frequency a video stays in peers compared between model and prop.

is at-peak phase is very small portion, then we can see that $z_{dl} - z_{ms}$ term will be in negative term if z_{dl} is before-peak phase or 0 if z_{dl} is after-peak phase. If $z_{dl} - z_{ms} = 0$ then it is same with PROP. Since the not-cached events happen when a requested video phase is after-peak phase, we can get that $f(p_{ms}) - f(p_{dl}) < 0$. For the same situation and we compare to the PROP, the probability of u_{ms} less than u_{dl} in the model is higher than PROP. Therefore, we can see in the model that the events when a peer does not cache a video are more often than PROP.

Figure 6a, 6b, and 6c show the frequency of a video stay in peers compared between model and PROP. As all figure show the model has higher frequency than PROP to stay in peers except for the beginning rank of data where the model has same frequency with prop in first and second scenario. In the third scenario, in the beginning rank of data the model



(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. 7: Duration compared between model and prop.

TABLE II: Percentage of Video Phase for Model in cached and not-cached events

Scenario	Type/Events	Before-Peak	At-Peak	After-Peak
Scenario 1	Model-Cached	8.2%	1.2%	24.1%
	Model-Not-Cached	11.2%	0.8%	54.5%
Scenario 2	Model-Cached	6.2%	1.2%	29.8%
	Model-Not-Cached	5.2%	0.8%	56.8%
Scenario 3	Model-Cached	8.0%	1.8%	22.7%
	Model-Not-Cached	15.1%	0.8%	51.6%

has lower frequency than PROP, then around rank 1000 the model has higher frequency than prop until the end of data. The frequency a video stay in a video can also be viewed in fig 7a, 7b, and 7c, where in the model 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 we can maintain same peer contribution while reducing number of replicas. We found that there are no much different between the first scenario, the second scenario and the third scenario in peer contribution to deliver a video. We found that in the all scenarios, the model gives lower replicas than PROP. This is because in the model, we found that not-cached events are higher than cached events, more specifically, the probability of utility function a requested video in model is lower than PROP. Therefore, in the model 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: The energy trade off 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. More numerical experiments for other zipf shape parameters.

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] 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.

[23] 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