TOWARD MONETARY COST EFFECTIVE CONTENT PLACEMENT IN CLOUD CENTRIC MEDIA NETWORK

Yichao Jin[1], *Yonggang Wen*[1], *Kyle Guan*[2], *Daniel Kilper*[2] *and Haiyong Xie*[3]

[1]Nanyang Technological University, Singapore
[2]Bell Labs, Alcatel-Lucent, 791 Holmdel Road, Holmdel, NJ, US
[3]University of Science and Technology of China, China
{yjin3, ygwen}@ntu.edu.sg, {kyle.guan, dkilper}@alcatel-lucent.com, haiyong.xie@ustc.edu

ABSTRACT

In recent years, technical challenges are emerging on how to efficiently distribute the rapid growing user-generated contents (UGCs) with long-tailed nature. To address this issue, we have previously proposed cloud centric media network (CCMN) for cost-efficient UGCs delivery. In this paper, we further study the content placement problem in CCMN. Our objective is to minimize the monetary cost incurred by using cloud resources to orchestrate an elastic and global content delivery network (CDN) service. In particular, this objective is achieved via a two-step method. First, for a single content, we map it into a k-center problem, and find a logarithmic relationship between the mean hop distance from users to contents, and the reciprocal of replica number. Second, for multiple contents, we formulate a convex optimization with storage and bandwidth capacity constraints, which can be solved by our proposed algorithm. Finally, we verify the algorithm based on real-world traces collected from a popular video website in China. Our numerical results suggest that, the optimal number of replica for each content follows a power law in respect to its popularity, under feasible storage and bandwidth constraints, in a set of deployed backbone networks.

1. INTRODUCTION

In recent years, the ever-growing popularity of user-generated contents (UGCs) significantly challenges the existing content delivery network (CDN). It is predicted the UGCs consumers will reach 70% of the total Internet users in 2013 [1]. Those UGCs are currently replicated and distributed all over the world via CDNs. However, at present, the leading CDN providers tailor their systems and operations mainly for popular contents. Thus, they are inadequate to serve the ever-growing UGCs with long-tailed nature, [2, 3] making the economical UGCs delivery difficult.

To address this problem, we have proposed cloud centric media network (CCMN) as a novel network architecture [2, 3]. It leverages cloud computing to build an elastic CDN overlay on top of its underlying physical networks, so that the UGCs can be delivered efficiently. One major design objective of CCMN is to minimize the monetary cost when delivering UGCs. The content placement problem is one crucial design choice, to determine the monetary cost. Specifically, it involves a fundamental trade-off between storage and transport cost. On one hand, multiple UGCs replicas should be placed in different data centers, to reduce the distance from users to contents, and the associated transport cost. On the other hand, if too many copies are placed and their locations are not properly chosen, significant storage cost may be incurred with limited gains on transport cost reduction. Therefore, cost-effective content placement policies that balance the trade-off to minimize the overall monetary investment, need to be carefully studied.

There are some existing researches working on the content placement problem. T. Wauters *et al* analyzed the underlying ring based CDN topologies, to design the placement strategy [4]. The optimal content caching schemes in web proxies were studied by [5]. Based on topology analysis and optimal content placement policies, the energy efficiency of content centric networks and traditional CDN were compared by [6, 7]. However, none of them can be directly applied to solve the problem in CCMN for two reasons. First, content placement on more realistic network topologies (such as mesh topologies), should be considered for CCMN. Second, we aim to analytically characterize the optimal content placement policy, instead of using approximated solution.

In this paper, we formulate the cost effective content placement problem as a constrained optimization problem, under storage and bandwidth capacity constraints. The objective is to minimize the total monetary cost. We solve this problem in two steps. First, for a single content delivered over deployed networks, it is mapped into a k-center problem [8] to explicitly derive the optimal placement policy for a given number of replicas. Our analytical results suggest that, the mean hop distance H between the users and their closest copy in a N-nodes deployed network, is a logarithmic function of the reciprocal of replica number n, (i.e., $H \sim \log(N/n)$). This leads to a better understanding on the trade-off between storage and transport cost from the topological aspect. Second, for multiple contents, the problem is formulated as a convex optimization problem. We develop an algorithm to obtain the optimal replica number for each content. The numerical verifications on real traces suggest, under feasible storage and bandwidth constraints, the optimal replica number follows a power-law distribution to its popularity in a set of deployed networks. This offers guidelines to the real business operations in CDN environment.

The rest of the paper is organized as follows. In Section 2, we present the problem formulation. In Section 3, we characterize the k-center problem for single content in a set of deployed networks. In Section 4, we solve the convex optimization problem for multiple contents. In Section 5, we verify the solutions using real traces. In Section 6, we conclude this paper.

2. PROBLEM MODELING & FORMULATION

This section first presents the architecture of the cloud centric media network. Then we present our system assumptions for the problem

Table 1. 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_{st}	Per bit cost for storing content
c_{tr}	Per hop cost for transmitting per bit data
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

formulation. Finally, a constrained optimization problem is given. For clarity and ease of reference in the discussion, we summarize the important notations in table 1.

2.1. CCMN architecture

Figure 1 illustrates the architecture of the cloud centric media network (CCMN), which builds an elastic CDN overlay on top of its underlying physical networks, to provide efficient UGCs delivery. CCMN carves out storage, bandwidth and computation resources from data centers to provide content caching and media streaming services. One of the design objectives of CCMN is to minimize the monetary cost, while providing those services with desired QoS.

The monetary cost in CCMN is highly dependant on the content placement policy, which decides the number of replicas for each content and their locations in CCMN. Specifically, replicating content to different places can reduce the distance between users and contents, which leads to less bandwidth resources usage. However, if too many replicas are placed and their locations are not proper, it results in significant storage resource usage, and limited reduction on bandwidth resource usage. Therefore, an optimal placement policy should be in place, to minimize the total monetary cost.

2.2. System assumptions

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 provides storage and computation resources, and each link provides bandwidth resources. All those resources are limited in CCMN. This work will use four well-known deployed backbone topologies as shown in section 3.3 as examples to obtain some practical insights.

2) Content Model: We assume a catalog of M unique contents to be delivered in a time window t. For a content k, where $k = 1, ..., M$ is sorted by download times in descending order, both its size B_k and its popularity r_k are random variables. In particular, the content size follows the bounded Pareto distribution [9] with the cumulative distribution function as,

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

where B_L and B_U are the smallest and largest content size, and α is the shape parameter.

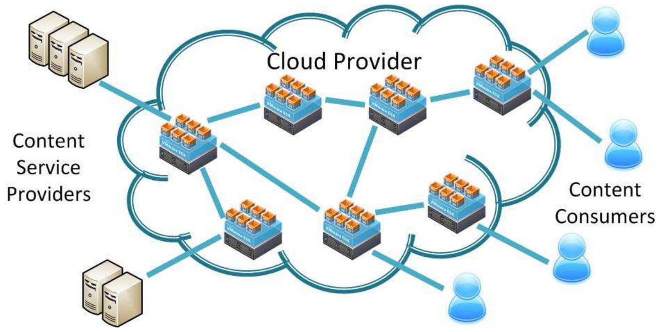


Fig. 1. Architecture of cloud centric media network

The popularity r_k in terms of download times R_k in a time window t, follows Zipf distribution [5], which is given by,

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

where $R_{tot} = \sum_{k=1}^{M} R_k$, and β is the shape parameter.

The two distributions will be verified by real traces in section 5.

3) User Request Model: Each node in CCMN is also a service entry point attached with an access network. We assume the user request is always served by the nearest node which holds a replica of the requested content. We assume an uniform traffic pattern, that is, the traffic load from each entry point is the same in a given time slot. This assumption is reasonable, since in most CDN systems including CCMN, load balancers roughly guarantee equal traffic load across different access points.

2.3. Problem formulation

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

Storage cost over a period t for content k is,

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

where B_k is the size, c_{st} is the per bit cost, n_k is the replica number.

Transport cost for content k with B_k size and R_k download times over a period t is,

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

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 content k.

We formulate the content placement problem as a constrained optimization problem, with an objective to minimize the combined storage and transport cost as,

$$\begin{aligned}
\min \quad & f(\mathbf{n}) = c_{st}\mathbf{B}^T\mathbf{n}t + c_{tr}\mathbf{B}^T\mathbf{HR} && (5)\\
\text{s.t.} \quad & g_i(\mathbf{n}) = 1 - n_i \leq 0, \;\; for \; i = 1, ..., M, && (6)\\
& g_{i+M}(\mathbf{n}) = n_i - N \leq 0, \;\; for \; i = 1, ..., M, && (7)\\
& g_{2M+1}(\mathbf{n}) = \mathbf{B}^T\mathbf{n} - S_{tot} \leq 0, && (8)\\
& g_{2M+2}(\mathbf{n}) = \mathbf{B}^T\mathbf{HR} - T_{tot} \leq 0, && (9)
\end{aligned}$$

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 constraints of (6) and (7) indicate the replicas can be placed at all the nodes at most, and a single node at least. The constraints of (8) and (9) capture the total storage and bandwidth capacity limitations respectively. Note we relax the integer constraint of n_k to achieve a lower bound solution.

We adopt a two-step method to solve this problem. First, we find optimal H for single content. Second, we use the result to solve the optimization problem for multiple contents.

3. SINGLE CONTENT PROBLEM

In this section, we cast the single content placement as a variant of k-center problem [8] as follows. Given a topology $G = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of data centers, and $\mathbf{E}$ is the set of links between them, we want to compute a subset of n vertices $\mathbf{R} \subseteq \mathbf{V}$ to minimize the mean hop distance H between any two nodes, then derive the minimal $H(n, G)$ for $n \in [1, N]$.

Due to the NP-completeness of this problem [8], the complexity of directly solving it, is prohibitive. Thus, we first solve it in generalized Moore graphs[1], then use the result to derive a lower and an upper bound of H in random regular graphs. Finally, we extend to the deployed networks, that share the same basic properties (i.e., the diameter) with random regular graph.

3.1. Generalized Moore graph

We first consider the optimal graph partition method, which is the key to characterize k-center problem [8], as described below. For a given replica number n and a d-degree, N-order, generalized Moore graph $\mathcal{G}(d, N)$, we can partition the $\mathcal{G}(d, N)$ into n sub-trees $\mathbf{S}_1, ..., \mathbf{S}_n$, where $\sum_{i=1}^{n} |\mathbf{S}_i| = N$. A replica is placed at the root of each sub-tree and serves the requests from all other nodes in this tree. The objective is to obtain an analytical form on mean hop distance H. Specifically, let the aggregated hop distance D_i from all the nodes to the root within one sub-tree S_i be the total hop distances from a replica to all its served users. By summing up D_i over $i = 1, ..., n$ and dividing it by N, we have H. Since in a generalized Moore Graph, each level of the spanning tree of each node is full (except probably the last level), we can calculate the height of each sub-tree to get H.

It has been shown [4], the uniform partition method can generate the minimized H in such problem. That is, if we partition $\mathcal{G}(d, N)$ into n separated sub-trees with the size of either $\lfloor N/n \rfloor$ or $\lfloor N/n \rfloor$+1, the overall height can be minimized. Note, such partition may not necessary be unique, since the leaf nodes of each tree may have the equal shortest distance to more than one replica. As a result, after the partitioning, there are r sub-trees, each serving r_0+1 nodes (i.e., $|S_1| = ... = |S_r| = r_0 + 1$, where r is the reminder of dividing N by n, and $r_0 = \lfloor \frac{N}{n} \rfloor$). It also indicates, there are $n - r$ replicas, each serving the requests from a group of r_0 nodes.

Following this rationale, we first compute the aggregated hop distance for each of the n sub-trees as,

$$D_d(h, o) = d \sum_{k=0}^{h-2} (k+1)(d-1)^k + ho, \tag{10}$$

where the first term is the aggregated hop distance from all the nodes excluding the last level. d nodes are in the first level, each has $d-1$ children. Thus, there are $d(d-1)^k$ nodes at $k+1$ hops away from the root, for $k = 0, ..., h-2$, where $h = \lceil \log_{d-1} \frac{dr_0 - 2r_0 + 2}{d} \rceil$ is the height of a full tree. The second term is the aggregated hop distance to the root from the last level nodes, where $o = r_0 - 1 - d\frac{(d-1)^{h-1} - 1}{d-2}$ is the node count in the last level. Note here, we compute each sub-tree has r_0 nodes.

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

$$\varphi(h, o) = h + \delta(d(d-1)^{h-1} - o), \tag{11}$$

where $\delta(x)$ is the indicator function that $\delta(x) = 1$ when $x = 0$, and $\delta(x) = 0$ otherwise.

Finally, we add a reminder γ to indicate the case that the nodes at the second furthest level are not filled up in some sub-trees. It can be seen $0 \leq \gamma \leq n-1$. As a result, we have the mean hop distance as,

$$H(n, \mathcal{G}_M(d, N)) = \frac{1}{N}(nD_d(h, o) + r\varphi(h, o) + \gamma). \tag{12}$$

This result will be used to derive both an upper and a lower bound for this problem in random regular graph.

3.2. Random regular graph

In random regular graphs, each node still has the same degree d, but not necessarily the same connectivity pattern. To extend the analysis in such graph, we first get a lower and an upper bound for H, then derive an approximation to capture the fundamental scaling of H as a function of N, d, and n.

A lower bound exists, when all the levels of each sub-tree are full except the last. This case is exactly the same as the generalized Moore graph. To further simplify the functional form of $\underline{H}(n, G(d, N))$, we make the simplification that N is dividable by n, (i.e., r_0=N/n and r=0). We also assume d is sufficiently large, such that $d \approx d$+1. Thus, we have the lower bound as,

$$\underline{H}(n, \mathcal{G}(d, N)) \approx \log_d \frac{N}{n} + c_1, \tag{13}$$

where c_1 is a constant which is independent of n and N.

To derive an upper bound, we use the upper bound for random regular graph obtained from [10], to get the upper bound height h_u of each sub-tree after partitioning as,

$$h_u \leq \log_d (2+\epsilon) r_0 \log r_0 < 2 \log_d c_2 r_0, \tag{14}$$

where c_2 is a constant as $\sqrt{2+\epsilon}$, and ϵ is a small number.

The Moore graph (a generalized Moore Graph where each node has a full d-ary routing spanning tree at all the levels) with its height at $h_u = 2 \log_d c_2 r_0$, holds an upper bound for the aggregated hop distance D of each sub-tree (note in this case, the sub-tree's order may be larger than r_0). By using the same assumptions when deriving the lower bound, we have an upper bound as,

$$\overline{H}(n, \mathcal{G}(d, N)) \approx (dc_2)^2 \log_d \frac{c_2 N}{n} + c_3, \tag{15}$$

where c_3 is a constant which is independent with n and N.

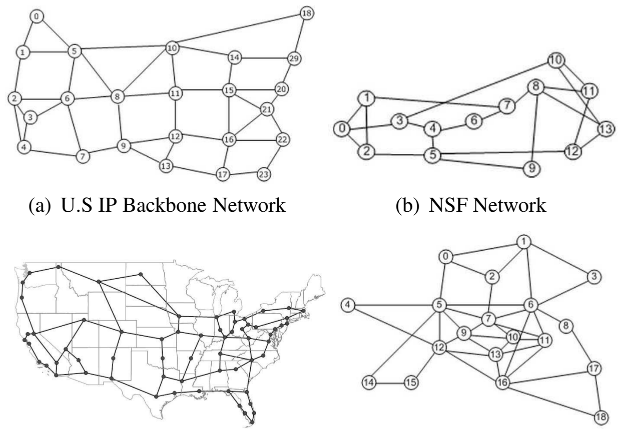
By capturing the scale in both Eq. (13) and (15), we have,

$$H(n_k, \mathcal{G}(d, N)) \approx A \log_d \frac{CN}{n_k}, \tag{16}$$

where A and C are topology-specific coefficients.

We will use this result to approximate the solution for the content placement problem in deployed networks.

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



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

(c) U.S 64 (d) EON

Fig. 2. Four deployed network topologies

3.3. Deployed networks

We consider four real deployed networks [11] as shown in figure 2, which share the same basic property (i.e., the diameter) with random regular graphs. We make this argument by comparing the diameter of those deployed networks with the analytical bounds [10] of specific random regular graphs, which have the same average node degree and node number with the deployed networks. Table 2 shows the comparisons on the diameters of those deployed networks.

Table 2. The diameters of deployed networks

	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

Using exhaustive numerical algorithms, we get the optimal placement of n replicas and the associated $H(n)$. Next we fit these results with Eq. (16). We compare the performance of our analytical results with the ones that use the power-law function [6, 7]. The accuracy of curve-fitting is evaluated by Mean Squared Error (MSE), which is denoted as

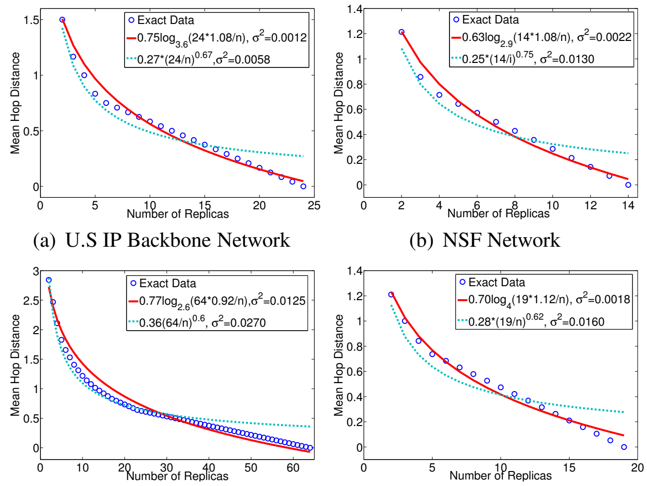
$$\tilde{\sigma}^2 = \frac{1}{N}\sum_{i=0}^{N}(\tilde{x_i} - x_i)^2, \tag{17}$$

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

Figure 3 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.

4. CONVEX OPTIMIZATION PROBLEM

By substituting Eq. (16) into the origin problem (i.e., Eq. (5)-(9)), we get a convex optimization problem. Specifically, the constraint (9) is convex, since each item, $H(n_k) = A\log_d(CN/n_k)$, for $k = 1, ..., M$ in matrix $\mathbf{H}$, is convex. Constraints (6)-(8) are all linear. And the objective function is also convex, which is the sum of a linear and a convex function.



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

(c) U.S 64 (d) EON

Fig. 3. Estimated functions for deployed networks

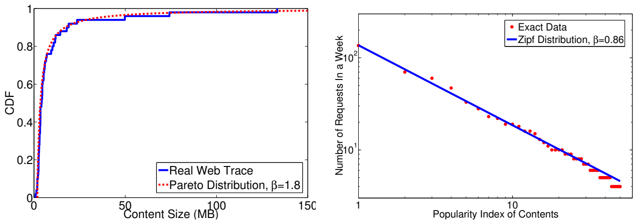
This section first uses a feasible direction algorithm to solve it in general condition. Then we present the optimal solution without capacity constraints to get some insights.

4.1. General Solution

We adopt the Topkis-Veinott's feasible direction method [12] as shown in Algorithm 1. It starts with a feasible point $\mathbf{n}_1$, then finds the reduced gradient direction $\mathbf{d}_k$ and the steps λ_k along this direction in each iteration. It is proved the new point $\mathbf{n}_{k+1} = \mathbf{n}_k + \lambda_k\mathbf{d}_k$ must be no worse than the previous one. This process repeats until the optimal solution is found.

Algorithm 1 The Feasible Direction Algorithm

Require:
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 $g_i(\mathbf{n}_1) \leq 0$
Ensure:
The optimal solution $\mathbf{n}^*$
Begin
(1) Initialize k=1;
(2) Let $(z_k, \mathbf{d}_k)$ be an optimal solution to the problem
$\min z_k$
$s.t.\ \nabla f(\mathbf{n}_k)^t\mathbf{d} - z_k \leq 0$
$\nabla g_i(\mathbf{n}_k)^t\mathbf{d} - z_k \leq -g_i(\mathbf{n}_k),\ \ for\ i = 1, ..., 2M + 2$
$-1 \leq d_j \leq 1,\ \ for\ j = 1, ..., 2M + 2$
if $z_k = 0$ **then** go to Step (4)
else go to Step (3)
(3) Let λ_k be an optimal solution to the problem
$\min f(\mathbf{n}_k + \lambda\mathbf{d}_k)$
$s.t.\ 0 \leq \lambda \leq \lambda_{max}$
$\lambda_{max} = \sup\{\lambda : g_i(\mathbf{n}_k + \lambda\mathbf{d}_k,\ for\ i = 1, ...2M + 2)\}$
Let $\mathbf{n}_{k+1} = \mathbf{n}_k + \lambda_k\mathbf{d}_k$
$k = k + 1$
go to Step (2)
(4) Finish, return $\mathbf{n}_k$ as $\mathbf{n}^*$
End



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

Fig. 4. The real-world trace from video sharing website

Our algorithm assures the convergence into the optimal point. It has been proved, the feasible direction method can guarantee the convergence to a Fritz John point [12]. Moreover, in this convex programming, the objective function and all the inequality constraints are continuously differentiable convex functions, and the Lagrangian multiplier on the gradient of the objective function cannot be zero. As a result, the global optimality of the solution is guaranteed.

4.2. Optimization without capacity constraints

The algorithm provides a way to solve the optimization problem under different constraints, but it is hard to obtain an analytical result. As a result, we consider the problem without capacity constraints, to obtain some fundamental insights.

By removing the constraint (8) and (9), it 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 $\nabla f(n_k) \leq 0$ when $n_k = N$, and $\nabla f(n_k) \geq 0$ when $n_k = 1$. By using the KKT conditions and letting $\nabla f(n_k) = 0$, we have the optimal replica number for each content as,

$$n_k^\star = \begin{cases} 1, & n_k \leq 1 \\ \frac{c_{tr} A R_k}{c_{st} \ln d}, & n_k \in (1, N) \\ N, & n_k \geq N \end{cases} . \tag{18}$$

In addition, by substituting the optimal replica number into the origin problem, we can get the optimal storage space as

$$S_{tot}^\star = \mathbf{B}^T \mathbf{n}^\star, \tag{19}$$

where $\mathbf{n}^\star = (n_1^\star, n_2^\star, ..., n_M^\star)$.

We also obtain the optimal network bandwidth as,

$$T_{tot}^\star = \mathbf{B}^T \mathbf{H}^\star \mathbf{R}, \tag{20}$$

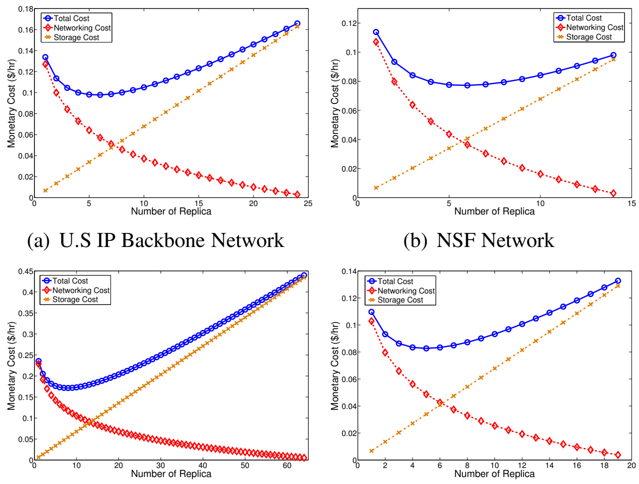
where $\mathbf{H}^\star = diag(A \log_d \frac{CN}{n_1^\star}, ..., A \log_d \frac{CN}{n_M^\star})$. The two optimal values indicate the capacity boundary that further increment on the resources can not bring with overall monetary cost reduction.

5. PERFORMANCE EVALUATION

This section uses real traces to evaluate the obtained solution.

5.1. Experimental settings

The real-world trace was captured from a leading video website in China. It contains the request history of 50 contents for a week. Figure 4 shows the distributions of content size and popularity. The Pareto distribution with its parameter $\alpha = 1.8$, and the Zipf distribution with its parameter $\beta = 0.86$, $B_U = 150$ MB and $B_L = 1.5$ MB



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

(c) U.S 64 (d) EON

Fig. 5. Monetary cost for the 10^{th} popular content

matches well with the real data trace. It verifies our assumptions on these two models as discussed in section 2.

We adopt the CDN pricing model from GoGird, a popular cloud service provider. The storage price is $c_{st} = \$0.60/GB$ per month, and bandwidth price is $c_{tr} = \$0.25/GB$.

We apply those deployed networks in section 3 to delivery the traces, and produce optimal solutions by our methods.

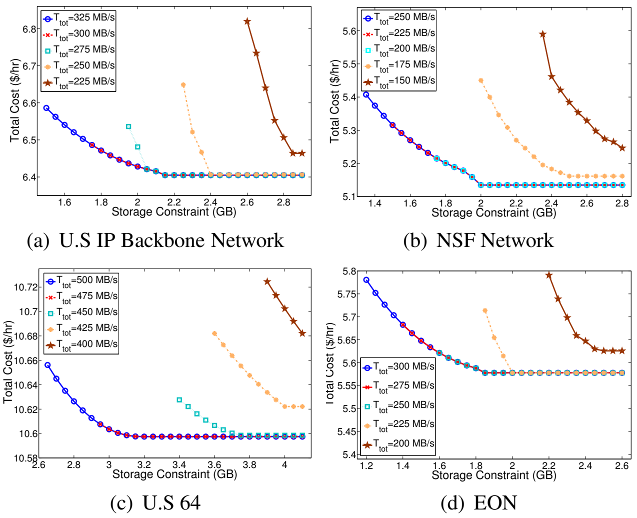
5.2. Fundamental trade-off for content placement

Figure 5 uses the 10^{th} popular content as an example to show the fundamental trade-off between storage and transport cost. In all four deployed networks, the storage cost is proportional to the replica number, and the transport cost follows a logarithmic function of the reciprocal of replica number. As the replica number grows, the storage cost increases while the transport cost decreases, and the total cost function is convex. Consequently, how to balance the trade-off and find the optimal replica number for each content under storage and bandwidth resource constraints, drives this work.

5.3. Optimal overall cost for multiple contents

Figure 6 illustrates the optimized overall monetary cost under different network topologies with various combinations of storage and network bandwidth constraints. The results in all the scenarios present almost the same characteristics.

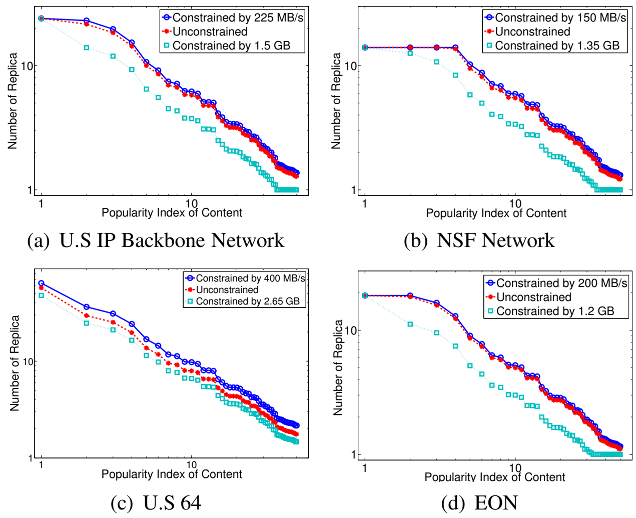
For a given T_{tot}, there are two phases as the growth of S_{tot}. In the first phase, the total monetary cost decreases as the storage capacity increases, because the storage constraint limits the optimal content placement solution. (e.g., $T_{tot} = 300$ MB/s and $S_{tot} \geq 2.2$ GB in U.S IP backbone network). In the second phase, as the storage capacity increases, the overall cost maintains at a certain level. This means the storage volume is no longer the dominating factor. Depending on the bandwidth constraint, this phase has two different implications. When the bandwidth is sufficient that $T_{tot} \geq T_{tot}^*$ (e.g., $T_{tot} \geq 267$ MB/s in U.S IP backbone network), the absolute optimal point $n^\star$ (i.e., the solution for unconstrained optimization problem as discussed in section 4.2) can be achieved, because both constraints are inactive. However, when $T_{tot} < T_{tot}^*$ (e.g., $T_{tot} < 267$ MB/s in



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

(c) U.S 64 (d) EON

Fig. 6. Overall monetary cost under different networks



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

(c) U.S 64 (d) EON

Fig. 7. The optimal distribution of content replica number

U.S IP backbone network), the bandwidth resource becomes active. The same trend can be found for a given S_{tot}.

In addition, we find, as the bandwidth capacity decreases, the minimal required storage volume grows accordingly, to have a feasible solution. For instance, in U.S IP backbone network, S_{tot} must $\geq$ 1.5 GB when $T_{tot} = 350$ MB/s, whereas S_{tot} must ≥ 1.75 GB when $T_{tot} = 300$ MB/s. This again verifies the trade-off between the storage and bandwidth resources.

5.4. Optimal replica number distribution

Figure 7 plots the optimal replica number for multiple contents. All the curves are roughly linear in a log-log graph, which suggest the optimal replica number is a power-law function to its popularity.

We also notice, when the bandwidth constraint is active, the optimal replica number is always the highest, while the storage constraint is active, it is the lowest. This still can be attributed to the trade-off between the two resources. When bandwidth resource is limited, we have to place more replicas to reduce the transport cost. When storage space is limited, less replicas are allowed to be placed. The optimal criteria of balancing this trade-off among different contents, and deciding which replicas should be adjusted, is ensured by the reduced gradient direction in our algorithm.

6. CONCLUSION

This paper investigated the content placement problem in CCMN, with the objective of minimizing the combined storage and transport cost. We formulated an optimization problem, under resource constraints, to balance the fundamental trade-off between storage and transport cost. This problem was solved in two steps. First, for a single content, we mapped it into a k-center problem, and obtained a logarithmic relationship between the mean hop distance from users to contents, and the reciprocal of replica number. Second, for multiple contents, we solved it by the feasible direction algorithm. Finally, we verified the obtained monetary cost effective strategy based on real traces. The results suggest that, the optimal replica number is a power-law function of content popularity in those deployed networks, under feasible storage and bandwidth constraints.

7. REFERENCES

[1] P. Verna, “A spotlight on ugc participants,” http://www.emarketer.com/Article.aspx?R=1006914, 2009.

[2] Y. Jin, Y. Wen, and et al., “Codaas: An experimental cloud-centric content delivery platform for user-generated contents,” in *IEEE ICNC*, 2012, pp. 934–938.

[3] Y. Wen and et al., “Designing an inter-cloud messaging protocol for content distribution as a service (codaas) over future internet,” in *ACM CFI’11*, 2011, pp. 91–93.

[4] T. Wauters and et al., “Replica placement in ring based content delivery networks,” *Computer Communications*, vol. 29, no. 16, pp. 3313–3326, June 2006.

[5] L. Breslau, P. Cao, and et al., “Web caching and zipf-like distributions: Evidence and implications,” in *IEEE INFOCOM*, 1999, pp. 126–134.

[6] K. Guan and et al., “On the energy efficiency of content delivery architectures,” in *IEEE ICC workshops on Green Communications*, June 2011, pp. 1–6.

[7] 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.

[8] 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.

[9] M. Crovella and et al., “Heavy-tailed probability distributions in the world wide web,” in *Statistical Techniques and Applications*, 1998, pp. 3–25.

[10] B. Bollobas and et al., “The diameter of random regular graphs,” *Combinatorica*, vol. 2, pp. 125–134, 1982.

[11] “Network topology lists,” http://www.optical-network.com/topology.php.

[12] 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.