Computer Communications xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Network delay guarantee for differentiated services in content-centric networking☆

Q1 Weibo Chu [a,*], Lifang Wang [a], Haiyong Xie [b], Zhi-Li Zhang [c], Zejun Jiang [a]

[a] Northwestern Polytechnical University, Xi'an 710049, China
[b] University of Science and Technology of China, Suzhou 215123, China
[c] University of Minnesota, Minneapolis, MN 55414, USA

A R T I C L E I N F O

Article history:
Received 8 October 2014
Revised 8 July 2015
Accepted 7 September 2015
Available online xxx

Keywords:
Content-centric networks
In-network caching
Content placement
Quality of Service
Delay guarantee

A B S T R A C T

The newly adopted built-in caching mechanism guarantees efficient content delivery for content-centric networking (CCN) as compared to the existing IP-based networks such as the Internet. However, it is a challenge at the same time for CCN to meet QoS requirements due to content caching. In this paper, we investigate the problem of providing network delay guaranteed services in CCN. More specifically, we study the problem of meeting network delay requirements for differentiated services (content providers) in CCN while at the same time optimizing the overall content delivery performance.

To support delay guarantee, we first present a simple and holistic network model which characterizes network delays of routing content to clients at different locations. By aligning network locations with content popularity, we ensure that each content provider has an optimized network delay of routing content to clients. We then derive analytical network delays for content providers by incorporating their content distribution models into the proposed holistic network model, and further formulate the delay guarantee task as a nonlinear integer programming (NIP) problem under the given network resources and traffic access patterns. We evaluate our mechanism and investigate the optimized network performance using different real/synthetic network topologies. With numerical studies, we analyze the process of competing for the network resources by different content providers, and investigate how various factors (e.g., content popularity, traffic volume, router storage capacity) affect this competition process. Our models and results presented in this paper provide guidance in designing resource provisioning and QoS mechanisms for CCN.

1. Introduction

Driven by the huge volume of content (e.g., video, audio, images), the usage of the Internet is increasingly focused around content delivery. Today users tend not to care where and how to obtain the content, but are more interested in fast and reliable content delivery. Moreover, content over the Internet is expected to grow even faster, i.e., it is believed that global IP traffic will increase threefold over the next 5 years [1]. This poses significant challenges for the Internet due to the mismatch between its host-to-host communication paradigm and the current content-oriented usage.

To address these challenges, content-centric networking (CCN) [2–8] as a clean-slate solution is proposed. CCN tackles the challenges by adopting two new mechanisms, namely, *name-based routing* and *systematic in-network caching*. Name-based routing refers to the mechanism that every piece of content is identified by an addressable name and requests for the content can be routed by network. As a result, users of CCN issue requests for the content (expressed as *interests*), and the network takes care of locating and retrieving the data. This naturally realizes the so-called *location-independent* (or *location-unaware*) content delivery.

Meanwhile, to provide users with efficient content delivery, CCN employs *systematic in-network caching*. Each CCN router can store the requested content in its local cache and then use the previously forwarded data to satisfy future requests. By typically storing popular content objects at the router, in-network caching inherently guarantees CCN to have lower bandwidth consumption, less congestion and fast response time to content fetching.

However, content caching at the same time raises many new challenges in both understanding and utilizing the built-in network caching capability. Typical research problems include modeling and

☆ The research is supported by the National Natural Science Foundation of China (General Program 61373120) and the Fundamental Research Funds for the Central Universities (3102014JSJ0016).

* Corresponding author. Tel.: +86 02988491593.
E-mail addresses: wbchu@nwpu.edu.cn (W. Chu), wanglf@nwpu.edu.cn (L. Wang), haiyong.xie@ustc.edu (H. Xie), zhzhang@cs.umn.edu (Z.-L. Zhang), claud@nwpu.edu.cn (Z. Jiang).

http://dx.doi.org/10.1016/j.comcom.2015.09.009
0140-3664/© 2015 Published by Elsevier B.V.

analysis of system dynamics under different caching hierarchies and with different cache replacement policies (e.g., LRU, RND, FIFO) [9–13], provisioning en-router content storage for network performance optimization [14–17], etc. While most previous work focus on these topics, in this paper we go one step further to explore QoS (Quality of Service) guarantee in CCN. More specifically, we investigate the problem of guaranteeing network delays for differentiated services (content providers) in CCN while at the same time optimizing the overall content delivery performance. This is a significant task for both network administrators/operators and service providers as network delay is a key metric of QoS due to the nature that different kind of content has different network delay requirements. For example, voice and videos are far more sensitive to long network latency than web and emails.

Guaranteeing network delays for differentiated services (content providers) in CCN is a new research problem. While most of existing mechanisms for supporting delay sensitive traffic in network are designed as end-to-end semantics [18–20], in CCN the concept of “end-to-end flows” or “connections” do not even exist. As a result, existing mechanisms for guaranteeing network delays are no longer applicable in the context of CCN.

Meeting network delay requirements for differentiated services in CCN is also challenging, mostly due to the following factors. First, end-users are generally distributed across network at different locations and have different delays of fetching content objects from content providers. For example, in a network with a tree-like topology, users located at lower-layer nodes often have longer delays of fetching content than those connected at upper-layers. To meet delay requirements for end-users with different locations, the network topology information should be taken into account and the delay guarantee mechanism needs to properly handle this user location diversity. Second, the request access profiles of end-users (e.g., request rate, content distributions) are not always consistent and are changing over time. This also raises significant challenges as long-term and stable access pattern is often required in resource allocation and content assignment.

Another challenge faced when one designs the delay guarantee mechanism in CCN is the huge computational cost. Existing models or approximate algorithms [11–13] for analyzing caching performance (e.g., cache hit/miss ratio) for a network of caches often require per-content state tracking and analysis, i.e., by adopting Markov models [24]. As a result, significant amount of computation are involved when there is a large number of content objects or routers/nodes in the underlying system, as in the real network. This also implies that most of existing models or approximate algorithms are no longer applicable to the task of network delay guarantee in CCN. The required mechanism or models, on the other hand, needs to be computationally feasible and scalable.

To address these challenges and achieve delay guarantee in CCN, in this paper we make the following contributions:

1. We present a simple and holistic network model which characterizes network delays of routing content objects to clients for content of all kinds, namely, *locally cached*, *remotely cached* and *uncached*, based on their locations. By assigning the same top ranked content objects in customer-facing routers as locally cached, and popular objects in peer routers as remotely cached, we ensure that end-users at different locations have a unified content access pattern. And this content access pattern is long-term and stable since the number of top ranked content objects cached in network is rather small as compared to the number of content objects delivered by the network.
2. In order for each content provider to have an optimized network delay for its content dissemination, we align network locations with their content popularities by assigning the top most ranked content objects in customer-facing routers and the popular objects in peer routers. We then combine the content distribution model with the proposed holistic network model, and derive an analytical optimized network delay for each content provider.
3. With the analytical network delay for each content provider, we further formally formulate the network delay guarantee task in CCN as a nonlinear integer programming (NIP) problem under the given network resources and traffic patterns of the underlying competing content providers. Rather than calculating the exact location for each content object, we approach the problem by specifying the number of top most ranked content objects that are cached locally and that are cached remotely. This significantly reduces the computational cost as compared to the existing models or approximate algorithms to content placement.
4. We evaluate our models and investigate the optimized network performance through numerical studies. Using different network topologies, we study how content providers compete for the network resources and how various factors (e.g., content popularity, traffic volume, router storage capacity) affect this competition process. Our results reveal interesting and important phenomena, for example, increasing content population does not significantly influence the competition process, but it degrades the overall network delivery performance; similarly, it is observed that increasing network storage improves the overall content delivery performance, but it almost does not affect the competition process, etc. We believe these results are highly valuable as they provide insights into designing QoS mechanisms for CCN.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 gives a detailed description of our models (network model, content distribution model and delay model) as well as the problem formulation. Section 4 presents our numerical studies and evaluation results. We conclude the paper in Section 5.

2. Related work

Network architectures with built-in storage [2–6] have received increasingly attention, and there is a large body of research in this field. In this section, we review some of the most well-known work.

One of the most important topics in this area is modeling and analysis of caching mechanisms. Researchers have proposed models and algorithms for analyzing caching effectiveness and characterizing caching dynamics. In [21], Busari and Williamson adopted both synthetic workload and trace-driven simulation to evaluate different cache management policies for a two-level Web proxy caching hierarchy. Che et al. in [22] developed an analytical modeling technique to analyze the caching performance in the context of web caches, and identified two hierarchical caching design principles to improve the caching performance. Caching dynamics and performance for content-centric networks was also studied. Psaras et al. in [24] developed a continuous time Markov-chains based model to assess the time a given content object is in a router, and extended their model to multiple routers with some simple approximations. Rossi and Rossini in [25] investigated the impact of several parameters such as content size, content request distribution, on the performance of caching. Rosensweig et at. in [11] proposed an approximation algorithms to evaluate caching dynamics for networks with general topologies. In [26], Dabirmoghaddam et al. proposed a computational framework to compare the performance of optimal on-path caching against the simple strategy of caching only at the edge of the network.

Recently, performance optimization for content-centric networks has attracted much attention. Rossi and Rossini in [27] considered various network topology aware policies to improve the overall cache hit rate in a network of caches. In [28], the authors proposed probabilistic caching schemes to increase the cache hit rate in a network of caches. Carlsson et al. in [32] investigated the problem of using geographically distributed cloud platforms to content delivery and proposed an optimization model for dynamic request routing. Badov et al. in [33] proposed a congestion-aware caching and search mechanism in CCN for the optimization of user-centric content-download delay. Yeh et al. in [34] proposed the VIP (virtual interest packets) framework which employs both a virtual control plane and an actual control plane for joint dynamic forwarding and caching in CCN.

In summary, although there are many studies in the literature, there exists very little work on QoS guarantee for content-centric networking. In fact, the only work we find is [36], where Khan et al. proposed a QoS aware path selection scheme for a multi-path content-centric network. We argue that this is probably due to the fact that many fundamental issues in CCN such as the concept of flows, the definition of fairness, to name a few, are still open problems.

Also note that our work differs substantially with [37] where the authors proposed comparative models to study the performance bounds of Content-Centric and Content-Distribution Networks by addressing the joint content placement and routing problems. The main differences are: (1) we focus on service differentiation and consider multiple content providers in network while the work in [37] does not distinguish services/providers; (2) we consider network delay constraints in the optimization model while these constraints are not included in [37]; (3) to keep the problem practically tractable, content assignment in caches is computed on a per class basis in our model instead of per content as in [37].

To the best of our knowledge, this is the first attempt to study QoS guarantee with respect to meeting delay requirements for CCN. We believe that our models provide a simple yet effective way to allocate network storage to different content providers, and the numerical results are important in designing QoS mechanisms for CCN.

3. Models and problem formulation

In this section, we present in details our models (network model, content distribution model and delay model) for the delay guarantee task in CCN. We then give mathematical problem formulation and further discuss some related issues (e.g., computational cost, implementation) of our mechanism.

Note that the network model was originally proposed in [30], and in this work we extend it by considering networks with both end-routers and transit routers, and for completely different purposes.

3.1. Network models

We consider a content-centric network that comprises of three different components: end-users, routers and original server *O*, as shown in Fig. 1. End-users issue requests for content. Routers are equipped with network storage and routing function, and can serve content request if data is cached in its storage, or otherwise forward requests to the original server *O*. The original server *O* contains all the content and therefore can always satisfy content requests if they are missed by the network (routers).

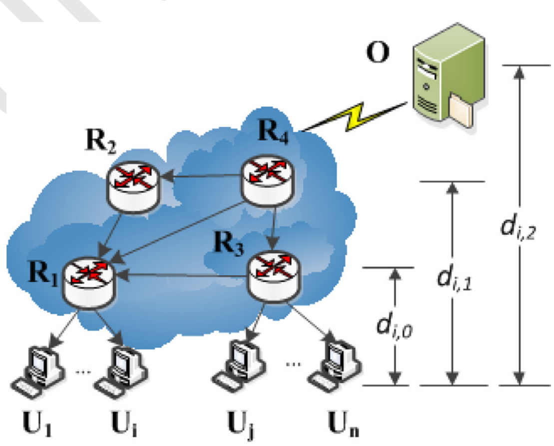


Fig. 1. A simple content-centric network model.

Note that in this conceptual model, the original server *O* is an abstraction of multiple origin servers. And in this work, it also denotes individual content providers.

Assumption. We focus on the network of a single administrative domain (e.g., an Autonomous system), and make the following assumptions.

Assumption 1. Each router in the network has a piece of content storage of size *C*.

Assumption 2. Following the common practice [10–12,17], we assume each content object is of equal size and is normalized to one unit with respect to router's storage capacity (see Assumption 1), which means that each router can hold at most *C* content objects in its storage.

Assumption 3. Traffic access pattern is consistent for users with different network locations. As the reader will see, this assumption is reasonable since in our model we cache the very small amount of top ranked content in routers and these content are considered to be rather long-term and stable.

In CCN, content requests can be satisfied by either the original server *O* or routers if the required data is cached. Moreover, routers with different network locations introduce different delays of fetching content. For example, content cached in end-routers (customer-facing routers) will have much smaller delay than those cached in intermediate (transit) routers or the original server *O*. Based on these observations, we classify requested content objects into three categories — *locally cached*, *remotely cached* and *uncached*. Locally cached objects refers to the objects that are cached in users' local routers, i.e., the first-hop routers (end-routers or customer-facing routers)[1]. These routers generally hold the most popular content in their storage as current cache replacement algorithms (e.g., LRU) tend to hold popular objects at routers closer to end-users. Remotely cached objects refer to the objects that are not cached in users' local routers, but instead are stored in other routers, i.e., peer routers. As a result, requests for these content objects are routed to and served by these peer routers. Uncached content refer to the content objects whose requests are missed by the network and are ultimately satisfied by the original server *O*.

The concept of local routers and peer routers can be demonstrated by taking the network shown in Fig. 1 as an example, where for end-users $U_1 \sim U_i$, router R_1 is their first-hop (customer-facing) router while router R_2, R_3 and R_4 are peer routers. It can be seen that local routers are end-routers at the same time. Meanwhile, since router R_3 is the local router for end-users $U_j \sim U_n$, we can see that end-routers can also be peer routers.

Note that our classification of content is actually based on their locations from end-users' perspective. Locally cached objects have the lowest network delay since their requests can be served directly by one-hop consumer-facing routers, while that for uncached content objects will experience the longest delay (i.e., several hops). The delay of the remotely cached content in peer routers, however, lies between the two.

Meanwhile, previous work have shown that coordinated caching mechanisms [40,41] where CCN routers store content in a coordinated manner allows more content objects to be cached and thus improves the overall content delivery performance of the network. We consider coordinated caching in our model and assume routers work

[1] We use *local routers*, *first-hop routers*, *end-routers* and *customer-facing routers* interchangeably.

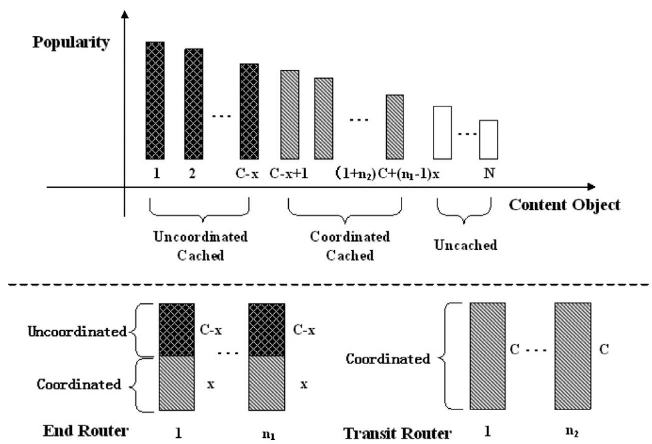


Fig. 2. Content assignment in routers' network storage.

collaboratively to decide which content object to store. More specifically, as in [30], we allocate an equally large size of network storage from each end-router for coordinated caching, say x out of C. The network storage of each end-router is thus divided into two parts, one for uncoordinated caching ($C - x$ out of C), and the other for coordinated caching (x out of C), as shown in Fig. 2.

Moreover, since the most popular content objects can be cached in each end-router without coordination (in uncoordinated caching storage), we specify coordinated caching storage of end-routers to store remotely cached content. To maximize the utilization of network caches, we also assume routers work collaboratively to cache *distinct content objects* in their coordinated caching storage so to hold as more content objects as possible, i.e., there is at most one copy of each content object in coordinated caching storage of the end-routers. As a result, n_1 end-routers in network will jointly cache $n_1 \cdot x$ distinct content objects in their coordinated caching storage.

Besides end-routers, there are also transit (intermediate) routers in network, i.e., R_2 and R_4 as in the network shown in Fig. 1. These routers do not have end-users connected. To hold as more *distinct content objects* as possible and improve the overall network delivery performance, in our model we allocate network storage of these transit routers solely for coordinated caching. Suppose there are n_2 transit routers, then the number of *distinct content objects* cached by these transit routers is $n_2 \cdot C$. To maximize the network performance, i.e., hold as more objects as possible, these distinct content objects in transit routers should also not *overlap* with that held by end-routers. As a result, totally $n_1 \cdot x + n_2 \cdot C$ distinct content objects are cached by all routers in network (including end-routers and transit routes).

Another important observation we have is that, although content popularity is generally dynamic in network, recent studies show that the top most popular content objects is rather long-term and stable [29]. For instance, hot videos from Youtube can last several hours or even days. Since the number of long-term popular content objects is quite small as compared to the number of existing objects on the Internet, and the network storage capacity are much smaller than the total number of content objects, we believe it would be reasonable for us to use the long-term popular content objects in our content assignment in an Autonomous System network. Based on these observations, in our model we specify all end-routers to cache the same top most ranked popular content objects (i.e., the top $(C - x)$ objects) in their uncoordinated caching storage. To minimize the overall network delay, we also have all routers (including end-routers and transit routes) jointly cache the next $n_1 \cdot x + n_2 \cdot C$ top ranked distinct content objects in their coordinated caching storage. The total number of unique content objects held by all routers, consequently, is $C - x + n_1 \cdot x + n_2 \cdot C$. The resulting content assignment in routers' network storage is illustrated in Fig. 2.

Overall, by allocating the same top most popular content objects in customer-facing routers (end-routers), popular objects in peer routers and unpopular ones uncached, we align network delays with the content popularity, and thus ensure each content provider will have an *optimized* network delay for its content dissemination. For arbitrary users at end-router i, this optimized network delay, denoted as $Delay_i$, can be calculated as follows:

$$Delay_i = \Pr\{most\ popular\} \cdot d_{i,0} + \Pr\{popular\} \cdot d_{i,1} + \Pr\{uncached\} \cdot d_{i,2} \quad (1)$$

where $\Pr\{most\ popular\}$, $\Pr\{popular\}$ and $\Pr\{uncached\}$ denotes the probability of fetching content objects from end-router i, peer routers and the original server O, respectively. And $d_{i,0}$, $d_{i,1}$ and $d_{i,2}$ denote the average network delay of fetching these content objects (see Fig. 1), respectively[2].

It can be seen from Eq. (1) that our delay model for each content provider incorporates both the network topology information and traffic access pattern. In fact, $d_{i,0}$, $d_{i,1}$ and $d_{i,2}$ are determined by the network structure, i.e., users at different locations will have different delays of fetching content objects from peer routers and the original server O. And $\Pr\{most\ popular\}$, $\Pr\{popular\}$ and $\Pr\{uncached\}$ depends on the content distribution (traffic) model.

Finally, it is noteworthy that in our content assignment model, users connected at different end-routers will have the same content access pattern, regardless of their locations. To be specific, since the same top most ranked content objects are cached at all of the local routers and the distinct objects cached remotely are uniformly distributed at peer routers, $\Pr\{most\ popular\}$, $\Pr\{popular\}$ and $\Pr\{uncached\}$ will be the same for end-users with different locations. This actually leads to a unified content access pattern for end-users, and which in turn greatly facilitates our characterization and computation of network delays for each content provider.

3.2. Traffic distribution model

In our delay model as shown in Eq. (1), we do not specify any detailed mathematical forms of the content distribution pattern. In fact, any content distribution model can be incorporated into our model.

Suppose that for an arbitrary content provider with N content objects, the top x_1 ranked content objects are locally cached in customer-facing routers, and the next top $x_2 - x_1$ ranked objects are remotely cached in peer routers (thus totally there are x_2 top ranked content objects cached by the network), the network delay of fetching content from this provider for end-users at end-router i, denoted by $D_i(x_1, x_2; N)$, can be calculated according to our delay model as follows:

$$D_i(x_1, x_2; N) = F(x_1; N) \cdot d_{i,0} + (F(x_2; N) - F(x_1; N)) \cdot d_{i,1} + (1 - F(x_2; N)) \cdot d_{i,2} \quad (2)$$

where $F(x_j; N)$ denotes the probability of requesting for the top x_j ranked objects.

Hereafter we assume that for each provider the content popularity distribution follows the Zipf distribution as shown in many studies [36,38,39]. Zipf's law predicts that out of a population of N elements, the probability of requesting for the top k ranked content objects is given by:

$$F(k; s, N) = \frac{\sum_{i=1}^{k} 1/i^s}{\sum_{j=1}^{N} (1/j^s)}, k = 1, 2, \ldots, N \quad (3)$$

where s is the Zipf's exponent.

[2] In this manuscript, the average network delay is adopted as a case study to evaluate our delay guarantee model. However, with the delay probabilities, one can easily express the delay requirement in a probabilistic form. Therefore, it can be seen that our model also works when the delay requirement is expressed in a probabilistic form.

Please cite this article as: W. Chu et al., Network delay guarantee for differentiated services in content-centric networking, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.009

Incorporating the above probability into Eq. (2), we derive an analytical formula for network delay for each content provider. With this analytical formula for network delay, we are then able to mathematically formulate the network delay guarantee problem where there are multiple content providers competing for the network resources.

3.3. Problem formulation

We now formally formulate the problem. Note that in the above network and delay model, we assume there is only one content provider. The delay guarantee problem we consider in this work, however, is much more complicated as there are multiple content providers and each one has its delay requirement. Our task is then how to allocate network resources to these content providers such that their delay requirements can be satisfied while at the same time the overall network delivery performance are optimized.

Problem formulation. Consider a network $G = (V, E)$ whose router set V consists of two parts: a set of end-routers U and a set of transit (intermediate) routers (V-U). Each router is equipped with a piece of content storage of size C. There are m content providers $CP = \{CP_1, CP_2, \ldots, CP_m\}$ running their businesses over the network. For each content provider j, let N_j be its content population and s_j the corresponding Zipf's exponent. Each content provider j has a specific delay requirement, i.e., the average network delay of fetching content objects from provider j for users at each end-router should not exceed its service-level agreement T_j. Meanwhile, for each end-router i, denote L_{ij} be its users' content access rate to provider j and $D_{ij}(x_{j1}, x_{j2}; s_j, N_j)$ be the network delay of fetching content objects from provider j, where x_{j1} and x_{j2} denote the number of top ranked content objects of provider j that are cached locally in customer-facing routers and that are cached by the whole network (including end-routers and transit routers). For each end-router i, denote $d_{i,0}$, $d_{i,1}$ and $d_{ij,2}$ the average network delay of fetching content from local routers, peer routers and the content provider j, respectively. $D_{ij}(x_{j1}, x_{j2}; s_j, N_j)$ thus can be calculated as follows:

$$\begin{aligned} D_{ij}(x_{j1}, x_{j2}; s_j, N_j) &= F(x_{j1}; s_j, N_j)\cdot d_{i,0} \\ &\quad + (F(x_{j2}; s_j, N_j) - F(x_{j1}; s_j, N_j))\cdot d_{i,1} \\ &\quad + (1 - F(x_{j2}; s_j, N_j))\cdot d_{ij,2} \end{aligned} \tag{4}$$

Given the above notations, the network delay guarantee task is then how to allocate network storage to different content providers, to be specific, determine the number of top ranked content objects that are cached locally in customer-facing routers and that are cached remotely in peer routers for each content provider, under the given network resources and traffic access patterns, so as to meet the delay requirements of the competing providers while at the same time minimize the overall content delivery latency. The problem can be mathematically formulated as follows:

$$\min \sum_{i\in U}\sum_{j\in CP} L_{ij}\times D_{ij}(x_{j1}, x_{j2}; s_j, N_j) \Big/ \sum_{i\in U}\sum_{j\in CP} L_{ij} \tag{5}$$

$$s.t. \begin{cases} D_{ij}(x_{j1}, x_{j2}; s_j, N_j) \le T_j, \forall i \in U, \forall j \in CP & (c1) \\ \sum_{j\in CP} x_{j1} \le C & (c2) \\ n_1 \cdot \sum_{j\in CP} x_{j1} + \sum_{j\in CP} (x_{j2} - x_{j1}) \le n\cdot C & (c3) \\ x_{j1}, x_{j2} \in Z, 0 \le x_{j1} \le x_{j2} \le N_j, \forall j \in CP & (c4) \end{cases} \tag{5}$$

The above optimization task is actually a nonlinear integer programming (NIP) problem due to the nature that network delay for each content provider is nonlinearly related to the integer decision variables x_{j1} and x_{j2}. Constraint (c1) denotes the delay requirement for each content provider. Constraint (c2) states that the total number of top ranked (locally cached) content objects from different providers should not overflow each end-router, and constraint (c3) requires that the total number of objects cached from different providers should not exceed the total amount of network storage. It can be seen that constraint (c2) and (c3) together describe the network resource limitations.

Nonlinear integer programming is mathematically NP-hard. To solve the problem efficiently, we convert it to a general nonlinear optimization problem by relaxing x_{j1} and x_{j2} to be two continuous variables. This is because: (1) the number of content objects served by each provider is generally very huge as compared to the network storage capacities, i.e., more than 120,000,000 videos are uploaded on Youtube every day and (2) for each end-router i and content provider j, the difference of the delays on any two consecutive integer points, i.e., $|D_{ij}(\lfloor x_{j1}\rfloor, \lfloor x_{j2}\rfloor; s_j, N_j) - D_{ij}(\lceil x_{j1}\rceil, \lceil x_{j2}\rceil; s_j, N_j)|$, is relatively small as compared to the delay requirement T_j. And in our numerical studies, we adopt PyOpt package [42] as the solver for the converted nonlinear programming problem.

3.4. Computational cost and implementation

(a) *Computational cost*. Content placement or replacement is often addressed by existing methods through solving complicated mathematic models on a per content basis (e.g., by adopting Markov models), and therefore a high computational cost is incurred under large-scale environment. In this work, instead of specifying the exact location for each content object, we adopt a simple and holistic network model for content assignment and approach the problem by specifying the number of top ranked content objects that are cached locally and that are cached remotely, which significantly decreases the computational cost. For the given network topology with 9 routers (each can accommodate 1000 content objects), 2 providers and 1000000 content objects served by each, it takes less than 1min to determine the cached objects in our numerical studies. Thus it can be seen that our delay guarantee mechanism is computational feasible in an AS environment.

(b) *Implementation*. With regard to implementation, we consider a centralized network management as in [40,41]. There is a server which periodically collects the content request information from each end-router, and then it estimates content popularity (distribution) and calculates the optimal content assignment. After that, the server indicates each router which content object to store in its storage. Meanwhile, to support network-wide caching as in our mechanism, we require some change on existing CCN's routing function to allow request to be forwarded to a router holding the content but which is not on the path from a requester toward the original provider. To be specific, we can adopt a new hash table to test whether the content in a received interest is selected in our content assignment. If so, the router updates the FIB (Forwarding Information Base) entry for the content and directs the request to the corresponding router. Otherwise, the interest is forwarded by the existing FIB entries.

4. Numerical studies and evaluation

In this section, we evaluate our delay guarantee mechanism through numerical studies. We mainly focus on the following two objectives: (1) illustrate how our delay guarantee model can be adopted to provide delay guaranteed services for different content providers; (2) based on the numerical results, study how different content providers (services) compete for the network resources and how various factors (e.g., content distribution, traffic volumes, router storage capacity) affect the competition process.

4.1. Evaluation methodology

In the numerical study we use different network topologies and synthetic content providers. For each given network topology, we calculate its network parameters (e.g., $d_{i,0}$, $d_{i,1}$ and $d_{i,2}$) based on its structure property. We then give a baseline setting of the parameters for the synthetic content providers and the underlying network, and vary each of these parameters (e.g., content population, router storage capacity) so as to assess their impact on the competition process. This allows us to identify the main influencing factors as well as figure out how these factors affect the competition process.

Meanwhile, for the performance evaluation we adopt the following four metrics:

1. x_{j1}, the amount of end-routers' storage allocated to a provider j.
2. x_{j2}, the amount of storage of all routers (including end-routers and transit routers) allocated to provider j.
3. $\max_{\forall i \in U}\{D_{ij}\}$, the maximum network delay of fetching content objects from provider j.
4. T, the overall average network delay of fetching content from providers.

Among the above four metrics, x_{j1} and x_{j2} denote the routers' storage allocated to a provider, and hence they can be used to evaluate the network resource allocation by our model. And $\max_{\forall i \in U}\{D_{ij}\}$ and T reflect the optimized network performance by our mechanism.

4.2. Evaluation setup

4.2.1. Network topologies

The Abilene network topology (11 routers, 14 links) shown in Fig. 3 is used in this study whose results are presented. Two end-users and content providers are randomly connected at different routers in the network representing both the providers' and the users' location diversity. Each router checks the requested object in its local cache before forwarding the request. If the requested content object cannot be found in the content storage, then the request will be forwarded to the next-hop router along the routing path determined by FIB. The request will be forwarded until either it reaches a node holding the content or the custodian (content providers).

Note that we also adopt other network topologies (e.g., the tree-like ISP network) and obtain similar results, so in this paper we only present the results for the Abilene network for brevity.

We use hop count as the network delay indicator, and assume content requests are routed via the shortest path between the requester and the provider. The network delay parameters for users at different end-routers are listed in Table 1. The parameters are defined and calculated as follows: (1) $d_{*,0}$ denotes the network delay of fetching content from end-routers and hence $d_{*,0} = 1$ hop for all end-users;

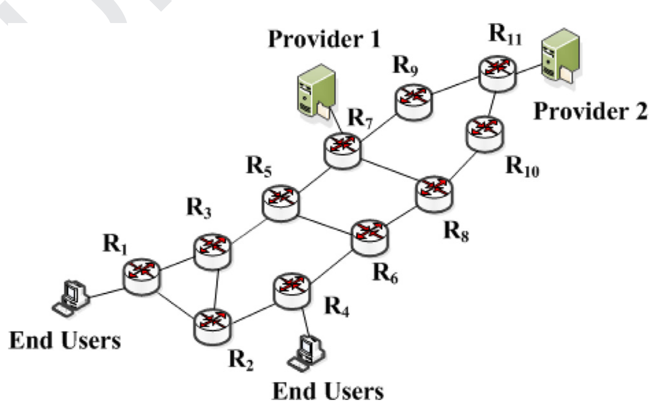


Fig. 3. Abilene network topology used for the numerical results.

Table 1
Network delay parameters.

Users	Provider	$d_{*,0}$(hops)	$d_{*,1}$(hops)	$d_{*,2}$(hops)
R_1	1	1	41/11	4
R_1	2	1	41/11	6
R_4	1	1	35/11	4
R_4	2	1	35/11	5

Table 2
Parameter settings.

Content provider	# Content objects	Zipf's exponent	Router storage capacity	Delay requirement	Request rate
Provider 1	800000	1.2	1000	3	1000
Provider 2	1000000	1.2	1000	4	2000

(2) $d_{*,1}$ denotes the average network delay of fetching content from peer routers and it is calculated in this way: for end user i, let $h_{i,j}$ denote the hop count of the shortest path between i and router j, $d_{i,1}$ are hence calculated as $d_{i,1} = \frac{\sum_{j \in V} h_{i,j}}{|V|}$ where V is the set of routers (peer routers) that i can reach; (3) $d_{*,2}$ denotes the network delay of fetching content from remote content providers, and it is the hop count of the shortest path between the user and the provider.

4.2.2. Content providers

We assume there are two content providers in the underlying network (see Fig. 3) and the content access pattern (content request rate and content distribution of each provider) at end-routers are identical. Note that this assumption is solely for ease of illustration, and our models apply to more complicated scenarios as in the real network, i.e., when the content request rate and content distribution of each provider at end-routers are completely different. Table 2 lists the parameter settings for the providers in our evaluation and this setting of parameters are then used as a baseline for our evaluation (each time we change one parameter and then look at its impact on the competition process).

4.3. Numerical results

4.3.1. Impact of content distribution

Given the parameter settings of the network and the two content providers described above, we first investigate how content distribution affects the competition process of the two providers. Fig. 4 shows the amount of routers' storage allocated to the two providers as well as the average network delay of fetching content objects when the Zipf's exponent of provider 1 varies. Clearly we can see that the amount of routers' storage allocated to provider 1 decreases when its Zipf's exponent becomes larger. We argue this is due to the fact that a larger exponent in a Zipf's law implies that the workload is more skewed (i.e., more workload is concentrated on a smaller set of populations). As a result, a smaller amount of storage is needed to satisfy the delay requirement for provider 1. Meanwhile, since the network storage is shared by two providers, one provider possessing smaller amount of storage will certainly lead to an increased amount of storage allocated to the other, and this in turn results in an improved network performance of fetching content from that provider, as shown in Fig. 4(c). In fact, from Fig. 4(c) we can see that both the network delay of fetching content objects from the two providers and the overall average network delay decrease when the content popularity becomes more skewed.

Another important phenomenon we observed from Fig. 4(a) and Fig. 4(b) is on the rate of decrease of network storage allocated to provider 1. More specifically, it can be seen that the amount of end-router's storage allocated to provider 1 decreases much more slowly (Fig. 4(a)) than that of peer routers' storage allocated to provider 1

(Fig. 4(b)). We believe that this is due to different roles of the two types of routers (end-routers and peer routers) in reducing network latency. Fetching content from end-routers incurs the lowest network delay (1 hop) while it takes several hops to retrieve content objects from peer routers. Thus we can see that, from the perspective of reducing network delay and saving network bandwidth, the storage of end-routers is more precious than that of peer routers. And in the competition process, every content provider has its priority to compete for end-routers' storage for its content dissemination. In other words, if a content provider has to make room for other providers, it will prefer to free its peer routers' storage than end-routers.

4.3.2. Impact of content population

Fig. 5 shows the network performance when the content population of provider 1 varies. From Fig. 5(a) and (b) we can see that the network resources allocated to each provider is insensitive to the number of content objects served by a provider. In other words, content population is not a factor that is likely to significantly influence the competition process of the underlying content providers.

However, from Fig. 4(c) we can see that while the average network delay of fetching content from provider 2 almost remains unchanged, the average network delay of fetching content objects from provider 1 as well as the overall average network delay of fetching content increases when the content population of provider 1 becomes larger. We believe that this is because more and more content objects are uncached by the network when the content population grows. Since the requests on these uncached content objects are served by the original provider, this leads to an increase in the average network delay of fetching content.

In short, Fig. 5 intuitively tells us that a content provider cannot benefit from increasing its content population in the competition process, but instead it suffers since doing this will lead to a decrease in its content delivery performance.

4.3.3. Impact of content request rate

Fig. 6 shows how the network performance changes when the request rate of provider 2 grows. Clearly we can see that as the workload increases, the network storage allocated to the corresponding provider also increases, as shown in Fig. 6(a) and (b). However, it is observed that the pattern on how the allocated storage changes is quite different from that shown in Fig. 4(a) and (b).

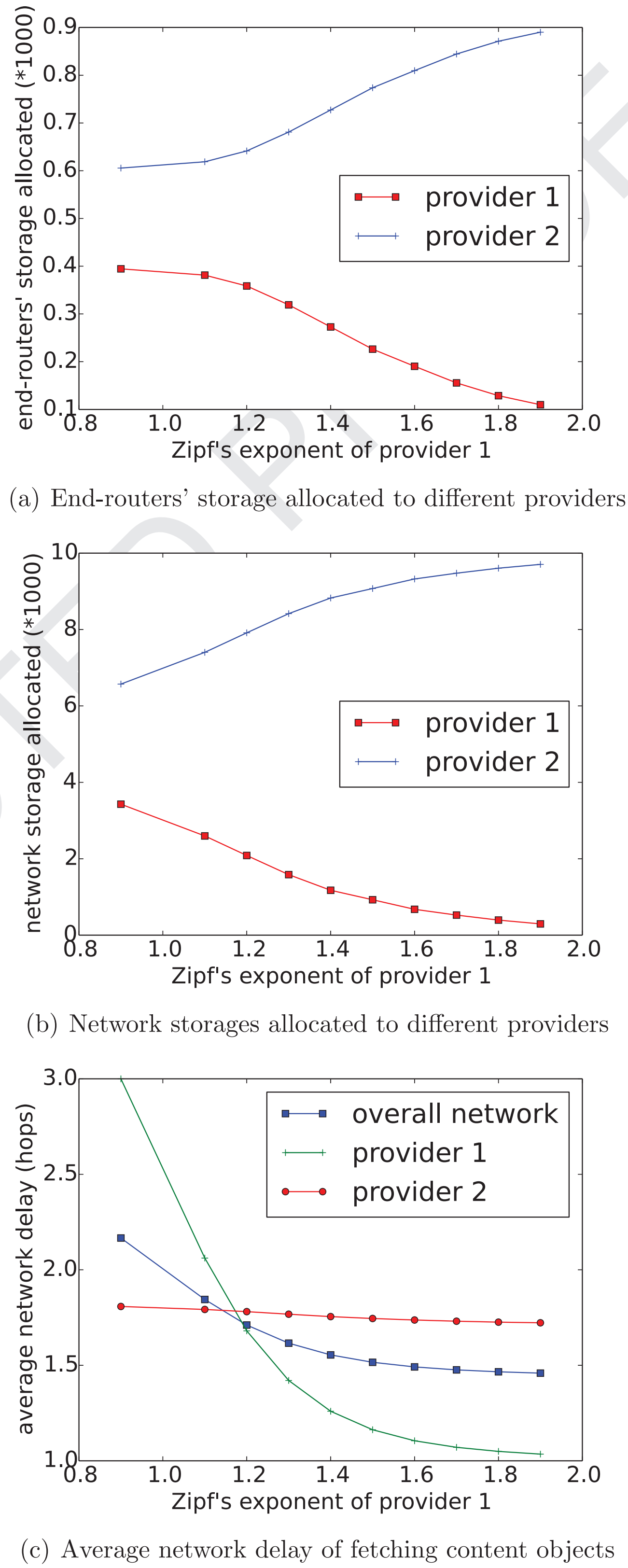
Fig. 6 (c) shows that the average network delay does not change monotonically with the increase of the content request rate, which is quite different from that shown in Fig. 4(c)) and Fig. 5(c)). We believe that this is because the metric of the average network delay is jointly determined by the delay of the competing providers and their content request rate.

Overall, from Fig. 6 we can see that content request rate is a factor that can significantly influence the competition process of the underlying providers. The larger content request rate to a provider (more popular), the more network storage allocated to that provider, and the smaller network delay of fetching its content (see Fig. 6(c)).

4.3.4. Impact of router's storage capacity

We then investigate how the network performance changes when router's storage capacity increases, and the results are shown in Fig. 7. From Fig. 7(a) and (b) we can see that both of the two providers obtain more network storage when the router's storage capacity becomes larger. However, it is interesting to observe that the network storage allocated to both content providers increase proportionally with the increase of router's storage capacity, which implies that increasing router's storage does not influence the competition process of the two content providers.

Fig. 7 (c) shows that as expected, the average network delay of fetching content drops when the network storage increases since more and more content objects are cached. However, it can be seen



(a) End-routers' storage allocated to different providers

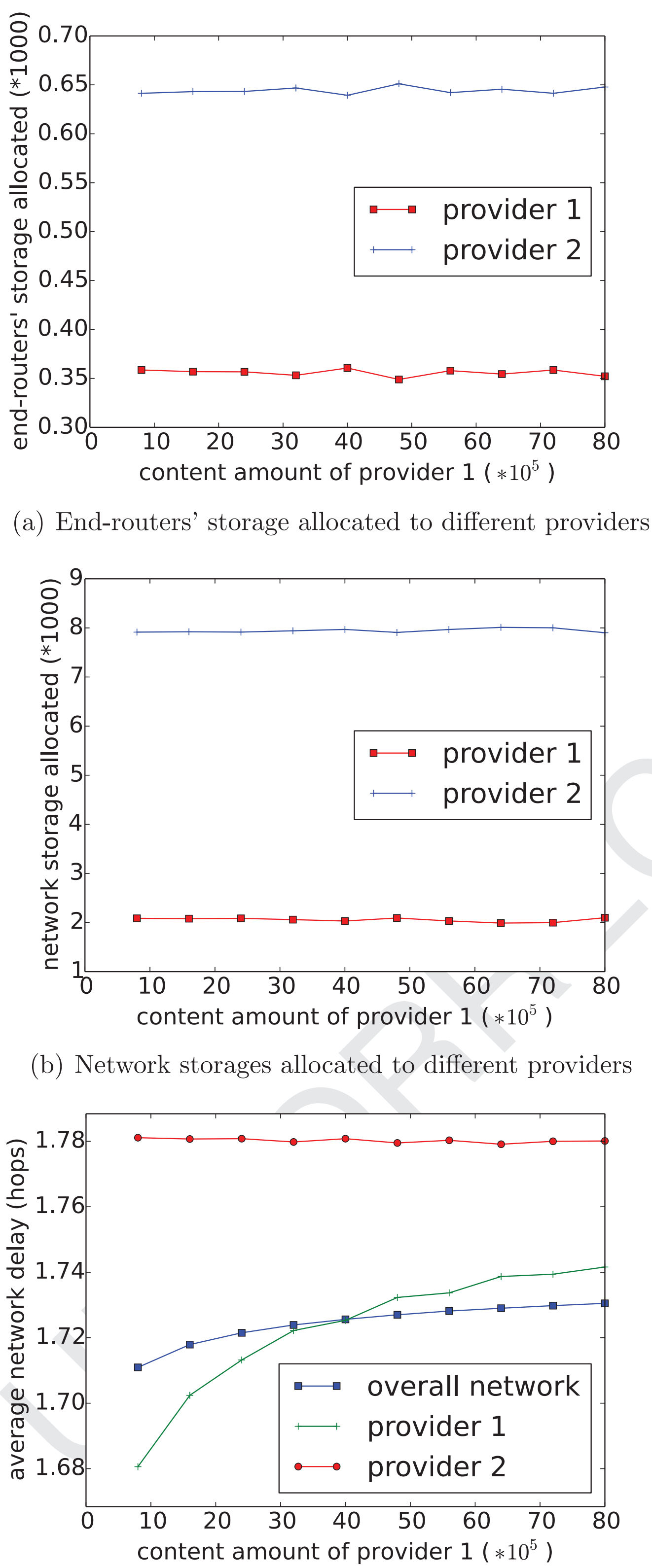
(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 4. Network performance when content distribution varies.

that the average network delay decreases rapidly at the very beginning and then decreases slowly. We argue that this is due to the nature of Zipf's distribution where almost 80% of requests are concentrated on 20% of content objects. When the network storage

Please cite this article as: W. Chu et al., Network delay guarantee for differentiated services in content-centric networking, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.009

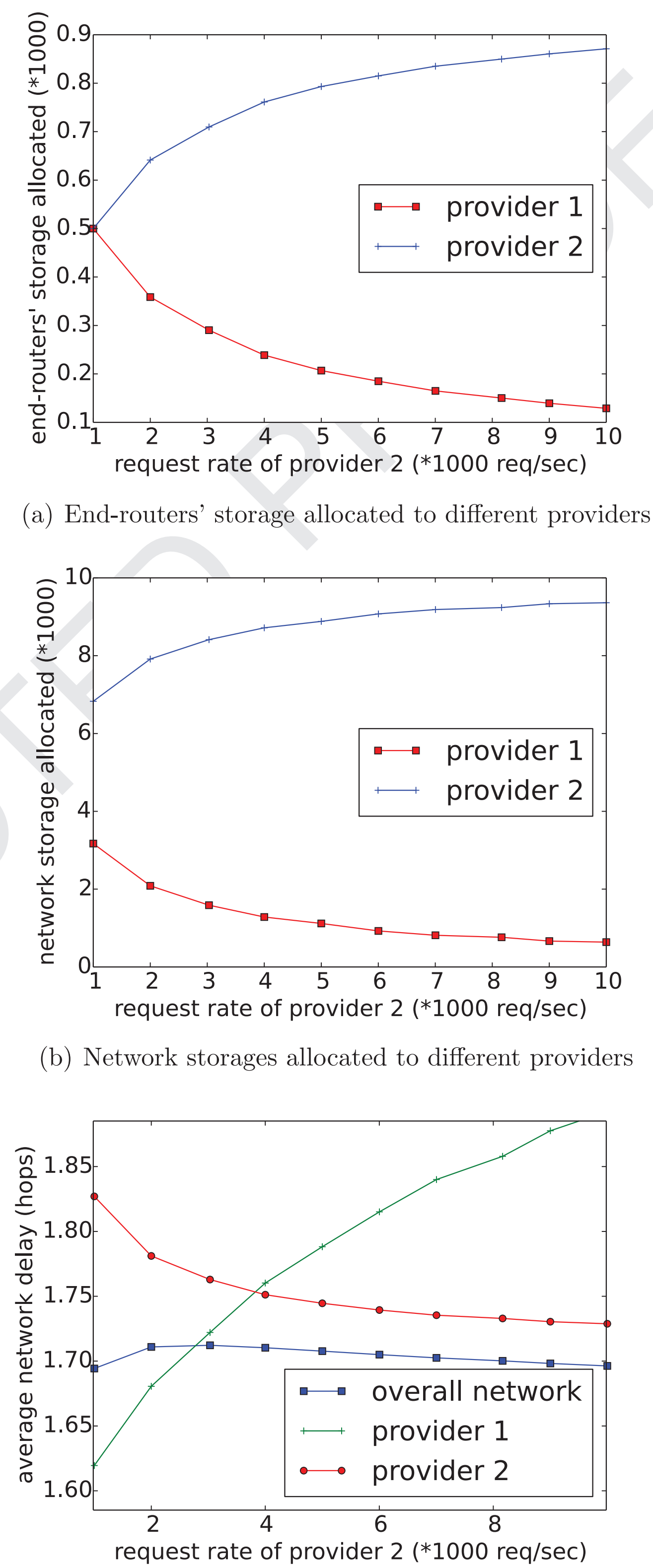


(a) End-routers' storage allocated to different providers

(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 5. Network performance when content population varies.



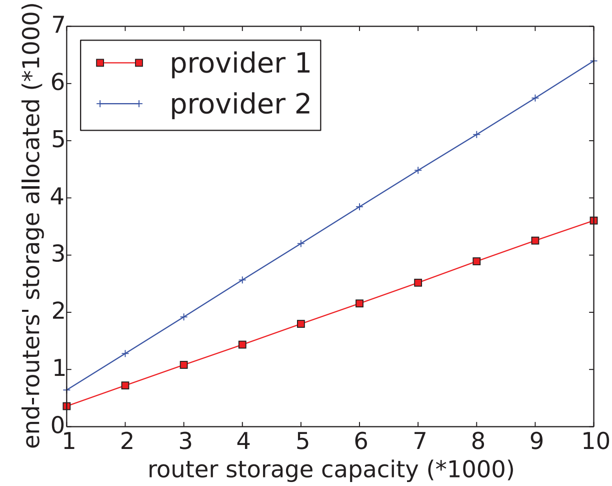
(a) End-routers' storage allocated to different providers

(b) Network storages allocated to different providers

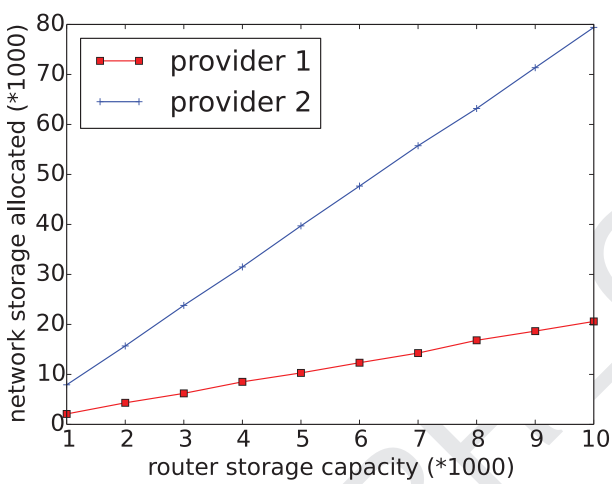
(c) Average network delay of fetching content objects

Fig. 6. Network performance when content request rate varies.

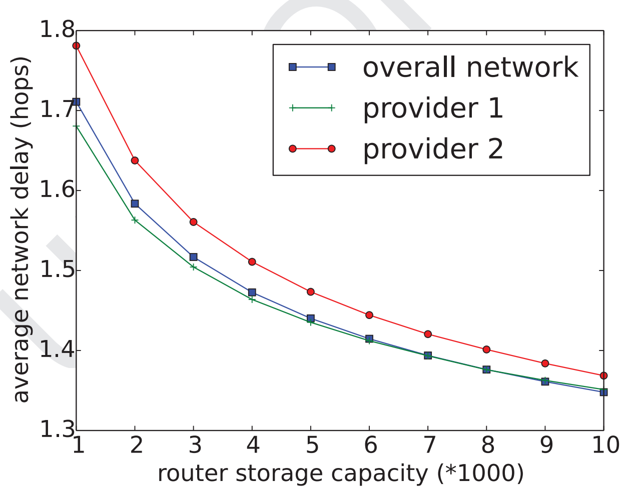
increases, more and more highly workload-concentrated content objects are cached by the network and hence the network delay drops rapidly. However, after these highly workload-concentrated content objects are cached, the average network delay will drop slowly since only a very small amount of workload is concentrated on the newly cached content objects. We believe this property is important as it provides insight on network resources provisioning and allocation for network administrators.



(a) End-routers' storage allocated to different providers

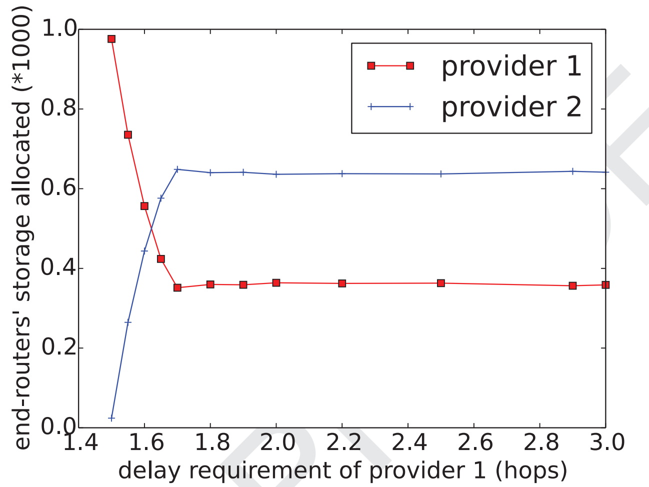


(b) Network storages allocated to different providers

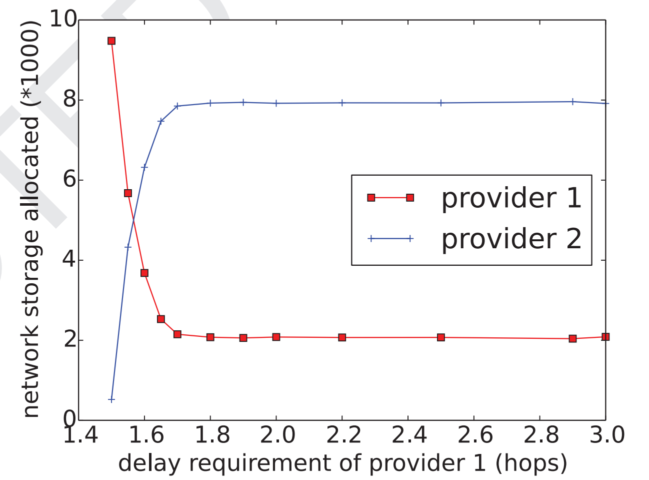


(c) Average network delay of fetching content objects

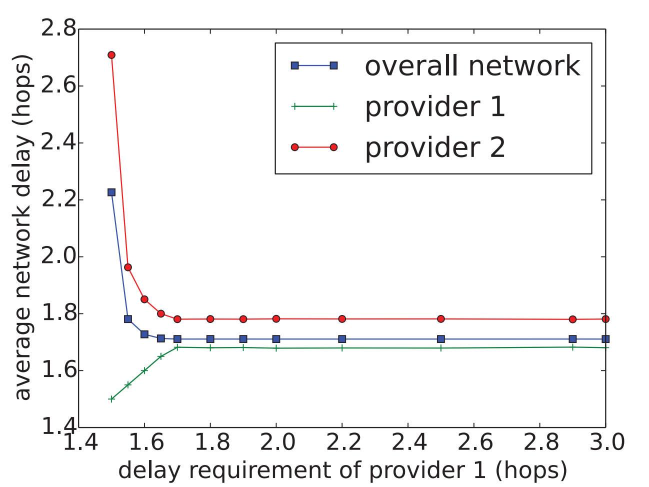
Fig. 7. Network performance when router's storage varies.



(a) End-routers' storage allocated to different providers



(b) Network storages allocated to different providers



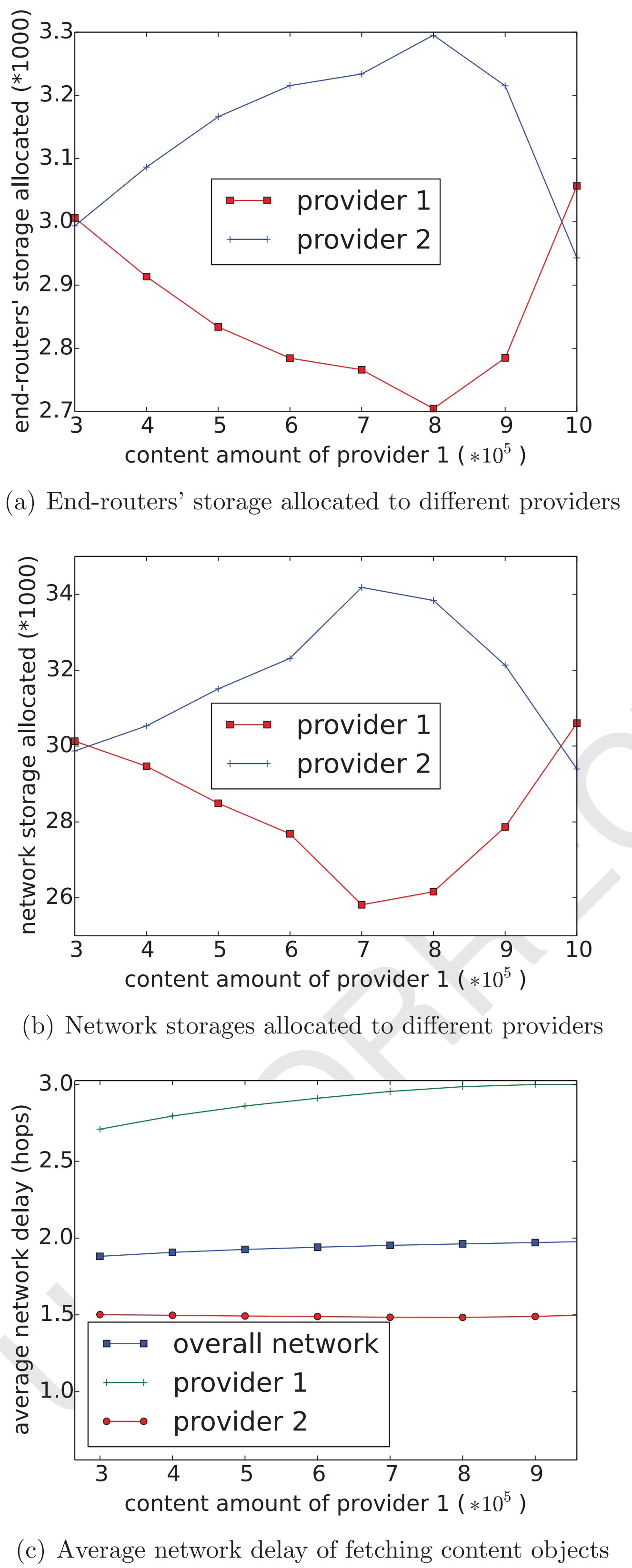
(c) Average network delay of fetching content objects

Fig. 8. Network performance when delay requirement varies.

4.3.5. Impact of delay requirement of content providers

Fig. 8 shows the network performance under different delay requirements of content provider 1. From these figures we can see that the network performance almost remains unchanged when the delay requirement of provider 1 is larger than 4 hops. However, it is observed that when the delay requirement becomes smaller than 3.5 hops and when it continues to decrease, the network storage allocated to provider 1 as well as the overall average network delay

Please cite this article as: W. Chu et al., Network delay guarantee for differentiated services in content-centric networking, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.009

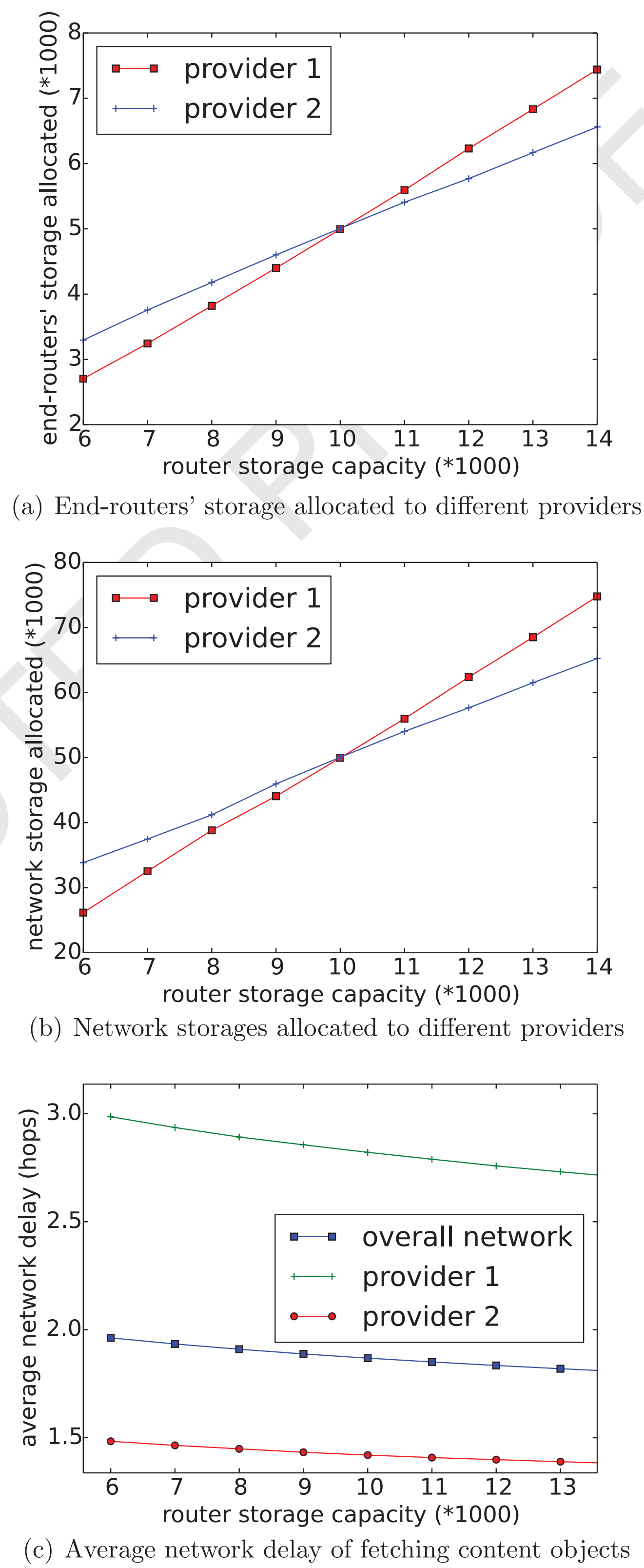


(a) End-routers' storage allocated to different providers

(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 9. Network performance when content population varies (Zipf's exponent < 1.0).



(a) End-routers' storage allocated to different providers

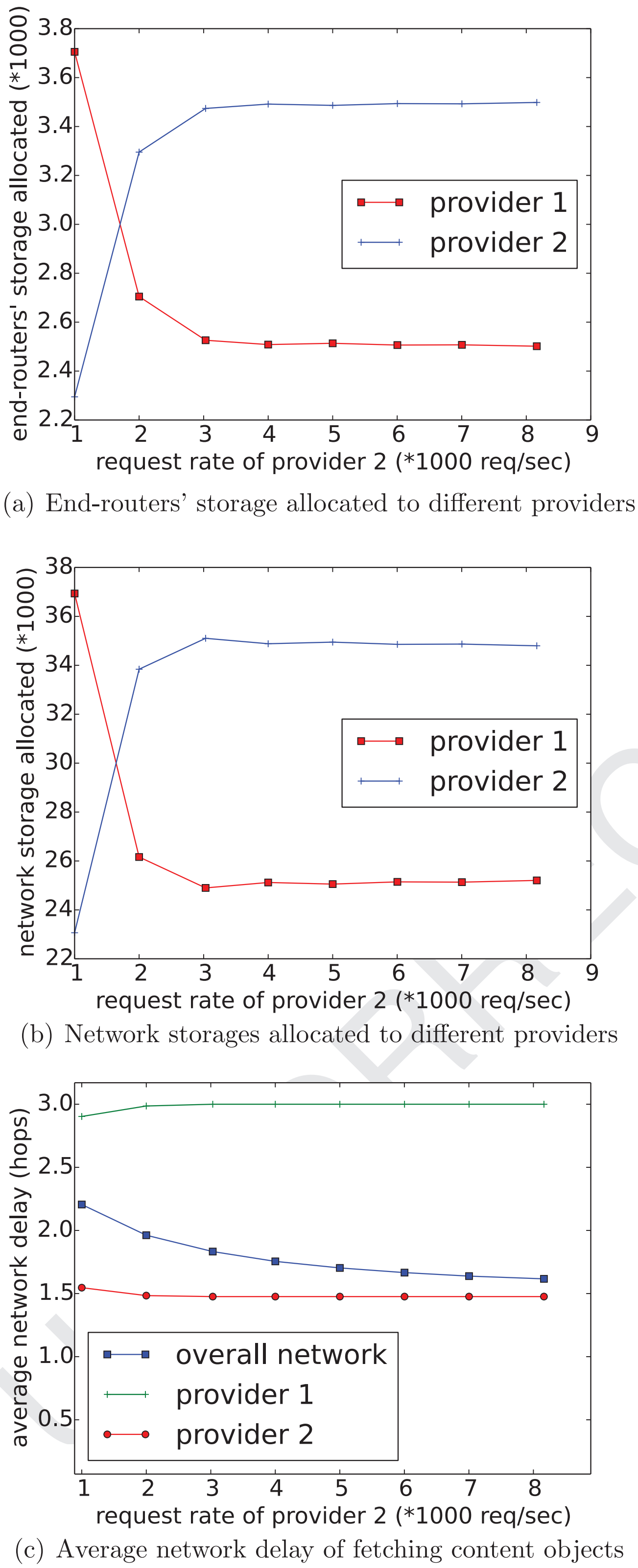
(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 10. Network performance when router's storage varies (Zipf's exponent < 1.0).

increase rapidly. After a careful examination on the inequality constraints in Eq. (5), we find that only when the delay requirement of content provider 1 is smaller than 1.7 hops do its constraints with equality hold. In other words, all the constraints with equality of content provider 1 are inactive when its delay requirement is larger than 1.7 hops. From this point of view, we can see that delay requirement of a content provider can influence the competition process, but the mechanism on how it impacts is much more complicated as compared to the other factors.

Please cite this article as: W. Chu et al., Network delay guarantee for differentiated services in content-centric networking, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.009

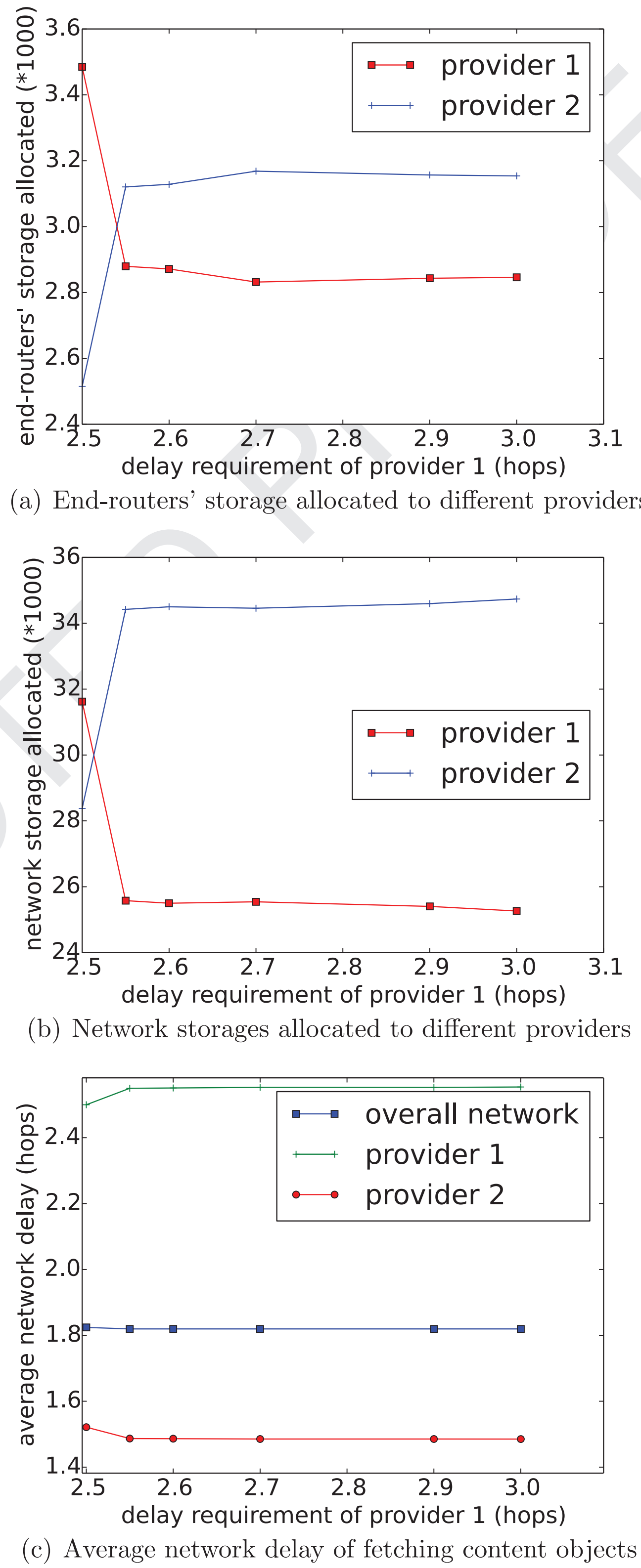


(a) End-routers' storage allocated to different providers

(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 11. Network performance when content request rate varies (Zipf's exponent < 1.0).



(a) End-routers' storage allocated to different providers

(b) Network storages allocated to different providers

(c) Average network delay of fetching content objects

Fig. 12. Network performance when delay requirement varies (Zipf's exponent < 1.0).

4.4. Numerical results when Zipf's exponent < 1.0

In the above subsection we give numerical results when content distribution of the two providers are highly skewed. For completeness, we also have evaluated our model on the scenario where Zipf's exponent is less than 1.0. To achieve this, we configured the Zipf's exponent of provider 1 to 0.8, the router's storage capacity to 6000, and the other parameters unchanged as in the baseline setting. The derived results are shown in Figs. 9–12.

Again from these figures we can observe similar trend, e.g., increasing the network storage improves the overall network delivery performance, but it almost does not influence the competition process of the underlying providers (see Fig. 10). However, we also observe different phenomena. One difference lies in the behavior of the two providers when the content population of provider 1 varies. As shown in Fig. 9, one can see that the behavior of the two providers are very sensitive to the content population, which is quite different from that shown in Fig. 5. More specifically, it is shown that at the very beginning, the network storage allocated to provider 1 decreases as its content population grows, but it then increases after the population exceeds some threshold. The initial decrease of network storage allocated to provider 1 is quite opposite to our expectation since in general, it needs more network resources to hold more content objects when content population grows, in order to guarantee network delay. Meanwhile, it is observed that for the two-provider competition process, the network resources allocated will keep unchanged once one provider reaches its maximum delay requirement, as shown in Fig. 11. These results indicate that one has to take into account the providers' content distribution type (i.e., whether it is skewed) when he/she designs QoS mechanisms for CCN.

5. Conclusion and future work

5.1. Conclusion

QoS guarantee for content-centric networks is a new but challenging research area due to the newly introduced built-in caching mechanism. Network delay is a key metric of QoS. In this paper, we investigate the problem of guaranteeing network delays for differentiated services (content providers) in CCN while at the same time optimizing the overall content delivery performance. To address the key challenges, in particular, the high computational cost incurred by conventional solutions such as Markov models, we propose a simple yet elegant network model for characterizing network delays of routing content to clients with different locations. We then derive analytical formula for network delay for each content provider by incorporating its content distribution pattern into the proposed network model. The network delay guarantee task is further formulated as a nonlinear integer programming (NIP) problem under the given network resources and traffic patterns of the underlying content providers. Finally, we evaluate our mechanism by numerical studies using different network topologies and investigate various factors (e.g., content popularity, traffic volume, router storage capacity) affecting the competition process. Our models and results in this paper provide guidance in designing mechanisms for QoS guarantee as well as other issues such as network resource provisioning and allocation in CCN.

5.2. Future work

There are several interesting directions for future research. First, as there are multiple content providers in the network and our mechanism tries to provide the best for all end-users while guarantee the delay requirements for each provider, it is interesting to study the fairness of resource allocation among these competing providers. Secondly, we adopt the network-centric metric—hop count, which is often used in the performance evaluation and optimization for CCN in most existing work. However, as the network capacity is limited, the actual delay (e.g., the content-download delay) will unavoidably be influenced by users' generated traffic volume. Therefore, to guarantee user-centric network delay, i.e., the content-download delay, one has to consider the network capacity (router storage, link bandwidth, etc) as well as the traffic characteristics (content popularity, traffic volume) into the optimization model. In our future work, we will focus on optimizing this user-centric content-download delay. Finally, our mechanism assumes that the demand is relatively stable with time. This is probably true in general but in case of a flash crowd the demand may vary drastically and make the allocation completely inefficient and break the SLA. It is therefore necessary to investigate the scenario of flash crowd and explore heuristics (e.g., a reactive algorithm) that ensures the optimal allocation is maintained and the delay is guaranteed.

Uncited references

Refs. [23,31,35]. Q3

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions which will certainly improves the quality of this paper.

References

[1] Cisco, 2014, Inc. Cisco Visual Networking Index: Forecast and Methodology. http://preview.tinyurl.com/3p7v28, accessed Sept. 22.
[2] T. Koponen, M. Chawla, B.G. Chun, et al., A data-oriented (and beyond) network architecture, ACM SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 181–192.
[3] A. Mark, Academic Dissemination and Exploitation of a Clean-slate Internetworking Architecture: The Publish-Subscribe Internet Routing Paradigm (2014). http://www.psirp.org/publications.html, accessed Sept. 22
[4] Scalable and Adaptive Internet Solutions. http://www.sail-project.eu/, accessed Sept. 22, 2014.
[5] Named Data Networking (NDN) Project. http://named-data.org/, accessed Sept. 22, 2014.
[6] K. Cho, J. Choi, D. Ko, et al., Content-oriented networking as a future internet infrastructure: Concepts, strengths, and application scenarios, Future Internet Technol. (2008).
[7] V. Jacobson, D.K. Smetters, J.D. Thornton, et al., Networking named content, in: CoNEXT09, Rome, Italy, 2009, pp. 1–12.
[8] J. Choi, J. Han, E. Cho, et al., A survey on content-oriented networking for efficient content delivery, IEEE Commun. Mag. 49 (3) (2011) 121–127.
[9] M. Gallo, B. Kauffmann, L. Muscariello, et al., Performance evaluation of the random replacement policy for networks of caches, ACM SIGMETRICS Perform. Eval. Rev. 40 (1) (2012) 395–396.
[10] P.R. Jelenkovic, Asymptotic approximation of the move-to-front search cost distribution and least-recently-used caching fault probabilities, Annal Appl. Probab. 9 (2) (1999) 430–464.
[11] E.J. Rosensweig, J. Kurose, D. Towsley, Approximate models for general cache networks, in: Proceedings of IEEE INFOCOM, 2010, pp. 1–9.
[12] L. Muscariello, G. Carofiglio, M. Gallo, Bandwidth and storage sharing performance in information centric networking, in: Proceedings of ACM SIGCOMM Workshop on Information-centric Networking. ACM, 2011, pp. 26–31.
[13] G. Carofiglio, M. Gallo, L. Muscariello, et al., Modeling data transfer in content-centric networking, in: Proceedings of the 23rd International Teletraffic Congress, International Teletraffic Congress, 2011, pp. 111–118.
[14] A. Jiang, J. Bruck, Optimal content placement for en-route web caching, Second IEEE International Symposium on Network Computing and Applications, IEEE, 2003, pp. 9–16.
[15] M. Korupolu, M. Dahlin, Coordinated placement and replacement for large-scale distributed caches, IEEE Trans. Knowl. Data Eng. 14 (6) (2002) 1317–1329.
[16] S. Borst, V. Gupta, A. Walid, Distributed caching algorithms for content distribution networks, in: Proceedings of IEEE Infocom, 2010, pp. 1–9.
[17] J. Li, H. Wu, B. Liu, et al., Popularity-driven coordinated caching in named data networking, Proceedings of the Eighth ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ACM, 2012, pp. 15–26.
[18] V. Sivaraman, F.M. Chiussi, M. Gerla, Traffic shaping for end-to-end delay guarantees with EDF scheduling, Eighth International Workshop on Quality of Service (IWQOS 2000), IEEE, 2000, pp. 10–18.
[19] C. Bouras, A. Sevasti, A delay-based analytical provisioning model for a QoS-enabled service, IEEE International Conference on Communications (ICC'06), IEEE, 2006, pp. 766–771.
[20] P. Lama, X. Zhou, Efficient server provisioning with end-to-end delay guarantee on multi-tier clusters, The 17th International Workshop on Quality of Service (IWQoS 2009), IEEE, 2009, pp. 1–9.
[21] M. Busari, C. Williamson, Simulation Evaluation of a Heterogeneous Web Proxy Caching Hierarchy, Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE, 2001, pp. 379–388.
[22] H. Che, Z. Wang, Y. Tung, Analysis and design of hierarchical web caching systems, in: Proceedings of IEEE Infocom, 2001, pp. 1416–1424.
[23] C. Williamson, On filter effects in web caching hierarchies, ACM Trans. Int. Tech. 2 (1) (2002) 47–77.
[24] I. Psaras, R.G. Clegg, R. Landa, et al., Modelling and evaluation of ccn-caching trees, in: Proceedings of IFIP Networking, 2011, pp. 78–91.
[25] D. Rossi, G. Rossini, Caching Performance of Content Centric Networks Under Multi-path Routing (2011).Technical Report

[26] A. Dabirmoghaddam, M. Mirzazad-Barijough, J.J. Garcia-Luna-Aceves, Understanding Optimal Caching and Opportunistic Caching at "The Edge" of Information-Centric Networks, in: Proceedings of 1st International Conference on Information-Centric Networking, ACM, 2014, pp. 47–56.

[27] D. Rossi, G. Rossini, On sizing ccn content stores by exploiting topological information, in: Proceedings of IEEE Infocom, NOMEN Workshop, 2012, pp. 280–285.

[28] I. Psaras, W.K. Chai, P. George, Probabilistic in-network caching for information-centric networks, in: ICN Workshop, 2012, pp. 1–6.

[29] Y. Kim, I. Yeom, Performance analysis of in-network caching for content centric networking, Comput. Netw. 57 (13) (2013) 2465–2482.

[30] Y. Li, H. Xie, Y. Wen, et al., Coordinating in-network caching in content-centric networks: model and analysis, 2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS), IEEE, 2013, pp. 62–72.

[31] G. Carofiglio, M. Gallo, L. Muscariello, Icp: Design and evaluation of an interest control protocol for content-centric networking, in: Proceedings 1st IEEE Intl Workshop on Emerging Design Choices in Name-Oriented Networking, 2012, pp. 304–309.

[32] N. Carlsson, D. Eager, A. Gopinathan, Z. Li, Caching and optimized request routing in cloud-based content delivery systems, Perform. Eval. 79 (0) (2014) 38–55.

[33] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, S. Nanda, Congestion-Aware Caching and Search in Information-Centric Networks, in: Proceedings of 1st International Conference on Information-centric Networking, ACM, 2014, pp. 37–46.

[34] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, D. Leong, VIP: A Framework for Joint Dynamic Forwarding and Caching in Named Data Networks, Proceedings of 1st International Conference on Information-centric Networking, ACM, 2014, pp. 117–126.

[35] L. Saino, C. Cocora, G. Pavlou, Cctcp: A scalable receiver-driven congestion control protocol for content centric networking, in: Proceedings of IEEE ICC, 2013, pp. 3775–3780.

[36] A.Z. Khan, B. Shahab, F.R. Dogar, QoS aware path selection in content centric networks, in: Proceedings of 2012 IEEE International Conference on Communications (ICC 2012), IEEE, 2012, pp. 2645–2649.

[37] M. Mangili, F. Martignon, A. Capone, A comparative study of Content-Centric and Content-Distribution Networks: Performance and bounds, in: Proceedings of (GLOBECOM), IEEE, 2013, pp. 1403–1409.

[38] L. Breslau, P. Cao, L. Fan, et al., Web caching and zipf-like distributions: Evidence and implications, in: Proceedings of IEEE INFOCOM, 1999, pp. 126–134.

[39] X. Cheng, C. Dale, J. Liu, Statistics and social network of youtube videos, in: IEEE IWQoS, 2008, pp. 229–238.

[40] A. Anand, V. Sekar, A. Akella, Smartre: an architecture for coordinated network-wide redundancy elimination, ACM SIGCOMM Computer Communication Review, ACM, 2009, pp. 87–98.

[41] K. Cho, M. Lee, K. Park, et al., Wave: popularity-based and collaborative in-network caching for content-oriented networks, Proceedings of IEEE Infocom Workshop on Emerging Design Choices in Name-Oriented Networking, IEEE, 2012, pp. 316–321.

[42] PyOpt. http://www.pyopt.org/, accessed Sept. 22, 2014.

Please cite this article as: W. Chu et al., Network delay guarantee for differentiated services in content-centric networking, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.09.009