How Much to Coordinate?—Optimizing In-Network Caching in Content-Centric Networks

Yanhua Li, *Member, IEEE*, Haiyong Xie, *Member, IEEE*, Yonggang Wen, *Senior Member, IEEE*, Chi-Yin Chow, *Member, IEEE*, and Zhi-Li Zhang, *Fellow, IEEE*

***Abstract*—In content-centric networks, it is challenging how to optimally provision in-network storage to cache contents, to balance the tradeoffs between the network performance and the provisioning cost. To address this problem, we first propose a holistic model for intradomain networks to characterize the network performance of routing contents to clients and the network cost incurred by globally coordinating the in-network storage capability. We then derive the optimal strategy for provisioning the storage capability that optimizes the overall network performance and cost, and analyze the performance gains via numerical evaluations on real network topologies. Our results reveal interesting phenomena; for instance, different ranges of the Zipf exponent can lead to opposite optimal strategies, and the tradeoffs between the network performance and the provisioning cost have great impacts on the stability of the optimal strategy. We also demonstrate that the optimal strategy can achieve significant gain on both the load reduction at origin servers and the improvement on the routing performance. Moreover, given an optimal coordination level ℓ^*, we design a routing-aware content placement (RACP) algorithm that runs on a centralized server. The algorithm computes and assigns contents to each CCN router to store, which can minimize the overall routing cost, e.g., transmission delay or hop counts, to deliver contents to clients. By conducting extensive simulations using a large-scale trace dataset collected from a commercial 3G network in China, our results demonstrate that our caching scheme can achieve 4% to 22% latency reduction on average over the state-of-the-art caching mechanisms.**

***Index Terms*—In-network caching, content-centric networks, coordinated caching.**

I. Introduction

INTERNET has become a ubiquitous, large-scale content distribution system. To date, not only traditional Web contents, but also an increasingly large number of video contents have been delivered through the Internet (see, e.g., [2], [3]); moreover, video content delivery over the Internet is expected to grow even more tremendously in the next few years [4], [5]. These have posed significant challenges to the Internet, e.g., how to store and disseminate the large-scale contents to support more robust, efficient, and expedited services for the users.

To address these challenges, content delivery networks (CDNs) with built-in large-scale, distributed content caching mechanisms have been adopted in the Internet. CDNs are typically deployed and operated independently by third-party CDN carriers (e.g., Akamai [6]), where CDNs are interdomain overlays spanning across multiple underlying networks; each of such underlying networks may be operated by different Internet service providers (ISPs). Some other CDNs are deployed by individual ISPs within their own networks for intradomain content dissemination (e.g., AT&T [7] and Level3 [8]). In both cases, content caching is one of the key mechanisms that make CDNs successful. However, content caching is only deployed as an overlay service rather than an inherent network capability, due to lack of the storage capability at individual routers.

Recently, in the emerging content-centric networking (CCN) architecture [9], [10], in-network storage (and caching) capacity exhibits a promising potential to significantly improve network resource utilization and energy efficiency [11]–[13], though it is still under debate whether the ubiquitous caching can improve the performance in CCN [14]. In this work, we consider the scenario, where CCN as a content-oriented future Internet architecture, content stores are available in routers; thus, content caching and dissemination become an inherent feature of routers. In such networks, users focus only on contents, rather than the physical locations from which contents can be retrieved. Moreover, the network routing and in-network storage are most likely provisioned by the same network carrier in a content-centric manner. It is worth noting that this networking model is fundamentally different from CDNs. In CDNs, the routing capability and the storage capability are completely separated and are provisioned by different entities. Content requests are fulfilled by routers which possess both the routing and the caching capabilities in CCN. On the other hand, such requests in CDNs are forwarded by routers which possess only the routing capability, and fulfilled by content servers which possess only the storage capability. Note that in many cases content servers are usually placed in networks different from where content requests are originated.

Content caching in CCN can be either coordinated or non-coordinated, similar to that in the Internet (see, e.g., [15], [16]).

Manuscript received April 6, 2015; revised July 10, 2015; accepted July 13, 2015. This work was supported in part by the U.S. National Science Foundation under Grants CNS-1117536, CRI-1305237, and CNS-1411636; by the U.S. Department of Defense Defense Threat Reduction Agency under Grant HDTRA1-14-1-0040; and by the U.S. Army Research Office Multidisciplinary University Research Initiative under Award W911NF-12-1-0385. An earlier version of this paper was presented in part at the 33rd IEEE International Conference on Distributed Computing Systems, Philadelphia, PA, USA, June 2013 [1]. The associate editor coordinating the review of this paper and approving it for publication was K. C. Almeroth.

Y. Li is with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: yli15@wpi.edu).

H. Xie is with the China Academy of Electronics and Information Technology, Beijing 100041, China (e-mail: haiyong.xie@gmail.com).

Y. Wen is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ygwen@ntu.edu.sg).

C.-Y. Chow is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: chiychow@cityu.edu.hk).

Z.-L. Zhang is with the Faculty of Computer Science and Engineering, University of Minnesota, Twin Cities, Minneapolis, MN 55455 USA (e-mail: zhzhang@cs.umn.edu).

Digital Object Identifier 10.1109/TNSM.2015.2458271

On one hand, coordinated caching mechanisms require that CCN routers store contents in a coordinated manner, which allows more contents to be efficiently cached in the "cloud" of content stores closer to the users, thus improving the overall content delivery performance. Hence, it is challenging how to trade the content *coordination cost* for network *routing performance* in CCN. The coordination cost refers to the cost incurred by provisioning storage capacity among routers, where the routing performance evaluates the performance to route the contents from one point to another. (See Section III-B for more details.) On the other hand, non-coordinated caching mechanisms store only the locally most popular contents at each CCN router, without coordination with other routers. Therefore, such mechanisms not only incur less coordination cost but also are more likely to store less distinct contents due to lack of coordination. Furthermore, studies have shown that the popularity of both Web and video contents follows the Zipf distribution [17], [18], and that user-generated contents distributed through social networks are expected to become one of the most significant contributors to Internet traffic [5]; hence, a dominant portion of contents are *not* popular. As a result, non-coordinated caching mechanisms are likely to suffer from the long-tail distribution, due to that contents are more likely fetched from distant origin servers that serve these contents.

Therefore, there exist clear trade-offs between the network performance and the coordination cost when designing in-network caching mechanisms for CCN. More specifically, coordinated caching mechanisms may trade the coordination cost for the network performance (e.g., lower average latency), while non-coordinated caching mechanisms may incur a significantly lower cost on provisioning in-network caching and may degrade the network performance due to lack of fine-grained control on where contents are cached, retrieved and routed to users.

In this paper, we focus on in-network caching mechanisms and their trade-offs in content-centric networks, where routers possess both the routing and the in-network storage capabilities. We make the first attempt to address the new challenge in CCN, namely, how to optimally provision CCN routers' storage capability and investigate the trade-offs between the network performance and the coordination cost.

More specifically, we develop a *simple holistic* model to systematically analyze the optimal strategy for provisioning the in-network storage capability. Our holistic model studies the intra-domain network as a whole, without bringing the information of each individual content and router, so it can provide a general and theoretical underpinning to understand the caching strategy design in content centric networks. We provide rigorous proofs for the existence and uniqueness of the optimal strategy, which guides us to investigate the trade-offs between the network performance and the coordination cost. We summarize our contributions as follows:

- We develop a simple holistic model to capture the network performance of routing contents to clients and the network cost incurred by globally coordinating the provision of the in-network storage capability.
- We derive the optimal strategy of provisioning the in-network storage capability to optimizes the overall network performance and cost, with mild conditions under which the optimal strategy is guaranteed to be unique.
- To further investigate how to realize the coordinated caching, i.e., placing contents to individual routers, we design a routing aware content placement (RACP) algorithm that computes the assignment of contents to each CCN router, with minimized overall routing cost.
- Through numerical analysis, we observe interesting phenomena that the stability of the optimal strategy is sensitive to key factors such as Zipf exponent of the content popularity distribution and the trade-off weights for the network performance and the coordination cost. Moreover, by evaluating the performance of our caching framework using a large-scale trace dataset collected from a commercial 3G network in China, our results demonstrate that our caching scheme can achieve 4% to 22% latency reduction on average over non-coordinated caching with least recently used (LRU) and least frequently used (LFU) eviction policies.

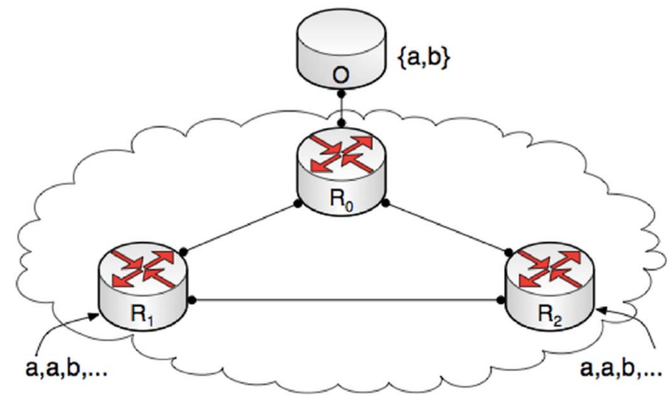


Fig. 1. A motivating example.

The rest of the paper is organized as follows. In Section II, we motivate our studies using a simple example. In Section III, we develop a holistic model that characterizes the overall network performance and cost to facilitate the analysis. In Section IV, we derive and analyze the optimal strategy for provisioning the in-network storage capability. In Section V, we introduce routing aware content placement (RACP) algorithm to assign contents among routers with minimized overall routing cost. In Section VI, we perform numerical analysis on the optimal caching strategy and conduct trace-driven evaluations on the network performances of our caching strategy. In Section VII, we present the related work. We conclude the paper with future work in Section VIII.

II. Motivation

We first motivate our study through an illustrative example shown in Fig. 1, which shows an intradomain network consisting of three routers R_0, R_1 and R_2, and one origin server O serving two content objects a and b. The network belongs to a single administrative domain (represented by the cloud). All

TABLE I
COORDINATED VS NON-COORDINATED STRATEGIES

	Non-coordinated caching	Coordinated caching
Load on origin	33%	0%
Routing hop count	≈ 0.67	0.5
Coordination cost	0	1

routers have the routing capability to forward contents to peer routers or clients. Moreover, R_1 and R_2 have storage capacity to store one single content object only, whereas R_0 does not have any available capacity for storing a or b.

We assume that there are two sets of clients (not shown in the figure) sending two request flows to their first-hop routers R_1 and R_2, respectively. The two request flows are identical, represented by a repeating sequence $\{a, a, b\}$. We assume that the performance (e.g., latency) of fetching contents from a peer router is much better than from the origin server O, since in real network, the origin may reside quite far away from the network. Then, apparently, storing contents at the routers R_1 and R_2 would reduce the overall delay and improve the network performance that clients experience. Due to the limited storage capacity, the problem is how to select contents to store at each router so as to improve the network performance and reduce the coordination cost.

We consider the following two in-network caching strategies and their trade-offs: (1) *Non-coordinated caching*: R_1 and R_2 work independently, where they both adopt the canonical caching policy based on frequency or historical usage. Assume that the content popularity distribution is consistent, and that routers R_1 and R_2 have already cumulated the information that a is requested more often than b. In this case, both R_1 and R_2 store a. (2) *Coordinated caching*: R_1 and R_2 work jointly and always prefer each other over the origin server whenever possible. In this case, R_1 and R_2 may store a and b respectively. Without loss of generality, we assume that R_1 stores a, and R_2 stores b. Then, on cache misses, a requested content will always be retrieved from either R_1 or R_2, rather than the origin server O.

We compare the coordinated and non-coordinated caching strategies when the network is in the steady state (i.e., the in-network storage at R_1 and R_2 has been steadily populated), by using three metrics: *the load on origin*, *the routing hop count*,[1] and *the storage coordination cost*. Note that the first two metrics can be used to measure the network performance, while the third metric can be used to measure the cost of provisioning the in-network storage capability. We summarize the comparison results in Table I.

First of all, the load on origin is measured by the percentage of all requests served directly by the origin server O. With the non-coordinated strategy, the requests for content a will be directly served by R_1 or R_2 (recall that both R_1 and R_2 store a in this case), while the requests for b will have to be served by the origin server. This means a total 1/3 of all requests from two flows incur the traffic load on the origin server. However, with the coordinated strategy, since both a and b are stored locally (i.e., at R_1 and R_2 respectively), all requests from the clients can be served by either R_1 or R_2. Hence, the load on origin is 0 when using the coordinated strategy, much less than that when using the non-coordinated strategy.

[1]We use hop counts as an simple example for measuring the routing performance where each hop has the same delay, and our results can be safely extended to actual delay time, by considering the link level delay time as a weight.

Secondly, the routing hop count is measured by the average number of network hops traversed when fetching contents (we focus only on the links between R_0, R_1, R_2 and O). Using the non-coordinated strategy, clients requesting for a can directly fetch a from R_1 or R_2 without going through any peer router, while requests for b have to go to the origin which is two hops away via router R_0 (i.e., the total hop count is 2). Therefore, the average routing hop count for non-coordinated strategy is $\frac{1}{3} \cdot 2 \approx 0.67$ per request. In contrast, using the coordinated strategy, only requests for b sent to R_1 and requests for a sent to R_2 trigger content fetching from their one-hop peer router, namely, R_2 and R_1, respectively. Hence, the average routing hop count is $\frac{2}{6} \cdot 1 + \frac{1}{6} \cdot 1 = 0.5$ per request.

Moreover, the coordination cost is measured by the number of messages that have to be exchanged among routers in order to reach consensus on the caching decision. Apparently, in the non-coordinated strategy,routers decide which contents to store purely based on their local information; therefore, non-coordinated caching does not incur any coordination cost. However, to implement coordinated caching, non-trivial communication costs are necessary to coordinate the caching decisions of both R_1 and R_2. In this example, to ensure that R_1 and R_2 store different contents, at least one message has to be exchanged between them.

In this example, the coordinated caching strategy leads to a lower load on origin and a lower routing hop count, while the non-coordinated strategy incurs a lower coordination cost. This suggests that there exist trade-offs between the coordination cost and the network performance. Hence, it is important to investigate how to provision the in-network caching capability and understand the trade-offs.

III. NETWORK AND PERFORMANCE-COST MODEL

In this section, we develop a holistic approach to quantifying the overall network performance of routing traffic and the cost of coordinating the in-network storage capability.

A. A Simple, Holistic Network Model

We consider a simple, holistic network model for content-centric networks. We focus on the network of a single administrative domain (e.g., an autonomous intra-domain system), where a set of routers with both the routing and storage capability serve content requests originated from end users, as shown in Fig. 2. The origin server O stores all content objects, referred to as the "*origin*"; therefore, requests for any content object can always be satisfied by O. Note that O is an abstraction of multiple origin servers (in practice, there are multiple origin servers hosting different contents).

We assume that there are n routers in the network and the number of contents N is sufficiently large. To simplify the analysis, we also assume that contents are equally large and all routers have the same storage capacity c; therefore, we are

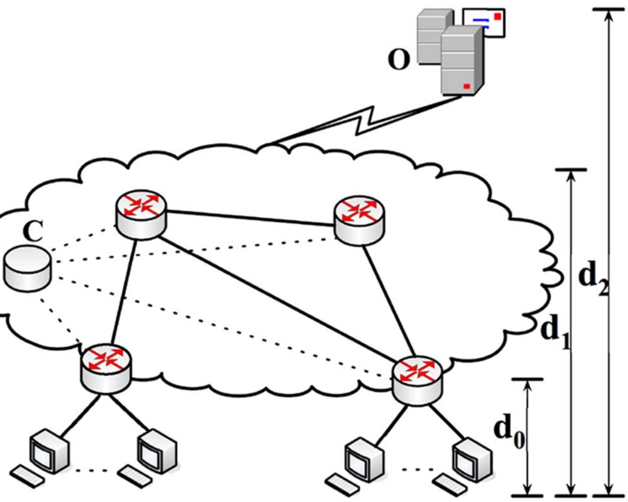


Fig. 2. A simple, holistic network model.

able to normalize the content size to one unit with respect to routers' storage capacity. Note that in the recently proposed content-centric networking architecture [9], [10], contents are segmented into smaller pieces, each of which is treated as an individually named content object, to allow flexible distribution and flow control. Content segmentation has also been adopted in many existing overlay content distribution systems, e.g., BitTorrent and eMule. These observations suggest that a homogeneous content model is reasonable in content-centric networks. Many studies have shown that the content popularity follows the Zipf distribution (see, e.g., [17], [18]). In particular, [19], [20] both show that after content segmentation, the popularity of content segments preserves the power-law distribution very well. The Zipf's law predicts that out of a population of N elements, the frequency of elements of rank i, denoted by $f(i; s, N)$, is

$$f(i; s, N) = \frac{1/i^s}{\sum_{j=1}^{N} (1/j^s)} = \frac{1/i^s}{H_{N,s}}, \quad i = 1, 2, \cdots \tag{1}$$

where s is the Zipf exponent and $H_{N,s} = \sum_{j=1}^{N} j^{-s}$ is the N-th generalized harmonic number of order s. Note that s is a key parameter of the Zipf distribution and is close to 1 but in general not equal to 1. We consider $s \in (0, 1) \cup (1, 2)$ in our analysis. In other words, $f(i; s, N)$ is the likelihood of the i-th ranked content object being requested.

The storage capability of CCN routers can be provisioned in either a non-coordinated or a coordinated manner. In the non-coordinated provision case, each router stores only the most popular contents locally so that clients can fetch them from directly connected routers. Since no coordination is necessary, routers make caching decisions independently and do not incur any cost of coordination. On the contrary, in the coordinated provision case, routers store popular contents in a coordinated and collaborative manner. Hence, more contents will be cached at peer routers in the network, and clients may experience less delay when fetching contents.

However, the coordination among routers comes at certain costs. Suppose that there exists a conceptually centralized coordinator (i.e., C) in Fig. 2. Then in order to manage the coordinated caching across the network, the coordinator has to collect the information of content stores from and disseminate necessary messages to all routers in the network. Note that the coordinator is conceptually centralized; in practice, it can be implemented in a fully distributed manner. For examples, various scalable controller designs discussed in [21], [22] (and the references therein) can be used to implement such coordinator in CCN.

In the following subsection, we will develop a performance-cost model to characterize the network performance and costs. Our model is general and unifies the coordinated and non-coordinated caching mechanisms.

B. Performance-Cost Model

We primarily consider the network routing performance and the coordination cost in CCN.

1) Routing Performance: The network routing performance refers to the performance of routing contents from one point to another in a content-centric network. The network carriers define their own routing performance metrics, for instance, the average total number of hops all traffic traverses in the network, or the average latency experienced by end users. In this paper, we use the average latency as the main routing performance metric. When there is no ambiguity, we also refer to the average latency as the routing performance. Note that our model is applicable to other metrics such as the average hop count.

As shown in Fig. 2, we denote by d_0 the average latency of serving requests from clients' closest routers (which store the requested contents locally). d_1 represents the average latency of serving a request from a peer router in the given network, namely, a directly connected router does not have the requested content but can fetch it from a peer router in the same administrative domain. Moreover, d_2 denotes the average latency of fetching contents from the origin. Note that d_1 includes two types of latency: the average latency between a client and its corresponding router (i.e., d_0), and the average latency of transferring contents from peer routers. Therefore, $d_1 > d_0$. Similarly, $d_2 > d_1$. Note also that d_0, d_1 and d_2 collectively reflect the average latency incurred by routing contents in the network. We further define $t_1 = \frac{d_1}{d_0}$ as the *first-tier latency ratio*, $t_2 = \frac{d_2}{d_1}$ as the *second-tier latency ratio*, and $\gamma = \frac{d_2 - d_1}{d_1 - d_0}$ as the *ratio of tiered latency* (or *tiered latency ratio* for short).

We consider a unified general model to formulate both coordinated and non-coordinated caching mechanisms by introducing a parameter $x \in [0, c]$, which denotes the amount of storage capacity allocated for coordinated caching mechanisms at each router. Each router stores in its local storage (i.e., the $c - x$ portion) the top ranked contents in a non-coordinated manner, and all routers collaboratively store $n \cdot x$ contents that are ranked from $c - x + 1$ to $c - x + nx$. We use $f(k; s, N)$ to characterize the probability of the k-th ranked content. Moreover, we compute the overall probability of requesting for the top k contents by

$$F(k; s, N) = \sum_{i=1}^{k} f(i; s, N) = \frac{H_{k,s}}{H_{N,s}}, \quad k = 1, 2, \cdots$$

where $H_{k,s}$ and $H_{N,s}$ are the k-th and N-th harmonic numbers of order s. Therefore, the average latency of serving a content request is

$$\begin{aligned}T(x; d_0, d_1, d_2) = &F(c-x; s, N) \cdot d_0 \\ &+ [F(c-x+xn; s, N) - F(c-x; s, N)] \cdot d_1 \\ &+ [1 - F(c-x+xn; s, N)] \cdot d_2. \qquad (2)\end{aligned}$$

The rationale is that each router uses the $c - x$ portion of its storage to store the most popular contents, and use the remaining x portion to store (distinct) contents in a coordinated manner. As a result, the total number of unique contents stored in all routers is $(c - x) + x \cdot n$ (recall that the content object size is normalized to 1).

2) Coordination Cost: The coordination cost refers to the cost incurred by coordinated provisioning of the storage capability among all participating routers. We consider three types of costs incurred to coordinate the in-network content caching decisions, including the *computational cost* of calculating the optimal storage provisioning policy for all routers and all contents, the *communication cost* of collecting statistics from and distributing optimal policies to all routers, and the *enforcement cost* of implementing the optimal policy at each individual router.

Among these three types of costs, the communication cost is a function of x. More specifically, the states of the coordinated storage at each router should be communicated to other routers in order for all routers to collectively compute the optimal policy. Such communication cost can contribute non-negligible amount of traffic. Many studies suggest that ISPs tend to define their own piece-wise linear functions to capture such cost (see, e.g., [23]); therefore, we adopt a linear function to capture the communication cost.

Note that the computational cost is dependent on the number of contents (i.e., N), coordinated contents per router (i.e., x), and many other factors such as the network topology and content popularity distribution. Recall that the number of contents is typically extremely large and is most likely to dominate other factors. Note also that the enforcement cost is independent of x. For instance, when hash-based algorithms are used for matching requests to stored contents, the complexity of operations, such as insertion, deletion, and search, is $O(1)$, which does not depend on the number of stored contents. Therefore, we consider both the computational cost and the enforcement cost as constants, and characterize the overall coordination cost in CCN by

$$W(x; w, \hat{w}) = w \cdot n \cdot x + \hat{w}, \qquad (3)$$

where $\hat{w}$ is the invariant computational and enforcement cost, w is the expected communication cost per content per router (referred to as the *unit coordination cost* for short), and $w \cdot n \cdot x$ is the overall communication cost.

IV. Problem Formulation and Analysis

In this section, we formulate the problem of how CCN routers' storage capability should be provisioned as an optimization problem, and systematically study the optimal solution, i.e., the optimal provisioning strategy for the in-network storage capability. More specifically, we provide a rigorous proof for the existence and uniqueness of the optimal strategy.

A. Problem Formulation

In practice, the network routing performance and the coordination cost may not be well aligned. Inspired by many studies where there exist multiple types of network performance and costs (see, e.g., [24], [25], [26]), we introduce a *trade-off weight parameter* $\alpha \in [0, 1]$ and formulate the overall performance/cost as a convex combination of the routing performance[2] and the coordination cost:

$$\begin{aligned}T_w(x; \alpha, w, \hat{w}, d_0, d_1, d_2) = &\alpha \cdot T(x; d_0, d_1, d_2) \\ &+ (1-\alpha) \cdot W(x; w, \hat{w}). \qquad (4)\end{aligned}$$

The goal of coordinating in-network caching is to find the optimal x^* that minimizes T_w, namely,

$$x^*(\alpha) = \arg\min_x T_w(x; \alpha, w, \hat{w}, d_0, d_1, d_2). \qquad (5)$$

We define $\ell(\alpha) = \frac{x(\alpha)}{c}$ as the *coordination level* and refer to $\ell^*(\alpha) = \frac{x^*(\alpha)}{c}$ as the *optimal strategy*, namely, the optimal percentage of coordinated storage.

In order to ease the analysis and derive meaningful results, we apply the assumption that N is sufficiently large and approximate $F(x; s, N)$ using a continuous function

$$F(x; s, N) \approx \frac{\int_1^x t^{-s} dt}{\int_1^N t^{-s} dt} = \frac{x^{1-s} - 1}{N^{1-s} - 1}, \; s \in (0, 1) \cup (1, 2). \quad (6)$$

B. Existence of Optimal Strategy

By checking the existence of the first-order derivative and the positivity of the second-order derivative of $Tw(x; \alpha, w, \hat{w}, d_0, d_1, d_2)$, both with respect to x, we can formally prove the following lemma, which suggests the existence of the optimal strategy.

Lemma 1: $T_w(x; \alpha, w, \hat{w}, d_0, d_1, d_2)$ is a convex function of x. The optimal solution to (5) exists, if the following conditions for system parameters hold:

- $0 \le x \le c$ and $c > 0$,
- The number of contents is sufficiently large ($N \gg 1$),
- the number of routers $n > 1$,
- $0 < s < 2$ and $s \neq 1$, and
- $d_0 < d_1 \le d_2$.

Proof sketch: By using the approximation in (6), $F(x; s, N)$ is differentiable. We accomplish the proof by checking the existence of the first-order derivative and the positivity of the second-order derivative of $T_w(x; \alpha, w, \hat{w}, d_0, d_1, d_2)$. □

We remark that the conditions for guaranteeing the existence of the optimal strategy are reasonable and are most likely to hold in practice. The number of contents is typically large, i.e., $N \gg 1$, and s is typically a positive number between 0 and 2 (see, e.g., [17], [18]). The number of routers n could range from a dozen to a couple of hundred in an administrative domain.

[2]Recall that we use the average latency to measure the routing performance.

Additionally, as far as the latency is concerned, the condition $d_0 < d_1 \leq d_2$ is most likely to hold in realistic networks. First, d_0 can be approximated by the latency between the end users and their first-hop routers. Its typical values are about 100 milliseconds in cellular networks (see, e.g., [27]), 10–20 milliseconds in cable access networks (see, e.g., [28]), and 30 milliseconds in ADSL access networks (see, e.g., [29]). Second, $d_1 - d_0$ can be approximated by the latency between routers in the same administrative domain, and its values typically range from a few to 20 milliseconds on average, depending on the geographical coverage of the network (e.g., [29]). Last, d_2 typically ranges from more than one hundred to a couple of hundred milliseconds with heavy-tailed distribution (see, e.g., [30]).

C. Uniqueness of Optimal Strategy

The following lemma characterizes the optimal strategy ℓ^*.

Lemma 2: The optimal strategy ℓ^* satisfies the following equation:

$$a\ell^{-s} = (1-\ell)^{-s} + b, \tag{7}$$

where $a \approx \gamma \cdot n^{1-s}$ and $b \approx \frac{1-\alpha}{\alpha} \cdot \frac{N^{1-s}-1}{1-s} \cdot \frac{nw}{d_1-d_0} c^s$, $\alpha \in [0, 1]$, $\gamma > 0$, $s \in (0, 1) \cup (1, 2)$, $n > 1$, $N > 0$, $c > 0$, $w > 0$, and $d_1 - d_0 > 0$.

Proof: By letting the first-order derivative of $T_w(x; \alpha, w, \hat{w}, d_0, d_1, d_2)$ equal to zero, we have

$$\frac{(1-s)\alpha}{N^{1-s}-1}\Big[(d_1-d_0)(c-x)^{-s} \\ -(d_2-d_1)(n-1)\left(c+(n-1)x\right)^{-s}\Big] + (1-\alpha)wn = 0,$$

$$\gamma(n-1)^{1-s}\left(\frac{1}{n-1}+\ell\right)^{-s} = (1-\ell)^{-s} + b,$$

where $b \approx \frac{1-\alpha}{\alpha} \cdot \frac{N^{1-s}-1}{1-s} \cdot \frac{nw}{d_1-d_0} c^s$. Since $n \gg 1$ holds, we have $n-1 \approx n$ and $1/(n-1) \approx 0$, which yields eq. (7). □

We next apply Lemma 2 to prove the uniqueness of ℓ^* in Theorem 1 as follows.

Theorem 1: There exists a unique solution to (7).

Proof: Given a particular trade-off weight parameter α, we define $y(\ell) = a\ell^{-s}$ and $z(\ell) = (1-\ell)^{-s} + b$, respectively. Firstly, we show that within $\ell \in (0, 1)$, both $y(\ell)$ and $z(\ell)$ are continuous and monotonically decreasing and increasing, respectively.

Continuity: Since for any $\ell \in (0, 1)$, both derivatives of $y(\ell)$ and $z(\ell)$ exist, namely, $\frac{\mathbf{d}y(\ell)}{\mathbf{d}\ell} = -a \cdot s\ell^{-s-1}$ and $\frac{\mathbf{d}z(\ell)}{\mathbf{d}\ell} = s(1-\ell)^{-s-1}$. Thus, $y(\ell)$ and $z(\ell)$ are continuous with respect to $\ell \in (0, 1)$.

Monotonicity: Given any $0 < \ell_1 < \ell_2 < 1$ and $s \in (0, 1) \cup (1, 2)$, we have

$$y(\ell_1) - y(\ell_2) = a\left(\ell_1^{-s} - \ell_2^{-s}\right) = \frac{a}{\ell_1^s \ell_2^s}\left(\ell_2^s - \ell_1^s\right) > 0,$$

thus $y(\ell)$ monotonically decreases, when increasing $\ell \in (0, 1)$. Similarly, we have $1 > 1 - \ell_1 > 1 - \ell_2 > 0$ and the following inequality holds

$$z(\ell_1) - z(\ell_2) = \frac{\ell_1^s - \ell_2^s}{(1-\ell_1)^s(1-\ell_2)^s} < 0,$$

which in turn proves that $z(\ell)$ monotonically increases, when increasing $\ell \in (0, 1)$.

Moreover, we observe that $\lim_{\ell\to 0} y(\ell) = \infty$, $\lim_{\ell\to 1} y(\ell) = a$, $\lim_{\ell\to 0} z(\ell) = 1 + b$, and $\lim_{\ell\to 1} z(\ell) = \infty$. Hence, $y(\ell)$ and $z(\ell)$ must have a unique intersection point in the range (0,1). □

D. Optimal Strategy for Routing Performance Optimization

We next focus on the optimal strategy when the routing performance is the dominant concern (i.e., $\alpha = 1$). We will derive the closed-form optimal strategy and analyze the impacts of various system parameters.

Theorem 2: When $\alpha = 1$, the unique optimal strategy for (5) is

$$\ell^* = \frac{x^*}{c} \approx \frac{1}{\gamma^{-\frac{1}{s}} n^{1-\frac{1}{s}} + 1}. \tag{8}$$

Proof: When $\alpha = 1$, we have $b = 0$. Eq. (7) becomes $a\ell^{-s} = (1-\ell)^{-s}$. The solution to this equation yields eq. (8). □

It is important to note that ℓ^* is a function of the tiered latency ratio γ (i.e., ratio between $d_2 - d_1$ and $d_1 - d_0$), rather than the absolute values of the individual latencies (such as d_0, d_1 and d_2). We refer to this property as the *latency scale free* property (or *scale free* for short). This property is particularly desirable and helpful in designing, deploying and provisioning storage capability optimally in a network.

Note that in real networks, the average latencies (e.g., d_0, d_1, and d_2) are all bounded in a few to a hundred milliseconds; thus, γ is bounded between 1 and 100 in general, while the number of routers n can scale up dramatically as the network size increases. Hence, we consider how increasing n impacts the optimal strategy ℓ^* while taking γ as a bounded constant. More specifically, when $s \in (0, 1)$, the optimal strategy ℓ^* quickly approaches 1 as n increases; in other words, all routers should dedicate all their storage capacity to coordinated caching when the number of routers is large. However, when $s \in (1, 2)$, the optimal strategy ℓ^* converges to 0 as n increases, meaning that all routers' storage capacity should be dedicated to non-coordinated caching instead.

This observation reveals that $s = 1$ is a singular point; $s \in (0, 1)$ and $s \in (1, 2)$ lead to opposite optimal strategies. We will evaluate and discuss in more details how various factors affect the optimal strategy ℓ^* in the next section.

E. Performance Gain

We now quantify the performance gain as a result of the optimal strategy ℓ^*. We consider two types of performance gain, the origin load reduction G_O from the origin server's perspective, and the routing performance improvement G_R from the network carrier's perspective.

1) Origin Load Reduction G_O: G_O is the total load reduction on the origin server, namely, the improvement on the total traffic load incurred on the origin server under the optimal strategy compared to the non-coordinated caching strategy. Based on the assumption of unit-size contents, the traffic load that the origin server sees can be directly expressed as the ratio between the number of contents served by the origin server when using the optimal strategy and when using the non-coordinated strategy.

More specifically, the traffic demand at the origin using the optimal caching strategy is $1 - F(c + (n-1)\cdot x^*; s, N)$, while the demand at the origin using the non-coordinated caching strategy is $1 - F(c; s, N)$. Therefore, the ratio of expected load at the origin with the optimal strategy over the non-coordinated strategy is

$$G_O = 1 - \frac{1 - F\left(c + (n-1)x^*\right); s, N)}{1 - F(c; s, N)} = \frac{\left(c + (n-1)x^*\right)^{1-s} - c^{1-s}}{N^{1-s} - c^{1-s}}$$

2) Routing Performance Improvement G_R: G_R is the total improvement on the routing performance, namely, the improvement on the overall routing performance under the optimal strategy versus the non-coordinated strategy.

Note that when routers are non-coordinated (i.e., $x = 0$), the routing performance in (2) is

$$T(0; d_0, d_1, d_2) = \frac{(N^{1-s} - c^{1-s})\cdot d_2 + (c^{1-s} - 1)\cdot d_0}{N^{1-s} - 1}.$$

Therefore, the overall routing performance improvement is

$$G_R = 1 - \frac{T(x^*, d_0, d_1, d_2)}{T(0, d_0, d_1, d_2)}.$$

V. Content Placement Algorithm: Challenges and Implementation

Given the holistic model we developed, an optimal coordination level, x^*, can be obtained to leverage the router storage capacity between coordinated vs non-coordinated caching strategies. As a holistic model, we look at the problem at a coarse-grained granularity, without bringing the infomation of each individual content and router. In this section, we further investigate the problem how to realize the coordinated caching, namely, how to place contents to individual routers. To solve this problem, a fine-grained granularity is considered with the content local popularity distribution being taken into account, which enables us to figure out the delay of fetching a particular content from a particular router. We design a routing aware content placement (RACP) algorithm that runs on a centralized server. The algorithm computes and assigns contents to each CCN router to store, which can minimize the overall routing cost, e.g., transmission delay or hop counts, to deliver contents to clients.

A. Content Request Model

In many large scale networks, such as YouTube and Twitter, the popularity distributions of contents, e.g., videos and Twitter topics (hashtags), change slowly over time, thus can be considered relatively stable.

Recall the network model illustrated in Fig. 2. There are in total n CCN routers in an administrative domain, serving N equal-size contents to their clients. There exists a conceptually centralized coordinator C in the administrative domain, which keeps track of the requests received by routers in the past, namely, the routers report the content requests received.[3] Let T denote a time interval, e.g., half a hour, where within each T the "local" content requests are aggregated to generate a "global" content popualirty ranking, thus within each T, we consider the "local" and "global" content popularities are invariant.[4] At a router i ($1 \le i \le n$), we denote $r_1^i, \cdots, r_N^i$ as the numbers of requests for each content received by i (within a time interval T), which represents the router i's "local" popularity ranking for all N contents. By aggregating the information, the centralized coordinator C has both the local popularity ranking of contents and a global popularity ranking of contents, i.e., $R_1 \ge \cdots \ge R_N$, where for a content $1 \le k \le N$, $R_k = \sum_{i=1}^{n} r_k^i$.

B. Routing Aware Content Placement

Since the global popularity ranking follows Zipf distribution and is likely stable over time [17], [18], we assume a fixed Zipf distribution parameter s. Hence, to minimize the overal network cost eq. (4), a unique optimal coordination level x^* can be obtained as the intersection point of the left vs right hand side in eq. (7), which infers that the top ranking $c - x^*$ contents (denoted as content set $\mathcal{C}_0 = [1, c - x^*]$) should be stored in all CCN routers (i.e., in a non-coordinated manner), and the next nx^* (globally) top ranking contents (denoted as content set $\mathcal{C}_1 = [c - x^* + 1, c + (n-1)x^*]$) are stored in a coordinated fashion, namely, each router stores x^* contents of them. The rest contents (i.e., $\mathcal{C}_2 = [c + (n-1)x^* + 1, N]$) are not stored at any CCN router.

While the holistic model focuses on how to provision the router storage capacity into coordinated vs non-coordinated cashing strategies, i.e., outputing coordination level x^*, the freedom of coordinated caching in placing those nx^* contents among CCN routers is still unaddressed. Now, we study how to place them in routers, so as to reduce the overall routing cost to deliver contents to clients.[5] Recall d_0, d_1, and d_2 ($d_0 < d_1 < d_2$) as the routing costs, e.g., transmission latency or hop counts, for a client to fetch a content from its directly connected router, a peer router, and the origin, respectively.

Let an n by N binary matrix $X = [X_{ij}]$ denote the content placement matrix, with each entry $X_{ij} = 1$ indicating that router $1 \le i \le n$ stores content $1 \le j \le N$, and $X_{ij} = 0$, otherwise. Clearly, for $j \in \mathcal{C}_0$, $X_{ij} = 1$ holds, which represents the non-coordinated caching of the most popular contents. Each request for these contents can be process with latency d_0, thus the overall latency incurred for contents $j \in \mathcal{C}_0$ is a constant, $D_0 = d_0 \sum_{j \in \mathcal{C}_0} R_j$. Similarly, for $j \in \mathcal{C}_2$, $X_{ij} = 0$ holds, which

[3]Note that since routers have limited storage resouces, the routers should report the local requests to the coordinator in a timely manner, such that no local request information is discarded before reporting to the coordinator.

[4]Note that the popularity ranking indeed change over time in CCN [31]. Studies such as [31] proposes a dynamic in-network caching algorithm by considering real-time content placement. However, frequent content migration would lead to oscillation of the content placement configuration, thus incurs high migration cost. Instead, we divide the time dimension into managable intervals, i.e., T, during which we keep the content popularity unchanged, to both account the popularity dynamics and avoid the unneccessary ocillation.

[5]Since x^* is fixed, the coordination cost $W(x^*, w, \hat{w})$ in eq. (3) is a constant, and we only consider the routing cost in the content placement problem.

corresponds to those non-popular contents that are not stored in any router. They incur a total latency as a constant too, $D_2 = d_2 \sum_{j \in \mathcal{C}_2} R_j$. For contents $j \in \mathcal{C}_1$ stored with coordinated caching, X_{ij} indicates the content placement configuration, and the total latency for these contents are

$$\begin{aligned} D_1 &= \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_1} \left(d_0 r_j^i X_{ij} + d_1 r_j^i (1 - X_{ij}) \right) \\ &= \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_1} \left(d_1 r_j^i - (d_1 - d_0) r_j^i X_{ij} \right) \\ &= \sum_{j \in \mathcal{C}_1} d_1 R_j - (d_1 - d_0) \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_1} r_j^i X_{ij}. \end{aligned}$$

Hence, the total latency for processing all content requests can be written as D in eq. (9) below.

$$\begin{aligned} D &= D_0 + D_1 + D_2 \\ &= \sum_{k \in \{0,1,2\}} d_k \sum_{j \in \mathcal{C}_k} R_j - (d_1 - d_0) \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_1} r_j^i X_{ij}. \end{aligned} \tag{9}$$

Hence, the goal of the content placement is to assign contents in $\mathcal{C}_1$ to n routers so as to minimize the total routing cost (i.e., latency) in eq. (9) to fetch contents for all requests. By skipping the constant part in the objective eq. (9), the problem is formally formulated as the following integer linear programming (ILP) problem.

Routing Aware Content Placement Problem (Primal):

$$\max_X \sum_{i=1}^{n} \sum_{j \in \mathcal{C}_1} r_j^i X_{ij} \tag{10}$$

$$\text{s.t.}: \sum_{i=1}^{n} X_{ij} \leq 1 \; j \in \mathcal{C}_1 \tag{11}$$

$$\sum_{j \in \mathcal{C}_1} X_{ij} \leq x^* \; 1 \leq i \leq n \tag{12}$$

$$X_{ij} \in \{0, 1\} \; 1 \leq i \leq n, \; j \in \mathcal{C}_1 \tag{13}$$

The constraint (11) examines that each content is stored for no more than one copy at routers in the administrative domain. The constraint (12) states that each router collaboratively store x^* contents. The primal problem (10)–(13) is an ILP problem due to the binary constraint (13), where ILP is in general NP-hard. Especially, 0-1 integer linear programming is one of Karp's 21 NP-complete problems [32], which means that in general there is no known polynomial time algorithm to locate a solution for the problem. The primal problem can be rewritten in matrix form as follows.

$$\max_{\mathbf{x}} \mathbf{r}^T \mathbf{x} \tag{14}$$

$$\text{s.t.}: A\mathbf{x} \leq \mathbf{b} \tag{15}$$

$$\mathbf{x}_k \in \{0, 1\} \; 1 \leq k \leq n|\mathcal{C}_1|, \tag{16}$$

where $\mathbf{x}$ and $\mathbf{r}$ are the vector form of X_{ij}'s and r_j^i's, with length of $n|\mathcal{C}_1|$, and A is the coefficient matrix of the constraints eqs. (11) and (12). b is the vector with the right hand sides of the constraints eqs. (11) and (12).

Optimal Solution: When the integer constraint eq. (16) is ignored, the relaxed problem becomes a linear programming problem, which can be solved in polynomial time.

$$\max_{\mathbf{x}} \quad \mathbf{r}^T \mathbf{x} \qquad \text{s.t.}: \qquad A\mathbf{x} \leq \mathbf{b}. \tag{17}$$

A total unimodular matrix (TU matrix) is a matrix for which the determinant of every square submatrix has value -1, 0, or $+1$. A totally unimodular matrix needs not to be square itself. From the definition, it follows that any totally unimodular matrix has only 0, $+1$ or -1 entries. Let A be partitioned into two disjoint sets B_1 and B_2. Then, the following four conditions together are sufficient for A to be totally unimodular (See [33]):

- Every column of A contains at most two non-zero entries;
- Every entry in A is 0, $+1$, or -1;
- If two non-zero entries in a column of A have the same sign, then the row of one is in B_1, and the other in B_2;
- If two non-zero entries in a column of A have opposite signs, then the rows of both are in B_1, or both in B_2.

In LP problem eq. (17), A can be partitioned into two components, B_1 and B_2, corresponding to the constraints in eqs. (11) and (12), and it is easy to check that all four conditions above hold. Thus, A in the LP problem eq. (17) is total unimodular. From [33], if a linear programming problem in form of eq. (17) has a total unimodular coefficient matrix A and an integer vector b, then all vertex solutions of the LP problem are integer. Therefore, the linear relaxation of the ILP problem (eqs. (10)–(13)) has all integer solutions, which in turn yields the optimal solution of our routing aware content placement problem.

VI. EVALUATIONS

In this section, we first perform numerical analysis on the optimal strategy and the performance gain using four real network topologies. Then, we conduct trace-driven evaluations on the performance of our optimal caching strategy with routing aware content placement (RACP) algorithm using a large-scale trace dataset collected from a commercial 3G network in China.

A. Numerical Analysis

Below, we first introduce the four real network datasets and parameter settings in the numerical analysis. Then, we analyze how various factors affect the optimal strategy ℓ^* and the performance gain obtained when applying ℓ^*.

1) Topologies and Parameter Setup: We use four real network topologies in our evaluations, namely, Internet2 (the Abilene Network) [34], CERNET [35], GEANT [36], and an anonymized tier-1 network carrier US-A in North America.

In particular, Abilene is a high-performance backbone network established by the Internet2 community in the late 1990s. The old Abilene network was retired and became the Internet2 network in 2007. It has 11 regional network aggregation points and the backbone connections among them are primarily OC192 or OC48. CERNET is the first nation-wide education and research network in China. It is funded by the government and managed by the Ministry of Education in China.

TABLE II
TOPOLOGIES USED IN EVALUATIONS

Topology	$\|V\|$	$\|E\|$	Region	Type
Abilene	11	28	North America	Educational
CERNET	36	112	East Asia	Educational
GEANT	23	74	Europe	Educational
US-A	20	80	North America	Commercial

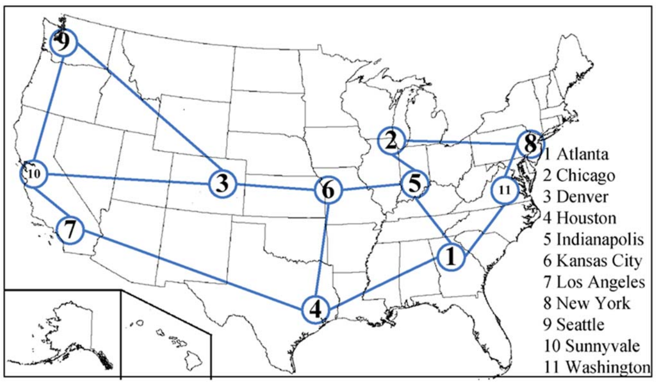


Fig. 3. The Abilene network topology.

It is constructed and operated by Tsinghua University and other leading universities in China, with 36 aggregation points and OC192 links. GEANT is a pan-European data network dedicated to the research and education community. Together with Europe's national research networks, GEANT connects 40 million users in over 8,000 institutions across 40 countries. GEANT has 23 aggregation points with links ranging from OC3 to OC192.

Each network topology, denoted by $G = (V, E)$, has the location information for each router $i \in V$ (the total number of routers $n = |V|$). We also obtain the pair-wise latency d_{ij} for every pair of routers $i, j \in V$ in each topology.

Let d_{ij} denote the average latency between two routers i and j. We estimate the unit coordination cost w by taking the maximum expected latency among routers, namely, $w = \max_{i,j \in V} d_{ij}$, since the communications among routers (or between the conceptual centralized coordinator and all routers) can be implemented in parallel, and the maximum latency plays a key role in determining the speed of converging to the optimal strategy.

Additionally, let h_{ij} denote the hop count of the shortest path between i and j. The average routing performance, measured by the average hop counts of the shortest paths among router pairs, is $(d_1 - d_0) = \frac{1}{|V|^2} \sum_{i,j \in V} h_{ij}$. Note that the routing performance can also be measured by the other metrics, e.g., the average pair-wise latency $(d_1 - d_0) = \frac{1}{|V|^2} \sum_{i,j \in V} d_{ij}$. In our evaluations, we applied both metrics and observed similar results; thus we only present the results for the routing performance measured by the hop count.

We summarize the statistics of the four networks in Table II. We show the topological structure of the Abilene network in Fig. 3, and omit the other three networks for brevity. Table III lists the topological parameters obtained from four real networks.

TABLE III
TOPOLOGICAL PARAMETERS

Topology	n	w (ms)	$d_1 - d_0$ (ms)	$d_1 - d_0$ (hops)
Abilene	11	22.3	14.3	2.4182
CERNET	36	33.3	16.2	2.8238
GEANT	23	27.8	16.0	2.6008
US-A	20	26.7	15.7	2.2842

We list in Table IV the general empirical ranges of network parameters, as well as detailed parameter settings in our evaluations. Note that we choose n, w, and $d_1 - d_0$ from the real network topologies, as listed in Table III. We obtain similar results for all four network topologies, so we only present the results for the topology of US-A for brevity. Moreover, in order to investigate how the topological parameters affect the optimal strategy, we also vary the number of routers (n) and the communication cost (w) in our evaluations.

We comment that the exact values that each parameter takes can vary over time across different networks; however, the overall trends are less likely to change.

2) Optimal Strategy ℓ^*: We first evaluate how various parameters affect the optimal strategy ℓ^*.

Trade-Off Parameter α: We first investigate the impacts of the trade-off weight parameter α to the optimal strategy ℓ^* in Fig. 4.

We observe that when α increases, namely, the routing performance is weighted more than the coordination cost, the optimal strategy ℓ^* increases monotonically from 0 to 1. This happens because as the routing performance becomes more dominant in the objective function (4), an increasingly larger portion of the storage should be dedicated to the coordinated caching in order to optimize the overall network performance and cost.

We also observe that for the same α, a higher γ leads to a higher level of coordination. Moreover, given a certain γ, when α is relatively small, ℓ^* increases slowly over α. However, when α is sufficiently large, ℓ^* grows rapidly and becomes more sensitive to changes of α.

These interesting phenomena suggest that α should be adjusted carefully when it is in the sensitive range, which is governed by other parameters, e.g., γ. For example, as shown in Fig. 4, when $\gamma = 2$, the sensitive range is around $\alpha \in [0.2, 0.4]$, and the range shifts to $[0.6, 0.8]$ when $\gamma = 10$.

Zipf Exponent s: We observe in Fig. 5 that as s increases, ℓ^* exhibits various trends over s. Note that $s = 1$ is a singular point, and is taken away from the range of s, because $s = 1$ leads to a constant routing performance $T(x; d_0, d_1, d_2) = d_2$, which is invariant to the coordination level ℓ. Zipf exponent s can usually be obtained from the trace data, and it can provide stable network performance, given the content popularity is stable over time. We make the following observations. Firstly, for $\alpha = 1$, i.e., only the routing performance is considered, the optimal strategy ℓ^* decreases from 1 to 0.35, as s changes from 0 to 2. This observation confirms our theoretical results presented in Theorem 2, namely, for $s \in (0, 1)$ (resp. $s \in (1, 2)$), ℓ^* converge to 1 (resp. 0), with an increasing n.

Secondly, when $\alpha < 1$, the optimal strategy ℓ^* converges to 0, namely, non-coordinated caching mechanisms are more

TABLE IV
SYSTEM PARAMETERS USED IN ANALYSIS

Parameters	α	γ	s	n	N	c	w (ms)	$d_1 - d_0$ (hops)
Ranges	$[0,1]$	$1 \sim 10$	$(0,1) \cup (1,2)$	$10 \sim 500$	$10^6 \sim 10^9$	$10^3 \sim 10^6$	$10 \sim 100$	$1 \sim 10$
Figure 4, 8, 12	$(0,1)$	$\{2,4,6,8,10\}$	0.8	20	10^6	10^3	26.7	2.2842
Figure 5, 9, 13	$[0.2,1]$	5	$[0.1,1) \bigcup (1,1.9]$	20	10^6	10^3	26.7	2.2842
Figure 7, 11	$[0.2,1]$	5	0.8	20	10^6	10^3	$10 \sim 100$	2.2842
Figure 6, 10	$[0.2,1]$	5	0.8	$10 \sim 500$	10^6	10^3	26.7	2.2842

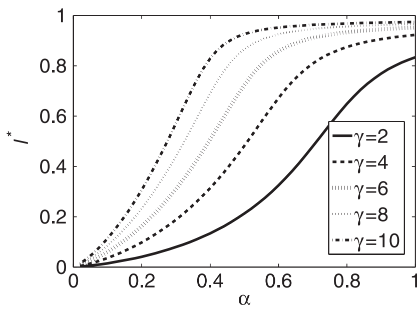


Fig. 4. Trade-off parameter.

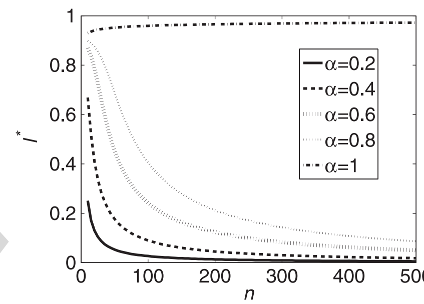


Fig. 6. Network size.

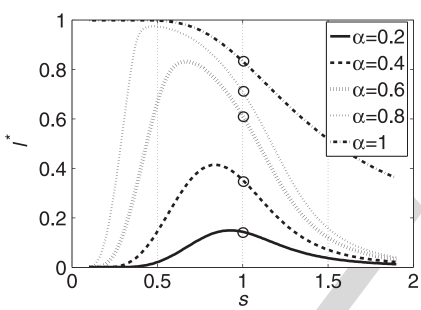


Fig. 5. Zipf exponent.

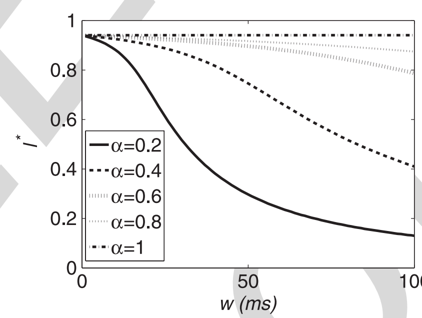


Fig. 7. Coordination cost.

preferred, when s approaches 0. This happens because caching is becoming less effective (due to less contents are popular enough to stay in routers' storage) and the coordination cost is gradually dominating the routing performance when using coordinated caching mechanisms. Moreover, for $0 \leq \alpha < 1$, there exists a maximum ℓ^* around $0.5 \sim 0.9$; while in reality, s turns out to be approximately around $0.5 \sim 0.9$ (see, e.g., [17], [18]). This illustrates that in practice, the optimal strategy ℓ^* usually indicates a higher coordination level.

Lastly, the optimal strategy ℓ^* decreases when α is decreasing; namely, the higher the weight on the coordination cost is, the lower the optimal coordination level is. This means that when the coordination cost is the major concern, non-coordinated caching mechanisms are more preferred.

Network Size n: Fig. 6 shows how ℓ^* changes with a varying size of an intradomain network (i.e., the number of routers n).

We observe that the optimal strategy ℓ^* decreases as n increases, because the more routers a network has, the higher the coordination cost is. Moreover, for a given network size, ℓ^* increases drastically as we put a higher weight on the routing performance (i.e., α increases), suggesting that a higher coordination level can help to reduce more traffic and thus to further improve the routing performance.

Unit Coordination Cost w: We observe in Fig. 7 that when the routing performance dominates in (4), i.e., $\alpha = 1$, ℓ^* is a constant close to 1, whereas for small α, e.g., $\alpha < 0.4$, ℓ^* decreases drastically as the unit coordination cost w increases. This suggests that a low coordination level can help improve the overall network performance and cost when w is large. Moreover, a larger α leads to a larger ℓ^* for the same w, which confirms the results presented in Fig. 4. This trend is also similar to the observation we made in Fig. 6.

We also numerically evaluate how the router caching capacity c (ranging from 10^3 to 10^6) and the total number of contents N (ranging from 10^6 to 10^9) affect the optimal coordination level ℓ^*. The results are similar to that of w in Fig. 7, namely, larger c and N lead to smaller ℓ^*, when $\alpha > 0$; and ℓ^* keeps unchanged over c or N, when $\alpha = 0$. For brevity, we omit these two plots.

3) Performance Gain: We next evaluate the performance gain of the optimal strategy from both the origin's and the carrier's perspectives.

Origin Load Reduction G_O: We observe in Fig. 8 that as the trade-off parameter α increases, the gain on origin load reduction increases, due to the fact that a higher ℓ^* allows routers to store more contents. Note that a higher γ leads

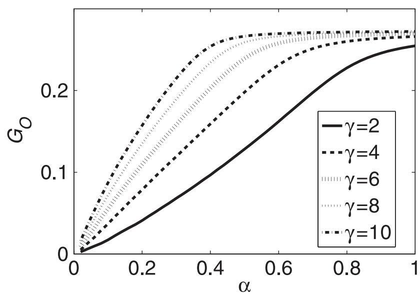


Fig. 8. α vs. G_O.

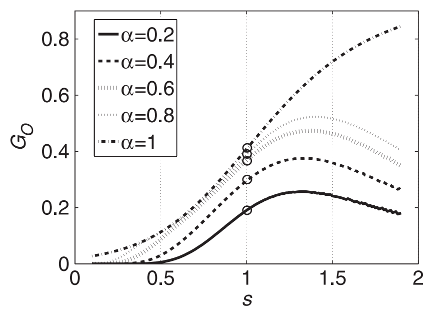


Fig. 9. s vs. G_O.

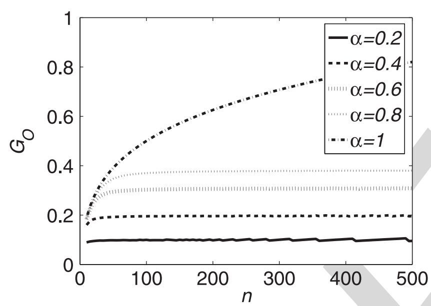


Fig. 10. n vs. G_O.

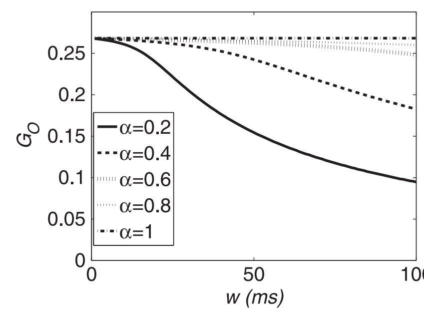


Fig. 11. w vs. G_O.

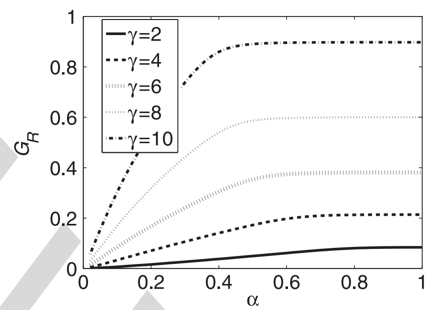


Fig. 12. α vs. G_R.

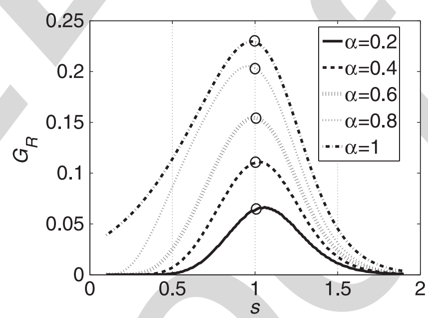


Fig. 13. s vs. G_R.

to a higher overall origin load reduction. We also observe in Fig. 9 that for a relatively smaller α, the overall origin load reduction is higher and reaches the maximum at around $s = 1.3$. Note that $s = 1$ is a singular point.

Fig. 10 illustrates how the total number of routers affects the load reduction at the origin server. When α is relatively small, the origin load reduction stays roughly constant over n, and a higher α leads to a higher origin load reduction. However, when α is approaching 1, the effect of the network size emerges; namely, the origin load reduction increases with an increasing n. This observation indicates that when the coordination cost is not dominated by the routing performance (i.e., α is small), the network size n has nearly no effect on the origin load reduction.

Moreover, Fig. 11 indicates that when α is small (e.g., $0 \leq \alpha < 0.4$), the origin load reduction decreases rapidly as the unit coordination cost increases. The reason is that when the unit coordination cost increases, the optimal coordination level ℓ^* decreases drastically, meaning that routers can store a much smaller number of distinct contents, and eventually the origin server has to serve more requests due to cache misses at routers. This phenomenon implies that for a large w, the gain on origin load reduction is low. In addition, when α is relatively large, or in other words the routing performance is weighted more, the origin load reduction becomes almost invariant with respect to a varying unit coordination cost.

Routing Performance Improvement G_R: We observe in Fig. 12 that as we increase the weight of the routing performance (i.e., α increases), the overall routing performance improvement G_R increases, and a higher γ will further raise the overall level of improvement. In particular, the routing performance improvement can be as significant as 60–90% when the trade-off parameter and the tiered latency ratio are reasonably large (e.g., $\alpha \geq 0.5$ and $\gamma \geq 8$).

Additionally, Fig. 13 shows that when s is further away from 1, i.e., closer to 0 or 2, the routing performance improvement is smaller; whereas for s close to 1 ($s = 1$ is a singular point), the routing performance improvement is large (reaching the maximum at around $s = 1$), which suggests that for those scenarios with the Zipf exponent s closer to 1, the optimal strategy is more efficient since more significant improvement on the routing performance can be achieved. When varying the parameters w and n, we observe results similar to Figs. 10 and 11, therefore we omit them here for brevity.

TABLE V
STATISTICS OF THE DATASET

Parameter	N	n	d_0	$d_2 - d_1$	s
Value	729,527	14	830.2 ms	141.3ms	0.6966

TABLE VI
GLOBAL CONTENT POPULARITY DISTRIBUTION

Index	1	2	3	4	5
# Requests	14,959	12,931	12,062	11,152	11,012
6	7	8	9	10	Total
9,827	9,263	9,102	8,776	6,939	1,748,276

B. Trace-Driven Performance Evaluation

We analyze the performance of our optimal caching strategy with routing aware content placement (RACP) algorithm in this subsection using a large-scale trace dataset collected from a commercial 3G network in China, with records of 3G users requesting and downloading various Internet contents. By applying both our content caching strategy (using RACP) and the non-coordinated caching strategy, our extensive evaluation results demonstrate that our strategy with RACP can achieve 4% to 22% latency reduction on average. Below, we first provide details about the trace data we used in the evaluations. Then, we present the evaluation settings and comparison results obtained.

1) Trace Dataset From a 3G Network: The dataset we used was collected from a Gateway GPRS Support Node (GGSN) in a 3G network in 2010 for a week (7 days) at a small town in China. The mobile phone users' private information was all anonymized by the carrier. A GGSN serves as a gateway between the cellular network and the Internet, which covers a set of base stations. All 3G requests from mobile users who were connected to those base stations were aggregated to the GGSN, through which the requests were sent to the hosts in the Internet and the contents were returned to the mobile users. A GGSN usually covers a large region with a few geo-distributed servers (i.e., Serving GPRS Support Nodes, SGSN), each of which governs all aggregated requests from users in a subregion. Our dataset indicates that there were in total 14 servers (distinguished by IP addresses) processing the 3G content queries from mobile users. Each 3G request from a mobile user looks for a content from the Internet, where the content could be a URL (for HTTP web contents) or an IP address with the directory and the content name. After finding the corresponding contents in the Internet, the content is returned through GGSN back to the mobile user.

In our simulation, we consider those servers (i.e., SGSNs) as CCN routers with caching capacity, and the Internet as the content origin, which stores all contents mobile users may request. Assuming that each server has a caching capacity ranging from $c = 100$ to $c = 1000$. We store the contents among the servers, using our caching scheme with RACP algorithm, and non-coordinated caching with least recently used (LRU) and least frequently used (LFU) [17] as the eviction policies, respectively.

This trace contains 1,748,276 content requests at all servers for in total 729,527 contents. The basic statistics of the dataset is summarized in Table V. Note that d_0 is evaluated as the average latency between the client and the first server (i.e., SGSN) it connected to. $d_2 - d_1$ value was evaluated by averaging the latency between the servers on the boundary of the network, from which the request was forwarded out to the source of the content in the Internet.

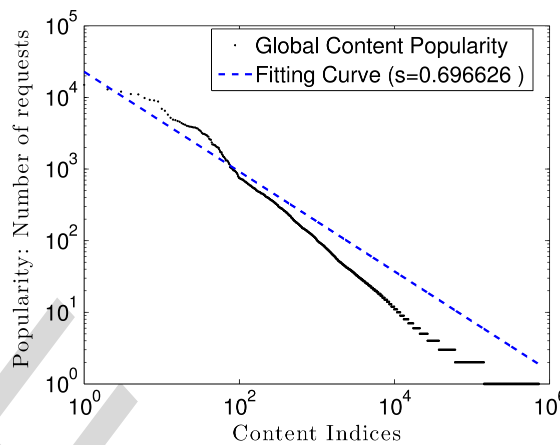


Fig. 14. Content popularity distribution.

We index the contents based on their popularity, namely, the numbers of total requests received (see Table VI). AQ1 Fig. 14 presents the overall popularity distribution of the contents. It exhibits power law distribution, with $s = 0.696626$. The local popularity distribution of requests at each server exhibit power law phenomenon as well, where the local content ranking lists at servers differ significantly from each other. We observe that some contents recevied high volume of requests at almost all servers, whereas some other contents received requests from only one server.

2) Performance Evaluation Results: In the simulation, we evaluate our content caching strategy with RACP algorithm on the trace data with such diverse request patterns of the contents, and compare the total routing performance, i.e., the average latency (eq. (9)) with the non-coordinated caching using least recently used (LRU) and least frequently used (LFU) as eviction policies. In the evaluations, we set the updating time interval to be half an hour for all caching schemes, namely, every half a hour, our caching scheme recomputes the optimal coordination level and corresponding content placement configuration, while non-coordinated caching schemes update routers caches using LRU and LFU eviction policies, respectively.

We change three parameters as follows, including server capacity c, tiered latency ratio $\gamma = \frac{d_2 - d_1}{d_1 - d_0}$, and the total number of servers n in the simulations to examine their impacts on the average latency of processing requests.

- Server (router) capacity c: Ranging from 100–1000 contents;
- Tiered latency ratio γ: Ranging from 1–10;
- Number of servers (routers) n: 2–14. To preserve the power law distribution of requests, we gradually reduced the number of servers by merging them in pair, i.e., treating two servers as a single super server.

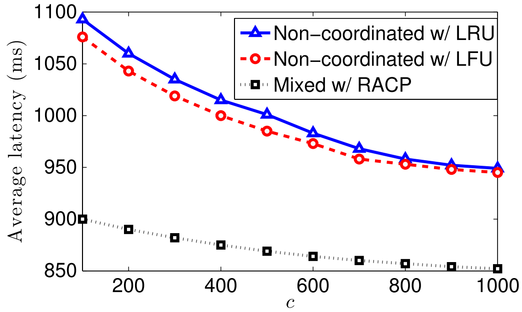


Fig. 15. Average latency over c.

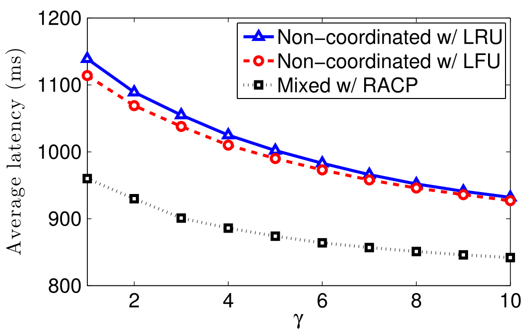


Fig. 16. Average latency over γ.

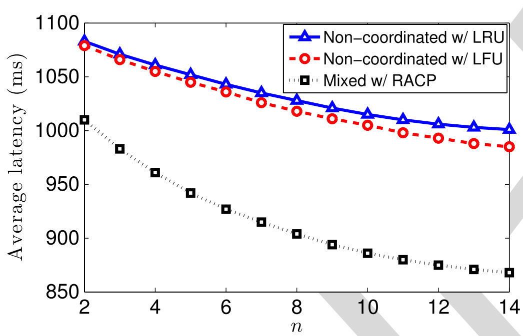


Fig. 17. Average latency over n.

The evaluation results are presented in Figs. 15–17. We compare our mixed caching scheme with RACP algorithm to non-coordinated caching schemes with LRU and LFU eviction policies, and measure the performance in terms of the average latency. Denote d_{mix} (resp. d_{LRU} and d_{LFU}) as the average latency when using our caching scheme with RACP algorithm (resp. non-coordinated caching schemes with LRU and LFU). The average latency reduction is computed as $R_{latency} = 100\% \times (d_{LRU} - d_{mix})/d_{LRU}$ for LRU ($R_{latency} = 100\% \times (d_{LFU} - d_{mix})/d_{LFU}$ for LFU), which indicates the reduced average latency in percentage when using our caching scheme.

Fig. 15 shows the effect of the router capacity c, with fixed total number of routers as 14 and the tiered latency ratio as $\gamma = 5$. As c increases from 100–1000, the latency decreases drastically for all caching schemes. The latency reduction ratio of our scheme (denoted as "Mixed w/ RACP" in the figure) over non-coordinated caching is from between 12% and 22%, which decreases as c increases, which happens because for a smaller router capacity c, i.e., very limited space to cache contents, there is more room for our caching scheme to improve the average latency. Moreover, the two non-coordinated caching schemes with LRU and LFU have similar latency, where LFU performs a bit better with lower latency.

In Fig. 16, we evaluate how the tiered latency ratio γ affects the average latency of our caching scheme and non-coordinated caching with LRU and LFU. Since $d_2 - d_1 = 141.3$ ms and $d_0 = 830.2$ ms are fixed (from Table V), changing $\gamma = \frac{d_2-d_1}{d_1-d_0}$ leads to the change on $d_1 - d_0$. The results in Fig. 16 show that as the tiered latency ratio γ increases, the average latency for all caching schemes decreases. This happens because larger γ corresponds to smaller $d_1 - d_0$, thus smaller d_1 and d_2. Moreover, the latency reduction ratio of our caching scheme over non-coordinated caching decreases (ranging from 11%–19%) as γ increases, because when γ is larger, d_1 and d_2 are closer to d_0, thus leaves smaller room for our scheme to reduce the average latency.

In Fig. 17, we evaluate how the network size n, i.e., number of routers, affects the average latency reduction. In the 3G network trace file, there are in total $n = 14$ routers. We reduce the number of routers by merging and aggregating the content query logs from two routers to one router. This way, we can still preserve the power law distribution of local content popularity. Fig. 17 shows the average latency decreases as the number of routers increases, which is because more routers lead to more caching capacity. Moreover, when the network size is small, the average latency reduction is small, e.g., $R_{latency} = 8\%$ for $n = 2$. On the other hand, when the network size is large, the average latency reduction gain is higher, e.g., around 18% when $n = 14$. These happen because when the network size is smaller, the local popularity distributions are less diverse, thus the routing performances are more similar between our caching scheme and non-coordinated caching schemes.

Numerical Analysis vs Trace-Driven Evaluations: Comparing the results in Sections VI-A.1 and B, our trace-driven evaluations demonstrate consistent results to that of numerical analysis, namely, a maximum of 25% routing performance improvement.

VII. Related Work

In this section, we discuss two topics that are closely related to our work and highlight the differences from them, including (1) content caching and placement, and (2) content-centric networking.

Content Caching and Placement: Content caching has been a key component of Internet-based services for many years (see, e.g., Akamai [6]), and there have been many studies in the literature on content caching techniques (see, e.g., [15]). The content placement problems with applications in CDNs, CCNs, P2P networks, wireless networks, and web server replicas aim to identify the right number of replicas and locations for the contents to achieve design objective, such as minimizing the delay, bandwidth, energy consumed, etc.

In particular, coordinated (or collaborative) content caching has been studied extensively. Researchers have investigated the effectiveness of collaborative caching (see, e.g., [15]) and proposed numerous collaborative caching schemes for both general networks and networks with specific structures, including general Internet-based content distribution (see, e.g., [37]), delivering special types of contents (e.g., [16]), and etc. Moreover, a common formulation employed in these studies is integer linear programming (ILP), which is in general NP-hard. LP relaxation techniques are widely used as a practical method to approximate the optimal solution [11], [31], [38], [39].

Our work differs from these studies in two ways. First, our network model for content-centric networks is novel, where we formulate the problem by focusing on the overall network performance and cost from the network carriers' perspectives. Thus, our model considers the routing performance and the coordination cost, and investigates the trade-offs between them. Secondly, by decoupling the coordinated vs non-coordinated caching strategies, the content placement is simplified and only performed for coordinated caching part. Thus, a nice property, total unimodularity holds, which allows polynomial time algorithm to find the provably optimal solution.

Content-Centric Networking: There exists a line of recent work on emerging Content-Centric Networking [9] and Named Data Networking (NDN) [10], where content storage becomes an inherent capability of network routers. CCN and NDN are closely related, with the latter focusing more on fundamental research. CCN/NDN has become one of the representative alternatives for the future Internet architecture. Both CCN and NDN have attracted much attention. There has been an increasingly large body of literature on CCN and NDN, to name a few, naming and name resolution (e.g., [40]), flow and traffic control (e.g., [41]), caching (e.g., [42], [43]), and etc. In particular, in [42], Xie *et al.* proposed a traffic-engineering-guided content placement and caching algorithm for CCN; and in [44], Sourlas *et al.* proposed content placement and caching algorithms to minimize overall traffic cost of content delivery, specifically designed for CCN. However, none of the existing work addresses the optimal strategy of coordinated content caching and investigates the trade-offs between the routing performance and the coordination cost in the context of CCN/NDN. To the best of our knowledge, our work is the first attempt to formally investigate and providing insights in addressing these issues.

VIII. Conclusion

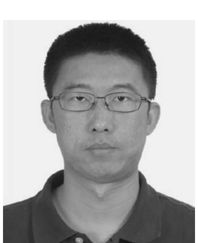
In content-centric networks, routers possess both the routing and the in-network storage capability, which raises new challenges in network provisioning, namely, how to optimally provision individual routers' storage capability for content caching, so as to optimize the overall network performance and provisioning cost.

In this paper, we developed a holistic model to quantify the overall network performance of routing contents to clients and the overall provisioning cost incurred by coordinating the in-network storage capability. Based on this model, we derived the optimal strategy for optimizing the overall network performance and cost, and evaluated the optimal strategy using real network topologies. Evaluation results demonstrated significant gains on both the load reduction at origin and the improvement on routing performance. To further investigate how to realize the coordinated caching, namely, placing contents to individual routers, we design Routing-Aware Content Placement (RACP) algorithm that computes and assigns contents to CCN routers to store, with minimized overall routing cost. By evaluating the performances of our caching scheme with RACP algorithm using a large scale trace dataset collected from a commercial 3G network in China, our results demonstrate that our caching scheme can achieve 4% to 22% latency reduction on average over non-coordinated caching.

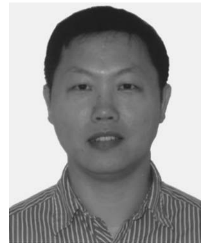
References

[1] Y. Li, H. Xie, Y. Wen, and Z.-L. Zhang, "Coordinating in-network caching in content-centric networks: Model and analysis," in *Proc. IEEE ICDCS*, 2013, pp. 1–11.
[2] Netflix. [Online]. Available: http://www.netflix.com
[3] J. Zhou, Y. Li, V. K. Adhikari, and Z.-L. Zhang, "Counting youtube videos via random prefix sampling," in *Proc. ACM IMC*, 2011, pp. 371–380.
[4] M.-F. Moens, J. Li, and T.-S. Chua, *Mining User Generated Content*. Boca Raton, FL, USA: CRC Press, 2014.
[5] "The Zettabyte era-trends and analysis," Cisco, San Jose, CA, USA, Cisco White Paper, 2013.
[6] Akamai Technologies. [Online]. Available: http://www.akamai.com
[7] "AT&T content acceleration service," AT&T, Wilmington, DE, USA, Product Brief, 2011.
[8] Level3 Communications. [Online]. Available: http://www.level3.com
[9] V. Jacobson *et al.*, "Networking named content," in *Proc. ACM CoNEXT*, 2009, pp. 1–12.
[10] L. Zhang *et al.*, "Named Data Networking (NDN) project," Palo Alto Research Center (PARC), Palo Alto, CA, USA, Tech. Rep. NDN-0001, 2010.
[11] N. Choi, K. Guan, D. C. Kilper, and G. Atkinson, "In-network caching effect on optimal energy consumption in content-centric networking," in *Proc. ICC. IEEE*, 2012, pp. 2889–2894.
[12] U. Lee, I. Rimac, D. Kilper, and V. Hilt, "Toward energy-efficient content dissemination," *IEEE Netw.*, vol. 25, no. 2, pp. 14–19, Mar./Apr. 2011.
[13] K. Guan, G. Atkinson, D. C. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *Proc. IEEE ICC Workshop*, 2011, pp. 1–6.
[14] A. Ghodsi *et al.*, "Information-centric networking: Seeing the forest for the trees," in *Proc. 10th ACM Workshop Hot Topics Netw.*, 2011, pp. 1–6.
[15] A. Wolman *et al.*, "On the scale and performance of cooperative web proxy caching," *SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, pp. 16–31, Dec. 1999.
[16] J. Ni and D. Tsang, "Large-scale cooperative caching and application-level multicast in multimedia content delivery networks," *IEEE Commun. Mag.*, vol. 43, no. 5, pp. 98–105, May 2005.
[17] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, 1999, pp. 126–134. AQ2
[18] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Proc. IEEE IWQoS*, 2008, pp. 229–238.
[19] S. Tang, Y. Lu, J. M. Hernández, F. Kuipers, and P. Van Mieghem, "Topology dynamics in a P2PTV network," in *Proc. NETWORKING*, 2009, pp. 326–337.
[20] H.-C. Hsiao and H. Liao, "A novel peer-to-peer overlay for live media broadcasting," in *Proc. IEEE ICPPW*, 2007, pp. 54–54.
[21] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, Feb. 2013.
[22] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," in *Proc. HotSDN*, 2013, pp. 7–12.
[23] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, pp. 519–528.
[24] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network," in *Proc. ACM SIGMETRIC*, 2009, pp. 239–250.
[25] Y. Li, Z.-L. Zhang, and D. Boley, "The routing continuum from shortest-path to all-path: A unifying theory," in *Proc. IEEE ICDCS*, 2011, pp. 847–856.

[26] Y. Li, Z.-L. Zhang, and D. Boley, "From shortest-path to all-path: The routing continuum theory and its applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1745–1755, Jul. 2014.
[27] C. Serrano *et al.*, "Latency in broad-band mobile networks," in *Proc. IEEE VTC—Spring*, 2009, pp. 1–7.
[28] N. Barakat and T. Darcie, "Delay characterization of cable access networks," *IEEE Commun. Lett.*, vol. 11, no. 4, pp. 357–359, Apr. 2007.
[29] K. C. Kang, K.-D. Nam, D. J. Jeon, and S. Y. Kim, "Delay characteristics of high-speed Internet access network," in *Proc. APNOMS*, 2003, pp. 1–12.
[30] B. Huffaker, M. Fomenkov, D. J. Plummer, D. Moore, and K. Claffy, "Distance metrics in the Internet," in *Proc. IEEE Int. Telecommun. Symp.*, 2002, pp. 1–6.
[31] J. Llorca *et al.*, "Dynamic in-network caching for energy efficient content delivery," in *Proc. IEEE INFOCOM*, 2013, pp. 245–249.
[32] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York, NY, USA: Plenum, 1972.
[33] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY, USA: Dover, 1998, sec. 13.2.
[34] Abilene, Internet2 IP IGP Metrics. [Online]. Available: http://noc.net.Internet2.edu/i2network/maps-documentation/maps.html
[35] "CERNET Topology," China Educ. Res. Netw. (CERNET), China. [Online]. Available: http://www.edu.cn/20060111/3170194.shtml
[36] GEANT, "GEANT Topology." [Online]. Available: http://archive.geant.net/upload/pdf/Topology_Oct_2004.pdf
[37] D. Wessels and K. Claffy, "Internet cache protocol (ICP), version 2," Internet Eng. Task Force (IETF), Fremont, CA, USA, RFC 2186, 1997.
[38] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless p2p networks: Design, implementation, and evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 2, pp. 229–241, Feb. 2010.
[39] W. Wu, J. Cao, and X. Fan, "Design and performance evaluation of overhearing-aided data caching in wireless Ad Hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 450–463, Mar. 2013.
[40] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in content-oriented architectures," in *Proc. ACM ICN Workshop*, 2011, pp. 1–6.
[41] S. Oueslati, J. Roberts, and N. Sbihi, "Flow-aware traffic control for a content-centric network," in *Proc. IEEE INFOCOM*, 2012, pp. 2417–2425.
[42] H. Xie, G. Shi, and P. Wang, "TECC: Towards collaborative in-network caching guided by traffic engineering," in *Proc. IEEE INFOCOM*, 2012, pp. 2546–2550.
[43] V. Sourlas, L. Gkatzikis, and L. Tassiulas, "On-line storage management with distributed decision making for content-centric networks," in *Proc. EURO-NGI*, 2011, pp. 1–8.
[44] V. Sourlas, L. Gkatzikis, P. Flegkas, and L. Tassiulas, "Distributed cache management in information-centric networks," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 3, pp. 286–299, Sep. 2013.



Yanhua Li (S'09–M'13) received the Ph.D. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the Ph.D. degree in computer science from the University of Minnesota at Twin Cities, Minneapolis, MN, USA, in 2013. From 2011 to 2013, he worked as an Intern with Bell Labs, NJ, Microsoft Research Asia, and Huawei Research Labs of America. From August 2013 to December 2014, he worked as a Researcher with Huawei Noah's Ark Lab, Hong Kong. He is currently an Assistant Professor with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include network performances and data analytics in many contexts, such as communication protocols design, and urban network data analytics and management. He served on Technical Program Committees of INFOCOM 2015-2016, ICDCS 2014-2015, and SIGSPATIAL GIS 2015. He is the Cochair of SIMPLEX 2015.



Haiyong Xie (S'05–M'09) received the B.S. degree from the University of Science and Technology China, Hefei, China, in 1997, and the M.S. and Ph.D. degrees in computer science from Yale University, New Haven, CT, USA, in 2005 and 2008, respectively. He is currently the Executive Director for the Cyberspace and Data Science Laboratory, China Academy of Electronics and Information Technology, Beijing, China. His research interest includes content-centric networks, software-defined networks, future Internet architecture, and traffic engineering.

Yonggang Wen (S'99–M'08–SM'14) received the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He has worked at Cisco to lead product development in content delivery network, which had a revenue impact of $3 billion globally. He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore.

Chi-Yin Chow (S'03–M'04) received the Ph.D. degree from the University of Minnesota at Twin Cities, Minneapolis, MN, USA, in 2010. In the summer of 2008, he was an Intern with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include geographic information systems, location-based services, wireless sensor networks, and mobile computing. He served as the Cochair of ACM SIGSPATIAL MobiGIS 2012 and 2013.

Zhi-Li Zhang (M'97–SM'11–F'12) received the B.S. degree in computer science from Nanjing University, Nanjing, China, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts Amherst, Amherst, MA, USA, in 1992 and 1997, respectively.

From 1987 to 1990, he conducted research with the Department of Computer Science, Aarhus University, Aarhus, Denmark, under a fellowship from the Chinese National Committee for Education. Since 1997, he has been with the Faculty of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA, where he is currently a Professor. He has held visiting positions with Sprint Advanced Technology Labs, Burlingame, CA, USA; IBM T. J. Watson Research Center, Yorktown Heights, NY, USA; Fujitsu Labs of America, Sunnyvale, CA, USA; Microsoft Research China, Beijing, China; and INRIA, Sophia-Antipolis, France.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = Table VI was uncited in the body. Therefore, its citation was inserted here. Please check if appropriate, otherwise, provide necessary corrections.

AQ2 = Note that reference [17] and [37] are the same. Therefore, reference [37] was deleted from the list. Citations were renumbered accordingly. Please check.

AQ3 = Provided URL in Ref. [36] was not found. Please check.

END OF ALL QUERIES