This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

Towards Cost-Efficient Content Placement in Media Cloud: Modeling and Analysis

Yichao Jin, Yonggang Wen, *Senior Member, IEEE,* and Kyle Guan, *Member, IEEE*

***Abstract*—Cloud-centric media network (CCMN) was previously proposed to provide cost-effective content distribution services for user-generated contents (UGCs) based on media cloud. CCMN service providers orchestrate cloud resources to deliver UGCs in a pay-per-use style, with an objective to minimize the operational monetary cost. The monetary cost depends on the actual usage of cloud resources (e.g., computing, storage, and bandwidth), which in turn, is affected by the content placement strategy. In this paper, we investigate this cost-optimal content placement problem. Specifically, it is formulated into a constrained optimization problem, in which the objective is to minimize the total monetary cost, with respect to the resource capacity. We tackle this problem via a two-step strategy. The first step focuses on the placement for a single content, which is mapped into a k-center problem. Using a graph-theoretic approach, we derive and verify a logarithmic model between the optimal mean hop distance from viewers to contents, and the optimal number of content replicas. The second step leverages this analytical result to solve the cost optimization problem, via a feasible direction method. The analysis is substantiated via numerical simulations, using a set of data traces from a top content website. This investigation suggests that, the optimal number of content replica for each title follows a power-law distribution in respect to its popularity rank. Moreover, it reveals a fundamental trade-off between the storage and bandwidth cost. Finally, compared to existing heuristics, our proposed algorithm is able to obtain the optimal placement strategy, with lower computational complexity.**

***Index Terms*—Content Delivery, Content Placement, Media Cloud, Cost Optimization**

I. Introduction

OVER the last decade, User Generated Contents (UGCs) are growing rapidly, triggering a rethinking of efficient content distribution approaches. Specifically, UGC refers to the contents that are created and published freely by end-users, such as YouTube videos. Owing to the ubiquitous penetration of high-speed Internet access and the advances in the field of media technologies, a tremendous volume of UGCs had been created, published and consumed by an increasing number of users. In 2012, around 500 million UGCs had been uploaded to YouTube. These contents had attracted more than 1.5 trillion views from billions of UGC consumers [2]. Only in the U.S, the amount of UGC consumers were 115.7 million (i.e., 60% of total Internet user) in 2008, and this number rapidly climbed to 154.8 million (i.e., 70% of total Internet user) in 2013 [3]. The huge amount of UGC uploading and consuming has significantly stressed the Internet capability. In response to this challenge, efficient mechanisms are urgently sought to distribute UGCs, preferably at a low cost.

At present, Content Distribution Network (CDN) is the main solution to distribute UGCs, which however, suffers from a mismatch between its operational model and the properties of UGCs. First, the service model of existing CDN is financially prohibitive for small providers. In traditional approaches, content providers usually strike a long-term contract with the CDN service provider at a relatively high price [4]. Second, the traditional CDN operational model did not well match with UGC's long-tailed nature (i.e., most UGCs are not popular) [5]. In particular, existing CDN providers mainly tailored their architectures and operational models only for those popular contents [6], and they did not allow the content service providers to customize the operational model [7]. This results in an inefficient distribution scheme for UGCs.

These problems motivated us to propose Cloud Centric Media Network (CCMN) [8], [9] as a novel cloud-based CDN architecture. Specifically, the key idea of CCMN is to orchestrate a virtual Content Delivery Service (vCDS) by carving out storage, bandwidth, and computation resources from the cloud, to build an elastic CDN overlay. As a result, CCMN comes up with significant flexibility to distribute UGCs in a pay-per-use manner. Therefore, it allows small content providers to build their own cost-efficient virtual CDNs to facilitate large-scale UGCs distribution.

An important objective of designing CCMN is to minimize the monetary cost incurred by using cloud resources to orchestrate elastic CDN services. To achieve this target, content placement problem is one of the critical issues. In particular, it requires a careful evaluation on a trade-off between the bandwidth and storage resources. On one hand, placing a large number of content replicas at different geo-locations would increase storage-related cost [10]. But at the same time, it brings the content closer to the end users, thus reducing the delay and saving the bandwidth usage. On the other hand, the planners can choose to deploy fewer replicas to save the storage resources, albeit at the expense of increased delays and bandwidth costs. Therefore, a cost-effective content placement policy needs to be thoroughly studied to balance the trade-off between storage and bandwidth.

In this paper, we formulate this content management problem in CCMN as a constrained optimization problem. The objective is to minimize the monetary investment associated with the usage of bandwidth, storage and computation resources.

Yichao Jin and Yonggang Wen are with School of Computer Engineering, Nanyang Technological University, Singapore, 639798. Email: {yjin3, ygwen}@ntu.edu.sg.

Kyle Guan is with Bell Labs, Nokia, 791 Holmdel Road, Holmdel, NJ 07733. Email: kyle.guan@nokia.com.

Part of this work has been published in IEEE International Conference on Multimedia and Expo (ICME 2013) oral session [1].

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

This problem is solved in two steps. First, for a single content, we map it into the k-center problem [11] to explicitly derive the optimal placement policy for a given number of replica, on deployed networks. Second, for a catalog of contents, the problem is formulated as a convex optimization problem. By proposing an algorithm to solve this problem, we obtain the optimal number of replicas for each content title.

The contributions of this paper are multi-fold:

- Our solution on single content problem suggests that, the mean hop distance H between users and their closest replica, can be modelled as a logarithmic function with respect to the reciprocal of the number of replicas n (i.e., $H \sim \log(N/n)$) in a set of deployed networks with N nodes. This logarithmic model represents the fundamental trade-off between storage and bandwidth resources, offering an analytical framework to evaluate the cost associated with different placement strategies.
- The numerical analyses based on real traces suggest that, under feasible bandwidth and storage constraints, the optimal number of replicas follows a power-law distribution to its popularity in a set of deployed networks. This offers guidelines for the real-word operations of CCMN.
- We compare the performance of our approach with a list of different heuristics. The results indicate our approach can provide the near-optimal content placement solution in CCMN with reasonable low complexity.

The reminder of the paper is organized as follows. In Section II, we review related works. In Section III, we present the system models and the problem formulation. In Section IV, we characterize the k-center problem for single content placement. In Section V, we solve the convex optimization problem for a catalog of contents. In Section VI, we verify the obtained solutions using real traces. In Section VII, we compare the performance of our approach with a set of heuristic algorithms. In Section VIII, we conclude this paper.

II. Related Work

The content placement problem, due to its complexity, was addressed by using heuristics in most existing works. Qiu *et al.* [12] focused on selecting M replicas among N different potential locations to minimize the bandwidth cost, and investigated the performances of several different algorithms. Kangasharju *et al.* [13] studied the object replication strategies on CDN nodes with finite storage capacity by developing four natural heuristics. Karlsson [14] formulated the minimal replication cost problem as an integer programming problem, and proposed a heuristic method for choosing a replica placement. Tang *et al.* [15] investigated this problem by using several heuristic algorithms, for QoS-aware content distribution. Differed from these works, we formulate a convex optimization problem to theoretically derive the optimal policy and obtain analytical insights. The convex optimization is better than heuristics for several reasons. First, the analytical results from convex optimization can always find the global optimal solution, whereas the heuristics usually fails to do so. Second, by substituting the parameters into the analytical result, one can directly get the optimal solution, whereas the computational complexity of heuristics usually is significantly higher. Finally, by checking the analytical result, one can clearly have the relationship between different parameters and the result, whereas the heuristics cannot explicitly reveal such relationship.

TABLE I
Notation Table

Symbol	Definition
n_k	The number of replica for content k
B_k	The size of content k
R_k	The downloaded time of content k over t period
G	The topology of underlying networks
c_{tr}	Per hop cost for transmitting per bit data
c_{st}	Per bit cost for storing content per unit of time
Δ	The node degree of a graph
M	The total number of unique contents to be delivered
N	The total number of datacenters
H	The mean hop distance between users and replicas
S_{tot}	The total storage capacity constraint
T_{tot}	The total bandwidth capacity constraint

Some other researches formulated the content placement problem in specific or arbitrary network topologies as a graph theory problem, aiming to derive an analytical solution. For specific network topologies, the analytical solutions on optimal content placement for ring based and tree based CDN were discussed by [16] and [17] respectively. For arbitrary network topologies, Laoutaris *et al.* [18] considered both k-center problem and facility location problem for optimal content placement in large-scale networks. Liu *et al.* [19], [20] investigated the content caching problem in a peer-to-peer environment, and develop a theoretical framework to derive the global optimal solution. However, they mainly focused on how to select the optimal locations for a fixed number of replica, whereas we aim to analytically find the relationship between the number of replicas and the mean hop distance from any of those replicas to the users. In addition, these works either solved an unconstrained problem [12], [16], or missed one of the resource constraints [13], [18], whereas we address this optimization problem by jointly considering the storage and bandwidth capacity constraint.

The novelty of this work includes several aspects. First, we mathematically establish a logarithmic model to represent the relationship between the number of replicas and the mean hop distances in large-scale networks. Second, we analytically derive the optimal number of replicas for each content with different popularity. Finally, we consider the optimization problem in a more real scenario, where both storage and bandwidth resources are constrained.

III. Problem Modeling & Formulation

In this section, we first describe the architecture of CCMN to provide necessary background. Then, we present key system models for CCMN. Finally, based on these models, we formulate the content placement problem in CCMN as a constrained optimization problem. For clarity in the discussion, we summarize the important notations in table I.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

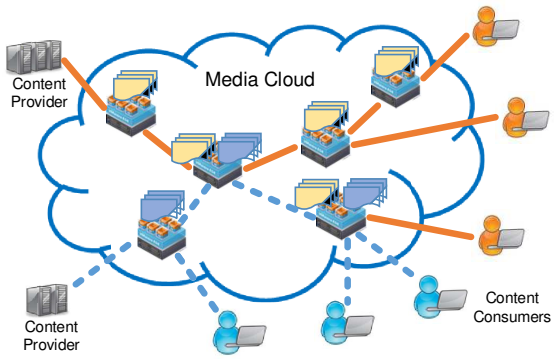


Fig. 1. Architecture of cloud centric media network. Content providers use the media cloud [21] to distribute UGCs to users as requested. Within the media cloud, the storage resources can be allocated to different edge servers to replicate contents in an on-demand manner.

A. CCMN Architecture

Figure 1 shows a reference of CCMN architecture, which consists of three parts, including cloud service provider, content service providers, and a list of content consumers. The cloud providers own a list of virtual machines (VMs), which are carved out in geographically distributed data centers. Those VMs work collaboratively to build an elastic CDN overlay on top of its underlying physical networks. The content service providers usually offer users a platform to publish their own generated contents. Typically, those contents are stored and processed in origin servers, with limited physical resources. Due to the enormous growth of UGCs, CDN services would be necessary to maintain the video distribution at a certain QoS level. However, it is not easy for them, specially small companies, to establish their own CDNs, or sign long-term contract with large CDN companies. To address this gap, in CCMN, they are encouraged to use cloud resources, to achieve content replication and media streaming services, on a pay-per-use model. As a result, the content consumers from different regions globally can be served at a lower price.

How to minimize the monetary cost incurred by using cloud resources to efficiently serve user requests, is one of the critical design objectives for CCMN. Indeed, the monetary cost highly depends on the content placement policy, which decides the number of replica for each content and their locations. Specifically, because the storage cost is usually a little cheaper than the bandwidth cost, it is advantageous to replicate content to different places, so that the average distance between users and contents, and the bandwidth cost can be reduced. However, if too many replicas are placed and their locations are not properly chosen, especially for those large amount of unpopular contents, it results in significant storage usage and limited reduction on bandwidth usage. Therefore, an optimal placement policy should be in place, to balance the trade-off between bandwidth and storage cost.

B. System Models

In this subsection, we present three system models, including a topology model, a content model and a user request model for the problem formulation.

1) Topology Model: The underlying physical infrastructure that supports CCMN can be viewed as a connected graph G by modeling N data centers as the nodes and the network connectivities as the edges. Each node is associated with storage and computation resources, and each link is associated with bandwidth resources. All those resources are finite in CCMN. This work will use four well-known deployed backbone topologies as shown in section IV-E as examples.

2) Content Model: The content model defines the basic properties of a catalog of M unique contents to be delivered in a time window t. For each content $k = 1, ..., M$, both its size B_k and its popularity r_k are random variables.

In particular, the content size follows a long-tailed distribution [22], [23], [24]. In this paper, we use the bounded Pareto distribution with the cumulative distribution function as,

$$F(x) = \frac{1 - B_L^{\alpha} x^{-\alpha}}{1 - (\frac{B_L}{B_U})^{\alpha}}, B_L \leq x \leq B_U, \quad (1)$$

where B_L and B_U are the smallest and largest content size, and α is the shape parameter having $\alpha > 0$. The larger α is, the smaller the percentage of large files are.

The popularity r_k can be measured by the download times R_k in a period t. It follows Zipf-like distribution [25], [26] as,

$$r_k = \frac{R_k}{R_{tot}} = \frac{k^{-\beta}}{\sum_{s=1}^{M} s^{-\beta}}, \quad (2)$$

where $R_{tot} = \sum_{k=1}^{M} R_k$ is the total amount of requests for all the contents, and β is the shape parameter of Zipf distribution. The large β indicates the relatively small percentage of very popular ones. The two distributions will be verified by real traces in section VI.

3) User Request Model: Each site in CCMN is also a service entry point attached with an access network. Each user request is served by the nearest node which holds a replica of the requested content. We assume a uniform traffic pattern, that the traffic load from each entry point is almost the same in a given time window t. This assumption could be reasonable, because when deploying CDN sites, a fundamental principle is to balance the traffic among them, so that the performance can be optimized [27]. In other words, more sites are available for the locations with more user requests, resulting in a uniform traffic pattern. For example, in the US IP backbone network and U.S 64 backbone network [28], the density of nodes beyond the west and east coast of the U.S. is much higher than the one in the middle area.

C. Problem Formulation

The objective is to minimize the monetary cost associated with the usage of bandwidth, storage and computation resources, which depends on the content placement strategy. Specifically, the storage cost depends on the content size and the number of replicas. The bandwidth cost depends on the hop distance from users to replicas. And the computation cost is incurred by processing each user request, which is independent with the number of replicas. As a result, in this paper, we only focus on the storage and bandwidth cost as follows.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

Storage cost for content k over a time duration t is a function of n_k as,

$$C_{st}^k(n_k) = c_{st} B_k n_k t, \tag{3}$$

where B_k is the size, c_{st} is per-bit cost per unit of time.

Bandwidth cost for content k with B_k size and R_k download times over the same time duration t is a function with respect to the number of replicas n_k and the underlying topology G, as,

$$C_{tr}^k(n_k, G) = R_k B_k c_{tr} H(n_k, G), \tag{4}$$

where c_{tr} is per hop cost for transmitting per bit data, $H(n_k, G)$ is the mean hop distance between an user and the closest replica of content k.

Note that, this paper considers the on-demand cost model of utilizing cloud resources in media cloud. Although there are a few alternative models, such as reserved model (i.e., pay a fixed amount of cost to reserve cloud resources in prior to real usage) and spots model (i.e., bid to use cloud resources), on-demand model is most suitable when the content popularity is unknown in advantage. Specifically, in this case, the spot instance cannot guarantee QoS, and the reserved resources could easily be wasted.

Based on the system models and our assumptions, we formulate the content placement problem as a constrained optimization problem, with an objective to minimize the combined storage and networking cost as,

$$\min \quad f(\mathbf{n}) = c_{st}\mathbf{B}^T\mathbf{n}t + c_{tr}\mathbf{B}^T\mathbf{H}\mathbf{R} \tag{5}$$

$$\text{s.t.} \quad g_i(\mathbf{n}) = 1 - n_i \leq 0, \quad for\ i = 1, ..., M, \tag{6}$$

$$g_{i+M}(\mathbf{n}) = n_i - N \leq 0, \quad for\ i = 1, ..., M, \tag{7}$$

$$g_{2M+1}(\mathbf{n}) = \mathbf{B}^T\mathbf{n} - S_{tot} \leq 0, \tag{8}$$

$$g_{2M+2}(\mathbf{n}) = \mathbf{B}^T\mathbf{H}\mathbf{R} - T_{tot} \leq 0, \tag{9}$$

where M is the number of unique contents, N is the number of data centers, $\mathbf{H}$=$diag(H(n_1, G), ..., H(n_M, G))$, $\mathbf{n}$=$(n_1, ..., n_M)$, $\mathbf{B}$=$(B_1, ..., B_M)^T$, and $\mathbf{R}$=$(R_1, ..., R_M)^T$.

The constraint (6) indicates that there must be at least one version of each content stored by the origin server within the media cloud, and (7) indicates that the number of replicas cannot exceed the number of all cloud nodes. The constraints (8) and (9) capture the total storage and bandwidth capacity limitations respectively. Note that, in this formulation, we relax the integer constraint of n_k (i.e., n_k can be decimal, satisfying $n_k \in [1, N]$), for a lower bound solution. This simplification can be reasonable because the video contents are usually sliced into a set of small segments. And for different content, the slicing scheme and the block size could be different. As a result, this allows the cloud nodes to cache any fraction of every content. Besides, we also do not consider the initial cost of placing the replica from the origin server to the cloud node, because for each content, such cost is only incurred for the first time.

We will adopt a two-step method. First, for a single content k, we will find the function of optimal mean hop distance H in terms of n_k in section IV. Second, we will use the obtained result to address the optimization problem for a catalog of content in section V.

IV. Optimal Single Content Placement

In this section, we cast the single content placement as a variant of k-center problem [11] as follows. Given a network topology $G=(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of data centers that $N = |V|$, and $\mathbf{E}$ is the set of undirected links between them. If n ($n \leq N$) replicas of a content item are cached at a set of V_R different nodes ($n = |V_R|$), other N-n nodes have to access a content copy from the nearest hosting node via one or more hops. We aim to find the relationship between the number of replicas n and the average hop distance H for an user. Here we formally define H as,

$$H = \frac{1}{N}\sum_{i \in V} \min_{j \in V_R} h_{ij}, \tag{10}$$

where h_{ij} is the shortest path from node $i \in V$ to node $j \in V_R$.

Due to the NP-completeness of this problem [11], the complexity of directly solving it can be prohibitive, especially when the size of network is large. We thus look into analytical functional form that relates H and n for random regular graphs, which are good models for the dynamic network topology in CCMN. We first study a few examples of generalized Moore Graph[1], such as ring topology, the Petersen graph, and the Heawood Graph. The symmetry and well-defined structures of these graphs enable us not only to derive analytical form of $H(n)$ but also to gain understandings of the efficient topological structures that allow for efficient content delivery. Using these properties, we further derive a lower and an upper bound of H for a random regular graph with N nodes and a degree of Δ. Finally, we provide close estimates for irregular deployed networks, for which, the exact analytical solutions are difficult to obtain. The generated results serve as a foundation to solve the optimization problem in section V.

A. Ring Graph

We first study ring graph, which can be treated as a trivial case of Generalized Moore Graphs. It can be observed that for any given number of replica in ring topology, the optimal replica placement is to maximize the hop distances among those replicas. Thus, for a placement of n replicas on a ring graph of N nodes, we have,

$$N = \lfloor \frac{N}{n} \rfloor n + r = r_0 n + r, \tag{11}$$

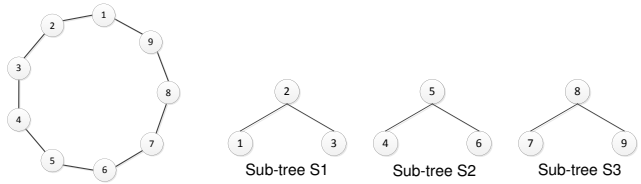
where r is the reminder of dividing N by n. The parameter r_0 indicates that there are $n - r$ replicas, each serving the download requests from a group of r_0 nodes. And each of the remaining r replicas serve $r_0 + 1$ nodes. Figure 2 gives an example, where $N = 9$, $n = 3$, $r = 0$, and $r_0 = 3$.

We adopt the divide and conquer approach to derive the solution. In particular, we separately consider two cases (i.e., N is odd and even).

[1]Generalized Moore Graph is a Δ-degree regular graph, where each node has a Δ-ary spanning tree that is full at each level, except probably the last.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



(a) 9-node Ring Graph (b) Optimal Partition when n=3

Fig. 2. Demonstration of optimal partition for 3 replicas in ring topology. (a) An example of ring topology with 9 nodes; (b) The root of each partition refers to the node holding the replica, and other nodes forward the content from the root to its users.

When N is odd, a ring topology is a trivial case of Moore Graph[2] with node degree $\Delta = 2$, that,

$$H(n) = \frac{1}{N}[n(\frac{r_0-1}{2}+1)\frac{r_0-1}{2} + r\frac{r_0+1}{2}] = \frac{r_0+1}{2}[1-\frac{n}{N}\frac{r_0+1}{2}], \quad r_0 \; is \; odd. \tag{12}$$

The first term denotes the total hops for all n replicas serving r_0 nodes, and the second term denotes the hop distances from the additional one node served by r of the replicas.

When N is even, a ring topology is a generalized Moore graph. In this case, $H(n)$ can be derived in the same way as,

$$H(n) = \frac{r_0}{2}(1-\frac{n}{N}\frac{r_0}{2}), \quad r_0 \; is \; even. \tag{13}$$

By combining Eq. (12) and (13), we have,

$$H(n) = \lfloor\frac{r_0+1}{2}\rfloor(1-\frac{n}{N}\lfloor\frac{r_0+1}{2}\rfloor). \tag{14}$$

This approach (i.e., Eq. (11)) and results (i.e., Eq. (14)) provide guidelines for our further analysis.

B. Petersen Graph & Heawood Graph

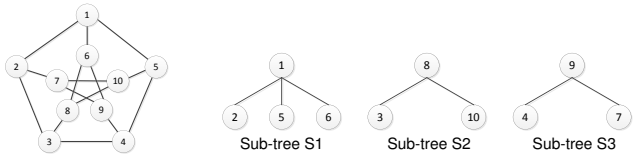
We further study the properties of content placement problem in Generalized Moore Graph of degree $\Delta = 3$. In particular we focus on Petersen graph (an example of Moore graph with $N = 10$), and Heawood graph (an example of Generalized Moore Graph with $N = 14$).

1) Numerical Results: To obtain the numerical results for a given number of replicas n in the two examples, we exhaustively calculate the average shortest distance H from n selected centers to all other nodes for all the possible combinations. After all the iterations are finished, the solution with the smallest H is then selected as the optimal result.

Figure 3 and Figure 4 present the optimal replica placement solutions when $n = 3$ for Petersen graph and Heawood graph, respectively. The key observation here is that, the minimum hop routing spanning trees are rooted at each of replicas.

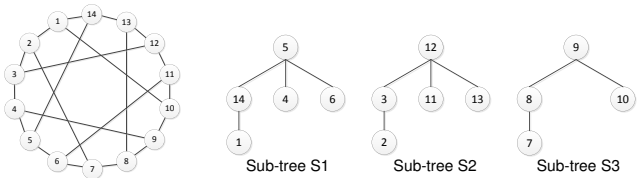
From the observation, the original replica placement problem can be transformed into a graph partition problem. That is, for a given number of replica n, and a Generalized Moore Graph $\mathcal{G}(\Delta, N)$ (where $N\Delta$ must be even), how to optimally partition $\mathcal{G}(\Delta, N)$ into n sub-trees $S_1, S_2, ..., S_n$,

[2]Moore Graph is a Generalized Moore Graph where each node has a full Δ-ary routing spanning tree at all the levels.



(a) Petersen Graph (b) Optimal Partition when n=3

Fig. 3. Demonstration of optimal partition for 3 replicas in Petersen Graph. (a) Petersen Graph; (b) The root of each partition refers to the node holding the replica, and other nodes forward the content from the root to its users.



(a) Heawood Graph (b) Optimal Partition when n=3

Fig. 4. Demonstration of optimal partition for 3 replicas in Heawood Graph. (a) Heawood Graph; (b) The root of each partition refers to the node holding the replica, and other nodes forward the content from the root to its users.

where $\sum_{i=0}^{n}|S_i| = N$, and $S_i \cap S_j = \emptyset$ for $i \neq j$, so that the aggregated hop distance, from the roots to all other nodes in all sub-trees can be minimized. Then the average hop distance can be calculated by dividing this aggregated hop distance by total node number N. Mathematically, by assuming A_i as the aggregated hop distance from all the nodes to the root in subtree S_i, we can obtain the mean hop distance H between any user and his nearest replica as,

$$H = \sum_{i=1}^{n} A_i/N. \tag{15}$$

2) A Lower Bound of Aggregated Hop Distance: To analytically obtain a lower bound of aggregated hop distances when partitioning a Generalized Moore Graph, we first derive two important properties, which are invariant under all cases.

Property 1: Each partition forms a Δ-ary spanning tree that is full at each level, except probably the last. (i.e., each sub-tree with its height at h, must have a root at its 0-th level, and $\Delta(\Delta-1)^{k-1}$ nodes at its k-th level, $k = 1, ..., h-2$).

Proof: See Appendix A for a completed proof. ■

Property 2: The node number of each sub-tree is either $\lfloor N/n \rfloor$ or $\lceil N/n \rceil$ (i.e., a uniform partition).

Proof: See Appendix B for a completed proof. ■

Following those properties, we find that the optimal solution should contain r sub-trees, each serving r_0+1 nodes (i.e., $|S_1| = |S_2| = ... = |S_r| = r_0 + 1$). There are also $n-r$ nodes serving r_0 nodes (i.e., $|S_{r+1}| = |S_{r+2}| = ... = |S_n| = r_0$), where r is the reminder of dividing N by n, and $r_0 = \lfloor N/n \rfloor$). For instance, Figure 3(b) illustrates the optimal graph partitions for Petersen graph when $n = 3$. Both sub-tree S_2 and S_3 have $r_0 = \lfloor 10/3 \rfloor = 3$ nodes, and sub-tree S_1 has $r_0 + 1 = 4$ nodes. Note that the lower bound may not be reached, because the node at the border of two partitions may have the same shortest distance to more than one replica (e.g., in Figure 4(b), the hop distances from node 3 to replica 8 and 4 are both 1).

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

3) Analytical Solution: We derive the analytical lower bound for Petersen graph and Heawood graph by considering two parts. First, we calculate the aggregated hop distances A_o from all n sub-trees, each is with r_0 nodes, and all its levels are full. For the second part, we compute the extra distance from the additional node in the last level of those r sub-trees.

To proceed the analysis, we first investigate the optimal height h of each sub-tree, where $\Delta \geq 3$, is the least integer that satisfies the following condition,

$$\Delta \frac{(\Delta-1)^h - 1}{\Delta - 2} \geq r_0 - 1. \tag{16}$$

The expression on the left side is the total number of nodes for a full tree with a height of h. The expression on the right side is the number of the leaves (excluding the root) in the subtree S_{r+1} to S_n.

From Eq. (16), we have h as a function of Δ and r_0 as,

$$h = \lceil \log_{\Delta-1} \frac{\Delta r_0 - 2r_0 + 2}{\Delta} \rceil. \tag{17}$$

Since the last level of those sub-trees may not be necessary full, we have to find the node count o in the last level at each sub-tree as,

$$o = r_0 - 1 - \Delta \frac{(\Delta-1)^{h-1} - 1}{\Delta - 2}. \tag{18}$$

Thus, the aggregated hop distance from each of the n trees with r_0 nodes for both the Petersen and the Heawood graph, A_o, can be expressed as,

$$A_o(h, o) = 2^{h-1}(3h - 6) + 3 + ho, \tag{19}$$

where the first term refers to the aggregated hop distance from the nodes excluding the first and the last level. The second term, '3' refers to the distance from all the nodes in the first level, and the last term refers to the distance from all the nodes in the last level.

Then we compute the aggregated hop distance φ from the additional node from each of the r trees, as,

$$\varphi(h, o) = h + \delta(\Delta(\Delta-1)^{h-1} - o), \tag{20}$$

where $\delta(x)$ is an indicator function, indicating whether those trees are full before adding the node, as,

$$\delta(x) = \begin{cases} 1, & x = 0 \\ 0, & otherwise \end{cases}. \tag{21}$$

Finally, the lower bound of the average hop distance $H(n)$ as a function of n in the Petersen graph can be solved as,

$$H(n) \geq \frac{1}{10}(nA_o(h, o) + r\varphi(h, o)). \tag{22}$$

Similarly, the lower bound for Heawood graph follows exactly the same format as that of Petersen graph as,

$$H(n) \geq \frac{1}{14}(nA_o(h, o) + r\varphi(h, o)). \tag{23}$$

It can be seen that, the only difference between Eq. (22) and Eq. (23) is the first parameter. This inspires us to extend the analysis to arbitrary Generalized Moore Graph by adopting the same rationale.

C. Generalized Moore graph

We first compute the aggregated hop distance A_o for each of the n sub-trees. Note that each sub-tree has r_0 nodes,

$$A_o(h, o) = \Delta \sum_{k=0}^{h-2} (k+1)(\Delta-1)^k + ho, \tag{24}$$

where the first term is the aggregated hop distance from all the nodes excluding the last level. There are Δ nodes in the first level, and each of those node has $\Delta - 1$ children. Therefore, it leads to the result, that there are $\Delta(\Delta-1)^k$ nodes locating $k+1$ hops away from the root, for $k = 0, 1, ..., h-2$. The second term is the sum of total hop distance to the root from the nodes on the last level. We note that in this term o still has the same expression as in equation (18).

Thus, we have a lower bound for Generalized Moore Graph $\mathcal{G}_M(\Delta, N)$ with N nodes and degree Δ, as,

$$H(n, \mathcal{G}_M(\Delta, N)) \geq \frac{1}{N}(nA_o(h, o) + r\varphi(h, o)). \tag{25}$$

This result will be further used to derive an upper bound and a lower bound in random regular graph.

D. Random Regular Graph

In random regular graph, each node still has the same degree Δ, but not necessarily the same connectivity pattern. To extend the analysis on content placement problem in such graph, we first get a lower and an upper bound for H. Then we derive an approximation that captures the fundamental scaling of H as a function of N, Δ, and n.

1) Lower Bound: There is a lower bound, when all the levels of each sub-tree are full except the last. This case is exactly the same as the analysis for Generalized Moore Graph. Therefore, this lower bound can be expressed as,

$$H(n, \mathcal{G}(\Delta, N))_l = \frac{1}{N}(nA_o(h, o) + r\varphi(h, o)). \tag{26}$$

To further simplify the functional form of $H(n, G(\Delta, N))_l$, we make the simplification that N is dividable by n (i.e., r_0=N/n and r=0). We also assume that $\Delta \approx \Delta + 1$. With these simplifications, we have $h \approx \log_\Delta r_0$, $o \approx 0$, and $A_o(h, o) \approx \sum_{k=1}^{h-1} k\Delta^k = \frac{((h-1)\Delta - h)\Delta^h + \Delta}{(\Delta-1)^2} > \frac{r_o(h(\Delta-1)-\Delta)}{(\Delta-1)^2}$. Finally, Eq. (26) can be reduced to the following form,

$$H(n, \mathcal{G}(\Delta, N))_l \approx c_1 \log_\Delta \frac{N}{n} + c_2, \tag{27}$$

where $c_1 \approx 1/(\Delta-1)$ and $c_2 \approx \Delta/(\Delta-1)^2$ are independent of n and N.

2) Upper Bound: We use the upper bound for random regular graph obtained from [29], to get the height upper bound h_u of each sub-tree after partitioning as,

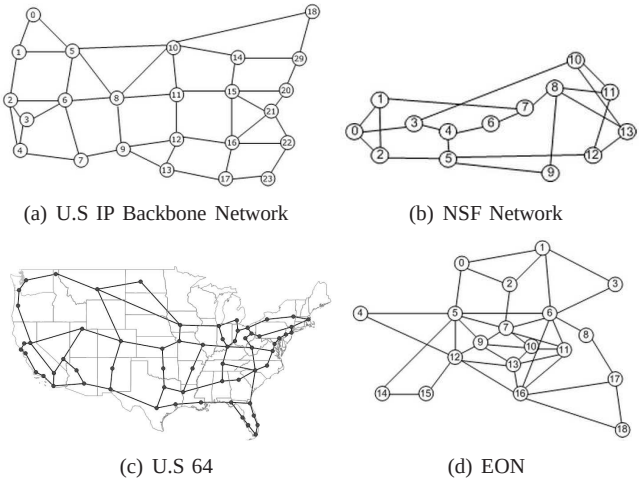
$$h_u \leq \log_\Delta (2+\epsilon) r_0 \log r_0 < 2 \log_\Delta c_3 r_0, \tag{28}$$

where c_3 is a constant as $\sqrt{2+\epsilon}$, and ϵ is a small number.

As a result, it is safe to draw the conclusion that, when each sub-tree with the height at $h_u = 2 \log_\Delta c_2 r_0$ is full at all the levels, it holds an upper bound for the aggregated hop distance (note in this case, the node number of each sub-tree may be

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 5. Deployed networks: (a) the U.S IP backbone network; (b) the National Science Foundation Network (NSFNET); (c) the hypothetical backbone network US64; (d) the European optical network (EON)

larger than r_0). By using the same approach on deriving the lower bound, we have an upper bound as,

$$H(n, \mathcal{G}(\Delta, N))_u \approx (\Delta c_3)^2 \log_\Delta \frac{c_3 N}{n} + c_4, \qquad (29)$$

where c_4 is a constant which is independent with n and N.

3) Approximation Result: By combining Eq. (27) and (29), we have the approximation form as,

$$H(n_k, \mathcal{G}(\Delta, N)) \approx C_1 \log_\Delta \frac{C_2 N}{n_k}, \qquad (30)$$

where C_1 and C_2 are topology-specific coefficients determined by the curve fitting. That is, in practice, for a specific topology, we can first pre-calculate the exact hop distance for a given number of replicas. Then based on these sampling data, we can approximate C_1 and C_2 using Eq. (30).

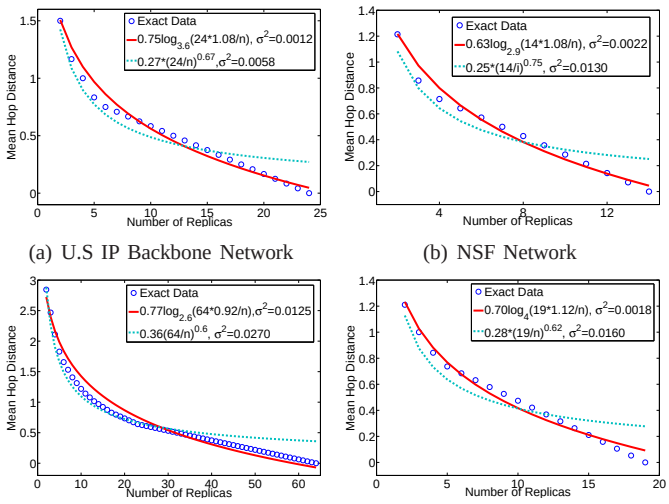
The result suggests that, assuming a uniform user request pattern, the mean hop distance between users and their closest replica is a logarithmic function with respect to the reciprocal of the number of replicas in random regular graph. Compared with existing approximation form in power-law function (i.e., $H = C_1(N/n)^{C_2}$) [30], [31], our solution can further capture the topological characteristics via Δ in logarithmic function, which is derived from the bound analysis. As a result, it is expected to be able to more accurately represent the relationship between H and n, for different network topologies. As to be shown in the following session, this convex form allows us to analytically solve content placement or similar resource allocation problem.

E. Deployed Networks

We study four representative deployed networks [28] as shown in figure 5, which roughly share the same basic properties with the random regular graph. We make this argument by evaluating the graph diameter (i.e., the maximum distance among all pairs of nodes in a graph G), which is one of the most important metrics [29]. Specifically, we evaluate the analytical lower bound (i.e., $a_l \approx \lfloor \log_{\Delta-1} N \rfloor + 1$) and upper bound (i.e., $a_u \approx \lceil \log_{\Delta-1} 2n \ln n \rceil + 1$) [29] of random regular graphs, which have the same degree (i.e., average node degree) and size with those in deployed networks, and compare the analytical results on random regular graph with the deployed network as shown in table II. It can be seen that, the diameters of those deployed graphs are between the lower and upper bound of their counterparts in random regular graph forms.

TABLE II
THE COMPARISON ON DIAMETERS

	U.S IP	NSF	U.S 64	EON
Lower Bound	4	4	10	3
Diameter	6	4	16	4
Upper Bound	7	8	16	6



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 6. Estimated functions for deployed networks. The obtained logarithmic function is more precise than the existing estimation in all four topologies.

To evaluate the performance of our solution in deployed networks, we compare the analytical results generated from our logarithmic function with the ones that use a power-law function (i.e., $H(n_k) \approx C_1(\frac{N}{n})^{C_2}$) [30], [31]. Specifically, we use an exhaustive approach (i.e., for every n, we try every possible combination, and use Dijkstra algorithm to get the shortest distance between every pair of nodes), to calculate the exact optimal placement of n replicas and the associated $H(n)$, for $n = 1, 2, ..., N$. Next we fit these results with both Eq. (30), and the power-law function from [30], [31]. The accuracy of the curve-fitting is evaluated by Mean Squared Error (MSE), denoted as,

$$\sigma^2 = \frac{1}{N} \sum_{i=0}^{N} (\tilde{x_i} - x_i)^2, \qquad (31)$$

where $\tilde{x_i}$ is the estimation, and x_i is the numerical result.

Figure 6 illustrates this comparison. In all the cases, our obtained logarithmic function is able to represent the real condition more precisely with much lower MSE. As a result, we use this function to estimate H, and drive the design of our algorithm to solve the content placement problem for a catalog of content in the next section.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

V. Optimal Placement for A Catalog of Content

By substituting Eq. (30) into the origin problem (i.e., Eq. (5)-(9)), we get a convex optimization problem. Specifically, the constraint (9) is convex, because each item, $H(n_k) = C_1 \log_\Delta(C_2 N/n_k)$, for $k = 1, ..., M$, in matrix $\mathbf{H}$, is convex. Constraints (6)-(8) are all linear functions with respect to the number of replicas. And the objective function is also convex, which is the sum of a linear function and a convex function.

Due to the convexity of this problem, this section first adopts a feasible direction algorithm to find the optimal solution in general condition. By using the Karush-Kuhn-Tucker (KKT) condition, we also present the optimal solution for the problem without capacity constraints to gain some fundamental insights.

A. General Solution

In this subsection, we apply the Topkis-Veinott's feasible direction method [32] to solve the convex optimization problem. Algorithm 1 describes the details of this method.

Algorithm 1 The Feasible Direction Algorithm

Input: The objective function $f(\mathbf{n})$
The constrains $g_i(\mathbf{n}),\ i = 1, 2, ..., 2n + 2$
One feasible solution $\mathbf{n}_1$ such that all $g_i(\mathbf{n}_1) \leq 0$
Output: The optimal solution $\mathbf{n}^*$

1: Initialize k=1;
2: Find a feasible descent direction $\mathbf{d}_k$ by solving an one-shot problem as,
$\min z_k$
$s.t.\ \nabla f(\mathbf{n}_k)^t \mathbf{d} - z_k \leq 0$
$\nabla g_i(\mathbf{n}_k)^t \mathbf{d} - z_k \leq -g_i(\mathbf{n}_k), \quad for\ i = 1, ..., 2M + 2$
$-1 \leq d_j \leq 1, \quad for\ j = 1, ..., 2M + 2$
if $z_k = 0$ **then** go to Step 4
else go to Step 3
3: Find the step λ_k along direction $\mathbf{d}_k$ by solving an one-shot problem as,
$\min f(\mathbf{n}_k + \lambda \mathbf{d}_k)$
$s.t.\ 0 \leq \lambda \leq \lambda_{max}$
$\lambda_{max} = \sup\{\lambda : g_i(\mathbf{n}_k + \lambda \mathbf{d}_k,\ for\ i = 1, ...2M + 2)\}$
Update $\mathbf{n}_{k+1} = \mathbf{n}_k + \lambda_k \mathbf{d}_k$
Update $k = k + 1$
go to Step 2
4: Finish, return $\mathbf{n}_k$ as $\mathbf{n}^*$

The optimality criteria of this algorithm is to iteratively move from one feasible point to another improved one. Specifically, the algorithm starts with one random feasible point $\mathbf{n}_1$, and in each iteration, finds the reduced gradient direction $\mathbf{d}_k$ by solving a one-dimensional optimization problem to determine how far to proceed along this direction. As a result, it can be proved that the new point $\mathbf{n}_{k+1} = \mathbf{n}_k + \lambda_k \mathbf{d}_k$ must be better or equal than the previous point $\mathbf{n}_k$. This process is repeated until the optimal solution is found.

Our algorithm also assures the convergence into the optimal point. To prove this argument, first, we show the generated solution is a Fritz John point (i.e., a feasible point that is local optimal) by using the proof from [32]. Specifically, it has been proved that in the feasible direction algorithm, $\mathbf{n_k}$ is a Fritz John point if and only $z_k = 0$, which is exactly the termination criterion of our algorithm. Second, we prove the obtained Fritz John point is also a KKT point (i.e., the feasible point that is global optimal) by checking those additional restrictions of KKT conditions. In particular, it has been proved that [33], if the objective function and all the inequality constraints are continuously differentiable convex functions, and the Lagrangian multipliers on the gradient of the objective function are nonzero, then the Fritz John point is also a KKT point. Because our problem meets those restrictions, the global optimality of the solution is guaranteed.

The complexity of this algorithm is $O(M)$. This statement can be proved as follows. First, it has been proved that [32], the feasible direction method can always find the optimal solution in finite iteration rounds C. For each iteration, the algorithm takes $2M+2$ steps to find the reduced gradient direction by solving the one-dimensional optimization problem with $2M+2$ constraints. As a result, the total steps are $C(2M+2)$, and the complexity is $O(M)$.

B. Optimization without Capacity Constraints

Algorithm 1 provides a way to solve the optimization problem under constraints, but it is difficult to generate an analytical result by this algorithm. As a result, we consider the optimization problem without capacity constraints, and obtain analytical solutions for fundamental insights.

By removing the constraints (8) and (9), the optimization problem becomes an optimization process over a box (i.e., $1 \leq n_i \leq N,\ i = 1, ..., M$). Therefore, we can turn it into an unconstrained problem, by ensuring and $\nabla f(n_k) \leq 0$ for $n_k = N$, and $\nabla f(n_k) \geq 0$ for $n_k = 1$. By considering the KKT first order conditions, we could find the optimal number of replicas for each content by letting $\nabla f(n_k) = 0$. Thus we have,

$$\nabla f(n_k) = c_{st} B_k - \frac{c_{tr} B_k R_k C_1}{n_k \ln \Delta} = 0. \tag{32}$$

By solving Eq. (32), and jointly considering the constraint (6) and (7), we have the optimal number of replicas for a catalog of content as,

$$n_k^\star = \max\{1, \min(N, \frac{c_{tr} A R_k}{c_{st} t \ln d})\}. \tag{33}$$

This result suggests that, under feasible constraints (i.e., $n_k \in (1, N)$, unconstrained storage and bandwidth resources), the optimal number of replicas $n_k^\star$ of content k is proportional to its downloaded times R_k in those deployed networks.

Moreover, by substituting the optimal number of replicas into the origin problem, we can get the optimal storage space and the optimal network bandwidth as,

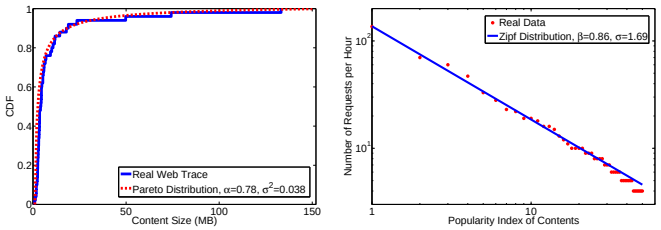
$$T_{tot}^\star = c_{tr} \mathbf{B}^T \mathbf{H}^\star \mathbf{R}, \tag{34}$$

$$S_{tot}^\star = \mathbf{B}^T \mathbf{n}^\star, \tag{35}$$

where $\mathbf{H}^\star = diag(C_1 \log_\Delta \frac{C_2 N}{n_1^\star}, ..., C_1 \log_\Delta \frac{C_2 N}{n_M^\star})$, and $\mathbf{n}^\star = (n_1^\star, ..., n_M^\star)$. In practice, any other resource allocation schemes, that are different with $T_{tot}^\star$ and $S_{tot}^\star$, will lead to either over- or under-provision.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



(a) The Distribution of Content Size (b) The Distribution of Popularity

Fig. 7. The real-world trace from a video provider. (a) The size of different contents follows Pareto distribution; (b) The popularity of different contents follows Zipf distribution.

VI. Performance Evaluation

This section uses real-world traces to evaluate the obtained monetary cost efficient content placement strategy.

A. Experimental Settings

The real-world trace was captured from one of the leading Chinese video providers. It contains the request history log of 50 contents over one week (i.e., from 10/Nov/2012 to 17/Nov/2012) from all over the world. Figure 7 shows the distributions of both the size and the popularity in terms of the downloaded times of this real data.

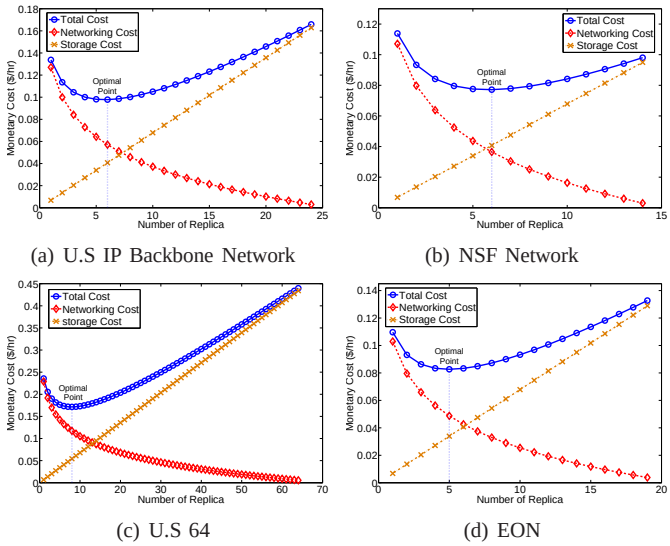
It can be seen that, the real trace agrees with the content size distribution model and the user request distribution model as discussed in section III. It verifies our assumptions on these two models. In particular, the bounded Pareto distribution with its parameter $\alpha = 0.78$, $B_U = 150$ MB and $B_L = 1$ MB matches well with the real content size distribution, where the estimation MSE is $\sigma^2 = 0.038$. And the Zipf distribution with its parameter $\beta = 0.86$ and $R_{tot} = 2.9 * 10^4$ represents the real popularity distribution of each content, where the estimation RMSE (root of MSE) is $\sigma = 1.69$.

We adopt pay-per-use CDN pricing model from GoGird [34], one leading cloud service provider, as an example to drive our experiment. Specifically, storage price is $c_{st} = \$0.6/GB$ per month and bandwidth price is $c_{tr} = \$0.25/GB$. Note, similar pricing models are widely used by other famous cloud service providers, such as Microsoft Azure ($c_{st} = \$0.19/GB$ per month and $c_{tr} = \$0.19/GB$) [35], and Rackspace ($c_{st} = \$0.18/GB$ per month and $c_{tr} = \$0.10/GB$) [36].

We apply those deployed networks in section IV to deliver the traces, and produce the optimal solutions by our methods.

B. Fundamental Trade-off for Single Content Placement

Figure 8 uses the 10^{th} popular content in the trace as an example to show the fundamental trade-off between storage and bandwidth cost for single content placement. It can be seen, in all the four deployed networks, the storage cost is in proportional to the number of replicas, and the bandwidth cost follows a logarithmic function with the respect to the reciprocal of the number of replicas. The trade-off here is that, as the number of replicas grows, the storage cost increases and the bandwidth cost decreases, while the total cost is convex. Consequently, how to balance this trade-off to find the optimal



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 8. Monetary cost for the 10^{th} popular content. The total cost is a convex function with respect to the number of replicas n in all cases.

number of replicas for each content under resource constraints, drives this work.

Besides, figure 8 also indicates the optimal solution depend on its underlying topology G. It can be seen, for the same content to be delivered over different networks, the minimal total cost can be achieved by different number of replicas. For example, if we do not consider the resource constraints, the optimal number of replicas for the 10^{th} popular content is 6 in U.S IP backbone network, 6 in NSF network, 8 in U.S 64 network, and 5 in EON.

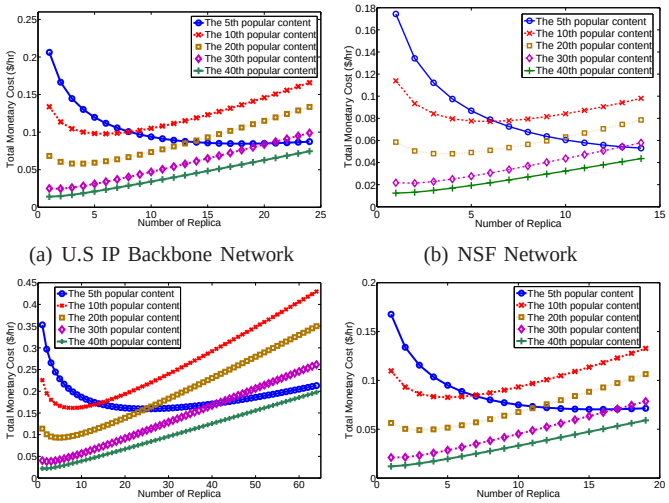
C. Fundamental Trade-off for A Catalog of Content

Figure 9 illustrates the total monetary cost with respect to the number of replicas for different contents with different popularity (i.e., the 5^{th}, the 10^{th}, the 20^{th}, the 30^{th}, and the 40^{th} popular content). It can be observed, as the content popularity decreases, the optimal number of replicas decreases accordingly. This indicates the trade-off for a catalog of content depends on the content popularity. For example, in U.S IP backbone network, if we do not consider the resource constraints, the optimal number of replicas is 12 for the 5^{th} popular content, 6 for the 10^{th} popular content, 3 for the 20^{th} popular content, 2 for the 30^{th} popular content, and 1 for the 40^{th} popular content.

Figure 9 also indicates, the total monetary cost depends on not only content popularity but also content size. For example, in NSF network, when the number of replicas for the 5^{th} popular content is less than 6, its overall monetary cost is the highest among all the contents, mainly because the enormous bandwidth cost incurred by insufficient replicas and the large amount of downloaded times. When the replica number of this content is more than 10, its overall cost is lower than the cost of the 20^{th} popular content, which attracts much less downloaded times (i.e., 14.38 times/hr) than the 5^{th} popular content does (i.e., 74.23 times/hr). This can be attributed to the relative small size of the 5^{th} popular content (i.e., 5.87

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

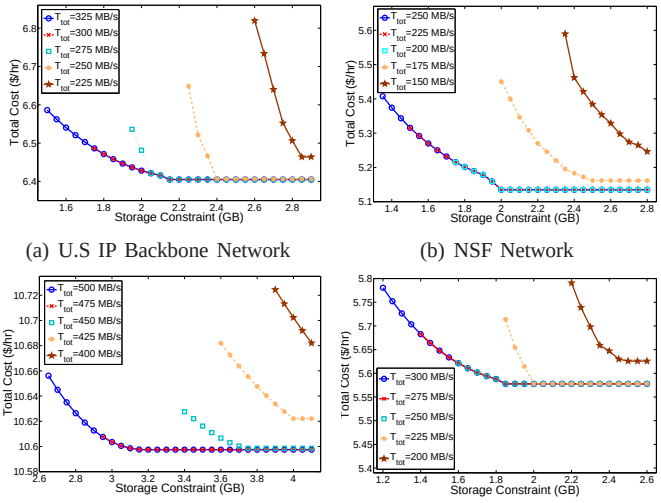
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 9. Total monetary cost for different content. The overall cost function for the content with different popularity is different.



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 10. Total monetary cost under different constraint combinations. The total cost decreases as the storage/bandwidth constraint increases until the other constraint becomes active.

MB) compared to the size of the 20^{th} popular content (i.e., 9.39 MB), and the consequent small storage cost.

D. Optimal Overall Monetary Cost

Figure 10 presents the optimal overall monetary cost under various combinations of storage and bandwidth constraints. We carefully select the range of these two parameters to emulate all possible conditions (i.e., one of the constraint is active, both are active, and both are inactive). The objective is to understand the impact of different constraints.

For a given T_{tot}, there are two phases as the growth of S_{tot}, depending on whether the storage constraint is active. Specifically, in the first phase, the limited storage capacity constrains the optimal solution. Thus, the total monetary cost decreases as the storage space increases (e.g., $T_{tot}=300$ MB/s and $S_{tot} \leq 2.2$ GB in U.S IP backbone). Whereas, in the second phase, the storage capacity is no longer the dominating factor, and the optimal cost stays the same (e.g., $T_{tot}=300$ MB/s and $S_{tot} \geq 2.2$ GB in U.S IP backbone). The threshold to separate these two phases decrease as the bandwidth capacity increases, until the bandwidth constraint also becomes inactive (e.g., in U.S IP backbone, the threshold is 2.2 GB when $T_{tot} \geq 300$ MB/s, and 2.4 GB when $T_{tot} = 250$ MB/s). It is also noticeable that, if both the storage and bandwidth capacity are insufficient (e.g., $S_{tot} \leq 1.5$ GB and $T_{tot}=350$ MB/s in U.S IP backbone network), there could be no feasible solution. This set of observations again verifies the trade-off between these two kinds of resources.

E. Distribution for Optimal Number of Replicas

Figure 11 plots the optimal number of replicas for a catalog of content in different conditions (i.e., unconstrained, constrained by storage resource, and constrained by bandwidth resource).

First, we find that, all the curves are roughly linear in a log-log graph. This suggests the optimal number of replicas for each content follows a power-law distribution to its popularity rank, regardless of whether the solution is constrained by resource limitations. Note, this insight only applies to $n_k \in (1, N)$.

Second, we notice that, when the bandwidth constraint is active, the optimal number of replicas is always the highest; while the storage constraint is active, the optimal number of replicas is the lowest. This still can be attributed to the trade-off between the two resources. Specifically, when bandwidth resource is limited, we have to place more replicas to reduce the bandwidth cost. And when storage space is limited, less replicas are allowed to be placed. The optimal criteria of balancing these competition factors among different contents, is ensured by the reduced gradient direction method.

VII. Algorithm Evaluation

To further evaluate the performance and optimality of our proposed algorithm, we also adopt three natural heuristics (i.e., random algorithm, popularity based algorithm and near-optimal algorithm) inspired by [12], [13] to solve the same problem. We then compare the results generated by those algorithms with our approach, in terms of the obtained total monetary cost and the computational complexity.

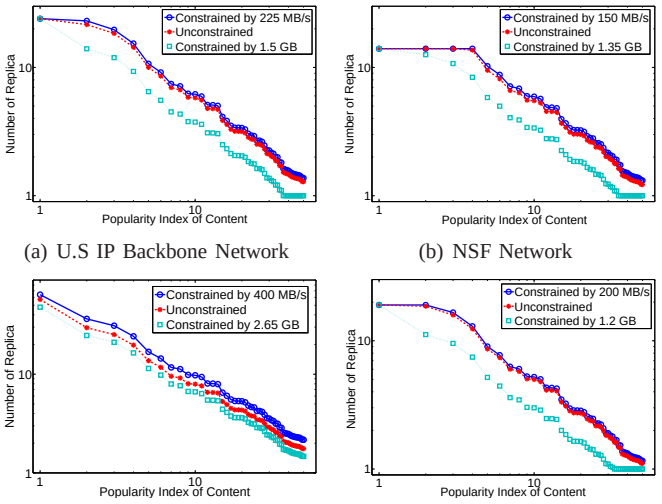
A. Heuristics Design

Since the design of those heuristics could be very complex if the problem is constrained by both bandwidth and storage resources, we only consider the storage constraint here. The detailed descriptions on these three heuristics are as follows.

Random Algorithm [12] assigns replicas of each content to different nodes randomly, subject to the total storage constraints. Specifically, we first allocate one replica to all the contents, to meet the lower bound of the number of replicas and the storage constraint (if the storage resource is still insufficient in this case, then there is no feasible solution). Then, for each content k, we randomly pick additional n_k

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

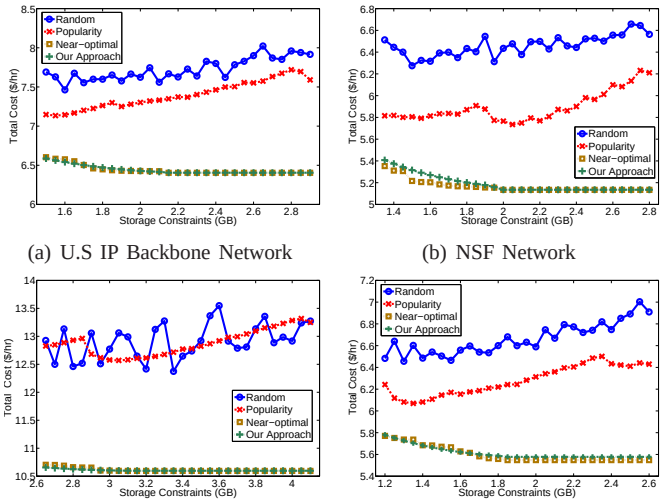
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 11. The optimal distribution of replica number follows a power-law distribution in terms of its popularity rank in all four topologies.



(a) U.S IP Backbone Network (b) NSF Network

(c) U.S 64 (d) EON

Fig. 12. Comparisons on the optimal total monetary cost generated by different algorithms, where the bandwidth resource is unconstrained. Our approach can find the near-optimal solution that is very closed to the optimal one, with much lower complexity.

replicas ($0 < n_k < min\{N, \lfloor S_{free}/B_k \rfloor\}$), where S_{free} is the remaining amount of free storage space. This process iteratively runs, until all the contents have been allocated the corresponding number of replicas, or the storage resource runs out. To improve the performance, we run this algorithm for 10 times, and pick assignment that yields the lowest cost in our experiment. Note this algorithm does not need any information on the content popularity and underlying topology.

Popularity Based Algorithm [13] allows the most popular contents have the upper bound for the number of replicas N, subject to the storage constraint, while the rest contents have only 1 replicas. Specifically, we still first assign one replica to every content, then iteratively assign $min\{N, \lfloor S_{free}/B_k \rfloor\}$ replicas to each content from the most popular one to the least popular one, until all the contents have been assigned, or the storage resource runs out. Note this approach uses content popularity information (which may not in a proper way), but it does not require the network topology information to calculate the hop distance between replica and clients.

Near-optimal Algorithm [12] iteratively calculates the total monetary cost generated by all the possible number of replicas for every single content k from the most popular one to the least popular one. When the storage space is sufficient, it picks the optimal number of replicas $n_k^\star$ with the lowest cost. Otherwise if storage space is not enough that $B_k * n_k^\star > S_{free}$, it picks the sub-optimal number of replicas $n_k^o = \lfloor S_{free}/B_k \rfloor$. Similarly, it starts by assigning one replica to each content, and ends when all the contents have been allocated proper number of replicas, or the storage resource runs out. Note this algorithm uses both content popularity and underlying topology information.

B. Total Cost Comparison

Figure 12 compares the total monetary cost generated by four different approaches based the same data set when the bandwidth resource is infinite. It can be seen, random algorithm is consistently the worst for different storage capability, in U.S IP backbone network, NSF network and EON network. Even in U.S 64 network, the result generated by random algorithm is still no better than the one yielded by popularity based algorithm. The second worst algorithm is popularity based algorithm. Compared to the results generated by near-optimal algorithm and our approach, popularity based algorithm yields the cost that is 15%-30% higher. Near-optimal method and our proposed approach perform almost the same (the difference is within 1%). They both generate the lowest total cost among all the four algorithms.

The observations can be understood as follows. Random algorithm is the simplest approach which just randomly picks the number of replicas for each content regardless of any context information (e.g., content popularity and topology). As a result, it performs the worst, and the generated results fluctuate for different rounds. Popularity based algorithm takes content popularity into consideration, but it fails to utilize it in an effective way. For example, in U.S 64 network, when the storage capacity is less than 2.85 GB, the total cost increases as the storage space grows. This is because the last assigned $\lfloor S_{free}/B_k \rfloor > n_k^\star$, that the reduced bandwidth cost cannot overcome the increased storage cost. Near-optimal algorithm finds the optimal solution based on the popularity and underlying topology information at each step. It has been proved such method can yield results very closed to the optimal ones [12]. Our approach uses the analytical relationship between the number of replicas and mean hop distance for deployed networks. It also successfully obtains the near optimal solution.

C. Computational Complexity Comparison

Table III shows the time consumed by different algorithms in different network topologies. We run those simulations implemented by C sharp codes on a desktop PC with 2.4GHz

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

TABLE III
THE TIME USED TO COMPUTE THE OPTIMAL NUMBER OF REPLICAS

	U.S IP	NSF	U.S 64	EON
Random	0.005	0.001	0.20	0.003
Popularity	0.007	0.001	0.22	0.004
Near-optimal	15.09	0.08	10600	0.33
Our Approach	0.042	0.003	3.73	0.010

dual-core CPU and 2 GB memory. Each simulation is executed for 10 times, and the average results are put on the table (except for near-optimal algorithm in U.S 64 case, which costs too much time). It can be observed that, both random and popularity-based heuristics carry light computing loads. While the complexity of our approach is slightly higher than them within a factor of 2.5-17, and the near-optimal algorithm costs significantly more time (within a factor of 80-50000).

To explain this observation, we analytically find the complexity of those four algorithms. First, both random and popularity algorithm assign replica to each content for at most one time. As a result, their complexity are both $O(M)$. Second, we have shown our solution also has a similar reasonable complexity at $O(M)$ in subsection V-A. Finally, near-optimal algorithm has to calculate all the possible combination for each content with each possible number of replicas, to determine the optimal assignment with the lowest cost. Even we use near-optimal method introduced by [12] to find the optimal solution for a given number of replicas n_k, the complexity of this algorithm can be still as high as $O(M^2N^2)$. Thus it is not practical when both M and N are large.

VIII. Conclusion and Future Work

This paper investigated the optimal content placement problem, in which content providers use cloud resources in pay-per-use model to orchestrate content delivery services. The objective is to minimize the combined storage and bandwidth cost. To balance this trade-off between them, we formulated a constrained optimization problem, which was then addressed by a two-step approach. First, for single content placement, we translated it into the k-center problem in graph theory, and obtained an approximation for the optimal mean hop distance between users and contents with respect to the number of replicas. Second, for the placement of a catalog of content, we formulated a convex optimization problem, and solved it by applying feasible direction algorithm. Finally, the analytical results from the two-step approach were verified by real traces driven simulations. Moreover, the optimality and the efficiency of our approach were also evaluated, by comparing it with three alternative heuristics.

There are a few avenues for our future. First, we want to explore the possibility of caching some content to end-users' mobile devices [37] to achieve a collaborative caching scheme in addition to the cloud CDN nodes. Second, we would like to capture the dynamics of user request pattern, and more specifically, to investigate how to optimally operate the cloud-based caching services when encountering flash crowd situations [38]. Finally, we are also interested in a QoE (Quality-of-Experience) aware strategy by adding QoE constraints into the formulation.

Appendix A
Proof of Property 1

In this section, we provide the proof of property 1.

For any given sub-tree with N_s node, there is a lower bound of its height h defined by the Moore bound (i.e., $h \geq \lceil \log_{\Delta-1} \frac{\Delta N_s - 2N_s + 2}{\Delta} \rceil$). This bound can be reached, when all the levels are full probably except the last in each sub-tree (i.e., Figure 3(b)). Thus, the aggregated hop distance within one sub-tree also reaches its lower bound.

Appendix B
Proof of Property 2

In this section, we provide the proof of property 2. Instead of directly proving the statement, we show that any other solution cannot yield the smaller aggregated hop distance than the one generated by the uniform partition.

Assuming there is such an optimal solution, where the node number difference between the largest sub-tree S_i and the smallest sub-tree S_j ($i \neq j$) is $N_i - N_j \geq 2$, and $N_i \geq \lceil N/n \rceil + 1$. According to property 1, the lower bound of the height h_k of any sub-tree S_k ($k = 1, ..., n$), is a monotonic increasing function of its node number N_k, for a given node degree Δ (i.e., $h_k(N_k + 1) \geq h_k(N_k)$). Thus, we have $h_i(N_i - 1) \geq h_j(N_j + 1)$. It indicates by re-partitioning the node at the last level of the largest sub-tree S_i to the last level of the smallest sub-tree S_j, the aggregated hop distance reduces or at least remains the same. This conflicts with the optimality of this solution, which aims to a lower bound of the aggregated hop distance.

Note, the uniform partition is only a sufficient condition to guarantee the optimality. Indeed, the optimal partition may not be unique, since the leaf nodes may have the equal shortest distance to more than one replica (e.g., in Figure 4(b), node 1 is 2 hops away from the root of either sub-tree S_1 or sub-tree S_3). As a result, they can be moved from one sub-tree to another one with the same height, while the aggregated hop distance remains the same.

Acknowledgment

This paper was supported in part by Singapore MOE Tier-1 (RG11/17), Cisco Systems, Inc., Microsoft Research Asia, SMART Innovation Grant.

References

[1] Y. Jin, Y. Wen, K. Guan, D. Kilper, and H. Xie, "Toward monetary cost effective content placement in cloud centric media network," in *IEEE ICME*, 2013, pp. 1–6.
[2] Z. Liu, Y. Ding, Y. Liu, and K. Ross, "Peer-assisted distribution of user generated content," in *IEEE P2P*, 2012, pp. 261–272.
[3] P. Verna, "A spotlight on ugc participants," http://www.emarketer.com/Article.aspx?R=1006914, 2009.
[4] F. Chen, K. Guo, J. Lin, and T. La Porta, "Intra-cloud lightning: Building cdns in the cloud," in *IEEE INFOCOM*, 2012, pp. 433–441.
[5] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking*, vol. 17, no. 5, pp. 1357–1370, 2009.
[6] E. Nygren, R. Sitaraman, and J. Sun, "The akamai network: A platform for high-performance internet applications," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 2–19, 2010.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia

[7] Z. Wang, W. Zhu, M. Chen, L. Sun, and S. Yang, "CPCDN: Content delivery powered by context and user intelligence," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 92–103, 2015.

[8] Y. Jin, Y. Wen, G. Shi, G. Wang, and A. Vasilakos, "Codaas: An experimental cloud-centric content delivery platform for user-generated contents," in *IEEE ICNC*, 2012, pp. 934–938.

[9] J. Tang, W. P. Tay, and Y. Wen, "Dynamic request redirection and elastic service scaling in cloud-centric media networks," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1434–1445, 2014.

[10] Y. Tian, R. Dey, Y. Liu, and K. W. Ross, "China's internet: Topology mapping and geolocating," in *IEEE INFOCOM*, 2012, pp. 2531–2535.

[11] O. Kariv and S. Hakimi, "An algorithm approach to network problems. i: the p-center," *SIAM Journal of Applied Mathmatics*, vol. 37, no. 3, pp. 513–538, 1979.

[12] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *IEEE INFOCOM*, vol. 3, 2001, pp. 1587–1596.

[13] J. Kangasharju, J. Roberts, and K. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, 2002.

[14] M. Karlsson and C. Karamanolis, "Choosing replica placement heuristics for wide-area systems," in *IEEE ICDCS*, 2004, pp. 350–359.

[15] X. Tang and J. Xu, "Qos-aware replica placement for content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 10, pp. 921–932, 2005.

[16] T. Wauters, J. Coppens, F. Turck, B. Dhoedt, and P. Demeester, "Replica placement in ring based content delivery networks," *Computer Communications*, vol. 29, no. 16, pp. 3313–3326, June 2006.

[17] I. Cidon, S. Kutten, and R. Soffer, "Optimal allocation of electronic content," in *IEEE INFOCOM*, 2001, pp. 205–218.

[18] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, "Distributed placement of service facilities in large-scale networks," in *IEEE INFOCOM*, 2007, pp. 2144–2152.

[19] F. Liu, S. Shen, B. Li, B. Li, and H. Jin, "Cinematic-quality VoD in a P2P storage cloud: Design, implementation and measurements," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 214–226, 2013.

[20] F. Liu, B. Li, B. Li, and H. Jin, "Peer-assisted on-demand streaming: Characterizing demands and optimizing supplies," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 351–361, 2013.

[21] Y. Wen, X. Zhu, J. Rodrigues, and C. Chen, "Cloud mobile media: Reflections and outlook," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 885–902, 2014.

[22] M. Crovella, M. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide web," in *Statistical Techniques and Applications*, 1998, pp. 3–25.

[23] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang, "The stretched exponential distribution of internet media access patterns," in *ACM PODC*, 2008, pp. 283–294.

[24] Q. Huang, K. Birman, R. van Renesse, W. Lloyd, S. Kumar, and H. C. Li, "An analysis of facebook photo caching," in *ACM SOSP*, 2013, pp. 167–181.

[25] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *IEEE INFOCOM*, 1999, pp. 126–134.

[26] B. Krishnamurthy, J. Wang, and Y. Xie, "Early measurements of a cluster-based architecture for p2p systems," in *ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 105–109.

[27] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, 2006.

[28] "Topology lists," http://www.optical-network.com/topology.php.

[29] B. Bollobas and W. F. D. I. VEGA, "The diameter of random regular graphs," *Combinatorica*, vol. 2, pp. 125–134, 1982.

[30] K. Guan, G. Atkinson, D. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *IEEE ICC workshops on Green Communications*, June 2011, pp. 1–6.

[31] K. Guan, D. Kilper, and G. Atkinson, "Evaluating the energy benefit of dynamic optical bypass for content delivery," in *INFOCOM workshops on Green Communications and Networking*, 2011, pp. 313–318.

[32] D. Topiks and A. Veinott, "On the convergence of some feasible direction algorithms for nonlinear programming," *J. SIAM Control*, vol. 5, pp. 280–294, 1967.

[33] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[34] "GoGird CDN pricing," http://www.gogrid.com/cloud-hosting/content-delivery-network.php.

[35] "Azure pricing," http://www.windowsazure.com/en-us/pricing/details/.

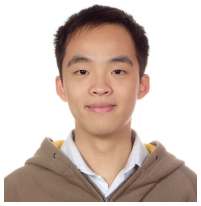
[36] "Rackspace pricing," http://www.rackspace.com/cloud/servers/pricing/.

[37] J. Dai, F. Liu, B. Li, B. Li, and J. Liu, "Collaborative caching in wireless video streaming through resource auctions," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 458–466, 2012.

[38] F. Liu, B. Li, L. Zhong, B. Li, H. Jin, and X. Liao, "Flash crowd in P2P live streaming systems: Fundamental characteristics and design implications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1227–1239, 2012.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2016.2537199, IEEE Transactions on Multimedia



Yichao Jin received the B.S and M.S degree from Nanjing University of Posts and Telecommunications (NUPT), China, in 2008 and 2011 respectively, and the Ph.D degree from the school of computing engineering, Nanyang Technological University (NTU), Singapore, in 2016. His research interests are cloud computing and content delivery networks.

Yonggang Wen is an Assistant Professor in the School of Computer Engineering at Nanyang Technological University (NTU) in Singapore. Prior to his present position, he has held R&D positions in networking companies in the USA, including Cisco and Lucent. He received his Ph.D degree in electrical engineering and computer science from Massachusetts Institute of Technology (MIT) in 2008. His research interests are cloud computing, content networking and green networks.

Kyle Guan received the Ph.D degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 2007. From 2007 to 2010, he was a Senior Research Engineer with BAE Systems, Wayne, NJ, USA. He joined Bell Labs, Nokia (formerly Alcatel-Lucent), NJ, in 2010. His current research interests include network security, energy-efficient network architectures, optical transmission systems, and network optimization.

1520-9210 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.