Tapping the Knowledge of Dynamic Traffic Demands for Optimal CDN Design

Guoming Tang, *Member, IEEE*, Huan Wang, Kui Wu, *Senior Member, IEEE*, and Deke Guo, *Member, IEEE, ACM*

***Abstract*— The content delivery network (CDN) intensively uses cache to push the content close to end users. Over both traditional Internet architecture and emerging cloud-based framework, cache allocation has been the core problem that any CDN operator needs to address. As the first step for cache deployment, CDN operators need to discover or estimate the distribution of user requests in different geographic areas. This step results in a statistical spatial model for the user requests, which is used as the key input to solve the optimal cache deployment problem. More often than not, the temporal information in user requests is omitted to simplify the CDN design. In this paper, we disclose that the spatial request model alone may not lead to truly optimal cache deployment and revisit the problem by taking the dynamic traffic demands into consideration. Specifically, we model the time-varying traffic demands and formulate the distributed cache deployment optimization problem with an integer linear program (ILP). To solve the problem efficiently, we transform the ILP problem into a scalable form and propose a greedy diagram to tackle it. Via experiments over the North American ISPs points of presence (PoPs) network, our new solution outperforms traditional CDN design method and saves the overall delivery cost by 16% to 20%. We also study the impact of various traffic demand patterns to the CDN design cost, via experiments with both real-world traffic demand patterns and extensive synthetic trace data.**

***Index Terms*— Content delivery network, time-varying traffic demands, optimal CDN design.**

I. Introduction

A CONTENT delivery network (CDN) is a globally distributed network system deployed across the Internet. Composed with geographically distributed cache servers, CDNs deliver cached content to customers worldwide based on their geographic locations. Extensively using cache servers, content delivery over CDN has low latency and high reliability, and supports better quality of experience (QoE) [1], [2]. Nowadays, CDNs are serving a big portion of Internet traffic. For example, the videos delivered by the Netflix CDN accounts for 37% of the total Internet traffic in North American during peak hours [3]; the leading CDN service provider Akamai claims to deliver more than 20% of the Internet Web traffic [4].

Due to CDNs' significant role in the current Internet ecosystem, companies like Google, Limelight, Microsoft, and Amazon, as well as Internet Service Provides (ISPs) like AT&T, Orange, Swisscom and KT, are planning or have already launched their own CDNs. In the current market, two predominant architectures are adopted [5]: the *scattered* CDN and the *consolidated* CDN. The scattered CDN has footprints across a huge number of Points of Presence (PoPs) while each of which has medium or low cache capacity, e.g., Akamai has deployed more than $100,000$ cache servers in more than $1,800$ locations covering thousands of autonomous systems (AS) [4]. In contrast, the consolidated CDN only chooses a small number of key PoPs co-located with high-capacity datacenters, e.g., YouTube employs seven datacenters that are connected to six tier-1 ISP ASes to deal with user demands [6].

Scattered or consolidated, CDN designers need to answer two critical questions:

1) **PoP Selection**: What are the optimal PoPs for cache deployment, given hundred of thousands potential venues all over the world?
2) **Cache Deployment**: What is the optimal caching capacity (the number of servers, storage volume, bandwidth, etc.) that should be configured at each selected PoP?

The above problems must be answered not only for new CDN entrants, but also for those existed ones that are growing and expending. The answers to the above problems benefit CDN operators with the most cost effective CDN design that satisfies desired QoE to end customers.

A. Motivation: From Spatial to Spatio-Temporal

Substantial research efforts have been devoted to seeking optimal solutions for CDN design, resulting in specific guidelines and principles in PoP selection and cache deployment [7], [8]. In prior work, the time-varying traffic demands at PoPs candidates were simplified to ease the optimal design. More often than not, the average demand at each PoP over a long term was used to represent the geographical distribution of traffic demands [7]. In other words, only spatial pattern was considered while the temporal domain was largely ignored.

As real-world measurements at two CDN PoPs, Fig. 1 shows their spatio-temporal patterns of Internet traffic demands. By aligning the two PoPs' traffic demands with the Greenwich Mean Time (GMT), we have two observations (that are also

Manuscript received November 11, 2017; revised September 11, 2018; accepted November 8, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor E. Yeh. Date of publication November 29, 2018; date of current version February 14, 2019. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, Ericsson Canada Inc. under a Collaborative Research and Development Grant CRDPJ 488453-15; in part by the National Natural Science Foundation of China under Grant 61802421 and Grant 61772544; and in part by the NUDT Research Plan under Grant ZK17-03-50. *(Corresponding authors: Guoming Tang; Deke Guo.)*

G. Tang is with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: gmtang@nudt.edu.cn).

H. Wang and K. Wu are with the Department of Computer Science, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: huanwang@uvic.ca; wkui@uvic.ca).

D. Guo is with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China, and also with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: dekeguo@nudt.edu.cn).

Digital Object Identifier 10.1109/TNET.2018.2881169

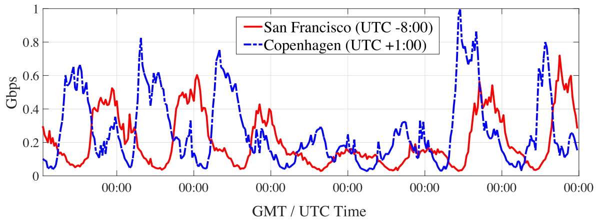


Fig. 1. One-week traffic demands of two Amazon cache servers at San Francisco, US and Copenhagen, Denmark, respectively.

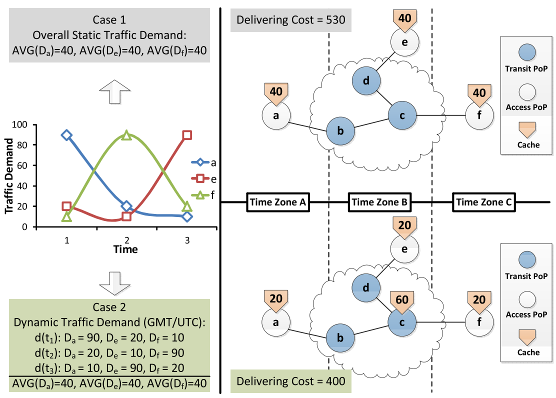


Fig. 2. An example for initial verification: static traffic analysis vs. dynamic traffic analysis.

applicable to other PoPs): i) the traffic demands at each PoP are time-varying and have large variations in each cycle (one day), and ii) there exists a shift of traffic demands at different PoP locations (or time zones), especially those with a large geographical distance. Nevertheless, such spatio-temporal patterns of distributed traffic demands have not been fully explored during the stage of CDN design. A systematic study of their impacts on CDN design, w.r.t. PoP selection and cache deployment is highly demanded.

In this paper, we refer to the spatial pattern used by prior work as *static traffic* and the spatio-temporal pattern shown in our context as *dynamic traffic*, because the former does not include time domain in the model while the latter does. By extending the design space from spatial to spatio-temporal, we will show that previous work has large room to improve.

B. Example: An Initial Verification

By considering the static traffic, existing work usually leads to a good solution where the selected PoPs are close to the end-users as much as possible and the deployed cache can locally satisfy the average traffic demand at each PoP. Nevertheless, the existing solutions can be further improved by exploring the statistical pattern in dynamic traffic.

Assuming an arbitrary target network across multiple time zones, Fig. 2 provides a simple yet concrete example to compare the optimal cache design when considering static traffic and dynamic traffic, respectively. In this example, the CDN design cost is defined as the total traffic-delivery cost, calculated as the traffic amount times its delivery hop distance. We solve two cost minimization problems under static and dynamic traffic, respectively. The detailed formulation of the optimization problem will be introduced in Sec. III. In case one, using the static traffic demands at the three access PoPs,[1] the CDN design only needs to follow the demands locally, resulting in the cache deployment shown on the right top topology. In case two, when the time-varying traffic demands are considered, with the result from our solution provided in Sec. III, 50% of the total cache deployed in case one will be re-distributed to the transit PoP c shown on the right bottom topology. As a result, the total delivery cost can be saved by 25%!

C. Challenges & Contributions

The above observation directly motivates our work in this paper. Actually, the benefit of using the dynamic traffic has been validated in other network economic domains [9], and in this work we revisit the CDN design problem by considering the dynamic traffic instead of the static one. Dynamic traffic analysis, however, poses challenges not existing before. First, how can we model the dynamic traffic and integrate it in the optimization problem to yield the optimal solution? Second, the newly introduced variables in the time dimension significantly increase the scale of the optimization problem. How can we effectively reduce the complexity and efficiently solve the large-scale optimization problem? Third, the CDN operator cannot obtain real spatio-temporal content demands from end users until the caches are deployed. Thus, it is important to solve the "cold start" problem caused by lack of traffic demand traces. In addition to the above challenges, more subtle issues need to be addressed, e.g., how much potential benefits (w.r.t. cost and delay) can be achieved by considering the dynamic traffic in CDN design? How does constraints like operating budget and number of candidate PoPs impact the design cost? Will the content type (i.e., demand pattern) impact the CDN design?

The paper addresses all the above questions and makes following contributions:

- Aiming at minimizing the total cost as well as the delivery latency, we model the dynamic traffic and integrate the model into the problem formulation of CDN design. Our system models are quite generic such that CDN operators can easily tailor the models to their own cases.
- To solve the optimal CDN design problem involving time-varying and integer variables, we first apply a special dimension-shrinking method to re-formulate the problem to a classic integer linear programming (ILP) problem; then we propose a greedy diagram for PoP selection and cache deployment to tackle the ILP problem efficiently.
- To deal with the "cold start" problem, we borrow the idea of transfer learning and utilize the knowledge of user demands from existing CDNs to the design of a new CDN. We conduct extensive experiments over the North America ISPs PoP networks. The results demonstrate that, compared with the traditional solution with static traffic demand analysis, ours can help the CDN operator save the overall delivery cost by 16% to 20%.
- We investigate the performance of our dynamic traffic-based solution under constraints of operating budget and various traffic satisfaction ratios, and test the robustness of our solution. The experimental results provide CDN operators with abundant knowledge for a better CDN design.
- We investigate CDN design with different content types and investigate how the characteristics of traffic demand patterns affect the traffic delivery cost. Specifically, we capture traffic demand patterns with mean value,

[1] According to the location of a PoP, it is called an *access PoP* if it peers to an access ISP or a *transit PoP* if it peers to a transit ISP. If a PoP peers to both access and transit ISPs, it is regarded as an access PoP.

variance and fluctuation index, which will be defined in Section VII. The impacts of these metrics on cost savings are studied and guidelines are provided to the CDN operators.

The rest of the paper is organized as follow: We review the related work and the state-of-the-arts in Sec. II. Then, we formulate the CDN design problem by considering dynamic traffic in Sec. III. The problem is transformed and re-formulated with a dimension-shrinking strategy and corresponding solution is developed in Sec. IV. In Sec. V, we set up experimental environment by extracting realistic ISPs PoP networks and collecting real-world traffic demands. Extensive experiments are conducted in Sec. VI to demonstrate the advantages of our solution and evaluate its performance and robustness under different situations. Sec. VII further investigates the relationship between CDN design cost and traffic demand pattern. Sec. VIII concludes the paper.

II. Related Work

Research on CDNs can be roughly grouped into two categories: (1) long-term network planning, including optimized CDN design that relates to PoP selection and cache deployment, and (2) short-term, run-time cache management, including content replacement and prefetching strategies in the CDN network. This paper belongs to the first category.

In the first category, the cache placement problem in a given network was investigated in [10], where both the general caches and transparent en-rout caches (TERCs) were taken into consideration. Instead of using real-world network topologies, the authors looked into the special type of network topologies (line and ring networks) and designed algorithms to minimize the overall traffic flow or average traffic delay. Furthermore, the per-week overall traffic requests were used in the experimental evaluation, which is different from our dynamic traffic-based analysis. Aiming at improving aggregate throughput of CDNs, the work in [8] proposed a simple model to quantify the tradeoff between selecting better traffic-delivery path and increasing the number of CDN PoPs. This investigation focuses mainly on the CDNs where an ISP has full control on both CDN servers and network paths, and thus can perform a joint optimization of server selection and path selection. Dynamic traffic, however, is not investigated in [8]. The trade-offs between the performance and the cost of CDN provision was explored in [11], in which the authors presented a decision support system to help CDN operators systematically investigate and evaluate different CDN design trade-offs. Their proposed decision support system, although providing CDN vendors with evaluations on the trade-offs at different levels, did not optimize the content delivery cost.

Prior researches also illustrate the benefits of considering dynamic traffic patterns in CDN design on both performance [12], [13] and economic aspects [14]. Shafiq *et al.* [13] conducted a large-scale measurement study to characterize the time-varying user request patterns and explored their impacts on the CDN performance metrics. Their results showed that the traffic patterns of different content types varies much and could impact the caching strategies greatly. To some degree, it validates the motivation of our work while lacking of detailed cache deployment strategies. Castro and Gorinsky [14] investigated the economic benefits of content delivery by proposing a hybrid ISP interconnection paradigm, considering the traffic patterns of different ISPs. Different from our work this paper focused on the interconnection strategy of multiple ISPs (so that the overall transit/peering costs of a specific ISP could be reduced).

As the most related work to ours, Hasan *et al.* [7] formulated the cache deployment optimization problem with a mix integer program and solved it for AS-level topologies. The key difference to our work is that it made use of static average traffic demand at each AS. Besides the general-purpose CDN network design and cache deployment, particular CDN systems were also explored for specific content delivering, such as the rich content multimedia [15] or live video stream [16].

In the second category, a broad spectrum of work focused on short-term content/replica placement and cache management. In [17], to find the optimal on-demand content delivering in short term (e.g., one-week), Applegate et al. formulated and solved the video content placement problem with the constraints of storage space and link bandwidth. The algorithm was evaluated with an arbitrary network and with trace demand data from a real-world VoD service. In [18], to minimize the bandwidth cost incurred by the VoD content in a distributed CDN cluster, Borst et al. formulated the content placement problem as a LP problem and developed light-weight cooperative cache management algorithms to solve the problem. Pedarsani *et al.* [19] proposed the online caching scheme for content placement and delivery, which reduced network load by creating and exploiting coded multicasting opportunities between users with different demands. This approach was further developed in [20] by eliminating the limitation of the central coordinating server and enabling content placement performed in a decentralized manner. Recent works related to the contemporary content replacement as well as load balance over the CDN networks were presented in [2], [21], and [22].

III. System Model and Problem Formulation

In this section, we model the time-varying traffic demands from the CDN users and formulate the distributed cache deployment optimization problem. For ease of reference, we list the notations in Table I.

A. Model of Target Network

We design a CDN over a target network composed by PoP candidates and PoP links. The PoP candidates are potential places for cache deployment. They usually peer to one or multiple ISPs (either access ISPs or transit ones) via the Internet eXchange Points (IXPs) [23], [24], and depending on the categories of their peering ISPs, they can be classified as access PoPs or transit PoPs. A pair of PoPs are linked to each other if their peering ISPs are linked.

Without loss of generality, we model the topology of the target network with a directed graph $G(V, E)$, where V is the set of vertices denoting the PoPs, and E is the set of edges denoting the transit and peering links between PoPs. We assume that the topology of the target network is known in advance and there are m PoPs across the target network, i.e., $|V| = m$.

B. Model of Traffic Demand

The end-users of the CDN are distributed across the target network and have time-varying content demands towards the CDN caches. At a time instant, we define the overall rate of content demands (e.g., measured in unit of Mbps or Gbps) from the end-users at PoP i as the traffic demands of PoP i at that time instant.

TABLE I
SUMMARY OF NOTATIONS

Notation	*Description*
G	graph of PoP network topology
V	vertices (PoPs) set of G
E	edges (links) set of G
T	a time period divided into consecutive slots
m	total number of PoPs
n	total number of time slots
t	time index under GMT/UTC
$d_i(t)$	traffic demand of PoP i at time t
D	peak traffic demand from all PoPs
α_{min}	minimum satisfaction ratio
$\alpha(t)$	satisfaction ratio of overall traffic demands at time t
x	PoP selection vector
y	cache capacity vector
z	traffic supply fraction matrix
$z_{ij}(t)$	traffic supply fraction of PoP i from PoP j at time t
K	budget of cache capacity cost
N	threshold of the total number of cached PoPs
Ψ	overall traffic-delivery cost
r_i	unit cache price at PoP i
h_{ij}	hop distance from PoP j to PoP i
$\ell_i(t)$	average delivery distance of PoP i at time t
$\overline{\ell}(t)$	average delivery distance over the network at time t
$\overline{\ell}$	average delivery distance over the network in period T

Consider a time period of interest. We evenly split the time into n consecutive slots,[2] $T = \{1, 2, \cdots, n\}$. We capture the traffic demands of PoP i at time t using the following generation model [25]:

$$d_i(t) := p(i) \cdot q(t), \quad \forall i \in V, \forall t \in T, \tag{1}$$

where $p(\cdot)$ represents the population of specified PoP location and $q(\cdot)$ is the normalized traffic profile whose values can be related to specified traffic content (e.g., text, picture or video).

In the stage of CDN design, due to lack of knowledge at time dimension, i.e., $q(t)$ in (1), the traffic demands at one PoP are usually simply estimated with its population $p(i)$, which leads to the static traffic analysis in prior work. In Sec. V-B, we will disclose how the idea of transfer learning can be utilized to obtain time-varying traffic demands at distributed PoPs.

Remark 1: The time index t, ($t \in T$) in this paper is labeled based on the Greenwich Mean Time (GMT) under the Coordinated Universal Time (UTC) standard. Thus, given the same time index, the local time of different PoPs can vary depending on their located time zone, e.g., the time index denoting 12:00pm (GMT/UTC) means 8:00pm for a PoP at Beijing, China (UTC+08:00), but 7:00am for a PoP at New York, US (UTC-05:00).

Given the dynamic traffic at each PoP in the target network, the *peak traffic demand* over the whole network during the n time slots can be determined as:

$$D = \max_{t \in T} \left\{ \sum_{i \in V} d_i(t) \right\}. \tag{2}$$

Moreover, we assume a *minimum satisfaction ratio* α_{min}, $0 \le \alpha_{min} \le 1$, of the overall traffic demands that should be guaranteed by the CDN operator in each time slot. The value of α_{min} being 1 indicates that the traffic demands to any PoP need to be completely (100%) satisfied.

[2]In our later experimental evaluation, each time slot is one hour.

C. Model of PoP Selection & Cache Deployment

To model the PoP selection and cache deployment processes, we first construct a *PoP selection vector* to indicate which PoPs are selected as targets for cache deployment (named cached PoPs), denoted by:

$$x := [x_1, x_2, \cdots, x_m]^\intercal, \tag{3}$$

where $x_i \in \{1, 0\}$ indicating whether or not PoP i is cached.

Then, we further construct a *cache capacity vector* to represent the capacity of caches deployed at cached PoPs by:

$$y := [y_1, y_2, \cdots, y_m]^\intercal, \tag{4}$$

in which y_i denotes the cache capacity deployed at PoP i. The cache capacity here refers to the capability (provided by server and network deployment) to satisfy the traffic demands of PoPs and can be measured by the overall rate of traffic supply. When $y_i > 0$, PoP i has been cached and $x_i = 1$; otherwise, PoP i is not cached and $x_i = 0$. Note that both x_i's and y_i's are key variables that should be determined by the CDN design.

With the peak traffic demand D and minimum satisfaction ratio α_{min}, the upper bound of total demands that need to satisfy is $\alpha_{min}D$. In other words, the total deployed cache size should be no less than $\alpha_{min}D$. To save cost, we set

$$\sum_{i \in V} y_i = \alpha_{min} D. \tag{5}$$

Here we assume that all (assigned) caches are accessible during one time slot, since compared with the length of time slot (e.g., one-hour in our later evaluation), the time of content delivery between PoP nodes (usually in unit of millisecond) is negligible.

With the total cache capacity deployment determined by (5), the satisfaction ratio of overall traffic demands that can be provided by the CDN operator at any time is:

$$\alpha(t) = \begin{cases} \dfrac{\alpha_{min} D}{\sum_{i \in V} d_i(t)}, & \text{if } \alpha_{min} D < \sum_{i \in V} d_i(t), \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

D. Model of Traffic Supply

Next, we define a three dimensional variable named *traffic supply fraction* as:

$$z := \{z_{ij}(t), \forall i, j \in V, \forall t \in T\}, \tag{7}$$

where $z_{ij}(t)$ represents the fraction of traffic demands at PoP i supplied by PoP j at time t. Note that each PoP can also supply traffic to itself (i.e., $i = j$), which results in the "on-net" traffic [7].

Obviously, the value of traffic supply fraction with respective to the cahced PoP is between $[0, 1]$, and the PoP without cache deployment has a traffic supply fraction value of zero. We model these two facts by the following constraint:

$$0 \le z_{ij}(t) \le x_j, \quad \forall i, j \in V, \forall t \in T, \tag{8}$$

where x_j is boolean and defined in (3).

Moreover, for any time instant t, i) the satisfaction ratio of overall traffic demands should be no less than $\alpha(t)$, and ii) the traffic supply of each cached PoP should not exceed its cache capacity. Thus, we have the following two constraints:

$$\sum_{j\in V} z_{ij}(t) \geq \alpha(t), \quad \forall i \in V, \forall t \in T, \tag{9}$$

$$\sum_{i\in V} d_i(t) z_{ij}(t) \leq y_j, \quad \forall j \in V, \forall t \in T. \tag{10}$$

Remark 2: It is well known that a CDN, after being deployed, relies on some proprietary load balance strategy to re-direct traffic from one PoP to another. This does not mean, however, the fraction of traffic demands at PoP i supplied by PoP j at time t can only be determined at CDN run time. To avoid confusion, it is worthwhile to point out the practical meaning of z during the CDN design stage: the values in z provide a guideline on the overall traffic amount (or a total budget) that one PoP can serve another PoP in the long term.

E. The Problem of Optimal CDN Design

The main cost of a CDN system can be roughly divided into two components: traffic-delivery cost and facility cost [8]. The first cost occurs when contents are delivered across the network, as the CDN operator needs to pay the ISPs for bandwidth at the peering points or transit links (the so-called customer-provider links). The second cost is related to PoPs where the cache servers are deployed, as the CDN operator needs to pay for the equipment and energy [26].

On the current utility market, the facility cost is relatively stable and fairly predictable for the CDN operator. The traffic-delivery cost, however, can differ significantly depending on the chosen CDN design strategies (i.e., PoP selection and cache deployment). Bad PoP selection and unbalanced cache deployment will lead to longer content delivery paths, and thus higher traffic-delivery cost is compounded when the longer paths are used repeatedly over time. *Due to the above considerations, in this paper we focus on minimizing the traffic-delivery cost across the target network.*

Meanwhile, we deal with the facility cost of CDN design by considering the following two more constraints.

- Cache Budget Constraint: the overall cost of deployed cache capacity is upper-bounded by a financial budget K, formulated by:

$$\sum_{i\in V} r_i y_i \leq K, \tag{11}$$

  where r_i is the unit cache price at PoP i. Note that the unit cache price r_i may vary from one PoP to another, depending on the local market, and we regard r_i as a known parameter.
- PoP Selection Constraint: the total number of cached PoPs is upper-bounded by a threshold N, given by:

$$\sum_{i\in V} x_i \leq N, \tag{12}$$

  where N is a natural number predefined by the CDN operator, mainly determined by the expected scale and financial investment of the designed CDN.

We ignore the diversity of price schemes of transit/peering links and simply treat all links equally. This simplification eases the analysis, but does not compromise the power of our model, because our model can be easily extended by adding different weights to the links.

Thus, during a time period T, the traffic-delivery cost can be calculated by the total traffic-delivery amount (i.e., the traffic burden over target network):

$$\Psi := \sum_{t\in T}\sum_{i\in V}\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t), \tag{13}$$

where h_{ij} denotes the shortest hop distance between PoPs i and j, with values computed by the shortest path algorithm (e.g., the Dijkstra algorithm is used in our implementation) in the target network.

Note that with the knowledge of billing models, the objective function can be reformulated to reflect the monetary expenditure, which is the primary interest of the CDN operator. For example, with known pricing schemes of the target network, the hop distance ($h_{i,j}$) can be replaced by a weighted (priced) "distance" (e.g., denoted by $w_{i,j}$) between PoP i and PoP j, indicating the monetary cost of traffic-delivery in unit of $\$/Gbps$. Thus, the objective function, in form of the product of $w_{i,j}$ and $d_i(t)z_{i,j}(t)$, will result in the measurement directly revealing the traffic-delivery cost, in unit of $\$/Gbps \cdot Gbps = \$$. The details of billing models, however, are CDN specific, and different CDN operators may use quite different pricing strategies to promote their market share. Reformulating the objective function in unit of momentary cost to jointly study different marketing strategies is thus an interesting future research topic.

The goal of our optimal CDN design problem is to minimize the traffic-delivery cost of the target network over a time period, i.e.,

$$\min_{\{x,y,z\}} \ \Psi \tag{14a}$$

$$\text{s.t.} (5), (8) \sim (12). \tag{14b}$$

Thus, by incorporating the time-varying traffic demands, we have formulated the optimal CDN design (i.e., PoP selection and cache deployment) problem with the above integer linear program (ILP) problem. The time-varying factors and integer variable make the problem not easy to tackle, and we will show our solution in the following section.

IV. Model Analysis and Solution Methodology

In this section, we first analyze the optimization problem in (14) and its hardness. Then we re-formulate it using a dimension-shrinking strategy and solve it by applying a greedy diagram.

A. Delivery Cost vs. Delivery Latency

At any time instant t, the average delivery distance for traffic towards PoP i can be estimated by [7]:

$$\ell_i(t) := \frac{\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t)}{\sum_{j\in V} d_i(t) z_{ij}(t)}, \tag{15}$$

where the traffic delivery (PoP hop) distance to PoP i at time t is weighted by the fraction of traffic volume along the delivery path. Thus, the average delivery distance of the overall traffic at time t can be computed by:

$$\bar{\ell}(t) := \frac{\sum_{i\in V}\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t)}{\sum_{i\in V}\sum_{j\in V} d_i(t) z_{ij}(t)}, \tag{16}$$

and the average delivery distance of the overall traffic across the whole network during the considered time period T can be measured by:

$$\bar{\ell} := \frac{\sum_{t\in T}\sum_{i\in V}\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t)}{\sum_{t\in T}\sum_{i\in V}\sum_{j\in V} d_i(t) z_{ij}(t)}. \tag{17}$$

Lemma 1: In the optimization problem (14), the objective of minimizing the traffic-delivery cost is equivalent to the objective of minimizing the average delivery distance of the overall network traffic, i.e., $\Psi \propto \bar{\ell}$.

Proof: Note that the volume of delivered traffic from all PoPs at time slot t should be equal to that of the total traffic demands multiplying the satisfaction ratio at the corresponding time slot. Thus, we have:

$$\sum_{i\in V}\sum_{j\in V} d_i(t) z_{ij}(t) = \alpha(t)\sum_{i\in V} d_i(t), \tag{18}$$

where $\alpha(t)$ is the traffic satisfaction ratio at time t given by (6). Then, for the whole time period of T, we have:

$$\sum_{t\in T}\sum_{i\in V}\sum_{j\in V} d_i(t) z_{ij}(t) = \sum_{t\in T}\left(\alpha(t)\sum_{i\in V} d_i(t)\right). \tag{19}$$

Substituting (19) into (17), the average delivery distance of the overall traffic during time period T can be rewritten as:

$$\bar{\ell} = \frac{\sum_{t\in T}\sum_{i\in V}\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t)}{\sum_{t\in T}\left(\alpha(t)\sum_{i\in V} d_i(t)\right)}. \tag{20}$$

Recall that in the objective function of problem (14), only $z_{ij}(t)$ are variables and thus all the others are known parameters (i.e., the denominator in (20) is a constant value). Hence:

$$\Psi = \sum_{t\in T}\sum_{i\in V}\sum_{j\in V} h_{ij} d_i(t) z_{ij}(t) \propto \bar{\ell}. \tag{21}$$

□

Remark 3: Assuming that the traffic-delivery distance is proportional to the traffic latency, Lemma 1 implies that the objective of problem (14) is equivalent to minimizing the traffic latency in the target network. This indeed aligns with the latency-oriented traffic delivering of real-world CDNs design.

For the optimization problem of (14), we have realized that it is not readily solvable due to the following two reasons: i) a large number of time-varying factors ($z_{ij}(t)$ and $d_i(t)$, $\forall t \in T$) exist and make the problem much complex, and ii) the boolean variables (x_j, $\forall j \in V$) make the problem an integer program and NP hard.

To deal with the above problems, in the following sections, we first eliminate the time-varying factors and reformulate the problem via a network transformation scheme. Then a greedy approach is proposed to solve the (relaxed) integer programming problem using the off-the-shelf LP solver.

B. Transformation of Time-Varying Factors

Consider the traffic supply fraction $z_{ij}(t)$ defined in (7), where $i, j \in V$, $t \in T$ and $|V| = m, |T| = n$. We rearrange the three-dimensional variable and convert it to a two-dimensional matrix with a specific layout as follow:

$$\hat{z} := [z(1), z(2), \ldots, z(n)]^\intercal, \tag{22}$$

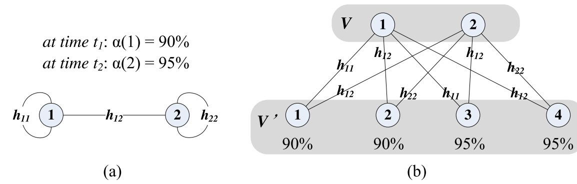


Fig. 3. Transform a two-node PoP network to a virtual one with (25) and (27). (a) original network G. (b) virtualized network G'.

where $z(t)$ represents the traffic supply matrix at time t and is formulated as:

$$z(t) := [z_{ij}(t)]_{m\times m}. \tag{23}$$

Thus, the original three-dimensional variable of traffic supply fraction is converted to a two-dimensional matrix with size of $(m \times n)$-by-m.

The new formulation of traffic supply fraction in (22) can be interpreted as a new (virtual) networking scenario: in a (virtual) network G' composed by two PoP sets V and V', where $|V| = m$ and $|V'| = m \times n$, the cached PoPs in V need to supply traffic to the PoPs in V' and satisfy their demands *at one time*.

In the new scenario, the traffic demands of the PoPs within V' can be denoted by:

$$\hat{d} := [\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_{m\times n}]^\intercal, \tag{24}$$

where $\hat{d}_i$ denotes the traffic demands of PoP $i, i \in V'$.

We denote the hop distance between PoP i in V' and PoP j in V as $\hat{h}_{ij}$. As the virtual network G' is formed with n (equal-size) slides by concatenating the original network for n times (as indicated by Equation (22)), $\hat{h}_{ij}$ is actually equal to the hop distance between PoP k and PoP j in the original target network, where k is the remainder of $i \mod m$, i.e.,:

$$\hat{h}_{ij} := h_{kj}, \forall i \in V', \forall j \in V, \tag{25}$$

in which

$$k = \begin{cases} i \mod m, & \text{if } i \neq m, \\ m, & \text{if } i = m. \end{cases} \tag{26}$$

Furthermore, we set the traffic demand satisfaction ratio of each node in V' as:

$$\hat{\alpha}(i) := \alpha(\lceil \frac{i}{m} \rceil), \quad \forall i \in V', \tag{27}$$

where $\alpha(\lceil \frac{i}{m} \rceil)$ corresponds to the overall traffic demand satisfaction ratio at time $\lceil \frac{i}{m} \rceil$ in the original problem.

Example 1: To help understand the above transformation, consider a two-node PoP network as a simple example. Given the traffic demands of each PoP in two time slots (t_1 and t_2), Fig. 3 illustrates the transformation from the original target network G considering the dynamic traffic to the virtual network G' where the time-varying factor is removed.

C. Problem Reformulation & Solution

After the above transformation, we rewrite the conditions with new notations of (22), (24), (25), and (27). Then, the traffic-delivery overhead over the (virtual) network can be minimized by the following optimization problem:

$$\min_{\{x,y,\hat{z}\}} \sum_{i\in V'} \sum_{j\in V} \hat{h}_{ij}\hat{d}_i\hat{z}_{ij} \tag{28a}$$

$$\text{s.t.} 0 \le \hat{z}_{ij} \le x_j, \quad \forall i \in V', \forall j \in V, \tag{28b}$$

$$\sum_{j\in V} \hat{z}_{ij} \ge \hat{\alpha}(i), \quad \forall i \in V', \tag{28c}$$

$$\sum_{i\in V} \hat{d}_{((k-1)m+i)}\hat{z}_{((k-1)m+i,j)} \le y_j, \quad \forall k \in T, \forall j \in V, \tag{28d}$$

$$\sum_{j\in V} y_j = \alpha_{min} D, \tag{28e}$$

$$\sum_{j\in V} r_j z_j \le K, \tag{28f}$$

$$\sum_{i\in V} x_i \le N. \tag{28g}$$

Specifically, constraint (28d) means that the traffic demands of the k-th slide (i.e., the k-th time slot) in the virtual network G' satisfied by PoP j in the original network G should be no larger than the cache volume deployed at PoP j. Thus, with (28), the time-varying factors are eliminated from the original formulation, and meanwhile we have the following Theorem.

Theorem 1: The optimization problem (28) is equivalent to problem (14).

Proof: Note that Equations (22) and (23) convert a three-dimensional variable (i.e., $z_{ij}(t)$) to a two-dimension matrix (i.e., $\hat{z}$), where the third dimension of $z_{ij}(t)$ is concatenated along the first dimension of $\hat{z}$.

Thus, each element in the objective function of (14) has a corresponding element in that of (28). Specifically, using the transformation defined with (22), (24), (25), and (27), we have:

$$h_{ij}d_i(t)z_{ij}(t) = h_{ij}\hat{d}_{i\times t}\hat{z}_{i\times t,j}, \quad \forall i, j \in V, \ \forall t \in T.$$

As the size of the virtual network is equal to the size of the original network times the number of time slots i.e., $|V'| = |V| \times |T|$, the following equation holds:

$$\sum_{t\in T}\sum_{i\in V}\sum_{j\in V} h_{ij}d_i(t)z_{ij}(t) = \sum_{i\in V'}\sum_{j\in V} \hat{h}_{ij}\hat{d}_i\hat{z}_{ij}.$$

Hence, the objective functions of (14) and (28) are equivalent. Following the same analysis, we can also verify the equivalence of each constraint. Therefore, the two optimization problems are equivalent. □

The integer programming problem in (28) belongs to the Capacitated Facility Location (CFL) problem that is NP hard [27], where the facilities correspond to deployed caches and candidate locations correspond to all PoPs. We find an approximate solution using a greedy method shown in Algorithm 1.

Note that in Line 3 of Algorithm 1, by relaxing the integer variables to real values, the optimization problem becomes a classic LP problem and can be easily solved with existing LP solvers, e.g., CVX [28] and Gurobi [29]. To speed up the process, instead of only selecting one PoP at each iteration, we can choose the first w PoPs ($1 < w \ll N$) with the largest cache capacities in Line 5. Thus, the number of iterations in Algorithm 1 depends on the chosen value of w and equals $\lceil N/w \rceil$.

Algorithm 1 Greedy PoP Selection & Cache Deployment

Input: $G(V, E)$ and parameters given in problem (28)
Output: PoP selection vector x; cache capacity vector y
1 **while** $|U| < N$ **do**
2 Initialize the set of cached PoP U as empty: $U = \varnothing$;
3 Update x and y with U: $x_i = \hat{x}_i$, $y_i = \hat{y}_i$ for $i \in U$;
4 Solve problem (28) by i) relaxing the integer constraints and ii) ignoring the PoP number constraint (28g);
5 Select the PoP with the largest cache capacity (not including those already in U). Assuming that the newly selected PoP is i, set $\hat{x}_j = 1$ and set $\hat{y}_j$ to resulted cache capacity;
6 Update U by adding the newly selected PoP: $U = U \cup \{\hat{x}_j, \hat{y}_j\}$;

As shown in our later experiments with the real-world PoP network in Sec. VI-C, we observe that a small number of key PoPs are always selected as cache, while others have only minor influence to the overall traffic-delivery cost. The above greedy approach works great in our context, since the key PoPs are most likely to be found in the first w PoPs with the largest cache capacities.

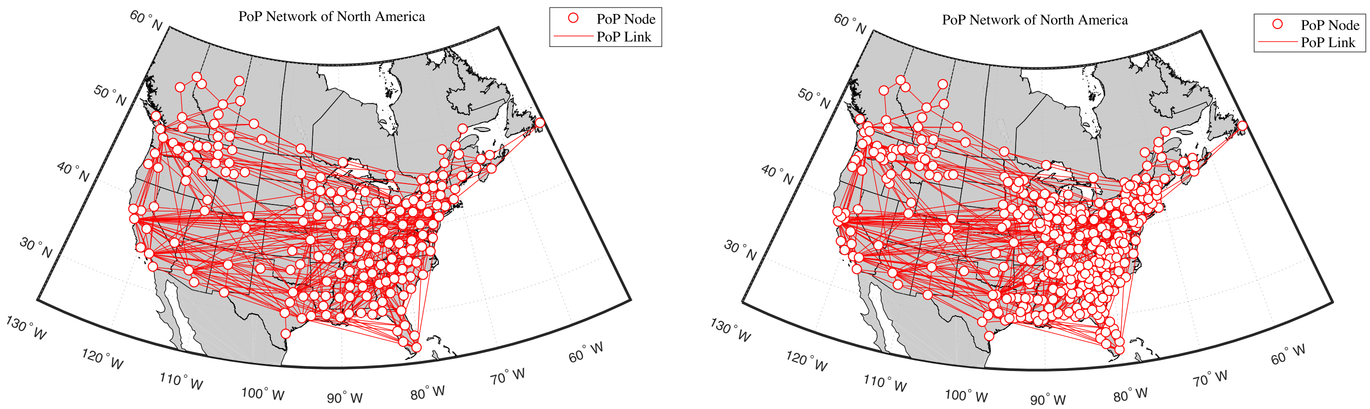
V. Experimental Configuration

In this section, we set up an experimental environment so that CDN designers or CDN operators can evaluate their design. For this purpose, we extract the PoP networks of various ISPs across North America as our target network and collect real-world traffic data, which will be used to validate our model and solution proposed in previous section.

A. Target Network

The datasets from *The Internet Topology Zoo* [30] provide topology information of over 250 ISPs' PoP networks all over the world. For our experiment, we carefully select 77 ISPs' PoP networks across North America (including ISPs like AT&T, Bell and Cogent) to construct the target network. We make use of PoP networks for the following two reasons: i) the PoPs are usually peered with (one or multiple) neighbour ISPs and thus preferred choices for cache deployment; ii) the PoP network provides more detailed geographic information of traffic demands than the AS-level network [31], and thus the traffic delay estimated by the PoP hop distance should be more accurate than that estimated by the AS hop distance.

In the PoP-level target network construction, we first extract the PoP nodes (including the longitude and latitude of the PoP locations) and PoP links from each of the 77 ISPs' PoP networks, and then merge all the nodes and links into one big graph. After that, we remove redundant nodes and links to form the target network used for our experiments. In addition, to avoid selecting too many PoPs in a small area, we control the node density of selected PoPs. For example, with node density $\le 1/35$ miles^2, we select PoPs such that no more than two PoPs are within a 35 miles^2 area. By changing PoP density, we can obtain various networks. Fig. 4(a) and Fig. 4(b) show a 194-node target network and a 332-node target network, respectively, which are utilized in our later evaluation.



(a) (b)

Fig. 4. The target PoP network topology across North America. (a) 194-node case: node density $\leq 1/(69 \text{ miles})^2$. (b) 332-node case: node density $\leq 1/(35 \text{ miles})^2$.

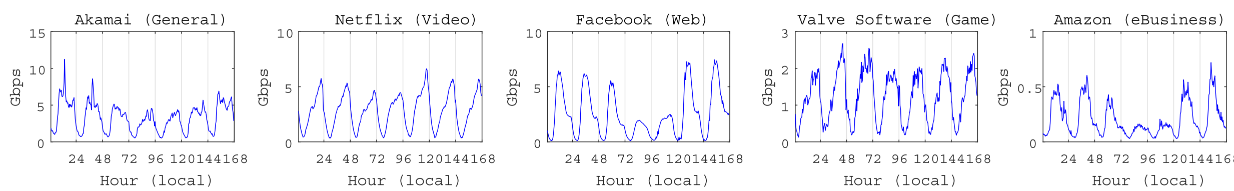


Fig. 5. Traffic demands of five content operators at NORDUnet nodes (one week time interval: Mar. 10-16, 2016).

B. Traffic Demand

For the design of a new CDN system, the operator usually faces the "cold start" problem due to the lack of traffic demand knowledge from end users. According to the principle of transfer learning[3] [32], the knowledge learnt from one entity (i.e., existing CDNs in operation) can be transferred to another similar entity (i.e., the new CDN where there is no traffic data before its deployment). A broad spectrum of related work on traffic pattern prediction can be found, either for the long-term network traffic forecasting [33] or for the short-term crowd flow prediction [34].

We refer to the transfer learning idea and make use of traffic demand traces extracted from existing CDNs for the design of new CDNs. Specifically, we collect the traffic data from the NORDUnet [35], which is a research and education oriented network infrastructure with cache servers/clusters deployed at various peering points and collect traffic information in Europe and North America. Using these real-world trace data, we generate synthetic traffic demands at PoP nodes in our target networks, for both general and categorized content delivery.

1) Traffic Demands for General Content: To investigate the traffic demands of general contents, we use the traffic data collected at a CDN cluster of Akamai, since Akamai is one of the largest CDN operators for general content delivering. The traffic demands towards the Akamai cluster in one-week period is shown in the first chart of Fig. 5.

[3]This paper only intends to point out the potential of transfer learning to address the "cold start" problem instead of offering a comprehensive study of transfer learning in this context, including source/target CDN comparison, feature construction and evaluation, and so on. A thorough investigation on the benefit of transfer learning is beyond the focus of the paper.

2) Traffic Demands for Categorized Content: To further investigate user traffic demands towards different Internet contents, we look into the traffic demands towards Netflix (video), Facebook (web), Valve Software (game), and Amazon (eBusiness), respectively. The second to fifth charts in Fig. 5 show one-week traffic demand curves of the four types of contents. It is interesting to see the differences among the four types of contents: the peak demands of Netflix (i.e., video/movie watching) and Valve Software (i.e., gaming) usually appear at night, while those of Facebook (blog browsing) and Amazon (online shopping) appear in day time.

Referring to the realistic traffic demand patterns as above, we generate synthetic traffic demands required by the dynamic traffic model introduced in Sec. III-B. First, we normalize the traffic demand curve of each content as the traffic profile for that type of content, i.e., $q(t), \forall t \in T$. Then, we collect the population of each city where the PoP nodes are localized, i.e., $p(i), \forall i \in V$, from the data published online [36], [37]. With $q(t)$ and $p(i)$, we generate the synthetic traffic demands at each PoP node for each type of Internet content, using the demand generation model (calculated with formula (1)). The generated traffic demands are used for the evaluations in the next section.

VI. Experimental Evaluation

Based on the implementations in the previous section, we evaluate the optimal CDN design with dynamic traffic demands and illustrate its advantage over CDN design with static traffic demands.

A. Evaluation Methodology

First, the diverse Internet content demands illustrated in Fig. 5 are utilized to generate synthetic traffic demands

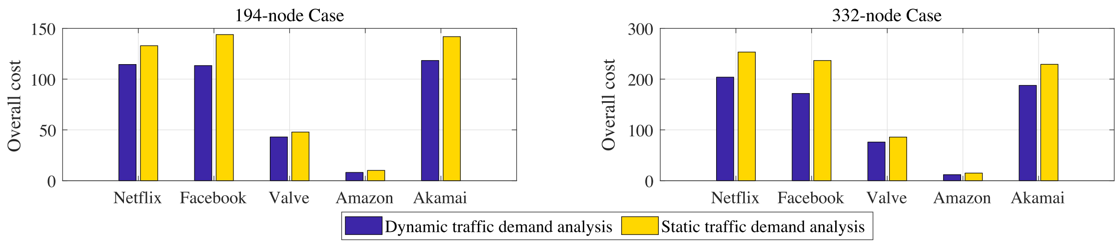


Fig. 6. Overall traffic-delivery cost of five different content demands over two target networks.

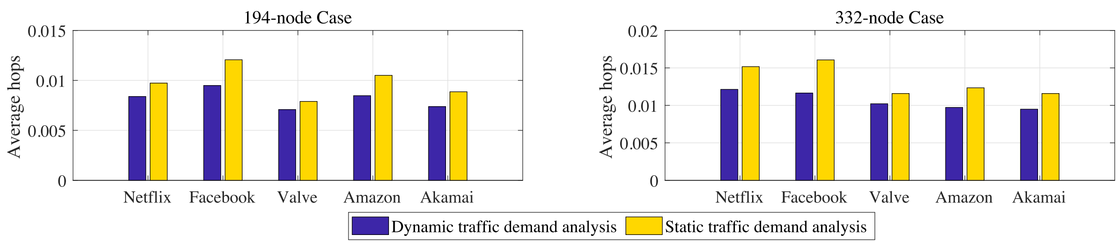


Fig. 7. Average traffic-delivery distance (PoP hops) of five different content demands over two target networks.

TABLE II

TRAFFIC DELIVERY COST AND TRAFFIC-DELIVERY DISTANCE USING DYNAMIC TRAFFIC VS USING THE STATIC TRAFFIC

		194-Node Target Network					**332-Node Target Network**				
		Netflix	Facebook	Valve	Amazon	Akamai	Netflix	Facebook	Valve	Amazon	Akamai
Delivering Cost (*Gbps·hop*)	*static*	133	144	47.9	10.1	142	253	237	85.9	14.9	229
	dynamic	114	113	43.0	8.11	118	204	172	76.1	11.8	188
	saving	14%	21%	10%	20%	17%	20%	28%	12%	21%	18%
Delivering Distance ($\times 10^{-3}$*hop*)	*static*	9.73	12.1	7.89	10.5	8.86	15.2	16.1	11.6	12.3	11.6
	dynamic	8.39	9.49	7.08	8.47	7.38	12.1	11.6	10.2	9.73	9.49
	shortening	14%	21%	10%	20%	17%	20%	28%	12%	21%	18%

Note: the value of cost saving for each content type is identical to the reduction of delivery distance, which further validates Lemma 1.

for each content type and each PoP node shown in Fig. 4, with the model introduced in Sec. V-B and Sec. VII-B. Then, the generated dynamic traffic demands are used as input to the optimization problem in (28). The results are recorded as the *optimal CDN design using dynamic traffic*. In addition, we calculate the static traffic demand of each PoP node by averaging its dynamic traffic demands over time and deploy a cache, whose size is the same as the average demand, at the PoP. In other words, the variables in c in problem (28) are replaced with the static values, and then the simplified problem is solved to obtain the *optimal CDN design using static traffic*.

Comparing the CDN design using dynamic traffic and that using static traffic, we first evaluate the advantages of the former regarding the traffic-delivery cost and the delivery hop distance (or correspondingly delivery latency). Then, considering the constraint of PoP footprint in CDN design, we study the influence of PoP number on the traffic-delivery cost, under dynamic and static traffic demands analysis, respectively. Furthermore, we also investigate the impacts of operating budget and satisfaction ratio, as well as the robustness of our solution.

B. Savings of Delivering Cost and Delivering Distance

A PoP is called transit if it does not host any CDN cache. The *transit ratio* refers to the percentage of transit PoPs over all PoPs in the target network. To test the impact of the transit ratio for CDN design, we randomly choose transit PoP nodes in the target networks by varying the value of the transit ratio. In this experiment, we also assume that the CDN operator is free to choose PoP location without PoP footprint constraint and has enough operating budget. The situations with a limited PoP footprint and operating budget will be evaluated in the following experiments.

The average traffic-delivery costs for the five content types over 194-node and 332-node networks are shown in Fig. 6. The average traffic-delivery (PoP hop) distances for different traffic types are shown in Fig. 7. The overall cost saving with the dynamic traffic-based CDN design is 16% over the 194-node network and 20% over the 332-node network. Furthermore, detailed values of delivery cost, delivery distance, and the saving for each traffic type are summarized in Table II. Note that the traffic-delivery cost saving of our solution can be up to 21% in the 194-node case and 28% in the 332-node case, respectively, compared with the CDN design under static traffic demand analysis. Such saving is significant for a CDN operator, especially that with large PoP footprint all over the world.

C. Constraint of PoP Footprint

Given a group of PoP candidates, the CDN operator may not want to choose all of them as caches, due to financial limitation or other considerations. Therefore, to control the cached PoP footprint during the CDN design, we limit the number of selected PoPs and choose the optimal PoP subset by making use of the PoP selection vector (defined in (3)) and the greedy PoP selection diagram (given by Algorithm 1).

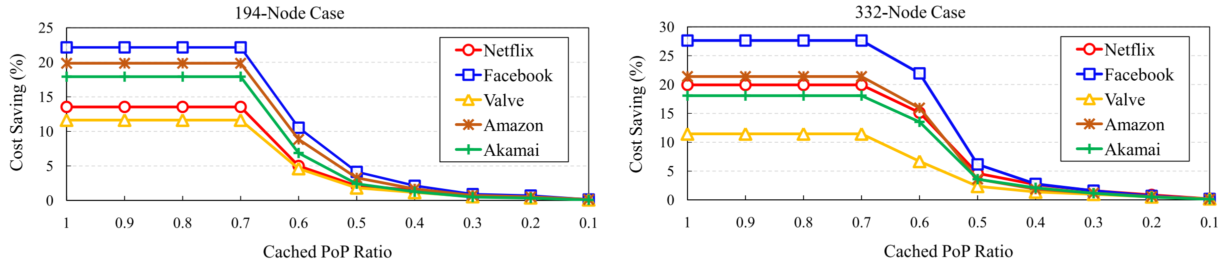


Fig. 8. Traffic-delivery cost savings with varying cached PoP ratio over the 194/332-node networks, respectively: satisfaction ratio = 1, transit ratio = 0.5.

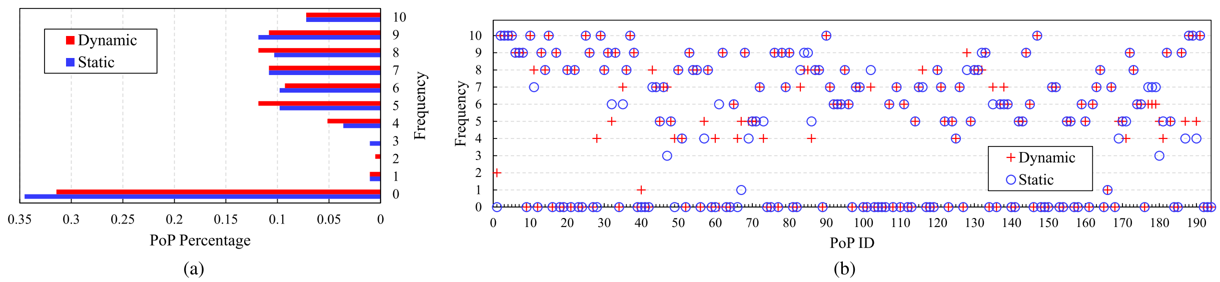


Fig. 9. Frequency of cached PoPs from dynamic and static traffic demand analysis, respectively: 194-node case, content type Netflix, satisfaction ratio = 1, transit ratio = 0.5. (a) Distribution of cached PoPs frequencies. (b) Frequency of cached PoP vs. PoP ID.

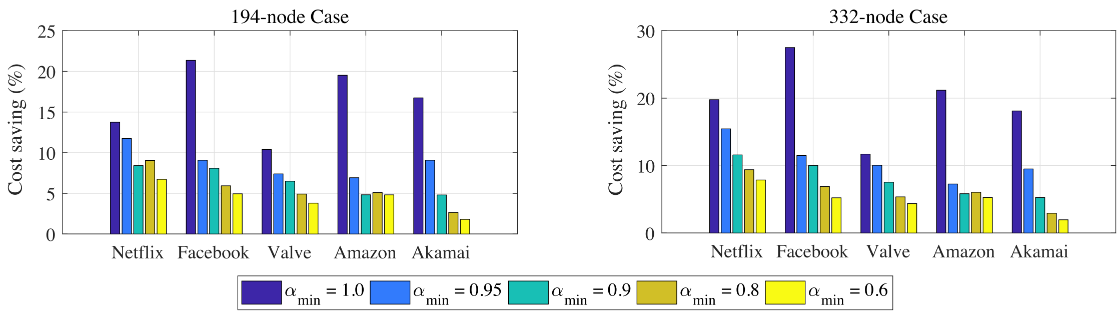


Fig. 10. Savings from dynamic demand analysis: delivery cost/distance with various min. satisfaction ratios over 194-node and 332-node networks, respectively.

We define the *cached PoP ratio* as the percentage of cached PoPs over all PoPs in the target network, and traverse its value from 100% (i.e., all PoPs can be cached) to 10% (i.e., only ten percent of the total PoPs can be cached) by every ten percent.

The results for the five types of content from two target network scenarios are illustrated in Fig. 8. From the results we can see that, when the cached PoP ratio decreases below a threshold (0.7 in our case), the cost saving of our dynamic traffic demand analysis over the static one becomes smaller. To explain this phenomenon, we keep record of the cached PoPs in each experiment and summarize the selection frequency for each PoP. The statistic results are shown in Fig. 9, from which we can observe that a small fraction of PoPs are key nodes in the target network and will be cached with much a larger probability than others. When the number of cached PoP decreases, the difference between the solution from our dynamic demand analysis and that from the static demand analysis becomes smaller, because a small number of cached PoP leaves very little room for obtaining better CDN design.

D. Constraints of Operating Budget & Satisfaction Ratio

In the previous experiment, we assume that the CDN operator has enough operating budget by ignoring the constraint (28f) in problem (28). In this experiment, we consider the situation where the CDN operator is constrained with operating budget and can only partially satisfy the user demands. For simplicity, we assume that the unit cache price (r_i in inequality (11)) of all PoPs are the same and denoted as r_0. Consequently, the constraint (28f) is equivalent to the constraint (28e) when $\frac{K}{r_0} = \alpha_{min} D$, or only one of them takes effect when $\frac{K}{r_0} \neq \alpha_{min} D$. Therefore, we only consider constraint (28e) and change the value of α_{min} to investigate the situations where the cache is under-provisioned and the operating budget poses a constraint.

Besides the situation where $\alpha_{min} = 1$, we evaluate the cases where the cache is under-provisioned by setting $\alpha_{min} = 0.95, 0.9, 0.8$ and 0.6, respectively. Correspondingly, the savings of the traffic-delivery cost and traffic-delivery distance are illustrated in Fig. 10. The overall savings for the four under-provisioning cases are $9\%, 7\%, 6\%$ and 4% over the 194-node network, respectively, and $11\%, 8\%, 6\%$ and 5% over the 332-node network, respectively.

The under-provisioning situations result in less savings. This is due to the relaxation of demand satisfaction. With $\alpha_{min} < 1$, the traffic demands with percentage of $1 - \alpha_{min}$ will not be delivered under the objective of delivery cost minimization in

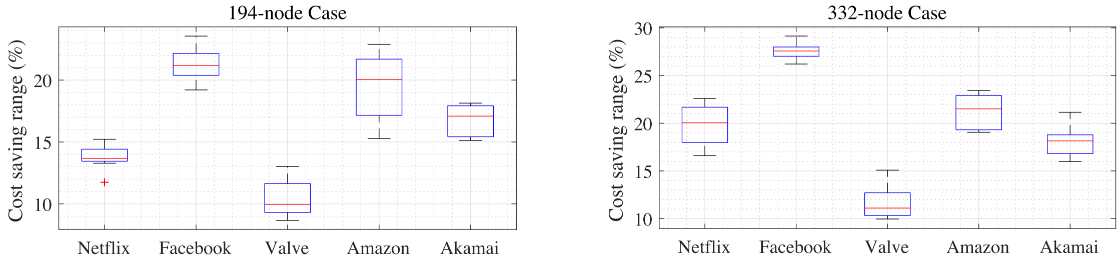


Fig. 11. Ranges of savings with dynamic traffic-based design with varying transit ratios over the 194-node and 332-node networks, respectively.

problem (28). Thus, the advantage of dynamic traffic-based design is weakened, leading to less savings.

E. Resiliency to Transit Ratio

We further investigate whether or not the transit ratio has a large impact on the cost saving. For this purpose, we vary the value of the transit ratio from 0 to 0.9 and record the cost savings with dynamic traffic-based CDN design.

The box plots are shown in Fig. 11, where the ranges of cost saving for each content type are demonstrated. According to the results, the variations of the performance are within 5% (or $\pm 2.5\%$), indicating that the dynamic traffic-based CDN design is advantageous over static traffic-based design regardless of the number of transit nodes.

VII. Further Study: How Does Demand Pattern Affect Cost Saving?

In this section, we look into CDN design with different content types and investigate how the characteristics of traffic demand patterns affect the design cost. For this study, we assume a situation where the traffic demand is completely satisfied and PoP footprint is unlimited.

A. Study With Real Traffic

We first look into the five traffic patterns shown in Fig. 5, denoting the content demands towards Akamai, Netflix, Facebook, Valve Software and Amazon, respectively. With varying transit ratios, corresponding cost savings for the five content types over 194-node network and 332-node network are illustrated in Fig. 12. With the results in Table II and Fig. 12, we can find that the content types with different demand patterns can result in different cost savings. Specifically, we have the following observations:

- Users' content demands towards web blog (Facebook) and online shopping (Amazon) correspond to higher cost saving percentages with our dynamic demand analysis, as shown by the blue and brown lines in Fig. 12.
- Users' content demands towards video watching (Netflix) and online gaming (Valve Software) lead to relatively smaller cost saving percentages from the dynamic demand analysis.
- General content demands like those over Akamai lead to moderate saving (less than those with Facebook and Amazon but higher than those with Netflix and Valve).

Our preliminary investigation shows that the cost saving is related to the variation of the traffic demand pattern. Nevertheless, more traffic demand patterns should be studied before we draw conclusion on the relationship between the cost saving and traffic demand pattern, because i) the number of content types (curve patterns) in our experiments is still small and ii) the cost saving percentages are relative measurements, and as such a higher cost saving percentage may not translate to higher absolute saving when the total traffic amount is small. In next section, we will i) generate synthetic traffic demands that show more diverse patterns and ii) investigate the characteristics of demand patterns and their impacts on CDN design.

Algorithm 2 Synthetic Demand Pattern Generation (Basic Shapes)

Input: $bs \leftarrow$ basic shape of the curve (peak or valley); $ip \leftarrow$ inflection point of the curve; $tp \leftarrow$ time period of the curve; $[tp\_start, tp\_end] \leftarrow$ start and end points of tp; $[r\_bottom, r\_top] \leftarrow$ magnitude range of the curve

Output: synthetic demand curve (vector) f

$stretch\_v \leftarrow r\_top - r\_bottom$; //vertical stretching extent

$stretch\_h\_p1 \leftarrow \frac{\pi}{2*(ip-tp\_start)}$; //horizontal stretching extent in $[tp\_start, ip]$

$stretch\_h\_p2 \leftarrow \frac{\pi}{2*(tp\_end-ip)}$; //horizontal stretching extent in $[ip, tp\_end]$

if bs *is peak* **then**

  $f\_p1 = stretch\_v * \sin(stretch\_h\_p1 * (tp - tp\_start)) + r\_bottom$; //curve in $[tp\_start, ip]$

  $f\_p2 = stretch\_v * \cos(stretch\_h\_p2 * (tp - ip)) + r\_bottom$; //curve in $[ip, tp\_end]$

else

  $f\_p1 = stretch\_v * (\cos(stretch\_h\_p1 * (tp - tp\_start) + \frac{\pi}{2}) + 1) + r\_bottom$; //curve in $[tp\_start, ip]$

  $f\_p2 = stretch\_v * (\cos(stretch\_h\_p2 * (tp - ip) + \pi) + 1) + r\_bottom$; //curve in $[ip, tp\_end]$

$f = [f\_p1, f\_p2]$;

B. Study With Synthetic Traffic

To further explore the impact of traffic demand pattern to the CDN design cost, we produce more synthetic demand curves. In detail, we first create **basic shapes** with sine/cosine functions, which denote the trends of traffic demands during a time interval and play as the basic building blocks of

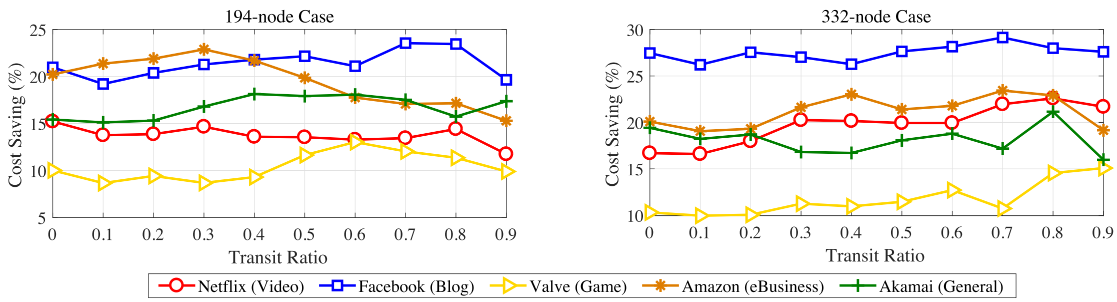


Fig. 12. Saving of the five types of content demands over the 194-node network and the 332-node networks, respectively.

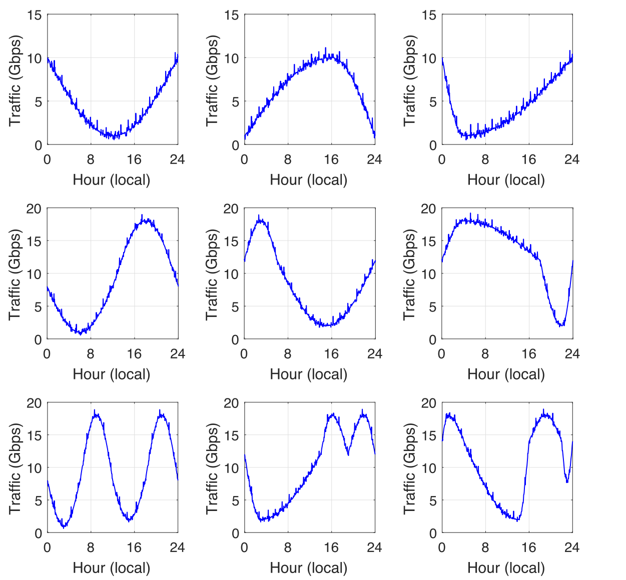


Fig. 13. Generated synthetic traffic demand patterns with diverse curve characteristcis.

the synthetic traffic demand curve. Then the basic shapes are compressed, stretched and combined to create diverse traffic demand curves within one period (24 hours in our implementation). At last, white noises are added into the curves to denote random and minor oscillations in demand patterns.

Algorithm 2 shows the process of generating basic shapes (single-peak/valley) of traffic demand curve, in which we split the generated curve into two parts (the sections before and after an inflection point).[4] With Algorithm 2 as the basic building block, we can create variety of demand curves (e.g., the multi-peak/valley curves). Some generated samples are illustrated in Fig. 13.

With massive demand curves generated, we are able to explore the impact of a specific characteristic on the cost of CDN design. For a generated traffic pattern/curve within one period (one-day time in our implementation), we use the following three metrics to measure its characteristics:

- Mean (μ): measuring the average magnitude of demand;
- Variance (σ^2): measuring the deviation of demand from mean value;

[4]An inflection point is a point on a curve at which the sign of the curvature changes, i.e., the second derivatives of the curve function before and after the point have opposite signs.

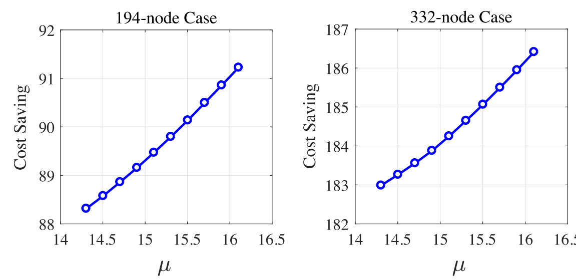


Fig. 14. Cost savings ($Gbps \cdot hops$) vs. mean values of traffic demand patterns: $\sigma^2 = 8$, $\eta = 1.2 \times 10^{-5}$.

- Fluctuation index (η): measuring the degree of major fluctuation in demand.

The details of calculating the metrics will be given later.

1) Impact of Mean Value: We first investigate the impact of mean value on the cost of CDN design with static and dynamic demand analysis. The mean value is defined as:

$$\mu := \frac{1}{n} \sum_{t=1}^{n} d(t), \tag{29}$$

where n is the number of time slots considered in one period and $d(t)$ is the value of traffic demand curve at time t. With the method of variable control, we keep the other two metrics (variances and fluctuation index) fixed and only change the mean value of the traffic demand. Along with different mean values of demand traffic patterns, the cost saving of CDN design with dynamic demand analysis compared with that of the static one is illustrated in Fig. 14, for 194-node case and 332-node case, respectively.

From Fig. 14, we can see that the cost saving of CDN design with dynamic demand analysis increases nearly linearly with the growth of mean traffic demand, both in 194-node case and 332-node case. This is easy to understand: as the absolute amount of delivered traffic within the network increased, the absolute saving provided by our solution should be increased. This result indicates that CDN operators serving more content demand of customers will benefit more from the CDN design with dynamic demand analysis.

2) Impact of Variance: We further investigate the impact of variance on the cost of CDN design. The variance is defined as:

$$\sigma^2 := \frac{1}{n} \sum_{t=1}^{n} \left(d(t) - \mu\right)^2, \tag{30}$$

where n is the number of time slots considered in one period, $d(t)$ is the value of traffic demand curve at time t, and μ is

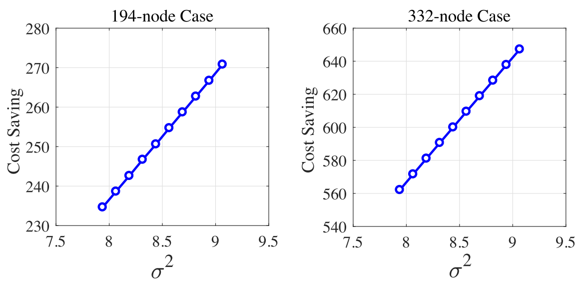


Fig. 15. Cost savings ($Gbps \cdot hops$) vs. variances of traffic demand patterns: $\mu = 12$, $\eta = 1.2 \times 10^{-5}$.

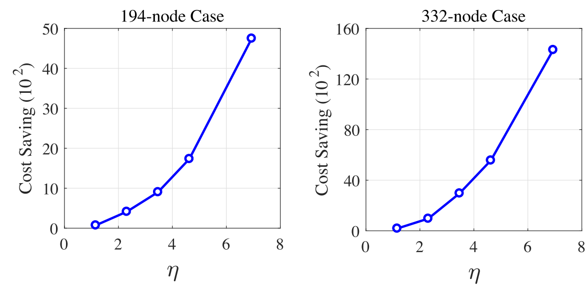


Fig. 16. Cost savings ($Gbps \cdot hops$) vs. fluctuation indexes of traffic demand patterns: $\mu = 15$, $\sigma^2 = 50$.

the mean value of traffic demand curve in one period (defined by (29)). We change the variances of traffic demand patterns while keeping the mean values and fluctuation index fixed. The cost saving from the CDN design with dynamic demand analysis is shown in Fig. 15, along with different variances of demand traffic patterns. We also perform experiments on the two target networks with 194 nodes and 332 nodes, respectively.

From Fig. 15, we find that the cost saving of CDN design with dynamic demand analysis also increases linearly with the increase of demand variances, both in 194-node and 332-node cases. This is caused by the "cache contest" among PoPs in peak (demand) time. The traffic demand curve with larger variance often has higher/sharper peaks, which leads to higher cache competition among PoPs during the peak time. Our CDN design with dynamic demand analysis takes the traffic demand in all time slots (including the peak time) into consideration while that with static demand analysis only considers the average traffic demand over time. Thus, the advantage of our solution with dynamic demand analysis becomes more apparent with the increase of variance in traffic demand. This also indicates that CDN operators serving content demands with larger variance can benefit more from the CDN design with dynamic demand analysis.

3) Impact of Fluctuation Index: We are also interested in the situation where the mean value and variance of traffic demand patterns are the same while the fluctuation degrees (e.g., the number of peaks/valleys) of the demand curves are different. To learn the fluctuation of the traffic demand pattern, we change the perspective from time domain to frequency domain of a curve. Given the Fourier transform of a specific traffic demand curve:

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k cos(k\Omega t + \psi_k), \tag{31}$$

where a_0 is the constant component and a_k is the magnitude of k-th component of the curve, the fluctuation index of the traffic demand curve is defined by the frequency of the component with the largest magnitude, i.e.,

$$\eta := \frac{k\Omega}{2\pi}, \quad \text{where } a_k = \max\{a_1, a_2, \cdots\}. \tag{32}$$

Intuitively, a traffic demand curve with larger fluctuation index means the curve has more waves (i.e., peaks and valleys). The results from the demand curves with different fluctuation indexes are demonstrated in Fig. 16, with 194-node and 332-node cases, respectively.

From Fig. 16, we can see that compared with the mean value and variance, the fluctuation index has even more apparent impact on the cost savings. The improvement of the cost saving is almost exponential with the increase of the fluctuation index in the traffic demand curves. This is because the curves with larger fluctuation indexes have more peaks, implying the length of peak (demand) time is longer and more cache competitions. As a consequence, CDN design using static traffic cannot handle the high cache competitions well and leads to sub-optimal cache decisions and high content delivery cost. In contrast, CDN design with dynamic demand analysis alleviates this problem and results in higher cost saving.

Note that, to illustrate our findings with charts, we only choose three specific cases with certain values of mean, variance and fluctuation index, respectively. Other cases with different settings of metrics result in the similar trends as those shown in Figs. 14-16. Recently, as another important factor, the network topology of the CDN was also investigated in affecting the traffic-delivery cost [38].

VIII. Conclusions

In this paper, we performed a comprehensive study on the CDN design problem and provided a better solution by considering distributed time-varying traffic demands, referred to as dynamic traffic. In particular, we formulated the distributed cache deployment optimization problem with an integer linear program (ILP), which takes dynamic traffic over a target network as input, to minimize the network traffic-delivery cost. We then transformed the original ILP problem into a scalable form and developed a greedy algorithm to solve the problem efficiently. To construct a realistic experimental environment, we extracted the ISPs' PoP networks across North America and generated synthetic data based on real-world traffic demands. Our results demonstrated that compared with static traffic-based CDN design, our dynamic traffic-based CDN design led to great savings in the overall delivery cost from 16% to 20%. With extensive experiments, we also quantified the impact of traffic demand patterns to the CDN design cost.

Acknowledgment

The authors thank Richard Brunner for his help in the early version of the paper.

References

[1] K.-K. Yap *et al.*, "Taking the edge off with espresso: Scale, reliability and programmability for global Internet peering," in *Proc. ACM SIGCOMM*, 2017, pp. 432–445.

[2] S. Manfredi, F. Oliviero, and S. P. Romano, "A distributed control law for load balancing in content delivery networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 55–68, Feb. 2013.

[3] Variety Digital News. (2016). *Netflix Bandwidth Usage Climbs to Nearly 37% of Internet Traffic at Peak Hours*. Accessed: Feb. 2016. [Online]. Available: http://variety.com/2015/digital/news/netflix-bandwidth-usage-internet-traffic-1201507187/

[4] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: A platform for high-performance Internet applications," *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, 2010.

[5] Imperva. *Essential CDN Guide*. Accessed: Mar. 30, 2017. [Online]. Available: https://www.incapsula.com/cdn-guide/

[6] V. K. Adhikari, S. Jain, and Z.-L. Zhang, "YouTube traffic dynamics and its interplay with a tier-1 ISP: An ISP perspective," in *Proc. ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2010, pp. 431–443.

[7] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of CDNs," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr./May 2014, pp. 460–468.

[8] M. Yu, W. Jiang, H. Li, and I. Stoica, "Tradeoffs in CDN designs for throughput oriented traffic," in *Proc. ACM Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2012, pp. 145–156.

[9] I. Castro, R. Stanojevic, and S. Gorinsky, "Using tuangou to reduce IP transit costs," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1415–1428, Oct. 2014.

[10] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 568–582, Oct. 2000.

[11] H. Yin *et al.*, "Tradeoffs between cost and performance for CDN provisioning based on coordinate transformation," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2583–2596, Nov. 2017.

[12] G. Xie, Z. Li, M. A. Kaafar, and Q. Wu, "Access types effect on Internet video services and its implications on CDN caching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1183–1196, May 2018.

[13] M. Z. Shafiq, A. X. Liu, and A. R. Khakpour, "Revisiting caching in content delivery networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 567–568, 2014.

[14] I. Castro and S. Gorinsky, "T4P: Hybrid interconnection for cost reduction," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Mar. 2012, pp. 178–183.

[15] J. Ni and D. H. K. Tsang, "Large-scale cooperative caching and application-level multicast in multimedia content delivery networks," *IEEE Commun. Mag.*, vol. 43, no. 5, pp. 98–105, May 2005.

[16] H. Yin *et al.*, "Design and deployment of a hybrid CDN-P2P system for live video streaming: Experiences with LiveSky," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2009, pp. 25–34.

[17] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system," in *Proc. ACM Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2010, p. 4.

[18] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Mar. 2010, pp. 1–9.

[19] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 836–845, Apr. 2016.

[20] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2014.

[21] K. Mokhtarian and H.-A. Jacobsen, "Coordinated caching in planet-scale CDNs: Analysis of feasibility and benefits," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 433–441.

[22] M. Leconte *et al.*, "Placing dynamic content in caches with small population," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 451–459.

[23] B. Schlinker *et al.*, "Engineering egress with edge fabric: Steering Oceans of Content to the World," in *Proc. ACM SIGCOMM*, 2017, pp. 418–431.

[24] V. Giotsas *et al.*, "Detecting peering infrastructure outages in the wild," in *Proc. ACM SIGCOMM*, 2017, pp. 446–459.

[25] A. Basta *et al.*, "SDN and NFV dynamic operation of LTE EPC gateways for time-varying traffic patterns," in *Proc. Int. Conf. Mobile Netw. Manage. (MONAMI)*. Cham, Switzerland: Springer, 2014, pp. 63–76.

[26] C. Ge, Z. Sun, N. Wang, K. Xu, and J. Wu, "Energy management in cross-domain content delivery networks: A theoretical perspective," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 3, pp. 264–277, Sep. 2014.

[27] V. Arya *et al.*, "Local search heuristics for k-median and facility location problems," *SIAM J. Comput.*, vol. 33, no. 3, pp. 544–562, 2004.

[28] *CVX Research*. Accessed: May 30, 2017. [Online]. Available: http://cvxr.com/cvx/

[29] *Gurobi Optimization*. Accessed: Jul. 30, 2017. [Online]. Available: http://www.gurobi.com/

[30] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011.

[31] B. Donnet and T. Friedman, "Internet topology discovery: A survey," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 4, pp. 56–69, 4th Quart., 2007.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[33] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1110–1124, Sep. 2005.

[34] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI*, 2017, pp. 1655–1661.

[35] *NORDUnet*. Accessed: Mar. 15, 2017. [Online]. Available: http://stats.nordu.net/connections.html

[36] *Census*. Accessed: Jun. 10, 2017. [Online]. Available: http://www.census.gov/

[37] *City Population*. Accessed: Oct. 10, 2017. [Online]. Available: http://www.citypopulation.de/

[38] G. Tang, H. Wang, K. Wu, D. Guo, and C. Zhang, "When more may not be better: Toward cost-efficient CDN selection," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 1–2.

Guoming Tang (S'12–M'17) received the bachelor's and master's degrees from the National University of Defense Technology, China, in 2010 and 2012, respectively, and the Ph.D. degree in computer science from the University of Victoria, Canada, in 2017. He joined the College of Systems Engineering, National University of Defense Technology, in 2017, where he is currently an Assistant Professor. Aided by machine learning and optimization techniques, his research mainly focuses on computational sustainability issues and distributed networking systems.

Huan Wang received the B.Sc. and M.Sc. degrees in computer science from Southwest Jiaotong University and the University of Electronic Science and Technology of China, in 2013 and 2016, respectively. He is currently pursuing Ph.D. degree with the Department of Computer Science, University of Victoria, BC, Canada. His research interests include content delivery networks, edge caching and computing.

Kui Wu (S'98–M'02–SM'07) received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, China, in 1990 and 1993, respectively, and the Ph.D. degree in computing science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science, University of Victoria, Canada, in 2002, where he is currently a Full Professor. His research interests include network performance analysis, mobile and wireless networks, and network performance evaluation.

Deke Guo received the B.S. degree in industry engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently a Professor with the College of Systems Engineering, National University of Defense Technology. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks. He is a member of the ACM.