Content Multi-homing: an Alternative Approach

Technical Report HKUST-CS14-01

Jason Min Wang[1], Jun Zhang[2], Brahim Bensaou[1]
[1]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
[2]Network and Computer Science Department, Telecom ParisTech
jasonwangm@cse.ust.hk, jun.zhang@telecom-paristech.fr, brahim@cse.ust.hk

***Abstract*—The CDN serves as an essential element in providing content delivery services on the Internet; however, limited by its footprint and influenced by variations of network conditions and user demands, no individual CDN can qualitatively fulfill delivery tasks anytime and anywhere. As such, large content providers often use multiple CDNs. Nevertheless, this approach is not only cumbersome to negotiate multiple business contracts, but also economically inefficient, especially for small content providers. A simpler alternative approach would be to let each content provider only sign one contract with an authoritative CDN, say CDN-0, and let CDN-0 handle the potential performance trap of individual CDN. To fulfill its business commitment, CDN-0 could either use its own infrastructure or rent services from other CDNs, which is sometimes indispensable. To make an optimal operating plan for CDN-0 operator, we propose a new problem, called MCDN-CM, whose objective is to minimize CDN-0's operating cost. MCDN-CM is a concave minimization problem and we take advantage of the special form of the practical CDN-pricing function—piece-wise linear concave—to arrive at an optimal solution through an iterative procedure. We conduct numerical experiments under realistic settings and show via the experimental results that CDN-0 can achieve a tremendous cost-saving by using our proposed algorithm.**

I. Introduction

By moving content "closer" to end users, content delivery networks (CDNs) greatly reduce the content distribution cost and significantly improve the content access experience. Increasingly, CDNs are recognized to play a fundamental role in supporting the large-scale content delivery over the Internet. Nowadays, large content providers (CPs) rely on the CDN to deliver their content in the global scale. For example, YouTube uses Google's private CDN to deliver videos; Netflix [1] and Hulu [2] provide their services via third-party CDNs, including Akamai, LimeLight, and Level-3. The ever-growing demand for videos drives the continual up-scaling of the CDN infrastructures.

From the content provider perspective, it is desirable that its content can be delivered to users with high quality, regardless of the geographical location of the users and the ISP network they attach to. However, two facts make this requirement difficult to achieve. First, it is hard for any single CDN provider, even the largest one like Akamai, to expand its footprint to every corner of the world. Typically a CDN presents distinct performances at different geographical areas. Second, due to the dynamics of network conditions and the variation of server loads, the performance of a CDN, even at the same area, fluctuates over time [1]–[4]. As a consequence, content providers like Netflix [1] and Hulu [2] use multiple CDNs to provide high quality services; this is referred to as *content multi-homing* [3].

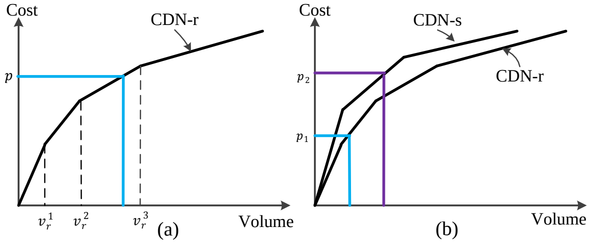

Fig. 1. (a) The piecewise linear concave pricing function of CDN-r, $f_r(\cdot)$. (b) *Motivating example*: when involving two CDNs (CDN-r and CDN-s), the content delivery performance gets improved but the total cost paid by the small content provider is greater than that using only CDN-r: $p_1 + p_2 > p$.

Content multi-homing enables a content provider to avoid the quality of experience (QoE) trap of any single CDN, whereas it could result in economic inefficiency for small or medium content providers. Such economic inefficiency arises from the widely-adopted charging policy of CDN: i) you pay as you go, and ii) the more you use, the cheaper the average price you get. Specifically, the CDN pricing function has a piecewise linear concave shape, as shown by Fig. 1(a). Under current charging policies, a small CP, when using multiple CDNs, has to pay the portions of traffic at a high charging rate for individual CDNs. This is illustrated by the example in Fig. 1. Note that for a large CP this problem is alleviated, since its charging volume at each CDN can easily arrive at the highest volume-price segment, thus making the payment at a lower average price. In addition, content multi-homing entails each CP to negotiate and sign business contracts with multiple CDN providers, which is cumbersome.

Motivation. To mitigate the above dilemma, an appealing approach consists in letting small CPs contact and deal with only one authoritative CDN, say CDN-0. As such, CPs no longer suffer from the negative effects of payment discount policy brought by multi-homing. It is now CDN-0's responsibility to fulfil the performance commitment, stipulated in the service level agreement (SLA). To this end, CDN-0 must rely on other CDNs in procuring *content multi-homing* transparently to the CP. This new approach is illustrated by Fig. 2. CDN-0 acts as a proxy, aggregating demands from registered CPs and acquiring the maximum discounts from other CDNs as a "giant CP".

Overview. Delivering any content item to a specific area can either use CDN-0's own servers or employ other CDNs. Two factors impact the decisions. One is that CDN-0 can not satisfy the service quality requirement of certain items, restricted by its footprint and regional service capabilities. In this case, it has

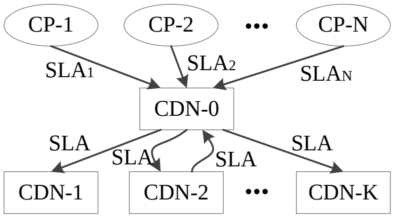

Fig. 2. Multiple content providers sign business contracts with one authoritative CDN with various SLA; the authoritative CDN-0, as a customer, rents services from other CDNs, besides using its own infrastructure.

to resort to other CDNs that are better implanted regionally. The other is that delivering some content items to certain areas, could be more cost-effective in "renting" other CDNs' services than doing it on its own network. For example, the average cost of running one replica server in a specific area could be so high that it hinders CDN-0 from running a portion of its replica servers there. Consequently, the operator of CDN-0 is inspired to carefully make an operating plan: how to realize the business commitments, in particular the QoE requirements, by using either its own infrastructure or other CDNs' services, whilst aiming to minimize its operating cost. In this paper, we help CDN-0 operator make the operating plan optimally, by modeling it as a concave minimization problem, referred as MCDN-CM. We solve the problem optimally by invoking an iterative procedure.

Contributions. First, to the best of our knowledge, we are the first to propose the MCDN-CM problem, which has a strong motivation for the operator of CDN-0 and also benefits the content providers, especially smaller ones, in both procedural and economical aspects. Second, we give an algorithm to solve MCDN-CM optimally. The solution can be directly translated into CDN-0's operating plan. Third, this work embraces the emerging trend of CDN-interconnection [5].

Outline. Section II describes the necessary background on CDN and formulates the MCDN-CM problem. We derive the optimal solution to MCDN-CM in Section III and present the experimental results in Section IV. Finally, we conclude the paper in Section V.

II. Content Multi-homing Model

We now present the new content multi-homing model. After introducing some basic terminology and notations, we formally define the MCDN-CM problem.

A. Preliminaries

CDN provider. A CDN provider deploys replica servers at strategically chosen "edge locations", aka point of presences (PoPs), scattered across the Internet. We view each PoP as an entity that hosts servers and has a massive amount of computing and caching power. Different CDNs have different landscapes in terms of deployed PoPs and service capacities. We denote P the set of PoPs owned by CDN-0.

Content provider. Each content provider (e.g., YouTube) services a set of objects to users at different areas via the CDN infrastructure. Suppose the set of content providers $\mathcal{N}$ have signed business contracts with CDN-0, shown in Fig. 2. For each CP $n \in \mathcal{N}$, I_n represents the set of served content objects. In view of the license issues, it is reasonable to assume that content sets of different providers are disjoint, i.e., $I_{n_1} \cap I_{n_2} = \emptyset, \forall n_1 \neq n_2$. For convenience, let $I = \bigcup_{n \in \mathcal{N}} I_n$.

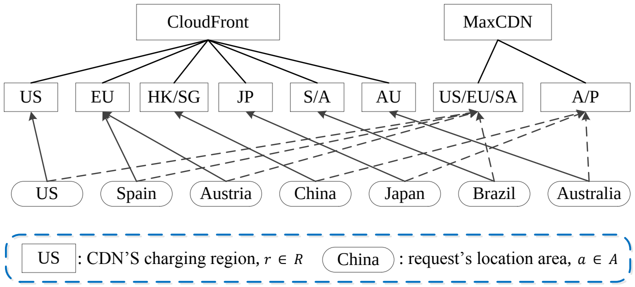

Fig. 3. The mapping from request location area $a \in A$ to charging region $r \in R_k$ of two CDNs: CloudFront and MaxCDN.

Content object. Each content object (e.g.,, a video) possesses various properties [3]. For content distribution, key properties involve the object size, the content popularity, and the performance requirement. Let o_i be the size of object $i \in I$. Since an object i can be requested by users in different areas, we denote A the set of geographical areas. Let d_{ai} be the number of times that object i has been requested by the users in area a during a period. Obviously, d_{ai} reflects the content popularity for this object. Note that d_{ai}s could be estimated according to historical access records. Thus, we assume these parameters are available to use in our problem. The property of performance requirement is discussed later.

CDN pricing. Some CDN providers (e.g., Amazon CloudFront and MaxCDN) offer their competitive prices in public, whereas other CDN providers (e.g., Akamai, Limelight, and Level3) do not publish their pricing information. Nevertheless, the pricing policies in the CDN market generally obey the rule that you pay as you go and the more traffic you use, the lower the average price you get. The CDN pricing has the following three properties: (1) *regional-based pricing*, with a price that varies across different regions; (2) *volume-based charging*, where charging is based on the volume of traffic, originating from location areas of a specific charging region, during the billing period (e.g., one month); and (3) *quantitative discount*, where the larger the volume is consumed, the larger the relative discount is given. For example, Amazon CloudFront divides its pricing into six charging regions [6], shown in Fig. 3. Among the six, the regional price of US is the cheapest; it charges 0.12$/GB for low volume and 0.02$/GB for high volume. Let $\mathcal{K}$ be the set of CDNs that CDN-0 leverages to fulfill its business commitments. Let R_k be the set of charging regions of CDN-k. Clearly, $R_{k_1} \cap R_{k_2} = \emptyset, \forall k_1 \neq k_2$. For simplification, let $R = \bigcup_{k \in \mathcal{K}} R_k$. Further, we denote $f_k^r(\cdot)$, as in Fig. 1.(a), the volume pricing function of CDN-k over region r.

QoE guarantee. The business contract stipulates the service-level-agreement (SLA) that outlines the expected QoE and costs associated with the business arrangement. The meaning of QoE, for different types of objects, varies. For example, the QoE with respect to a Web page or image could simply mean access latency, while for a YouTube video it could pertain to the bit-rate. Instead of specifying distinct QoE metrics for diverse objects, we adopt a simple unified characterization [3], *the fraction of times* that requests for an object can be served with sufficient QoE. We use $\hat{q}_{ai}$ to denote the performance target set by the CP. For example, for video i,

$\hat{q}_{ai} = 90\%$ means that 90 percent requests at area a should be served at the video's encoding rate. On the other hand, we use $q^{p}_{ai}, p \in P$ and $q^{r}_{ai}, r \in R_k$[1] to characterize the ability (fraction of times) of PoP p of CDN-0 and the servers in region r of CDN-k in serving requests at area a for object i with the targeted QoE, respectively. To reflect the QoE constraints, we define $Q_{ai} = \{p \mid p \in P, q^{p}_{ai} \geq \hat{q}_{ai}\}$ and $\tilde{Q}_{ai} = \{r \mid r \in R, q^{r}_{ai} \geq \hat{q}_{ai}\}$. Note that, it is possible that $Q_{ai} = \tilde{Q}_{ai} = \emptyset$. In this case, the best p or r will be added.

B. Cost Model and Problem Formulation

We model PoP p as a collection of replica servers used to cache content and serve client requests. As the upfront investment, we assume CDN-0 has deployed M_p servers at PoP $p \in P$. Given the demands from customers $\mathcal{N}$, the CDN-0 operator needs to figure out how requests for objects are served. To this end, we introduce two decision variables: i) x^{p}_{ai}, the fraction of requests at area a for object i to be directed to PoP p of CDN-0; and ii) $x^{r}_{ai}, r \in R_k$, the fraction of requests to be served by servers in region r of CDN-k. To support $\{x^{p}_{ai}\}$, we assume m_p servers have to be provisioned at PoP p. The goal is to minimize CDN-0's operating cost that consists of two parts: i) the running cost of its infrastructure, $\{m_p\}$ servers; ii) the cost paid to other CDNs in $\mathcal{K}$.

Running cost of P. In practice, the running cost of PoP p depends on several factors, like electricity, bandwidth, and tax. Cost like electricity even varies temporally. To simplify, we use h_p to represent the average hosting cost per server at PoP p. Then the running cost of PoP p becomes $m_p \cdot h_p$. Albeit simple, it has captured the fundamental aspects of CDN's operating cost, i.e., the cost increases with m_p; h_p varies across different areas. Further, we use b_p to denote the average processing power of one server at PoP p; b_p is normalized to be the average number of requests each server can respond to. Note that this simplified model was used in [7], [8] as well.

Cost paid to other CDNs. The portions of requests redirected to other CDNs aggregate to be the charging volume that CDN-0, as the customer, pays to other CDNs. The cost paid to CDN-k is $\sum_{r \in R_k} f^{r}_{k}(\sum_{a \in A} \sum_{i \in I} d_{ai} o_i x^{r}_{ai})$.

Problem formulation. Given the above cost model, the goal is to minimize the total operating cost of CDN-0. We call this the *Multiple CDNs Cost Minimization* (MCDN-CM) problem, defined as:

$$\min_{\mathbf{x},\mathbf{m}} \sum_{k,r} f^{r}_{k}\big(\sum_{a,i} d_{ai} o_i x^{r}_{ai}\big) + \sum_{p} m_p h_p \tag{1}$$

$$\text{s.t.} \sum_{p \in Q_{ai}} x^{p}_{ai} + \sum_{r \in \tilde{Q}_{ai}} x^{r}_{ai} = 1, \ \forall a \in A, i \in I \tag{2}$$

$$x^{p}_{ai} \geq 0, \ \forall a \in A, i \in I, p \in Q_{ai} \tag{3}$$

$$x^{r}_{ai} \geq 0, \ \forall a \in A, i \in I, r \in \tilde{Q}_{ai} \tag{4}$$

$$x^{r}_{ai} = 0, \ \forall a \in A, i \in I, r \notin \tilde{Q}_{ai} \tag{5}$$

$$\sum_{a,i} d_{ai} x^{p}_{ai} \leq m_p b_p, \ \forall p \in P \tag{6}$$

$$0 \leq m_p \leq M_p, \ \forall p \in P \tag{7}$$

(1) is CDN-0's operating cost. (2) ensures that any user request for object i at area a is served. The QoE requirements are captured via Q_{ai} and $\tilde{Q}_{ai}$, encapsulated in (3), (4), and (5). (6) states that the expected total load of servers in PoP p should not exceed the service capacity. (7) limits the number of active servers of PoP p. Although m_p is integral in essence, we may relax it to be continuous in MCDN-CM, in view of: i) a PoP could comprise hundreds of servers; and ii) b_p and h_p per se are rough estimates. Apart from the service capacity constraint (6), we could also add constraints like the bandwidth capacity constraint, $\sum_{a,i} d_{ai} o_i x^{p}_{ai} \leq m_p u_p$. The solution discussed hereinafter still works.

MCDN-CM is a *concave minimization* problem [9]. Since decision variables are well bounded and constraints are linear, the feasible set $\mathcal{D}$ is a bounded polyhedron. We assume $\mathcal{D}$ is non-empty. By *Weierstrass' Theorem*, MCDN-CM has a global optimal solution, which is attained at an extreme point of $\mathcal{D}$ [9]. However, instead of using a general approach to solve a concave minimization problem, we will take advantage of the special structure of $f^{r}_{k}(\cdot)$—piecewise linear concave.

C. Discussion

Liu et al. studied *content multi-homing* [3] from the perspective of large content providers. Their approach requires each content provider to sign multiple CDN contracts and is not economically efficient for small content providers. In this paper, we advocate the special role of the authoritative CDN-0. Thus, the multi-homing problem in [3] and MCDN-CM differ in the participation of CDN-0, with a different objective function and additional constraints. Although Liu et al. gave a special and efficient algorithm in solving the multi-homing problem, we cannot use their approach to solve MCDN-CM. This is because the extra server capacity constraints (5) (or the potential bandwidth capacity constraints) prohibit transforming MCDN-CM into the so-called *assignment problem* [3] in the first step, thus making the approach in [3] infeasible.

III. Global Optimization

To reach an optimal solution to MCDN-CM, we transform the concave minimization problem into a *bilinear program* by exploiting the special structure of the CDN pricing function and solve the bilinear program in an iterative manner.

A. Eliminating m_p

Closer observation to MCDN-CM reveals that the constraint (7) must be tight in the optimal solution $(\mathbf{x}^*, \mathbf{m}^*)$. That is, $\sum_{a,i} d_{ai} {x^{*}}^{p}_{ai} = {m^{*}}_{p} b_p, \forall p \in P$. With this property, we can eliminate the decision variable m_p from MCDN-CM, and get an equivalent problem MCDM-CM-E:

$$\min_{\mathbf{x}} \sum_{k,r} f^{r}_{k}\big(\sum_{a,i} d_{ai} o_i x^{r}_{ai}\big) + \sum_{p} c_p \sum_{a,i} d_{ai} x^{p}_{ai} \tag{8}$$

$$\text{s.t. } (2), (3), (4), (5)$$

$$\sum_{a,i} d_{ai} x^{p}_{ai} \leq s_p, \ \forall p \in P \tag{9}$$

where $c_p = h_p / b_p$ and $s_p = M_p b_p$. Note that c_p can be viewed as the normalized per request service cost in PoP p and s_p is the maximum service capacity of PoP p.

[1] In practice, $\{q^{p}_{ai}\}$ can be estimated based on the access logs of CDN-0; $\{q^{r}_{ai}\}$ can be estimated from the SLA that CDN-0 has signed with CDN-k.

B. Bilinear Transformation

Table I demonstrates the concrete pricing functions of CloudFront and MaxCDN. The pricing function is piece-wise linear concave (see Fig. 1(a)). Specifically, $f_k^r(\cdot)$ can be written in the following form

$$f_k^r(x_r) = \begin{cases} u_r^1 x_r + e_r^1, & x_r \in [v_r^0 = 0, v_r^1) \\ u_r^2 x_r + e_r^2, & x_r \in [v_r^1, v_r^2) \\ \cdots & \cdots \\ u_r^{n_r} x_r + e_r^{n_r}, & x_r \in [v_r^{n_r-1}, \infty) \end{cases} \tag{10}$$

where $u_r^1 > u_r^2 > \cdots > u_r^{n_r}$. n_r is the number of charging intervals and $u_r^l (1 \le l \le n_r)$ reflects the average price (per GB) in the volume interval $[v_r^{l-1}, v_r^l)$. Equivalently,

$$f_k^r(x_r) = \min_{1 \le l \le n_r} \{u_r^l x_r + e_l^r\}. \tag{11}$$

For brevity, let $x_r = \sum_{a,i} d_{ai} o_i x_{ai}^r$ represent the charging volume of region $r \in R$. Given the volume interval in which x_r falls, $f_k^r(\cdot)$ turns into a linear function, while MCDF-CM-E becomes a linear program. For this purpose, we introduce for each region r a set of binary variables y_r^l $(1 \le l \le n_r)$, indicating whether x_r falls in $[v_r^{l-1}, v_r^l)$. That is,

$$y_r^l = \begin{cases} 1, & \text{if } x_r \in [v_r^{l-1}, v_r^l) \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

We reformulate the problem MCDN-CM-E: i) modify the objective function (8) using binary variables y_r^l, $f_k^r(x_r) = \sum_l [u_r^l x_r + e_r^l] y_r^l$; ii) add extra constraints, $\sum_l y_r^l = 1$, $\sum_l v_r^{l-1} y_r^l \le x_r \le \sum_l v_r^l y_r^l$, and $y_r^l = \{0, 1\}$.

Relaxing the integer constraint of y_r^ls leads to the following problem, referred to as MCDN-CM-ER,

$$\min_{\mathbf{x},\mathbf{y}} \sum_{k,r} \sum_l y_r^l \big(u_r^l x_r + e_r^l\big) + \sum_p c_p \sum_{a,i} d_{ai} x_{ai}^p \tag{13}$$

$$\text{s.t.} \quad (2), (3), (4), (5), (9)$$

$$\sum_l v_r^{l-1} y_r^l \le x_r \le \sum_l v_r^l y_r^l, \ \forall r \in R \tag{14}$$

$$\sum_l y_r^l = 1, \ \forall r \in R \tag{15}$$

$$y_r^l \ge 0, \ \forall r \in R, 1 \le l \le n_r \tag{16}$$

Although we relax the integer constraint of y_r^l, Theorem 1 (see proof in Appendix) establishes the equivalence of MCDN-CM-ER and MCDN-CM-E. Therefore, we devote to solving MCDN-CM-ER instead.

Theorem 1: $(\mathbf{x}^*, \mathbf{y}^*)$ is an optimal solution to MCDN-CM-ER, if and only if $\mathbf{x}^*$ is an optimal solution to MCDN-CM-E.

C. Algorithm for Solving MCDN-CM-ER

MCDN-CM-ER is a bilinear program with jointly constrained feasible region. Along the line of "Dynamic Cost Updating Procedure" (DCUP) [10], we present an exact and continuous algorithm, Alg. 1, to solve MCDN-CM-ER. DCUP procedure is based on the observation that the feasible sets of variables $\{x_{ai}^r\}$ and $\{y_r^l\}$ are jointly constrained only by constraint (14). If we fix one type of variables, the remaining problem becomes an optimization with respect to the other type of variables and consists in a linear program. Specifically, by fixing $\mathbf{y}$, we get the following linear problem MCDN-LP($\mathbf{y}$),

Algorithm 1 DCUP for MCDN-CM-ER

```
1: Initialization: let $\mathbf{y}_r(0)$ be the initial vector of $\{y_r^l\}$ where
   $y_r^1(0) = 1$ and $y_r^l(0) = 0, \forall l \neq 1$; let $t \leftarrow 1$;
2: Iteration-$t$:
   2.1: Solve MCDN-LP$\big(\mathbf{y}(t-1)\big)$:
        $\mathbf{x}(t) \leftarrow \arg\min_{\mathbf{x}} \big\{\text{MCDN-LP}\big(\mathbf{y}(t-1)\big)\big\}$;
   2.2: Solve MCDN-LP$\big(\mathbf{x}(t)\big)$:
        $\mathbf{y}(t) \leftarrow \arg\min_{\mathbf{y}} \big\{\text{MCDN-LP}\big(\mathbf{x}(t)\big)\big\}$;
3: If $\mathbf{y}(t) = \mathbf{y}(t-1)$, then stop; otherwise, let $t \leftarrow t+1$ and
   go to Step 2;
```

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & \sum\limits_{k,r} x_r \sum\limits_l u_r^l y_r^l + \sum\limits_p c_p \sum\limits_{a,i} d_{ai} x_{ai}^p \\ \text{s.t.} & (2), (3), (4), (5), (9) \end{array} \tag{17}$$

Similarly, by fixing the variable $\mathbf{x}$, we get the other linear problem MCDN-LP($\mathbf{x}$),

$$\begin{array}{ll} \min\limits_{\mathbf{y}} & \sum\limits_{k,r} \sum\limits_l y_r^l \big(u_r^l x_r + e_r^l\big) \\ \text{s.t.} & (14), (15), (16) \end{array} \tag{18}$$

Alg. 1 proceeds by solving two (coupled) linear programs iteratively. It is worth noting that MCDN-LP($\mathbf{y}$) itself does not include the constraint (14). But when Alg. 1 terminates, the constraint (14) will be satisfied, because this constraint already appears in MCDN-LP($\mathbf{x}$). Theorem 2 establishes the correctness of Alg. 1 in solving MCDN-CM-ER and Theorem 3 establishes the convergence of Alg. 1 (see proof in Appendix).

Theorem 2: The final solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ given by Alg. 1 is an optimal solution to MCDN-CM-ER.

Theorem 3: The Alg. 1 stops in a finite number of iterations.

IV. Performance Evaluation

MCDN-CM-ER aims to minimize the operating cost of CDN-0, whilst fulfilling its business commitment, that requests from clients for content provided by registered CPs should be satisfied with sufficient QoE specified in the SLAs. It is natural to observe the cost achieved from different operating plans.

A. Experimental Setup

Datasets. To make the numerical experiments as realistic as possible, we used the Youtube Dataset provided by [11]. We ran the experiments on 5 groups of datasets. Since similar results were observed, we only report results of one group for brevity. Each group dataset consists of 5 days' records. We randomly distribute them to $|\mathcal{N}| = 10$ content providers (CPs). Typically, a 5-day dataset corresponds to several thousands of TB traffic, which we believe is a medium size[2] and is appropriate for our experiments. Important fields of the dataset

[2] The mathematical model of multiple-CDN cost minimization (MCDN-CM) per se does not depend on the fidelity of any input parameters.

TABLE I. REGIONAL PRICING FUNCTIONS USED IN THE EXPERIMENTS

CDN	Region	[0, 10TB)	[10TB, 50TB)	[50TB, 150TB)	[150TB, 500TB)	[500TB, 1PB)	[1PB, 5PB)	[5PB, ∞)
CloudFront	US	\$0.12 /GB	\$0.08 /GB	\$0.06 /GB	\$0.04 /GB	\$0.03 /GB	\$0.025 /GB	\$0.02 /GB
	EU	\$0.12 /GB	\$0.08 /GB	\$0.06 /GB	\$0.04 /GB	\$0.03 /GB	\$0.025 /GB	\$0.02 /GB
	HK/SG	\$0.19 /GB	\$0.14 /GB	\$0.12 /GB	\$0.10 /GB	\$0.08 /GB	\$0.07 /GB	\$0.06 /GB
	JP	\$0.201 /GB	\$0.148 /GB	\$0.127 /GB	\$0.106 /GB	\$0.085 /GB	\$0.075 /GB	\$0.065 /GB
	S/A	\$0.25 /GB	\$0.20 /GB	\$0.18 /GB	\$0.16 /GB	\$0.14 /GB	\$0.13 /GB	\$0.125 /GB
	AU	\$0.19 /GB	\$0.14 /GB	\$0.12 /GB	\$0.11 /GB	\$0.095 /GB	\$0.090 /GB	\$0.085 /GB
MaxCDN	US/EU/SA	\$0.07 /GB	\$0.06 /GB	\$0.05 /GB	\$0.04 /GB	\$0.035 /GB	\$0.03 /GB	\$0.02 /GB
	A/P	\$0.10 /GB	\$0.074 /GB	\$0.064 /GB	\$0.053 /GB	\$0.043 /GB	\$0.037 /GB	\$0.032 /GB

involve the video id, the views, the length, and the video rate (KB/s). The product of length and rate yields the video size. Based on the rate value, the videos are categorized in two types: *low bit-rate* and *high bit-rate*. We distinguish these two types, as each type has different QoE requirements.

Pricing and QoE settings. We deliberately choose CloudFront (CDN-1) and MaxCDN (CDN-2) to form the set $\mathcal{K}$, as their regional pricing is publicly available, thereby leading to a realistic setting of pricing functions $f_k^r(\cdot)$. The concrete pricing function is illustrated in Table I. On average, the pricing of MaxCDN is lower than that of CloudFront. Liu et al. in [3] conducted experiments via PlanetLab and measured the CDN's performance in servicing requests from 7 location areas (as are shown in Fig. 3). To make the settings of QoE parameters $\{q_{ai}^r\}$ realistic, we reuse their collected data and set A to hold those 7 areas as well. The detailed information on QoE parameter settings is summarized in Table II. In general, CloudFront presents a high quality in servicing both low bit-rate and high bit-rate videos, while MaxCDN presents varying performances in serving different types of videos at distinct areas. Finally, the QoE target $\tilde{q}_{ai}$ is set to 90%, which determines the values of Q_{ai} and $\tilde{Q}_{ai}$ in turn.

Setting of CDN-0. To model the footprint of CDN-0, we assume 3 PoPs have been deployed at each area in A. For example, P_1, P_2, and P_3 are three PoPs to serve requests from the US area. In total CDN-0 contains $|P| = 21$ PoPs. We assume that three PoPs at the same area present similar performance, i.e., $q_{ai}^{p_1} = q_{ai}^{p_2} = q_{ai}^{p_3}$. The concrete values are given in Table II. Furthermore, the general cost of running an active server, h_p, should reflect the regional cost variations. To this end, we set the values of $\{h_p\}$, referring to the pricing for Virtual Machines of Amazon EC2 [12]. The setting of $\{h_p\}$ is summarized in Table III, where different groups correspond to different levels of hosting prices. Lastly, we set M_p, the quota of servers in p, between $[20, 35]$ randomly.

TABLE II. MEASURED CDN PERFORMANCE SETTINGS

	US	Spain	Austria	China	Japan	Brazil	Australia
CDN-0	99[a]	99	97	96	99	99	94
	96[b]	97	42	91	47	12	96
CDN-1	99	99	99	99	99	100	100
	99	99	99	99	99	100	100
CDN-2	99	98	97	91	97	98	94
	98	96	96	24	95	70	89

[a] The first row corresponds to the performance in servicing low bit-rate videos.
[b] The second row corresponds to the performance in servicing high bit-rate videos.

TABLE III. AVERAGE HOSTING PRICE OF SERVER IN POP OF CDN-0

h_p group	US	Spain	Austria	China	Japan	Brazil	Australia
Group-1	122[c]	121	122	163	135	248	133
Group-2	243	242	243	296	270	495	266
Group-3	486	484	486	582	540	990	531
Group-4	517	531	517	685	593	1118	585

[c] in US dollars per month.

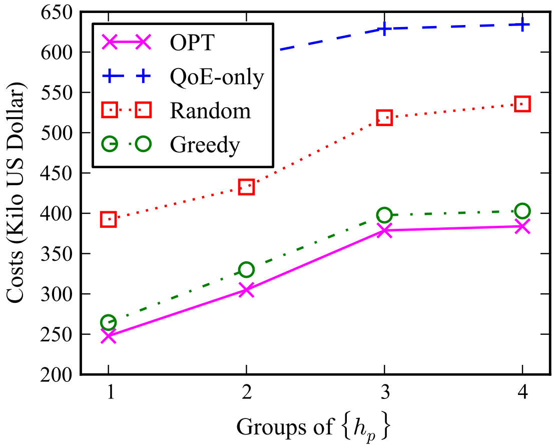

Fig. 4. The gross operating cost of CDN-0 using different methods, with varying groups of $\{h_p\}$.

B. Evaluation Methodology

The solution given by Alg. 1, referred as *OPT*, offers the detailed operating plan of CDN-0. Any feasible operating plan involves two aspects: i) PoP activation, that m_p servers of PoP p will be activated; and ii) request assignments[3], that x_{ai}^r ($r \in R_k$) portion of requests for object i at area a will be directed to region r of CDN-k and x_{ai}^p portion will be redirected to PoP p of CDN-0. Together $\{m_p\}, \{x_{ai}^r\}, \{x_{ai}^p\}$ determine the operating cost. To evaluate the performance of Alg. 1 in minimizing the operating cost, we compare it with three alternatives: i) *QoE-only*, which always choose the region r or PoP p that can provide the best QoE (maximum q_{ai}^r or q_{ai}^q); ii) *random*, which picks any one among those regions or PoPs that satisfy the QoE requirements; and iii) *greedy*, which selects from the QoE-satisfied candidates the one that leads to the lowest cost after current assignment. Note that the above three alternatives all proceed by making assignments for requests sequentially in a uniformly random order.

[3] The requests redirection may either use existing DNS-based approach or employ the emerging technique of software defined networking (SDN).

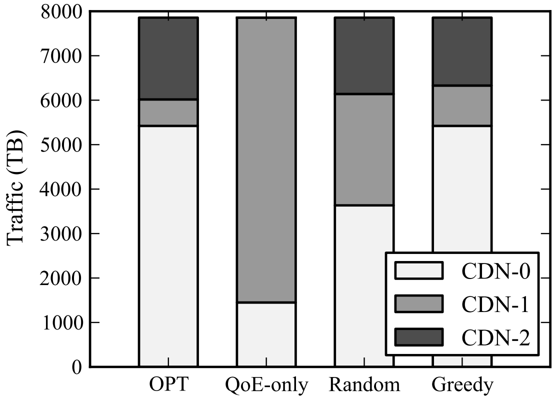

Fig. 5. Traffic distribution of four methods under Group-1 $\{h_p\}$.

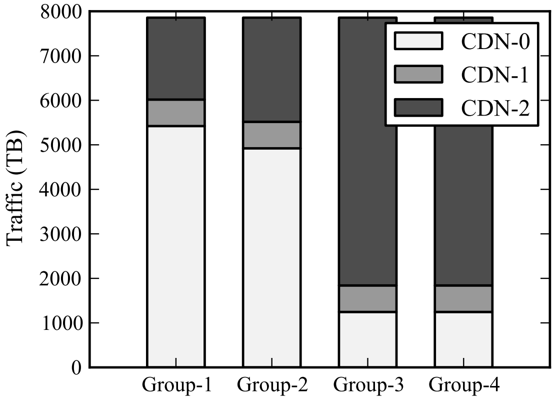

Fig. 6. Traffic distribution using *OPT*, under four groups of $\{h_p\}$.

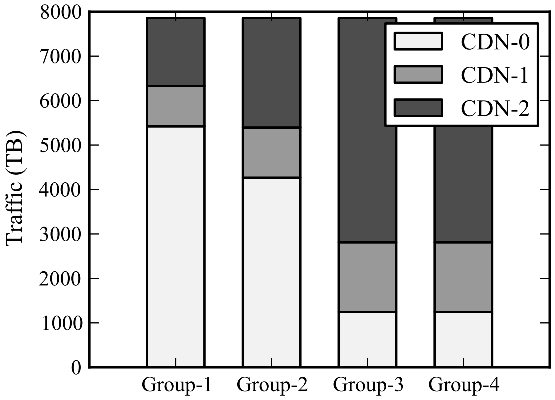

Fig. 7. Traffic distribution using *greedy*, under four groups of $\{h_p\}$.

We use the Python interface of Gurobi-5.1 (under the academic license) to solve MCDN-LP($\mathbf{y}$). With around 380$\sim$400 thousands of location-objects, Gurobi-5.1 can solve MCDN-LP($\mathbf{y}$) in about 10 minutes under a machine with 4 $\times$ AMD Opteron 844 (1.8GHz) CPU and 8GB RAM. In all our experiments, Alg. 1 terminates with less than 3 iterations.

C. Experimental Results

First, we observe the cost paid by CDN-0 for accomplishing its business commitment with full QoE grantees. We compare the values of the operating cost, resulting from the four different algorithms mentioned above. Fig. 4 shows the results with varying groups of $\{h_p\}$. As expected, *OPT* leads to the least monthly cost. In general, the *greedy* method performs much better than the other two alternatives: *QoE-only* and *random*. This is because, these two methods do not take into consideration at all the cost when making request assignments. Albeit, the curve of *greedy* seems to be close to that of *OPT*, this latter still brings a considerable saving. For example, under this particular parameter settings, OPT saves 16689.78, 25023.04, 18960.05, and 18960.49 US dollars for each month under the four groups of h_p separately, compared with *greedy*.

Next, we dissect how portions of traffic are distributed using different methods. From Fig. 5, we can see that *OPT* prioritize using CDN-0's own infrastructure under Group-1 $\{h_p\}$, since Group-1's hosting prices are relatively small. Further, *OPT* prefers CDN-2 to CDN-1, since the average price of CDN-2 is lower than that of CDN-1 in general. But it still has to seek help from CDN-1, since CDN-2 cannot satisfy the performance target in areas like China, Brazil, and Australia. This point is further reflected by the traffic distribution of *greedy*. From Table II, we observe that CDN-1 provides the best performance across all the areas, which arises from the far-flung footprint of CloudFront as a global CDN provider. By virtue of this fact, *QoE-only* heavily relies on CDN-1. On the other hand, *random* equally uses the three CDNs and will suffer from the negative effects of using multiple CDNs caused by the payment discount policy, when the total traffic is of medium size, as in this case.

Finally, we see how the traffic distribution changes when the average hosting price of CDN-0 increases. As shown in Fig. 6, when the hosting costs of PoPs increase, *OPT* begins to rely more on other CDNs, particularly CDN-2, in servicing requests. The *greedy* approach reflects such change as well, but not as evidently as does *OPT*, as shown in Fig. 7.

V. Conclusion

This paper is motivated by the content multi-homing problem in [3]. Content multi-homing is economically inefficient, particularly for the myriad of small content providers. We propose an alternative approach where each content provider only signs the contract with the authoritative CDN-0, and CDN-0 takes charge of handling the potential QoE trap of any single CDN, by utilizing other CDNs to fulfill its business commitment. We identify the MCDN-CM problem that helps the operator of CDN-0 make an optimal operating plan in terms of minimizing its operating cost. Under realistic settings, the experimental results show that our algorithm for solving MCDN-CM results in considerable amount of money saving. In the future work, we plan to study the impacts of renting other CDNs' services on defining the pricing plan of CDN-0. For example, can CDN-0 raise its price without loosing potential customers?

VI. Acknowledgments

This work was in part supported by UGC grant FS-GRF14EG24.

References

[1] V. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-cdn movie delivery," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1620–1628.

[2] V. Adhikari, Y. Guo, F. Hao, V. Hilt, and Z.-L. Zhang, "A tale of three cdns: An active measurement study of hulu and its cdns," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, March 2012, pp. 7–12.

[3] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian, "Optimizing cost and performance for content multihoming," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '12. New York, NY, USA: ACM, 2012, pp. 371–382.

[4] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind akamai: inferring network conditions based on cdn redirections," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1752–1765, Dec. 2009.

[5] N. B. Ben Niven-Jenkins, Francois Le Faucheur, "Content distribution network interconnection (cdni) problem statemen-rfc6707," http://datatracker.ietf.org/doc/draft-ietf-cdni-problem-statement/, 2012.

[6] Amazon. Amazon cloudfront pricing. http://aws.amazon.com/cloudfront/pricing/.

[7] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '12. New York, NY, USA: ACM, 2012, pp. 211–222.

[8] Q. Zhang, Q. Zhu, M. F. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *Proceedings of the 2012 IEEE 32nd International Conference on Distributed Computing Systems*, ser. ICDCS '12, 2012, pp. 526–535.

[9] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*. Springer, 1996.

[10] P. M. Nahapetyan, A. Pardalos, "A bilinear relaxation based algorithm for concave piecewise linear network flow problems," *Journal of Industrial and Management Optimization*, vol. 3, no. 1, pp. 71–86, 2007.

[11] X. Cheng, C. Dale, and J. Liu, "Dataset for statistics and social network of youtube videos," Simon Fraster University, Canada. http://netsg.cs.sfu.ca/youtubedata/, 2008.

[12] Amazon. Amazon ec2 pricing. http://aws.amazon.com/ec2/pricing/.

Appendix

Proof of Theorem 1: To prove the theorem, we will show that any optimal solution to one of the two problems is a feasible solution to the other with identical objective value.

Suppose $\tilde{\mathbf{x}}$ is an optimal solution to MCDN-CM-E. We may construct the vector $\tilde{\mathbf{y}}$ according to (12). That is, for each $r \in R$, if $\tilde{x}_r \in [v_r^{\ell-1}, v_r^{\ell})$, let $\tilde{y}_r^{\ell} = 1$; and $\forall l \neq \ell, 1 \leq l \leq n_r$, let $\tilde{y}_r^l = 0$. Since $\tilde{\mathbf{x}}$ satisfies constraints (2), (3), (4), (5), and (9), and $\tilde{\mathbf{y}}$ satisfies constraints (14), (15), and (16), $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is a feasible solution to MCDN-CM-ER. Further, $f_k^r(\tilde{x}_r) = u_r^{\ell}\tilde{x}_r + e_r^{\ell}$, where $\tilde{x}_r \in [v_r^{\ell-1}, v_r^{\ell})$. Thus, the objective value of MCDN-CM-ER given by the feasible solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is identical to the optimal objective value of MCDN-CM-E.

Next, we show the opposite direction. Suppose $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is an optimal solution to MCDN-CM-ER. First, it is easy to check that $\tilde{\mathbf{x}}$ is also feasible to MCDN-CM-E. Second, we need to show that the objective values of the two problems are identical. Note that the objective functions of the two—(8) and (13) – differs only in the first part. Thus, we only need to check whether $f_k^r(\tilde{x}_r)$ equals $\sum_l \tilde{y}_r^l\big(u_r^l\tilde{x}_r + e_r^l\big)$. Or put in another way, we examine whether $\tilde{y}_r^l$ respects (12).

By fixing the value of vector $\mathbf{x}$ as $\tilde{\mathbf{x}}$, the MCDN-CM-ER problem reduces to a linear program with respect to $\mathbf{y}$, that is, the problem LP($\tilde{\mathbf{x}}$). Note that LP($\tilde{\mathbf{x}}$) can be fully separated into $|R|$ subproblems for each $r \in R$. Hence, in the following, we only focus on a specific region r. Since $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is an optimal solution to MCDN-CM-ER, $\tilde{\mathbf{y}}$ must be the optimal solution to LP($\tilde{\mathbf{x}}$) as well. On the other hand, we assume that $\tilde{x}_r \in [v_r^{\ell-1}, v_r^{\ell}]$. That is, in the optimal solution $\tilde{\mathbf{x}}$, the value $\tilde{x}_r$ lies in the segment indicated by ℓ. Recall (11) and thus $f_r^l(\tilde{x}_r) = \min\limits_{1 \leq l \leq n_r} \{u_r^l\tilde{x}_r + e_r^l\} = u_r^{\ell}\tilde{x}_r + e_r^{\ell}$. Based on this, we construct a new value $\hat{\mathbf{y}}_r$ of vector $\mathbf{y}_r$: let $\hat{y}_r^{\ell} = 1$; and for any $l \neq \ell$, let $\hat{y}_r^l = 0$. Then the newly constructed vector $\hat{\mathbf{y}}_r$ satisfies: i) the constraint (14), because $\tilde{x}_r \in [v_r^{\ell-1}, v_r^{\ell}]$; and ii) $\hat{\mathbf{y}}_r = \arg\min\limits_{y}\{\sum\limits_l y_r^l(u_r^l\tilde{x}_r + e_r^l) \mid \sum\limits_l y_r^l = 1, y_r^l \geq 0\}$. Hence, $\hat{\mathbf{y}}_r$ is an optimal solution to the subproblem of LP($\tilde{\mathbf{x}}$) with respect to r. A similar result hold for any other $r \in R$. Therefore, $\hat{\mathbf{y}}$ is an optimal solution to LP($\tilde{\mathbf{x}}$). Since $\tilde{\mathbf{y}}$ is also an optimal solution to LP($\tilde{\mathbf{x}}$), we conclude $(\tilde{\mathbf{x}}, \hat{\mathbf{y}})$ is also an optimal solution to MCDN-CM-ER[3]. These two facts: i) $f_r^l(\tilde{x}_r) = \sum_l \hat{y}_r^l\big(u_r^l\tilde{x}_r + e_r^l\big)$, according to how we construct $\hat{\mathbf{y}}_r$; and ii) $\sum_l \hat{y}_r^l\big(u_r^l\tilde{x}_r + e_r^l\big) = \sum_l \tilde{y}_r^l\big(u_r^l\tilde{x}_r + e_r^l\big)$, because both $\tilde{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are optimal to LP($\tilde{\mathbf{x}}$), allow us to conclude that $f_k^r(\tilde{x}_r)$ is equal to $\sum_l \tilde{y}_r^l\big(u_r^l\tilde{x}_r + e_r^l\big)$. So we finish proving that the objective value of MCDN-CM-E given by $\tilde{\mathbf{x}}$ equals the objective value of the optimal solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ of MCDN-CM-ER. ■

Proof of Theorem 2:

Let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be the returned solution of Alg. 1. For notational simplification, let $g(\mathbf{x}, \mathbf{y})$ represent the objective function (13) of MCDN-CM-ER and let $\mathcal{D}$ be the feasible region. By virtue of the stopping condition in Step 3 of Alg. 1, we get: (i) $\tilde{\mathbf{x}} = \arg\min\limits_{\mathbf{x}}\{\text{MCDN-LP}(\tilde{\mathbf{y}})\}$; and (ii) $\tilde{\mathbf{y}} = \arg\min\limits_{\mathbf{y}}\{\text{MCDN-LP}(\tilde{\mathbf{x}})\}$.

(ii) leads to $\tilde{\mathbf{y}} = \arg\min\limits_{(\tilde{\mathbf{x}}, \mathbf{y}) \in \mathcal{D}} g(\tilde{\mathbf{x}}, \mathbf{y})$. Since $\tilde{\mathbf{y}}$ satisfies the constraint (14) already (even though (14) is missing in MCDN-LP($\tilde{\mathbf{y}}$)), combined with (i), we get $\tilde{\mathbf{x}} = \arg\min\limits_{(\mathbf{x}, \tilde{\mathbf{y}}) \in \mathcal{D}} g(\mathbf{x}, \tilde{\mathbf{y}})$. The above two equations show that $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \arg\min\limits_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} g(\mathbf{x}, \mathbf{y})$. Therefore, the solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ returned by Alg. 1 is an optimal solution of MCDN-CM-ER. ■

Proof of Theorem 3:

Theorem 3 can be proven in the same approach as Theorem-6 in [10]. ■

[3] Note that in this proof, we don't care whether $\tilde{\mathbf{y}} = \hat{\mathbf{y}}$.