Cost-aware Capacity Provisioning for Internet Video Streaming CDNs

Huajun He, Yang Zhao, Jinfu Wu and Ye Tian*

School of Computer Science and Technology,
University of Science and Technology of China, Hefei, Anhui 230027, China
Email: {jasons, qqtang, jfw}@mail.ustc.edu.cn, yetian@ustc.edu.cn

With the increasing popularity of the Internet video streaming services (e.g., YouTube and Netflix), content delivery networks (CDNs) are heavily used to stream video contents to users, and consume more and more power and bandwidths in recent years. In this paper, we investigate the problem of saving a video streaming CDN's operating expense, including both its energy cost and the traffic cost. From our measurement study on the CDN infrastructure of Youku, which is the largest video site in China, we find that there exists an inherent conflict between improving a video streaming CDN's energy efficiency for power saving, and maintaining the CDN's ISP-friendly server selection policy. To address this problem, we propose a cost-aware capacity provisioning algorithm, which dynamically plans the service capacities of a CDN's server clusters in numerous ISPs, and optimizes its overall operating cost regarding both the energy consumptions and the cross-ISP traffics. By using the workload derived from real-world measurement and applying actual power and bandwidth price parameters, we show with experiments that our approach can significantly reduce a video streaming CDN's overall operating cost, and avoid frequent server switches effectively. To our best knowledge, this work is the first one that identifies and resolves the inherent conflict between a CDN's energy efficiency and its ISP-friendly policy.

Keywords: Video Streaming CDN; Network Measurement; Capacity Provisioning

1. INTRODUCTION

Over the last decade, video becomes more and more prevalent on the Internet. Studies show that YouTube is accounted for 19.27%% and 13.19% of the Internet traffics in Europe and North America in 2014 [1], and it is estimated that by 2017, video will constitute 80 $\sim$ 90% of all the global IP traffic [2].

Perhaps the most important technological innovation that allows content providers to deliver videos to a global audience of viewers is the content delivery network (CDN) [3]. A CDN is a large distributed system that consists of hundreds of thousands of servers. Some CDNs, such as Google and Limelight, employ a few massive data centers that each comprises thousands of servers [4][5]. Such a CDN usually has its own autonomous system (AS), runs its "internal" backbone network that connects the data centers, and peers as many eyeball ISPs as possible. The other CDN design, with Akamai as example, deploys its server clusters at the "edges" of the Internet in many geographical locations and ISPs so as to be proximal to the clients. Usually an Akamai-like CDN does not possess its own data center, but places its clusters, which vary in size from a few to hundreds of servers, at the ISP or third-party data centers at as many locations and ISPs as possible. A detailed comparison between the two CDN designs can be found in [4].

Note that unlike peer-to-peer (P2P) video streaming networks [6], running a CDN for sustaining a large scaled video service is expensive. For example, it is estimated that Netflix has spent about 50 million US dollars on CDNs for video delivery in 2011 [7].

Many works for saving a CDN's operating expense are focused on reducing the clusters' or data centers' energy consumptions by "right-sizing" their service capacities [8][9][10]. The basic idea is that during the hours of lower workload, by scheduling the idle servers into the power-saving mode, the energy cost can be saved. For the CDNs employing massive data centers, such an energy-aware technique works well, as the energy cost constitutes a significant part of the data center's overall operating cost. However, when considering a CDN that adopts the "deep-into-ISPs" design (e.g., Akamai), more attentions should be paid to the cross-ISP video traffic that is incurred by the CDN. This is because for many ISPs, by placing the CDN clusters at their own data centers, ISP can avoid a significant part of its cross-ISP traffic and save its expense. When a CDN becomes

less friendly to the ISPs and incurs too much cross-ISP traffic, its price for leasing servers and racks at the ISP data centers will eventually rise so as to compensate for the ISP's increased traffic expense.

Unfortunately, there exists a conflict between simultaneously saving a CDN's energy cost and avoiding its cross-ISP traffic. On one hand, for saving the energy cost, we can keep a minimum number of the servers as long as the CDN's service level agreement (SLA) is met. On the other hand, to avoid the cross-ISP video deliveries, we need to minimize the chance that the workload demand from an ISP exceeds the service capacity provisioned by the clusters in that ISP, and has to be accommodated by servers from other ISPs. Since workload fluctuates over time, the more service capacity we have provisioned, the less likely we will incur cross-ISP video deliveries. Apparently, for reducing the overall cost, the inherent conflict between the energy efficiency and the ISP-friendliness requires people to carefully plan the service capacities of the CDN clusters.

In this paper, we consider both the energy and the cross-ISP traffic cost for a video streaming CDN. We present a capacity provisioning algorithm that is cost-aware by dynamically planning service capacities of the CDN clusters in numerous ISPs. Our work is motivated by the measurement study on the CDN infrastructure of Youku [11], which is the largest video site in China. Using workload derived from real-world traces and applying actual bandwidth and power price parameters, we show with experiments that our solution can balance a CDN's energy and bandwidth expenses, and significantly reduces its overall operating cost. In addition, our approach avoids unnecessary switches that toggling servers into and out of the power-saving mode, therefore can be practically applied on today's video streaming CDNs.

The contributions of this paper are in two-fold:

- **Measurement and analysis**: We carry out an extensive measurement study on Youku's CDN, and present an insightful analysis on its server selection behaviors. Our observations include: 1) Youku adopts a "deep-into-ISPs" design in its CDN network; 2) Youku generally follows an ISP-friendly server selection policy; and 3) When balancing excessive workloads from the ISPs, Youku violates the ISP-friendly policy, especially in the small ISPs where it has insufficient service capacities. Based on the observations, we identify that there exists an inherent conflict between improving a CDN's energy efficiency and maintaining its ISP-friendly server selection policy.
- **Solution and evaluation**: Based on the insights from the measurement study. We formulate the problem of saving a CDN's overall operating cost, including both the energy cost as well as the cross-ISP traffic cost, under the precondition of meeting the CDN's SLA. We show that the problem is a convex optimization problem, and can be solved greedily. We propose a practical algorithm that dynamically allocates the CDN's service capacity in numerous server clusters for saving its operating cost. Experimental results suggest that our solution can save a CDN's operating cost considerably, and avoid unnecessary frequent server switches. We also discuss the influence of the ISP-CDN business relationship on the system's energy efficiency.

To our best knowledge, this work is the first one that identifies and resolves the inherent conflict between energy efficiency and ISP-friendliness of a CDN with a "deep-into-ISPs" design approach. The remainder part of this paper is organized as the follows. Section 2 discusses the related works; in Section 3, we present our measurement study on Youku's CDN, analyze its server selection policy, and discuss the conflict between the CDN's energy efficiency and its ISP-friendly policy that motivates this work; we formulate the CDN's cost saving problem and present our solution in Section 4; Section 5 evaluates our proposed cost-aware capacity provisioning algorithm, and finally we conclude this paper in Section 6.

2. RELATED WORK

As an early example of a Platform-as-a-Service (PaaS) cloud, CDNs become more and more important in delivering Web contents, applications, and streaming media on today's Internet. Many works analyze and evaluate CDN from measurement-orient approaches. Huang *et al.* [4] study presentative CDNs by aggressively discovering their footprints, and point out that there exist two philosophies in a CDN's architecture design. Adhikari *et al.* [5] address the interplays between the CDN of Google, which carries YouTube's traffic, and a tier-1 ISP, and show that YouTube applied a location-agnostic load-balancing strategy when distributing the traffic, during the time when the service was initially integrated into Google's platform. Torres *et al.* [12] also focus on YouTube's server selection strategies, and show that the server-client RTT plays the most important role in YouTube's server selections. Khare *et al.* [13] discuss the potential conflicts between a CDN's server selection strategy and an ISP's traffic engineering policy, and present solutions for minimizing the CDN's traffic payment. Our work differs from these previous works in that we are the first to study the server selection policy of a large scale real-world CDN that adopts the "deep-into-ISPs" architecture from a measurement-oriented approach, and examines the scenarios when the CDN follows and violates the ISP-friendly policy.

As power cost becomes a major component of a data center's operating expense, many works are focused on reducing the energy cost of a cloud or a CDN in

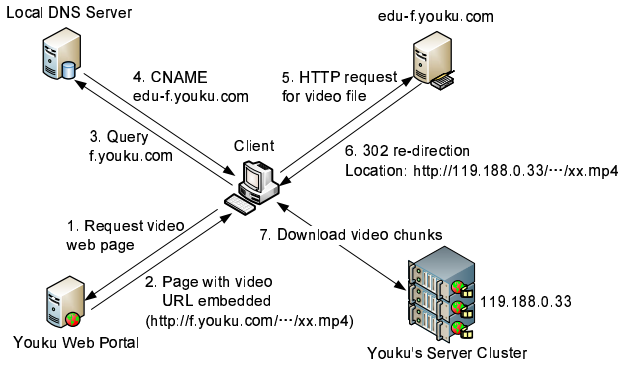


FIGURE 1. Demonstration of Youku's server selection steps.

recent years. Lin *et al.* [8] present an online algorithm that jointly optimize the servers' energy costs and their switching costs by dynamical "right-sizing" the server cluster. Mathew *et al.* [9] study the power saving problem under a CDN context, and present algorithms at the levels of the local and global load-balancing of the CDN. Palasamudram *et al.* [14] propose to use the distributed USP within a data center for saving a CDN's monetary power cost instead of saving its power consumptions. Tchernykh *et al.* [10] propose to balance the VoIP workload in a distributed computer environment with an adaptive algorithm. Our work differs from these previous works in that we are the first to identify the inherent conflict between a CDN's ISP-friendly server selection policy and its energy efficiency based on real-world measurement study, and present practical solution by jointly optimizing the CDN's energy cost as well as the cross-ISP traffic cost.

3. MEASUREMENT AND MOTIVATION

In this section, we carry out a measurement study on the CDN infrastructure of Youku, which is the largest video site in China and the second largest site in the world after YouTube. In particular, we present an insightful analysis on its server selection behaviors. We find that Youku follows an ISP-friendly policy in general, but the CDN also violates the policy when balancing the excessive workloads from the ISPs. We show that for a Youku-like CDN, there is an inherent conflict between the energy-aware capacity provisioning, which right-sizes the CDN's service capacity for saving its energy expense, and the ISP-friendly server selection policy that seeks to avoid the cross-ISP video traffic.

We describe our measurement methodology and unveil the CDN design in Section 3.1; the CDN's server selection policy is analyzed in Section 3.2; in Section 3.3, we characterize the energy-aware capacity provisioning techniques that aim to improve the system's energy efficiency; we identify the inherent conflict between a CDN's energy efficiency and its ISP-friendly policy in Section 3.4, and discuss its implications.

3.1. Measurement Methodology and CDN Design

There are two objectives in our measurement study on Youku's CDN: 1) to unveil the CDN's system design; and 2) to collect a rich set of the server selection samples from various geographical locations and ISPs for enabling a policy analysis.

To achieve the two objectives, we exploit Youku's built-in server selection mechanism. Figure 1 demonstrates the major steps in this mechanism: When a client's web browser parses a web page in which the video is embedded, a static video URL like "`http://f.youku.com/...`" is retrieved (step 1-2). The client then queries its local DNS server for the name `f.youku.com`, and gets a CNAME reply like `edu-f.youku.com` (step 3-4)[2]. After the DNS resolution, the client sends out an HTTP GET request to the host binding the CNAME (step 5). However, the host doesn't have the requested video, but replies with an HTTP 302 re-direction message, which contains the IP address of the content server that actually hosts the video file (step 6). Finally, the user client follows the re-direction and downloads video chunks from the content server (step 7).

With the understanding of Youku's server selection mechanism, we can see that if we emulate a client's video request through an HTTP proxy, as the proxy queries its local DNS server and uses its own IP address when forwarding the request, Youku will select a server that is "optimal" for the proxy, and returns it's IP address in the 302 re-direction message via the proxy to our measurement agent. In other words, we can collect a *sample* on Youku's server selection decisions from the proxy. Furthermore, by probing through many HTTP proxies distributed on a wide range of geographical locations and ISP networks, a large number of the samples can be harvested for analyzing the CDN's server selection policy.

Based on the methodologies above described, we carry out a measurement study on Youku's CDN. To filter out the influence of the content availability on CDN's server selection decisions, in each probe we only request the "headline" video, that is, the video posted at the headline position on Youku's web portal. The video could be a breaking news, an important social event, or any other content that Youku wishes to promote. As it is under promotion, the CDN typically caches the video to the maximum extent.

We carried out the measurement from Sep. 14, 2011 to Dec. 5, 2011, which lasted 83 days. In each day of the measurement, we collect more than one hundred HTTP

[2]Youku's authoritative DNS server resolves the DSN query with different CNAMEs, and besides `edu-f.youku.com`, we have observed four other CNAMEs when probing from different ISPs and locations

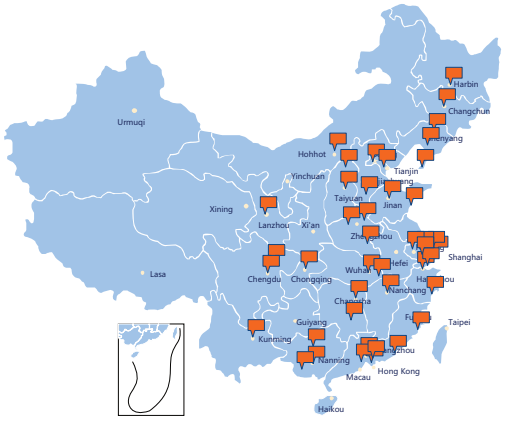


FIGURE 2. Geographic distribution of Youku's server clusters.

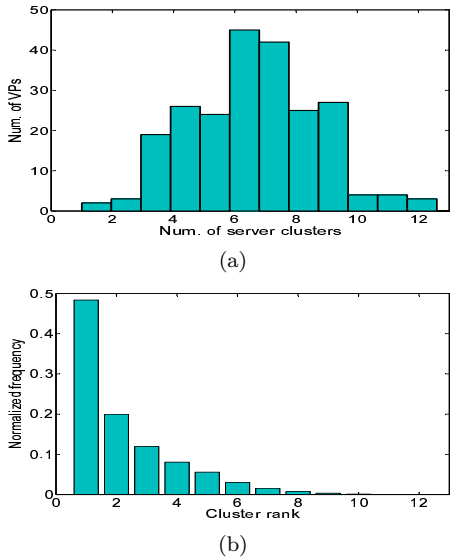


FIGURE 3. (a) Histogram of server clusters selected for VPs; (b) Normalized cluster selection frequencies across all the VPs.

proxies from various resources (e.g., websites, forums and blogs)[3], and use them to probe Youku's CDN. A total number of 2,997 distinct HTTP proxies were employed in our measurement. By using the Cymru IP to AS mapping tool [15] and applying the geolocating technique in [16], we find that the proxies are widely distributed in 267 cities and 55 ASes, covering nearly all the cities and ISPs in China. By probing from these proxies, we have found 759 distinct CDN server IP addresses in 43 cities and 17 ASes. We further group the server addresses that are in a same city and a same AS into a cluster, which we refer to as a CDN's *server cluster*. We have found a total number of 54 clusters and show their geographical distribution in Figure 2.

From Figure 2, one can see that Youku adopts a CDN degisn that deploys its server clusters at dozens of cities in China. In fact, unlike YouTube, which employs a few massive data centers [5], Youku does not possess any data center, but places its servers at the ISP or third-party data centers at as many locations and ISP networks as possible. Note that such a "deep-into-ISPs" design is representative, as it is found that many large scale CDNs in China and in the world, such as ChinaCache and Akamai, construct their networks in a similar way [4].

3.2. Server Selection Policy Analysis

We investigate Youku's server selection policy based on the samples collected from the proxies. Similar to the aggregation of the server clusters, we group the proxies that are in a same city and a same AS into a cluster, which we refer to as a measurement vantage point (or a *VP* for short). By filtering out the VPs with less than 15 samples, we have grouped 229 VPs, which are distributed in 142 cities and 32 ASes in China[4]. A total number of 24,533 server selection samples were collected from these VPs.

3.2.1. Server selection characteristics

We first examine the server clusters that were selected by Youku for the VPs. Figure 3 (a) presents the histogram of numbers of the clusters selected for the VPs in the 83-day measurement. From the figure one can see that there are considerable *dynamics* in Youku's server selections, as for most VPs, more than one clusters were selected over time.

We then focus on the selection frequencies. For a VP v, we compute the selection frequency $f_v(k)$ of its k^{th} most selected cluster c_k^v as the ratio of the times that the cluster got selected. For all the VPs under study, we compute a *normalized cluster selection frequency* for their k^{th} most selected clusters as

$$F(k) = \frac{\sum_v f_v(k)}{N},$$

where $k = 1, 2, \cdots$, and N is the total number of the VPs under study.

Figure 3 (b) presents the normalized cluster selection frequencies. From the figure one can see that although multiple server clusters on the CDN network were selected for a VP over time, Youku did not distribute the workload among them evenly, but route most requests to only a few clusters. In addition, we can see that the selection frequency decreases rapidly as the rank increases.

[3]These proxies were provided by individuals and organizations for different purposes, such as bypassing local ISP's control policies, or enabling anonymous web surfing, etc, examples can be found at http://www.cnproxy.com/.

[4]The reason that some proxies produce few samples is because they are functional for only a few hours, so that we cannot have sufficient samples from them.

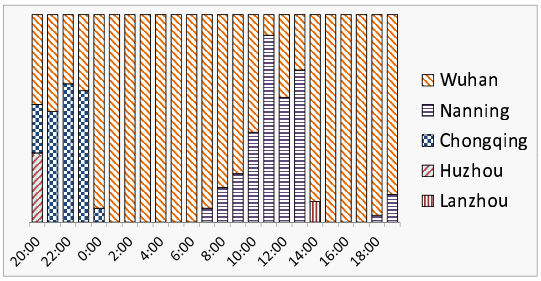


FIGURE 4. Clusters selected in each hour.

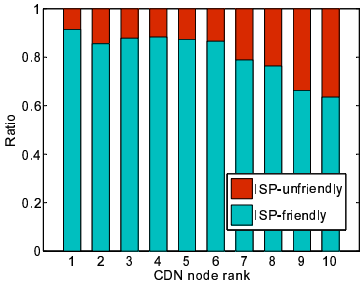


FIGURE 5. ISP-friendliness in Youku's server selections.

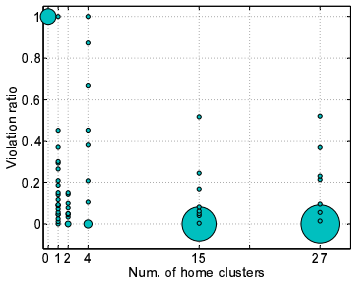


FIGURE 6. Correlation between violation ratio and home clusters.

TABLE 1. Ratios of the clusters being selected in 24 hours

Wuhan	Nanning	Chongqing	Huzhou	Lanzhou
0.748	0.143	0.090	0.014	0.004

3.2.2. Understanding server selection dynamics

We seek to understand the server selection dynamics exhibited by Youku through an experiment. In our experiment, we employ the methodology described in Section 3.1 to make frequent video requests to Youku from a measurement agent that is located at a fixed location and ISP network. We probed Youku in every two minutes, and collected the video server IP addresses that were returned in the 302 re-direction responses. The experiment lasted 24 hours from 20:00 Dec. 19, 2012 to 19:59 Dec. 20, 2012.

We have observed 15 distinct content server IP addresses in the 24-hour measurement. These addresses are coming from five different clusters in five different cities. Table 1 presents the ratios of the clusters being selected, from which we can see that most requests were directed to the Wuhan cluster. In addition, we find that all the CDN clusters are in the same ISP of the measurement agent.

We then analyze the server selections on an hourly basis. Figure 4 presents the numbers of the times that the clusters were selected in each hour. From the figure we can see that Youku exhibited different degrees of dynamics in different hours: in the periods of 1:00-6:00 and 15:00-17:00, Youku consistently selected the most preferred Wuhan cluster; however, during the other hours, the CDN made dynamic decisions by switching among two or three clusters. Note that the hours with the greatest server selection dynamics (i.e., 8:00-13:00 and 17:00-23:00) are in fact the times that Youku attracts the most visits in a day, we can see that during these times, Youku has to balance the CDN workload among several clusters, which leads to the observed dynamics.

3.2.3. ISP-friendliness

For investigating the ISP-friendliness in Youku's server selections, we employ an indicator function $I(\cdot)$ for labeling the relationship between a VP and a CDN's server cluster: for a VP v, if its k^{th} most selected cluster c_k^v is in the same ISP of v, then c_k^v is considered as a *home cluster* to v, and we have $I(v, c_k^v) = 1$; otherwise, $I(v, c_k^v) = 0$, and c_k^v is referred to as a *foreign cluster*. With the indicator function, for all the VPs under study, we can define an aggregated *ISP-friendly ratio* for their k^{th} most selected server clusters as

$$R(k) = \frac{\sum_v I(v, c_k^v) \cdot f_v(k)}{\sum_v f_v(k)}$$

Figure 5 presents the ISP-friendly ratios of the server clusters selected by Youku for all the 229 VPs, and we also present the ISP-unfriendly ratios as $1 - R(k)$. From the figure one can see that in most cases, the CDN follows an ISP-friendly policy by selecting home clusters. To better support this claim, we examine each VP's most preferred cluster: among the 229 VPs, only 20 of them select the foreign clusters as their most preferred clusters, and further examination shows that 15 of them are indeed in small stub ASes, in which Youku does not place any video servers. In other words, there are no home clusters on Youku's CDN for these VPs. An other observation from the figure is that the ISP-friendly ratio decreases as the cluster rank increases. Recall that Youku makes dynamic server selections for balancing the workloads during the busy hours, we can see that when a VP's home clusters are all overloaded, Youku will select the foreign clusters for accommodating the excessive workload and violate its ISP-friendly policy.

3.2.4. An ISP view of ISP-friendliness violation

Although it is observed that Youku follows an ISP-friendly policy in its server selections, however, in 6.2% of the samples, we find that the policy is indeed violated. Here we seek to study such a phenomenon from an ISP perspective. For each VP in our study, we define its *violation ratio* as the times that a foreign cluster was selected divided by the total number of the samples collected from the VP, and for each VP, we correlate its violation ratio with the number of its home clusters available for selection on the CDN network.

Figure 6 presents the correlation using a bubble graph, where each bubble on the figure represents a

group of the VPs in a same ISP. For each bubble, x-axis indicates how many home clusters available for the VPs in the group, y-axis shows the violation ratio observed from these VPs, and the bubble size reflects the group size. Two observations could be made from the figure: First, for the ISPs with many server clusters, nearly no violation is observed. For example, 87 of the 94 VPs in the ISP of ChinaNet (corresponding to the big bubble at (27, 0)) and 70 of the 78 VPs in the ISP of China169 (the big bubble at (15, 0)) have a zero violation ratio. Second, for some small ISPs with few clusters available for selection, there are considerable violations. The observations suggest that Youku may have insufficient service capacities in these small ISPs, which forces the CDN to violate its ISP-friendly policy by selecting foreign clusters for the clients in these ISPs.

We summarize our observation as the following:

1. We find that Youku employs the "deep-into-ISPs" design in its CDN network by deploying its server clusters in as many as 43 cities and 17 ASes in China;
2. Youku generally follows an ISP-friendly policy by preferring the server clusters that are in a same ISP for a client;
3. When handling the excessive workloads from the ISPs, the CDN violates the ISP-friendly policy, especially in the small ISPs in which Youku does not have sufficient service capacity.

3.2.5. Discussion

Through our measurement study, we have observed that Youku follows an ISP-friendly policy in selecting clients' server clusters. We also find that other large scale video streaming CDNs in China, like Tudou and Sina video, are employing similar policies. The reasons are in two-fold: on one hand, the ISPs have incentive to bring the CDN at "home" by hosting the CDN's server clusters at their data centers with relative lower price, as long as the CDN follows the ISP-friendly sever selection policy. On the other hand, when the CDN becomes less friendly to the ISPs, the ISPs can migrate the increased cross-ISP traffic cost to the CDN. For example, the ISP can directly charge higher prices on the CDN for renting servers and racks in their data centers; or the ISPs can throttle the CDN applications, which causes the CDN to have more failures in meeting the SLA requirement, and eventually lose its customers and profits. In either cases, CDN actually pays for their cross-ISP video deliveries. Therefore, it is very necessary for a CDN like Youku to select video servers in an ISP-friendly way for saving its operating cost.

3.3. Energy-aware Capacity Provisioning

As a large scale Internet infrastructure, a large portion of CDN's operating cost is on energy. In recent years, many works [8][9][10] were focused on improving a CDN or a cloud's energy efficiency with techniques that dynamically "right-size" the system's service capacity. Generally, an energy-aware capacity provisioning technique is based on two facts: First, for many Internet services, the workload patterns are periodic, and exhibit great fluctuations, for example, the average workload for a large cloud service could be only 40% of the peak load; Second, current servers are far from energy-proportional, for instance, an idle server consumes up to 60% of the energy consumed by a full-loaded server [17]. Therefore, during the hours with lower workloads, by scheduling idle servers into the power-saving mode, considerable energy cost can be saved.

A typical energy-aware service capacity provisioning algorithm works as the following: Given a period of interest $t \in \{1, ..., T\}$, the mean workload at interval t is denoted as $\mathbb{E}[x_t]$, the algorithm determines n_t, which is the service capacity provisioned during interval t, so that n_t is larger than the actual workload $\mathbb{E}[x_t]$, and the overall energy cost is minimized. Sometimes, it is also desirable to reduce the *server switches*, that is, to reduce the times that servers are toggled into and out of the power-saving mode.

3.4. Implication and Motivation

From the above analysis, we can see that for a Youku-like CDN, or a CDN following the "deep-into-ISPs" design, there are two major components in its operating cost: 1) the traffic cost for cross-ISP content deliveries, and 2) the energy cost for running its server clusters. Note that *it is conflicting to reduce the both costs simultaneously*. To avoid the cross-ISP traffic, for each ISP, we need to reduce the chance that workload from the ISP exceeds the service capacity provisioned by the clusters in that ISP. This can be achieved by allocating some surplus capacity that is larger than the predicted workload. However, the power cost also rises with the over-provisioned capacity.

On the other hand, to reduce the energy cost, we need to right-size the active servers in each clusters based on the predicted workload. However, as workload fluctuates, in cases that an ISP's actual workload exceeds the planned service capacity, the CDN has to violate the ISP-friendly policy by selecting servers from foreign clusters, and incurs cross-ISP traffic cost.

4. COST-AWARE CDN CAPACITY PROVISIONING

From the above section, we can see that for a video streaming CDN like Youku, there exists an inherent conflict between improving the CDN's energy efficiency and preserving its ISP-friendly server selection policy. In this section, we present a formal description of the problem, and propose a cost-aware capacity provisioning algorithm that dynamically plans the

TABLE 2. Notations used in problem and algorithm description

Notation	Meaning
$x_{i,t}$	CDN workload from ISP_i in interval t;
$f_{i,t}(x)$	Distribution of the workload of $x_{i,t}$;
$n_{i,t}$	Number of active servers planned in ISP_i in interval t;
$c_{ISP}(n_{i,t})$	CDN's operating cost in ISP_i in interval t as defined in Equation (1);
th_{SLA}	SLA in terms of the video service availability;
X_t	CDN workload from all the ISPs in interval t;
$f_{G,t}(x)$	Distribution of the workload of X_t;
N_t	Total number of active servers planned in interval t;
$C(n_{1,t},\cdots)$	CDN's operating cost in all the ISPs in interval t as defined in Equation (2);
$\bar{N}_t$	Lower bound of the global service capacity for meeting SLA in interval t;
$n^*_{i,t}$	Number of active servers for ISP_i in interval t by solving Equation (3);
$n^{(j)}_{i,t}$	Temporarily allocated service capacity for ISP_i during the j^{th} iteration in Algorithm 1;

CDN's service capacities in its clusters for saving the overall operating cost. The notations used in describing the problem and our proposed algorithm are listed in Table 2.

4.1. The CDN Capacity Provisioning Problem

4.1.1. The network model

We consider a Youku-like CDN that employs the "deep-into-ISPs" design. In particular, the CDN is composed of a number of server clusters, which are partitioned in K different ISPs. An ISP can have multiple clusters, but for simplicity we assume that a CDN cluster can only be in one single ISP, that is, no multi-homing clusters. Each ISP has a number of point-of-presences (PoPs), where a PoP represents a group of clients that impose a large volume of aggregated workload upon the CDN.

The CDN allocates its capacity in time intervals. An interval could be in several minutes (for example, 10 minutes). In interval t, we denote the workload imposed on the CDN from all the PoPs in an ISP, say ISP_i, as $x_{i,t}$. Note that $x_{i,t}$ can be viewed as a random variable, which fluctuates over time. At the beginning of each interval, the CDN plans the service capacities for its server clusters; more specifically, for ISP_i, the CDN determines $n_{i,t}$, the number of the servers that are scheduled to be alive in all its clusters in ISP_i, to provide the video streaming service.

The CDN under study follows an ISP-friendly server selection policy as observed in Section 3.2. That is, the CDN always selects a server from a home cluster for a client, as long as such a server is alive and has spare capacity. The CDN violates the policy only when all the servers in the client's home clusters are busy, and in that case, the CDN will deliver the video chunks from a server with spare capacity in a foreign cluster, and incurs cross-ISP traffic.

4.1.2. Cost function

With the assistance of the network model, we formulate a CDN's overall operating cost. For one particular ISP, say ISP_i, it is easy to see that the energy cost of all its server clusters during interval t is $c_1 \times n_{i,t}$, where c_1 is the power cost for running one active server per interval. For the CDN's cross-ISP traffic cost, note that cross-ISP video delivery happens only when the workload demand exceeds the service capacity in an ISP. By assuming one server providing one unit service capacity, the incurred cross-ISP traffic cost can be expressed as $c_2 \times \int_{n_{i,t}}^{\infty} x \times f_{i,t}(x)dx$, where $f_{i,t}(x)$ is the distribution of the workload $x_{i,t}$ during interval t, $\int_{n_{i,t}}^{\infty} x \times f_{i,t}(x)dx$ is the part of the workload that exceeds the service capacity in ISP_i, and c_2 is the cost for delivering one unit workload traffic in the cross-ISP way. Finally, the CDN's total cost for operating the clusters in ISP_i during interval t can be expressed as

$$c_{ISP}(n_{i,t}) = c_1 \times n_{i,t} + c_2 \times \int_{n_{i,t}}^{\infty} x \times f_{i,t}(x)dx \quad (1)$$

From the above formulation, we can see that the service capacity in each ISP need to be carefully planned: by increasing $n_{i,t}$, the cross-ISP traffic cost will be reduced, but at a higher energy cost; while decreasing $n_{i,t}$ for energy-saving will lead to a higher cross-ISP traffic cost.

4.1.3. Problem formulation

In our CDN capacity provisioning problem, we seek to plan and allocate the CDN's service capacity in each ISP, by determining $n_{i,t}$, to achieve the following objectives:

- **Meeting SLA**: The CDN should meet its service level agreement (SLA). A typical SLA for a CDN service is its availability [9]. In this work, we consider SLA as the availability of the video service. That is, in any interval t, SLA requires that

  $$\Pr[X_t < N_t] > th_{SLA}, \quad 0 < th_{SLA} \leq 1,$$

  where $X_t = \sum_{i=1}^{K} x_{i,t}$ is the overall workload from all the ISPs in interval t, $N_t = \sum_{i=1}^{K} n_{i,t}$ is the CDN's global service capacity, and th_{SLA} is the SLA requirement in terms of the service availability.
- **Saving the operating cost**: The CDN should be cost effective, that is, its overall cost for operating the server clusters in all the ISPs, which can be expressed as

  $$C(n_{1,t},\cdots,n_{K,t}) = \sum_{i=1}^{K} c_{ISP}(n_{i,t}), \quad (2)$$

should be minimized. Note that $c_{ISP}(n_{i,t})$ contains both the energy cost and the cross-ISP traffic cost as indicated in Equation (1).

Besides the two objectives, it is also expected that there are limited *server switches*, so as to reduce the wear-and-tear effects on the servers when switching them into and out of the power-saving mode.

4.2. Capacity Provisioning Algorithm

In this section, we propose our solution for the above described CDN capacity provisioning problem. The solution works in two steps: in the first step, we determine $\bar{N}_t$, the lower bound of the global service capacity, for meeting the CDN's SLA requirement; in the second step, we decide $n_{i,t}$, which is the number of the live servers in each ISP, for saving the CDN's overall operating cost. Obviously, $\sum_{i=1}^{K} n_{i,t} \geq \bar{N}_t$.

As in other capacity provisioning algorithms (e.g., [8][9][18]), our approach relies on predicting of the future workload. In particular, at the beginning of each interval, say interval t, the CDN scheduler predicts $f_{i,t}(x)$, the distribution of the workload from ISP_i, for each of the ISPs in which the CDN has deployed its clusters; and the scheduler also predicts the CDN's global workload distribution as $f_{G,t}(x)$.

We note that the prediction is feasible because of two reasons: First, recent studies show that by applying regression-based techniques and by employing sufficient history data, it is possible to accurately predict a video streaming service's average workload in a median-length interval like 10 minutes [19][18]. Second, studies show that the instantaneous workload can be well approximated as modified Poisson [20] or Gaussian [21]; in Appendix A, we also observe that Youku's instantaneous workload can be approximated as Gaussian, and similar to web traffic [22], we observe that the workload variance scales linearly with the mean workload. By combining these results, we can see that it is possible to predict the distribution of a CDN's global or cluster-wise instantaneous workloads with high accuracies.

4.2.1. Estimating global capacity lower bound

As it is required that $\Pr[X_t < N_t] > th_{SLA}$ for meeting the CDN's SLA, with the prediction of the global workload distribution $f_{G,t}(x)$, it is easy to see that the lower bound of the global service capacity $\bar{N}_t$ can be obtained by solving the following problem

$$\int_0^{\bar{N}_t} f_{G,t}(x)dx = th_{SLA}$$

4.2.2. Optimal ISP capacity allocation for cost-saving

For minimizing the CDN's overall operating cost as in Equation (2), we search for the optimal service capacities $\{n_{i,t}\}_{i=1}^{K}$ that are allocated in all the ISPs. The problem can be formulated as

$$\begin{array}{ll} Minimize & C(n_{1,t},\cdots,n_{K,t}) = \sum_{i=1}^{K} c_{ISP}(n_{i,t}) \\ s.t. & \sum_{i=1}^{K} n_{i,t} \geq \bar{N}_t, \quad i = 1,\ldots,K \end{array}, \tag{3}$$

where $c_{ISP}(n_{i,t})$ is the CDN's operating cost in ISP_i as expressed in Equation (1), and $\bar{N}_t$ is the lower bound of the CDN's global capacity.

The problem is generally difficult to solve under arbitrary workload, however, for many well-known workload models such as exponential [20] and Gaussian [21], we find that $C(n_{1,t},\cdots,n_{K,t})$ is convex, which makes the problem a convex optimization problem [23], whose global optimal solution can be obtained by greedily reducing the objective function until convergence. We use $\{n_{i,t}^*\}_{i=1}^{K}$ to denote the optimal solution for the convex optimization problem in Equation (3).

4.2.3. Allocating ISP capacities with reduced server switches

The optimal solution for Equation (3) can be used to determine the initial capacity of a CDN. However, during the regular operation, it is not applicable. This is because the algorithm is unaware of the server switches, therefore will frequently toggling servers into and out of the power-saving mode in consecutive intervals. To address this problem, we propose a heuristic algorithm that balances the cost saving with server switches.

After obtaining the global capacity lower bound $\bar{N}_t$, our proposed heuristic works in three phases iteratively to determine the capacity in each ISP. During each iteration, say iteration j ($j = 0, 1, 2, \cdots$), the algorithm updates the temporarily allocated service capacity $n_{i,t}^{(j)}$ for ISP_i. Following is the algorithm details:

1. **Phase I: Initialization**: In the initial iteration, we let

$$n_{i,t}^{(0)} = \begin{cases} n_{i,t-1}, & \text{If} \quad n_{i,t-1} \geq \lceil \mathbb{E}[x_{i,t}] \rceil \\ \lceil \mathbb{E}[x_{i,t}] \rceil, & \text{If} \quad n_{i,t-1} < \lceil \mathbb{E}[x_{i,t}] \rceil \end{cases},$$

   where $i = 1, 2, ..., K$, and $\mathbb{E}[x_{i,t}]$ is the predicted mean workload in interval t. That is, the initial planned capacity should be no less than the predicted mean workload of the incoming interval, and can be the capacity in the previous interval when it is greater than $\mathbb{E}[x_{i,t}]$.
2. **Phase II: Saving CDN cost**: In the jth iteration ($j \geq 0$), for ISP_i, compare the current capacity $n_{i,t}^{(j)}$ with the optimal solution $n_{i,t}^*$ for the problem in Equation (3):
   - If $n_{i,t}^{(j)} < n_{i,t}^*$, and the cost reduction by adding one more server, $\Delta c_{i,t}^{+} = c_{ISP}(n_{i,t}^{(j)}) - c_{ISP}(n_{i,t}^{(j)} + 1)$, is larger than a threshold θ, we let $n_{i,t}^{(j+1)} = n_{i,t}^{(j)} + 1$ for ISP_i;

- Similarly, if $n_{i,t}^{(j)} > n_{i,t}^*$, and the cost reduction by removing one server, $\Delta c_{i,t}^- = c_{ISP}(n_{i,t}^{(j)}) - c_{ISP}(n_{i,t}^{(j)} - 1)$, is larger than the threshold θ, we let $n_{i,t}^{(j+1)} = n_{i,t}^{(j)} - 1$.

For each ISP, repeat until no server can be added or removed any more.

3. **Phase III: Meeting SLA**: In this phase, compare the current global capacity $\sum_{i=1}^{K} n_{i,t}^{(j)}$ with the global capacity lower bound $\bar{N}_t$, if $\sum_{i=1}^{K} n_{i,t}^{(j)} < \bar{N}_t$, which means more servers should be alive for meeting the SLA requirement, the heuristic finds the ISP that has the minimum cost increase (or the maximum cost reduction) by adding one more server, that is, find

$$s = \arg \max_{i=1,\ldots,K} \{\Delta c_{i,t}^+\}$$

and add one more server for ISP_s by letting $n_{s,t}^{(j+1)} = n_{s,t}^{(j)} + 1$; repeat until $\sum_{i=1}^{K} n_{i,t}^{(j)} \geq \bar{N}_t$.

Note that in Phase II, we compare the expected cost reduction by adding or removing one server with threshold θ, and actually add or remove a server only when it is worthwhile. In fact, we can view θ as the wear-and-tear cost of a server for making a server switch, and the algorithm avoids unnecessary switches by hibernating or awakening a server only when the benefit is significant enough. A formal description of the algorithm can be found in Algorithm 1.

Algorithm 1 CDN capacity provisioning algorithm

$n_{i,t}^{(0)} \leftarrow \begin{cases} n_{i,t-1}, & n_{i,t-1} \geq \lceil \mathbb{E}[x_{i,t}] \rceil \\ \lceil \mathbb{E}[x_{i,t}] \rceil, & n_{i,t-1} < \lceil \mathbb{E}[x_{i,t}] \rceil \end{cases}$; ▷ Phase I
$j \leftarrow 0$;
repeat ▷ Phase II
  for $i \leftarrow 1, K$ **do**
    if $n_{i,t}^{(j)} < n_{i,t}^*$ **and** $\Delta c_{i,t}^+ \geq \theta$ **then**
      $n_{i,t}^{(j+1)} \leftarrow n_{i,t}^{(j)} + 1$;
    else
      if $n_{i,t}^{(j)} > n_{i,t}^*$ **and** $\Delta c_{i,t}^- \geq \theta$ **then**
        $n_{i,t}^{(j+1)} \leftarrow n_{i,t}^{(j)} - 1$;
      end if
    end if
  end for
  $j \leftarrow j + 1$;
until no server can be added or removed
while $\sum_{i=1}^{K} n_{i,t}^{(j)} < \bar{N}_t$ **do** ▷ Phase III
  $s \leftarrow \arg\max_{i=1,\ldots,K} \{\Delta c_{i,t}^+\}$;
  $n_{s,t}^{(j+1)} \leftarrow n_{s,t}^{(j)} + 1$; $j \leftarrow j + 1$;
end while
$n_{i,t} \leftarrow n_{i,t}^{(j)}$;

5. PERFORMANCE EVALUATION

In this section, we evaluate our proposed CDN capacity provisioning algorithm, and compare it with other schemes through simulation experiments. We use the workload derived from real-world measurement and the actual bandwidth and power price parameters in our simulation.

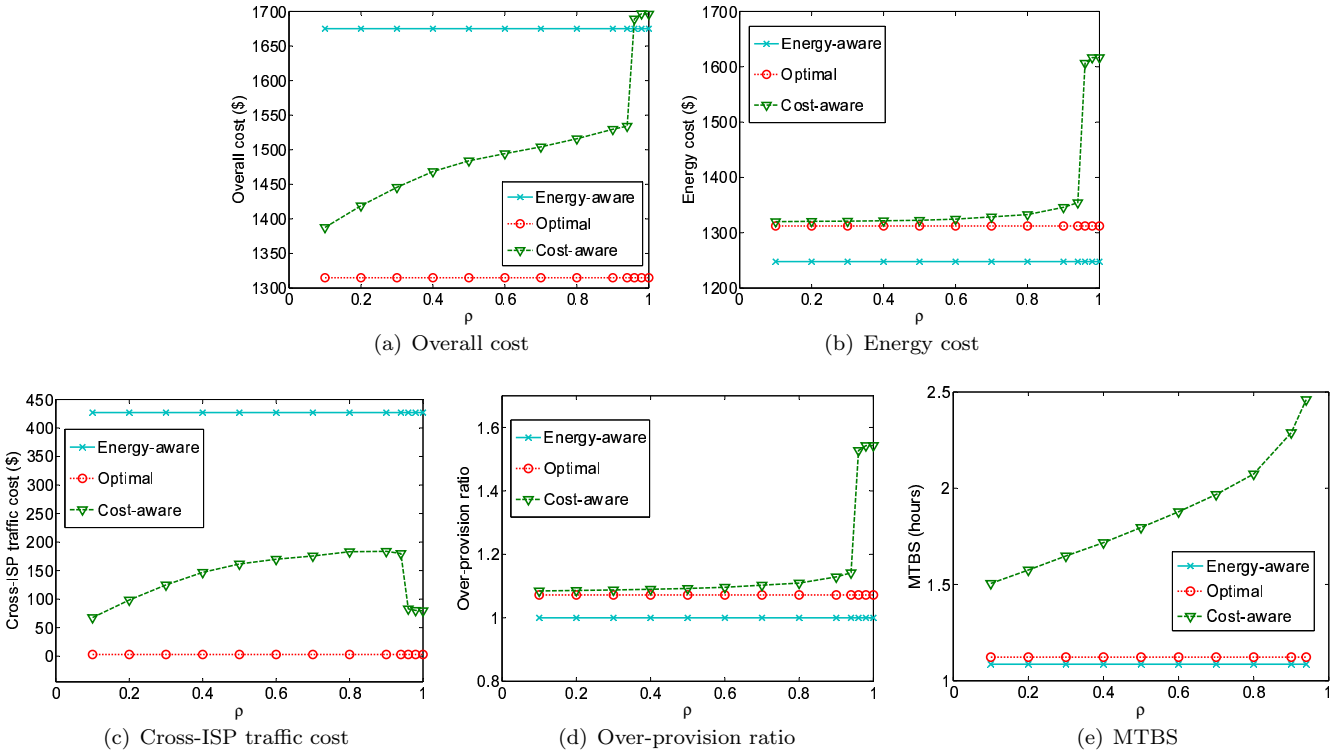
5.1. Experiment Setup

We emulate a CDN that deploys its video server clusters in 14 different ISPs. ISPs vary in size, and we consult the measurement result in Section 3.2 by assigning various numbers of PoPs in different ISPs, where the largest ISP has 88 PoPs and the smallest has only one PoP. Each PoP imposes a certain amount of video requests on the CDN per second, and the CDN follows the ISP-friendly policy by directing a request to an active server in the same ISP as much as possible. However, when all the servers are overloaded, the request will be handled by a server from a different ISP, and incurs some cross-ISP traffic cost.

The CDN plans its service capacity in intervals. An interval lasts for 10 minutes in our simulation. At the beginning of an interval, the CDN predicts the workload of the incoming interval, and applies one of the following capacity provisioning schemes to decide the number of the active servers for each ISP:

- The *energy-aware* capacity provisioning: In this approach, during each interval, after obtaining the lower bound $\bar{N}_t$ of the global service capacity, the CDN allocates a capacity for each ISP that is proportional to the predicted workload from the ISP, that is, for ISP_i, $n_{i,t} = \lceil \frac{\mathbb{E}[x_{i,t}]}{\mathbb{E}[X_t]} \times \bar{N}_t \rceil$. Since only the lower bound capacity is provisioned, the CDN's energy expense is minimized. Note that such a scheme consumes even less power than existing energy-aware solutions (e.g, [8] and [9]) as we do not consider reducing the server switches here.
- The *optimal* capacity provisioning: In this approach, the CDN allocates the service capacities for the ISPs by solving the convex optimization problem in Equation (3) in each interval.
- The heuristic *cost-aware* capacity provisioning: In this scheme, we use the optimal solution of Equation (3) for the initial interval, then apply the heuristic algorithm described in Section 4.2.3 to plan the CDN capacity in each ISP in the subsequent intervals. Note that the heuristic employs a threshold θ. Since it is the wear-and-tear cost of a server switch, θ can be expressed as ρ times of the unit chunk energy cost by letting $\theta = \rho \times c_1$, we consider various values of the parameter ρ in our simulation.

For the CDNs employing different capacity provision-



(a) Overall cost (b) Energy cost

(c) Cross-ISP traffic cost (d) Over-provision ratio (e) MTBS

FIGURE 7. (a) Overall costs, (b) energy costs, (c) cross-ISP traffic costs, (d) over-provision ratios, and (e) MTBS of the CDNs employing the energy-aware, optimal and cost-aware capacity provisioning schemes, under various ρ values.

ing schemes, we examine and compare the following performance metrics from one-day operating of the CDN:

- The CDN's monetary *operating cost*, including both the energy cost as well as the cross-ISP traffic cost. We will explain how to compute the two costs later in this subsection.
- The service capacity provisioned by the CDN. In particular, we are interested in the *over-provision ratio* of the CDN's capacity, which is defined as the ratio between the service capacity determined by the algorithm under study divided by the minimum capacity $\bar{N}_t$ for saving the energy cost only. In other words, the over-provision ratio indicates how many extra servers the CDN should keep alive for the objectives of saving its overall operating cost and reducing the server switches.
- The *mean-time-between-switch* (MTBS) for the servers in the CDN, which is the mean time between two consecutive switches of a video server.

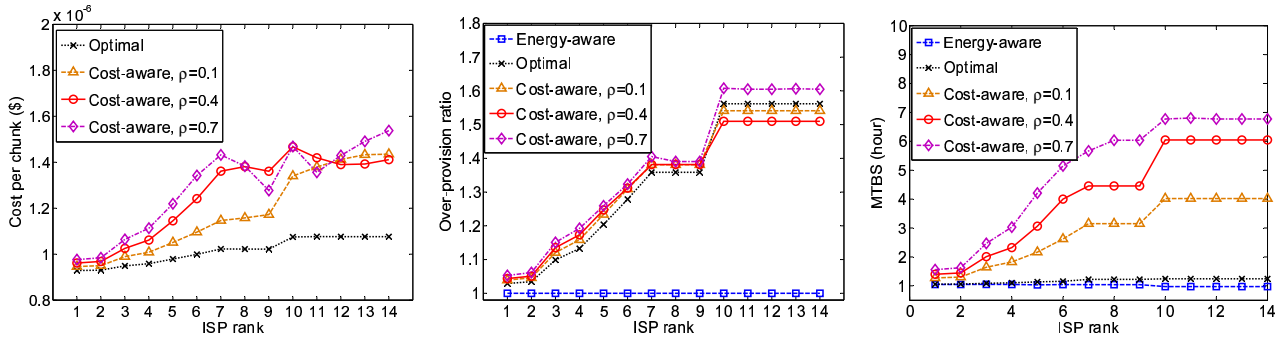
From the definitions, we can see that a preferred capacity provisioning scheme should be able to reduce the CDN's operating cost, but it should also avoid frequent server switches by achieving a medium-long MTBS.

We use synthetic workload traces derived from the real-world workload on Youku that we have collected from a campus network in Appendix A to feed our simulator. More specifically, we view each PoP as a mixture of 20 independent workload sources. Each source imposes a workload whose mean is randomly drawn from the empirical distribution of the campus trace in recent 30 minutes, and the workload fluctuates following a Gaussian distribution whose variance scales linearly with the mean workload, as shown in Appendix A.

We assume that videos are requested and delivered in 256KB-sized chunks, and each video server has a maximum data rate of 100Mbps. To compute the energy cost, we adopt the energy model in [9], which states that a server's energy consumption is $(63+29\times U)$ watts, where U is the server's utilization ratio. We then refer the current power price for data centers in China and compute that c_1, the energy cost for serving one video chunk, is about $9.14 \times 10^{(-7)}$ US dollars.

To compute the cross-ISP traffic cost, we need to decide the value of c_2, the cross-ISP traffic cost for delivering one video chunk. In fact, c_2 depends on the bandwidth price negotiated between the ISPs, and varies from a few to tens of the times of c_1. For simplicity, in our simulation we always let $c_2 = 5 \times c_1$ if not otherwise specified.

Finally, we require an availability of $th_{SLA} = 0.97$ as the CDN's SLA, since such an availability is typical in real-world video streaming services [24].



(a) Cost per chunk (b) Over-provision ratio (c) MTBS

FIGURE 8. (a) Costs per chunk, (b) over-provision ratios, and (c) MTBS for the ISPs under various capacity provisioning schemes.

5.2. Evaluation and Comparison

5.2.1. Overall performance

In our first experiment, we evaluate and compare the energy-aware, optimal, and cost-aware schemes. For the cost-aware scheme, we vary ρ from 0.1 to 1.0. Figure 7 presents the CDN's overall operating costs, the energy costs, and the cross-ISP traffic costs, under the three kinds of CDN capacity provisioning schemes respectively. Figure 7 (d) shows the capacity over-provision ratios and Figure 7 (e) gives the MTBS of the servers under the different schemes.

From Figure 7 (a-c), one can see that by jointly optimizing the energy cost and cross-ISP traffic cost, the optimal scheme saves 21.5% of the CDN's operating expense comparing with the energy-aware scheme. We also find that with moderate ρ values, the cost-aware scheme has a cost between the energy-aware and the optimal schemes, and achieves $8.5 \sim 17.2\%$ of the cost savings comparing with the scheme that focuses only on energy.

Furthermore, from the figures, we can see that for the cost-aware scheme, when ρ is small, the algorithm behaves more closely to the optimal scheme, with a lower over-provision ratio but a shorter MTBS; on the other hand, when ρ becomes larger, the algorithm is more insensitive to the workload dynamics, making the CDN to have a longer MTBS but at a price of higher over-provision ratio and operating cost. One can see that when ρ is larger than 0.94, the threshold for hibernating or awakening a server is too high, so that the CDN indeed rarely adjusts its capacity to cope with the workload dynamics, but keeps its initial capacity, which is planned according to the workload of the first interval, during the entire period of simulation.

In summary, from the experiment results, we can make two observations: 1) Comparing with the energy-aware scheme, the optimal and the cost-aware capacity provisioning schemes are effective in saving a CDN's overall operating cost; 2) By tuning the threshold parameter of ρ, the cost-aware scheme enables the CDN to trade its operating cost with the server switches and vice versa, therefore provides more flexibility in the CDN operation.

5.2.2. Performance from ISP perspective

We further examine the CDN's performances in different ISPs. In Figure 8 (a), we show the averaged monetary cost for serving one video chunk in each of the 14 ISPs, where ISPs are ranked according to their sizes from the largest to the smallest[5]. An ISP's averaged cost per chunk is computed by dividing the CDN's overall operating cost in this ISP with the total number of the served chunks that are requested from the PoPs in the ISP. In Figure 8 (b, c), we present the over-provision ratios and MTBS in different ISPs under various capacity provisioning schemes. For the cost-aware scheme, we choose moderate ρ values as $\rho = 0.1$, 0.4, and 0.7.

From Figure 8 (a) one can see that in a larger ISP, the CDN generally has a lower cost per serving a chunk, and Figure 8 (b) shows that the CDN also has smaller over-provision ratios in larger ISPs. This can be explained with the fact that as the variance of the CDN workload scales linearly with the mean workload, for a larger ISP, its workload is relatively less dynamic and more predictable, which enables the CDN to have a smaller over-provision ratio for the clusters in this ISP, and achieve a relatively lower energy cost as well as the cross-ISP traffic cost. Finally, from Figure 8 (c), one can see that the servers in a small ISP have a longer MTBS than the servers in a larger ISP. This is because in a small ISP with relatively lower workload, in many cases, the absolute values of the workload changes are not large enough for hibernating or awakening one video server. Our observations here suggest that for a CDN employing the "deep-into-ISPs" design, it is more cost-effective to deploy the server clusters in larger ISPs than in smaller ones.

[5]We do not plot the energy-aware scheme for the reason that its costs are much larger than the ones of the other solutions.

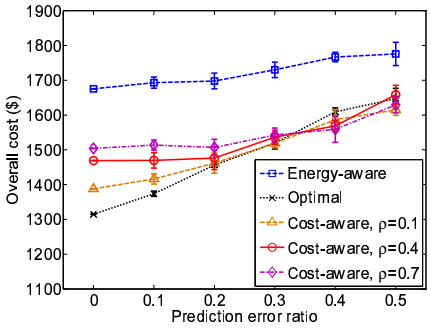


FIGURE 9. CDN's overall operating costs under erroneous workload predictions.

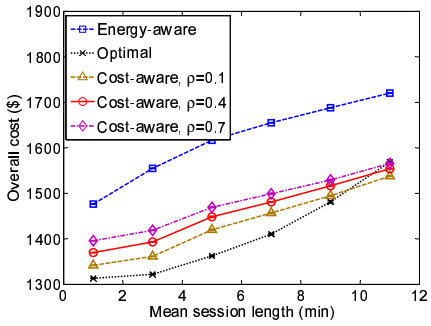


FIGURE 10. CDN's overall operating costs under varying video session lengths.

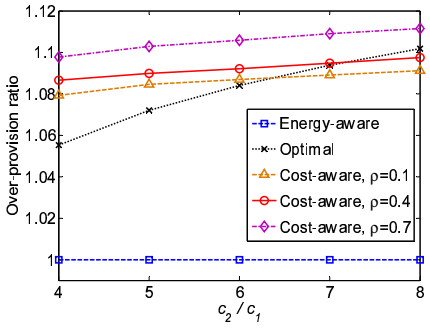


FIGURE 11. Over-provision ratios under different power and cross-ISP bandwidth prices.

5.2.3. Influence of erroneous workload predictions

In this experiment, we consider the influence of the errors in predicting the future workload on the performance of our proposed capacity provisioning algorithm. More specifically, in our simulation the CDN predicts the mean workload of the incoming interval t as $w_{i,t} = \mathbb{E}[x_{i,t}] \times (1 \pm e)$, where e $(0 \leq e < 0.5)$ is the error ratio indicating how far the prediction deviates from the actual workload.

Figure 9 presents the CDN's overall operating cost under the erroneous workload predictions. From the figure, one can see that the optimal and the cost-aware schemes can considerably save the CDN's overall cost under modest error ratios; however, when the error ratio becomes too large (e.g., when $e \geq 0.2$), the two schemes become less effective, and have very similar performance.

5.2.4. Influence of view session lengths

In our previous experiments, we assume that user requests video by chunks. In this experiment, we consider a scenario that a user only requests the video once, and the content streams to the client for a continuous period of time (which is defined as the user's *view session time*). A user's session time depends on many factors, such as the length of the video file, the level of interest held by the user on the content, etc., thus for each request, we can simply suppose its associated session length as random variable, which follows an exponential distribution. We vary users' mean session length, and investigate its influence on CDN's operating cost. However, for different mean lengths, we tune the video request rates so that the total workload imposed on CDN is constant.

We plot the CDN's overall operating costs under the varying mean session lengths in Figure 10. From the figure, one can observe that the CDN's operating cost rises as users tend to view the videos longer, and the cost-aware schemes constantly out-perform the energy-aware scheme with lower costs; it is also interesting to see that the cost for the optimal scheme grows faster than the cost-aware schemes, and will surpasses the latter when the mean session length is longer than 10 minutes.

We seek to explain the observations as the following: Since a user views a video for a random period of time, the workload imposed by one video request on the CDN is also random, therefore for the CDN capacity provision schemes, it is difficult to have an accurate workload prediction under varying view session lengths, and for the exponential distribution, the longer the mean session length is, the less predictable the workload will be. For this reason, we can see that when users tend to have longer session lengths, the CDN pays more for the cross-ISP traffics due to the increasingly inaccurate workload predictions; moreover, such inaccuracy influences the optimal scheme more seriously than the cost-aware approaches, as the latter uses a threshold to avoid some unnecessary server switches.

5.2.5. Influence of power and bandwidth prices

In the previous experiments we assume that by delivering a video chunk in a cross-ISP way, the incurred traffic cost is five times of the energy cost for serving the chunk, that is, $c_2 = 5 \times c_1$. In this experiment, we examine how the trends of the bandwidth and power prices influence the performances of the CDN.

In our experiment we suppose that the cross-ISP chunk delivery cost (c_2) varies from 4 to 8 times of the energy cost per serving a chunk (c_1). Figure 11 presents the CDN's over-provision ratios under the energy-aware, optimal and cost-aware schemes. From the figure one can see that when the cross-ISP bandwidth price is relatively higher than the power price, the CDN has to over-provide more capacities to avoid the cross-ISP content deliveries and save its overall operating cost.

We discuss the implication of Figure 11 as the following: Although in the long-term future, it is expected that the energy price will continue to rise and the cross-ISP bandwidth price will continue to fall [14], however, under the current power and bandwidth

prices, c_2 is still much higher than c_1, indicating that for a CDN with the "deep-into-ISPs" design, a significant part of its service capacity should be over-provisioned for avoiding the cross-ISP video deliveries. The observation also suggests that the current Internet business relationship among the ISPs [25] actually discourages a CDN to improve its energy-efficiency, and a new business model that is more energy-efficient and friendly to the environment should be negotiated between the ISPs and the CDNs.

6. CONCLUSION

In this work, we focus on the Internet video streaming CDNs that employ a "deep-into-ISPs" design, and address the problem of saving a CDN's overall operating cost. By studying the CDN infrastructure of the largest Internet video site in China, namely Youku, we find that the CDN employs an ISP-friendly policy in selecting servers for users, and there exists an inherent conflict between improving the CDN's energy efficiency and maintaining its ISP-friendliness. Motivated by the observation, we propose a practical solution that seeks to save both the energy cost as well as the cross-ISP traffic cost for a CDN. Simulation experiments show that our approach can significantly reduce a CDN's overall operating cost, and enable the system to avoid frequent server switches effectively.

APPENDIX A. CHARACTERISTICS OF CDN WORKLOAD

In this Section, we seek to capture the key characteristics of Youku's workload that can assist us to evaluate our proposed CDN service capacity provisioning scheme. More specifically, we testify if the workload models derived from other CDNs or web-based services can be applied on Youku.

Our study is based on a trace of the workload on Youku from a campus network. More specifically, we setup a traffic capturing program based on tcpdump at the gateway of a university network in China, and collected 24-hour TCP traffics in Jan., 2013. From the trace we have found 256,183 requests for Youku video chunks. The workload is presented in Figure 12. From the figure we can see that there is an obvious diurnal pattern, where the peak workload is over two times of the average. Furthermore, the workload is not smooth, but fluctuates dynamically, even within a short period of time.

Since the aggregated CDN workload can be viewed as generated from a large number of individual clients, one might expect its distribution to be close to Gaussian. Moreover, a recent study shows that CDN workload can be well approximated as Gaussian [21]. We use our campus workload trace to testify this argument, and find that it is indeed the truth. For example, in Figure 13 we consider 5-second workload samples in two intervals, each last 30 minutes, and compare the empirical workload distributions with the Gaussian ones. We can see that the two distributions are very close. In fact, it is observed that samples over time slots of 10 and 30 seconds and in other time intervals are also nearly normally distributed.

We then investigate how the workload fluctuates. Note that for Web traffic, it is observed that the Gaussian traffic variance scales linear with the mean traffic [22]. We expect that such a scaling law also applies for the workload on Youku's CDN. To testify this, we consider a time slot of one second, and for each minute, we compute the mean and variance of the chunk requests in its 60 seconds. The correlation of the mean and variance of the one-second workloads is presented in Figure 14. From the figure, one can see that the workload variance scales nearly linearly with the mean workload. In fact, by using linear regression [26], the relationship between the mean workload and the workload variance can be approximated as

$$\mathrm{Var}(x) \approx a \cdot \mathbb{E}[x],$$

where $a \approx 2.21$ is a constant.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China [61202405]; and the sub task of Strategic Priority Research Program of the Chinese Academy of Sciences [XDA06010301]

REFERENCES

[1] Sandvine (2014). Global Internet phenomena report. http://www.sandvine.com.

[2] Cass, S. (2014). IP traffic in 2017: 1.4 zettabytes. http://spectrum.ieee.org/telecom/internet/ip-traffic-in-2017-14-zettabytes.

[3] Nygren, E., Sitaraman, R. K., and Sun, J. (2010) The Akamai network: A platform for high-performance Internet applications. *ACM SIGOPS Operating Systems Review*, **44**, 2–19.

[4] Wang, Y. A., Huang, C., Li, J., and Ross, K. W. (2011) Estimating the performance of hypothetical cloud service deployments: A measurement-based approach. *Proceedings of IEEE INFOCOM 11*, Shanghai, China, 10-15 April, pp. 2372–2380. IEEE Press, Piscataway.

[5] Adhikari, V. K., Jain, S., and Zhang, Z.-L. (2010) YouTube traffic dynamics and its interplay with a tier-1 ISP: An ISP perspective. *Proceedings of Internet Measurement Conference (IMC 10)*, Melbourne, Australia, 1-3 November, pp. 431–443. ACM Press, New York.

[6] draft-ietf-ppsp-survey-09 (2014) *Survey of P2P Streaming Applications.* Internet Engineering Task Force (IETF). Fremont, CA, USA.

[7] Rayburn, D. (2013). Netflix announces major ISPs deploying their CDN caches. https://twitter.com/DanRayburn/status/288656716747374592.

[8] Lin, M., Wierman, A., Andrew, L. L. H., and Thereska, E. (2011) Dynamic right-sizing for power-proportional

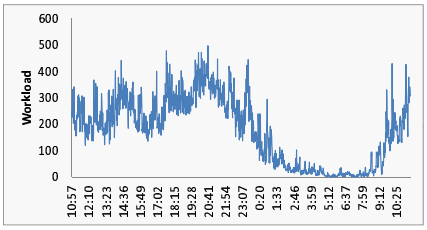


FIGURE 12. Workload from clients in a campus network imposed on Youku's CDN (in video chunk requests per minute)

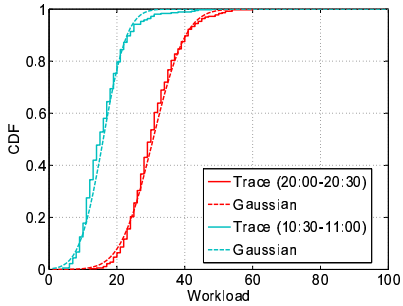


FIGURE 13. Distributions of 5-second workloads from 10:30-11:00 and 20:00-20:30 comparing with Gaussian distributions with the same means and standard deviations.

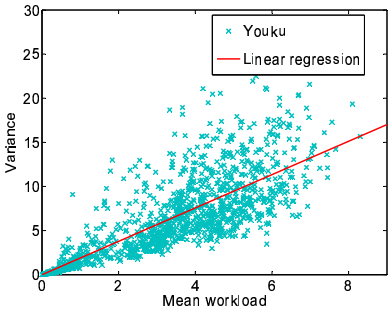


FIGURE 14. Correlation of 1-second mean workload (x-axis) and workload variance (y-axis).

data centers. *Proceedings of IEEE INFOCOM 11*, Shanghai, China, 10-15 April, pp. 1098–1106. IEEE Press, Piscataway.

[9] Mathew, V., Sitaraman, R. K., and Shenoy, P. (2012) Energy-aware load balancing in content delivery networks. *Proceedings of IEEE INFOCOM 12*, Orlando, FL, USA, 25-30 March, pp. 954–962. IEEE Press, Piscataway.

[10] Tchernykh, A. and et al. (2014) Adaptive energy efficient distributed voip load balancing in federated cloud infrastructure. *Proceedings of IEEE International Conference on Cloud Networking*, Luxembourg, 8-10 October, pp. 1–6. IEEE Press, Piscataway.

[11] Youku. http://www.youku.com.

[12] Torres, R., Finamore, A., Kim, J. R., Mellia, M., Munafo, M. M., and Rao, S. (2011) Dissecting video server selection strategies in the YouTube CDN. *Proceedings of ICDCS 11*, Minneapolis, MN, USA, 20-24 June, pp. 248–257. IEEE Press, Piscataway.

[13] Khare, V. and Zhang, B. (2011) Making CDN and ISP routings symbiotic. *Proceedings of ICDCS 11*, Minneapolis, MN, USA, 20-24 June, pp. 869–878. IEEE Press, Piscataway.

[14] Palasamudram, D. S., Sitaraman, R. K., Urgaonkar, B., and Urgaonkar, R. (2012) Using batteries to reduce the power costs of Internet-scale distributed networks. *Proceedings of ACM Symposium on Cloud Computing (SoCC 12)*, San Jose, CA, USA, 13-17 October, pp. 1–11. ACM Press, New York.

[15] Team cymru IP to ASN mapping. https://www.team-cymru.org/Services/ip-to-asn.html.

[16] Tian, Y., Dey, R., Liu, Y., and Ross, K. W. (2013) Topology mapping and geolocating for China's Internet. *IEEE Transactions on Parallel and Distributed Systems*, **24**, 1908–1917.

[17] Hameed, A. and et al. (2014) A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing*, **online preprint**.

[18] Niu, D., Feng, C., and Li, B. (2012) Pricing cloud bandwidth reservations under demand uncertainty. *Proceedings of ACM SIGMETRICS 12*, London, UK, 11-15 June, pp. 151–162. ACM Press, New York.

[19] Niu, D., Liu, Z., Li, B., and Zhao, S. (2011) Demand forecast and performance prediction in peer-assisted on-demand streaming systems. *Proceedings of IEEE INFOCOM 11*, Shanghai, China, 10-15 April, pp. 421–425. IEEE Press, Piscataway.

[20] Kang, X., Zhang, H., Jiang, G., Chen, H., Meng, X., and Yoshihira, K. (2010) Understanding Internet video sharing site workload: A view from data center design. *Journal of Visual Communication and Image Representation*, **21**, 129–138.

[21] Bak, A., Gajowniczek, P., and Pilarski, M. (2011) Gaussian approximation of cdn call level traffic. *Proceedings of the International Teletraffic Congress (ITC 11)*, San Francisco, CA, USA, 6-9 September, pp. 135–141. IEEE Press, Piscataway.

[22] Morris, R. and Lin, D. (2000) Variance of aggregated web traffic. *Proceedings of IEEE INFOCOM 00*, Tel Aviv, Israel, 26-30 March, pp. 360–366. IEEE Press, Piscataway.

[23] Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*. Cambridge University Press, Cambridge, UK.

[24] Liu, X., Dobrian, F., Milner, H., Jiang, J., Sekar, V., Stoica, I., and Zhang, H. (2012) A case for a coordinated Internet video control plane. *Proceedings of ACM SIGCOMM 12*, Helsinki, Finland, 13-17 August, pp. 359–370. ACM Press, New York.

[25] Gao, L. (2001) On inferring autonomous system relationships in the Internet. *ACM/IEEE Transactions on Networking*, **9**, 733–745.

[26] Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, MA, USA.