Computer Networks 57 (2013) 2465–2482

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

Performance analysis of in-network caching for content-centric networking

Yusung Kim, Ikjun Yeom *

Department of computer engineering, Sung Kyun Kwan University, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Available online 8 April 2013

Keywords:
Content centric networking
In-network caching
Optimal content assignment
Information-centric networking

ABSTRACT

With the explosion of multimedia content, Internet bandwidth is wasted by repeated downloads of popular content. Recently, Content-Centric Networking (CCN), or the so-called Information-Centric Networking (ICN), has been proposed for efficient content delivery. In this paper, we investigate the performance of in-network caching for Named Data Networking (NDN), which is a promising CCN proposal. First, we examine the inefficiency of LRU (Least Recently Used) which is a basic cache replacement policy in NDN. Then we formulate the optimal content assignment for two in-network caching policies. One is *Single-Path Caching*, which allows a request to be served from routers only along the path between a requester and a content source. The other is *Network-Wide Caching*, which enables a request to be served from any router holding the requested content in a network. For both policies, we use a Mixed Integer Program to optimize the content assignment models by considering the link cost, cache size, and content popularity. We also consider the impact of link capacity and routing issues on the optimal content assignment. Our evaluation and analysis present the performance bounds of in-network caching on NDN in terms of the practical constraints, such as the link cost, link capacity, and cache size.

1. Introduction

The Internet was developed based on a host-to-host conversation model in the 1960s and 1970s. Although the Internet Protocol (IP) has been widely deployed for facilitating ubiquitous interconnectivity, the host-to-host conversation model is no longer effective for handling the explosion of multimedia content [1,2]. The majority of current Internet traffic consists of requests and dissemination of content provided by popular websites such as YouTube and Facebook. Internet bandwidth is wasted by repeated downloads of identical content. To minimize waste and to deliver content more efficiently without changing the current Internet architecture, several application layer solutions such as the Content Distribution Network (CDN) and peer-to-peer systems have been deployed.

Recently, novel network architectures have been proposed for efficient content delivery. In these architectures, the Internet needs a fundamental paradigm shift from a traditional host-to-host conversation model to a content-centric communication model. In this model, it is no longer necessary to connect to a server to obtain content. Instead, a user can directly send a request for content to the network with the content name without considering the original content location.

- In DONA [3], content is published on the network by content sources. There is a resolution infrastructure consisting of resolution handlers. A user sends a request with the flat name of the content to a resolution handler. Resolution handlers route the request by name in

* Corresponding author. Tel.: +82 31 299 4595.
E-mail addresses: yskim525@gmail.com (Y. Kim), ijyeom@gmail.com (I. Yeom).

1389-1286/$ - see front matter
http://dx.doi.org/10.1016/j.comnet.2012.11.026

a hierarchical fashion and try to find the copy of the content that is closest to the user. When the request reaches the content source or a node holding a copy, the content is sent back through the reverse path.

- In PSIRP [4], content is published into Rendezvous by the content sources. A user subscribes to the content by a flat name, and the publication and subscription are matched by a rendezvous system. After the matching process, the rendezvous system creates a forwarding identifier and sends it to the content source. The content source starts to send the content with the forwarding identifier to the user. Each PSIRP node routes the content toward the user by using the forwarding identifier.
- In NetInf [5], a hybrid architecture is introduced to support name resolution as well as name-based routing to retrieve the content. Content sources publish their content through a Name Resolution Service (NRS) or by announcing the content in a routing protocol. A user can request content by a flat name from the NRS and the user will then receive a set of available sources. Alternatively, the user can directly send a request with the content name, and it will be forwarded to the source or a node holding a copy of the content.
- In CCN [6], each content has a hierarchical name, and the content is published by a routing protocol. Because of hierarchical naming, name aggregation and longest prefix matching are available in routing. A content request is forwarded toward a content source. If the request reaches a node that is a content source or is a node holding a copy of the content, data is routed back on the reverse path. All nodes on the path store the data, and they can answer subsequent requests for the same content.

In these architectures, storage for caching content is essential. All nodes have storage, and a content request can be satisfied by any node holding a copy of the content in its storage. With this in-network caching mechanism, we can avoid wasting network bandwidth due to the repeated delivery of popular content. From the users' perspective, in addition, we can expect a reduced response time for content by placing the content closer to users.

Among proposals of this type, the Content Centric Network (CCN) [6], now renamed Named Data Networking (NDN) [7], has received wide attention due to its simple and efficient content delivery mechanism. To maximize the benefits of NDN, it is necessary to develop an efficient cache management, scheme since cache size is likely to be very limited in order to contain all of the requested content. In the current literature on NDN [7], a very simple approach, such as Least Recently Used (LRU), is considered for replacement. In this approach, each router stores all of the content that is delivered, and replaces the content according to the LRU order. When the cache is sufficient to maintain a demanded cache hit ratio, this approach can be effective for reducing the bandwidth consumption. When the cache is insufficient, however, this approach frequently involves cache replacement on all routers along the data path. It produces multiple duplicated copies of content in a network, which results in a low cache hit ratio.

In this paper, we investigate the performance of in-network caching on NDN with a cache management scheme. First, we examine the inefficiency of LRU as a cache replacement policy. Then we formulate an optimal content assignment problem for NDN. To maximize the in-network cache hit ratio, it is sufficient to spread content to the network without duplicating. To minimize the average response time, however, a cache hit is insufficient. We have to consider the distance of popular content to be close to the users. Another consideration is routing protocol. If we allow a request to reach a router that is not along the path toward the content source, additional overhead is required in order to determine which router has a copy of the content, and to update the routing information on each router. Hence, we consider two different policies. One is for avoiding additional routing overhead. Content is supposed to be cached only along the path to the sources. We call this policy *Single-Path Caching*. The other policy, called *Network-Wide Caching*, is intended to improve the in-network cache hit ratio with additional routing overhead. Content can be cached at any place in a network. For both policies, we develop optimal content assignment models using a Mixed Integer Program (MIP) as a function of the popularity of the content and the internal and external link costs of a given network. Note that the internal link refers to a link within the network, and an external link refers to a link connected to outside networks, used for requesting content which is not within the network. Using these models, we evaluate and analyze the performance of NDN in terms of the total delivery cost for all requests, the in-network cache hit ratio, and the average response time. We also consider the impacts of the link capacity and routing issues on optimal content assignment. The complexity of MIP is high, and we do not claim here that the presented models are practical. The objective of this paper, rather, is to investigate the performance bounds of NDN with practical constraints, such as the internal and external link costs, cache size, and link capacity.

NDN with in-network caching is a new area, and performance analysis has not been investigated thoroughly. The primary objective of this paper is to provide the performance bounds of NDN. For that, we derive the MIP solution for the optimal content assignment to obtain the best performance with the limited cache size. Even though our MIP solution is theoretical, we believe that it can still be feasible. To implement our solution, we must have information regarding the popularity of all content items. Usually, the popularity of the content is considered to be changing, but recent studies show that the top most popular content is rather long-term [8,9]. Since the amount of long-term popular content is very small compared to the amount of existing content on the Internet, we think that it would be enough to use the long-term popular content in our optimal content assignment in an Autonomous System (AS) network. In Section 6, we present the implementation issues in more detail.

Through this paper, we make the following significant contributions: (a) We examine the limitations of the current approach for cache management in NDN, and address the need for efficient cache management schemes. To find the performance bound of in-network caching, we derive two different MIP models for optimal content assignment: *Single-Path Caching* and *Network-Wide Caching*. (b) We evaluate the performance of in-network caching with our optimal models in terms of the in-network cache hit ratio, the average response time, the average hop count, and the max link load. (c) We analyze the factors to produce the performance differences between *Single-Path Caching* and *Network-Wide Caching*. Practical constraints such as the external link cost and link capacity impact the performance (particularly on *Network-Wide Caching*), and there is a trade-off between improving performance and increasing overhead. (d) We investigate a nearly optimal version of *Network-Wide Caching* by grouping content. The grouping scheme reduces the computation time by two orders of magnitude, and mitigates scalability problems, including the number of requests that are broadcast and the size of the routing table.

The rest of this paper is organized as follows: Section 2 introduces related works. In Section 3, we briefly introduce NDN and describe the content assignment problem. We also present a preliminary evaluation of the current approach for cache management. In Section 4, we derive MIP models of the optimal content assignment for *Single-Path Caching* and *Network-Wide Caching*. In Section 5, we evaluate the performance of NDN based on content assignment using MIP models. In Section 6, we analyze the impact of the content assignment on NDN performance. We provide our conclusion in Section 7.

2. Related works

Caching is a mechanism for providing temporary storage to reduce bandwidth, server load, and response time. The performance of cooperative caching has been studied for several systems such as the World Wide Web, a peer-to-peer storage system, and a network file system. Cooperative caching has been used to improve the performance of a network file system by coordinating caches of multiple clients distributed on a LAN [10]. This differs from our system in the sense that it requires both read and write operations, while our system does not provide a write operation. A peer-to-peer (P2P) global storage system has been presented that replicates multiple copies of a file to improve availability, and supports cache management to minimize access latencies [11]. Cooperative Web caching systems have been analyzed including hierarchical, hash-based, and directory-based caching schemes [12]. Both P2P and Web cooperative caching systems run at end peers or proxies over an IP layer. Instead, we study the optimal performance of Named Data Networking (NDN), which adopts in-network caching in every router. The previous cooperative caching studies are difficult to deploy on NDN due to their application dependency and complexity. NDN requires application-independent, simple cache management to provide line-speed packet forwarding, since all traffic can be cached in routers universally. In this paper, our caching management is similar to the directory-based caching system, while that of the original NDN is similar to the hierarchical caching system.

The Content Distribution Network (CDN) is a globally distributed system that serves content to end users with high availability and performance. CDNs are in charge of serving web objects, media files, and streaming services. For example, *Akamai*, which is the largest CDN, serves about 20% of the web traffic using over 100,000 servers distributed in over 70 countries [13]. The performance of CDNs has been studied from non-commercial CDNs deployed on PlanetLab in [14,15]. CDNs are essentially a massive overlay infrastructure. The service is expensive, and is specific only to contracted applications that have been especially modified in order to use it. Instead, NDN overcomes the limitations of CDN and democratizes the content distribution, and it can run on top of any layer 2 technology or higher.

Packet-level redundant elimination has been shown in a network [16,17]. With redundancy-aware routing algorithms, redundancy elimination approaches can lower the link loads on both intra and inter domain networks. Since the emergence of Information-Centric or Content-Centric network architectures, the concept of in-network caching has received more attention [3–6]. These proposals adopt built-in caching in a network, which is similar to the redundancy elimination approaches, but the fundamental paradigm has shifted from a host-to-host conversation model to a content-centric communication model. The potential caching benefits of deploying content-centric networking were shown using measured BitTorrent traces over a real world topology for its simulation [18]. A prototype of NetInf was implemented to improve the performance of BitTorrent and results showed that the prototype achieved high performance gains in both static and mobile scenarios [19].

In [20–23], the performance of in-network caching is evaluated in simple cascade or tree topologies. More recently, the performance of in-network caching has been described for arbitrary networks in [24–26]. In [27], the impact of topological information is studied for the performance of in-network caching. It shows that the simplest metric, namely the degree centrality, which allocate cache size proportionally to the number of links, is sufficiently good, and that cache size heterogeneity does not significantly impact the performance of in-network caching. In [28], a two-layer caching model is studied for in-network caching when different types of traffic are mixed. In [29], an upstream router recommends the number of chunks to be cached at its downstream router, and the number is exponentially increased as the request count increases. In [30], a caching algorithm has been proposed which exploits the concept of betweenness centrality. Previous researchers evaluated the performance of in-network caching based on a default routing method such as the shortest path toward the content source (we call it *Single-Path Caching*). In this paper, we introduce *Network-Wide Caching* which allows a request to reach any router in a single AS network, if the router holds the corresponding content.

Content placement (or content assignment) using a Mixed Integer Program (MIP) has been studied in several areas, such as CDN [31] and Video on Demand Systems [9]. In [32], MIP is used to optimize the average link hop counts for in-network caching. In our paper, we minimize the total delivery cost as a function of the link cost. We also formulate an MIP model with not only a cache size constraint, but an external link cost and a link capacity constraint within a single AS network. We analyze the considerable difference between the optimization of *Network-Wide Caching* and *Single-Path Caching*, and provide a nearly optimal scheme by grouping content to mitigate scalability problem.

3. Background and motivation

Basically, NDN (Named Data Networking) employs a receiver-driven communication model in which communication is triggered by a request packet for content from a user, and is completed by delivery of the content from the network. In Fig. 1, we illustrate the communication model of NDN. To address content without knowledge of the location of the content source, all content has a unique and hierarchical name. Users who want to obtain content send a request packet (called an interest packet) with the name to the network. Each router has a content cache called Content Store to hold the content if it is forwarded by the router. Upon receiving an interest in content, a router delivers the content directly from its cache if it has a copy of the content. Otherwise, it forwards the interest to the next router, which is determined by a routing table called Forwarding Information Base (FIB). In FIB, the next router information heading to the corresponding content source is stored based on content names (longest prefix match). The interest is forwarded through the path toward the source until it reaches either a router that has the content in its cache or the content source, and then the content is delivered via the reverse path. To avoid repeated delivery of the same interest and the corresponding content, a router maintains a table called the Pending Interest Table (PIT) in which each entry stores the content name in the interest and a set of interfaces from which the interests have been received. When multiple interests for the same content arrive, only the first interest is forwarded toward the content source. When the content arrives, the router finds the matching PIT entry and forwards the content to all of the interfaces listed in the PIT entry.

The initial motivation of NDN is to avoid repeated delivery of identical content. Consequently, we can reduce Internet traffic significantly, and can expect to resolve congestion without additional network resources. Since content can be delivered from a router nearby, the response time is also expected to be reduced. The performance benefit of NDN highly relies on cache management, since the content that is missing from the in-network caches is eventually delivered from outside the network. In the current proposals in [6,7], a simple policy called LRU (Least Recently Used) is considered. In this policy, a cache contains recently used content, and removes the content in the order of the LRU. It is well-known that the LRU policy is effective when the requests are biased to a limited number of content items, and the cache size is sufficiently large to contain the popular content. It is also important that the LRU policy does not require any collaboration with neighbor caches, and works only with local information. These properties make LRU policy well-suited for Internet backbone routers. We can suppose that Internet backbone routers may be equipped with a sufficiently large cache, but it is hard to expect collaborated caching, since each of them can be administrated by different organizations. If we consider a single AS network such as the intranet of an institution or the access domain network of an ISP (Internet Service Provider), however, the LRU policy may not be effective, since applying the LRU policy to each router individually within a network results in multiple cached copies of the same content along the data path. The cache size of routers in these networks is likely to be much smaller than that of the backbone routers, and multiple copies of content degrade the in-network cache hit ratio. To increase the in-network cache hit ratio, we need to consider more effective cache management schemes than LRU for an AS network.

To understand the performance of LRU in more detail, we present a simple simulation on the tree topology shown in Fig. 2. There are 15 nodes (routers) forming a tree with R0 as a root, which is the gateway node in this topology. If the content does not exist in this network, then the interest is forwarded over R0 to reach the content source. R7 $\sim$ R14 are access routers, and interests are issued from the users attached to them. We suppose that there are 10,000 content items, and that they follow a Zipf popularity distribution.

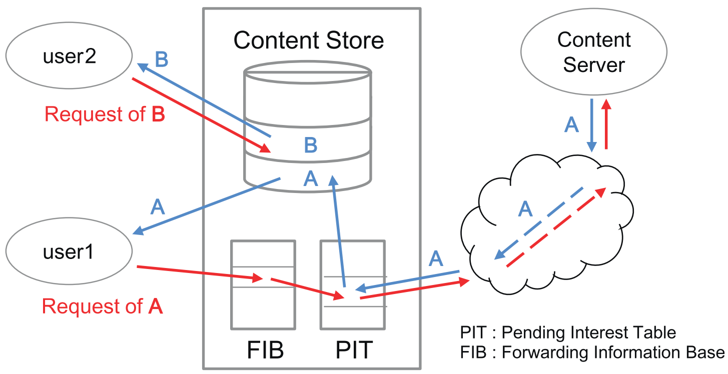


Fig. 1. Operation at an NDN node.

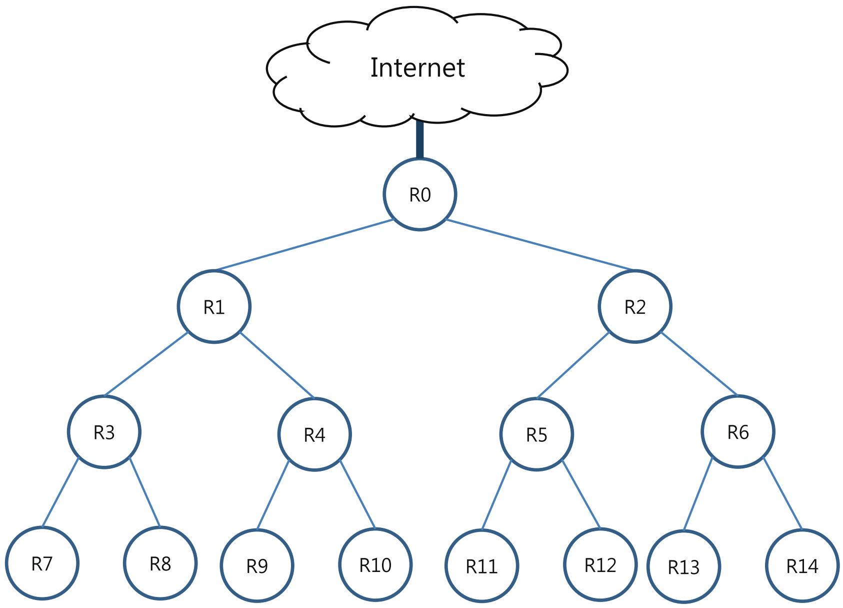


Fig. 2. A simple tree topology with LRU caching.

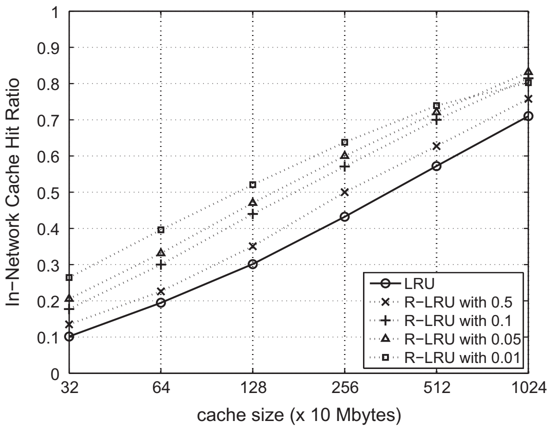


Fig. 3. The LRU cache hit ratio with different cache sizes.

We assume that the size of the content is 10 MBytes, which is the same as the average size of videos on YouTube [33]. The average rate of requests for each access router is 5 per second with a Poisson distribution. With this simulation scenario, we vary the cache size from 32 to 1024, where a unit is one content item, and we measure the in-network cache hit ratio. To investigate the impact of multiple copies of content along the path, we devise a simple variant of the LRU policy called random LRU. In the original LRU policy, whenever a router delivers new content that was not in its cache, it replaces a least recently used content with the new content. When the content is not in the network, it is delivered from its source. As a consequence, all of the routers on the path cache the content. In random LRU, each router caches the new content randomly with a given probability to reduce duplicated caching along the path. We vary the probability from 1% to 50%, and compare the results with the original LRU policy. The results are presented in Fig. 3.

As expected, when the cache size is small, the original LRU has a very low in-network cache hit ratio. The hit ratio increases as we decrease the replacement probability when the cache size is not sufficiently large. The hit ratio with a cache size of 32 is just 0.1, while the hit ratio of random LRU with the probability of 1% is around 0.27. This ratio is even higher than that of the original LRU with a cache size of 64. It is clear that frequent replacement produces duplicated copies of content, and this is harmful when the cache size is small. Here note that when the cache size is sufficient (for example, 1024), the hit ratio of random LRU becomes lower than that of random LRU with a probability of 10%. Slow replacement with a large cache may fail to maintain the cache with proper content for our simulation time. The in-network cache hit ratio may be impacted by other factors, such as the content popularity function, the amount of content, and the topology of the network. It is obvious that LRU is not the best choice for an AS network.

4. Optimization model for in-network caching

In this paper, we observe the impact of optimal content assignment in NDN. Rather than propose a new cache management scheme, we study the optimal content assignment with practical constraints, such as the cache size, internal and external link costs, and link capacity. For content assignment, we consider two policies for the routing path. The first policy is that the content is delivered only along the shortest path between users to the content sources. With this policy, a network can be modeled as a tree as we stated earlier, and most prior works on NDN use this policy.[1] We call this policy *Single-Path Caching*, and evaluate the performance of an NDN with the optimal content assignment using this policy. However, if we focus on an AS network, it can be effective to try to find content within the entire AS network, not only along the shortest path, especially when the external link cost is high. We call this policy *Network-Wide Caching*.

We formulate the mixed integer program (MIP) to find the optimal content assignment for both *Single-Path Caching* and *Network-Wide Caching*. We assume that the content popularity and the request rates are given. We consider that the content is to be a complete file consisting of a bundle of packets for convenience, but it can be applied to a chunk-level or packet-level approach. Our goal is to find the optimal content assignment that minimizes the total delivery cost while satisfying all content requests within the link capacity and cache size constraints.

4.1. Input parameters

Table 1 summarizes the parameters and their meanings. Let V denote the set of nodes, including both the routers and original (outside) servers. In this model, we use one node to represent the outside servers. L is the set of directed links between theses nodes, and C is the set of content items. $R\ (c, v)$ is the request rate of content $c \in C$ generated at an access router $v \in V$. Each router $v \in V$ has a cache size CS (v) and each link $l \in L$ has a link capacity of LC (l). For v, $w \in V$, PV (v,w) is the set of nodes and PL (v,w) is the set of links on the path from v to w. For $c \in C$, Size (c) is its size and Src (c) is its original (outside) server. Cost (c,v,w) is the sum of the link costs for $v \in V$ to get $c \in C$ from $w \in V$.

Table 1
Input parameters and decision variables in MIP.

	Meaning
Parameter	
V	Set of nodes (routers and outside servers)
L	Set of directed links
C	Set of content items
$R\ (c, v)$	Request rate of $c \in C$ at $v \in V$
CS (v)	Cache size of $v \in V$
LC (l)	Link capacity of l ∈ L
PV (v,w)	Set of nodes on path from $v \in V$ to $w \in V$
PL (v,w)	Set of links on path from $v \in V$ to $w \in V$
Src (c)	An original (outside) server $\in V$ for $c \in C$
Size (c)	Size of $c \in C$
Cost (c, v,w)	Sum of link costs for $v \in V$ to get $c \in C$ from $w \in V$
Decision variable	
$M\ (c, v)$	Binary variable indicating whether to store $c \in C$ at $v \in V$
$Y\ (c, v, w)$	Binary variable indicating whether it is the minimum cost for $v \in V$ to get $c \in C$ from $w \in V$

4.2. MIP model

We present two MIP models for *Single-Path Caching* and *Network-Wide Caching*. They have common variables: M (c,v) is a binary variable indicating whether to store $c \in C$ at $v \in V$. If c is cached at v, $M\ (c,v) = 1$; otherwise, $M\ (c,v) = 0$. $Y\ (c,v,w)$ is a binary variable indicating whether it is the minimum cost for $v \in V$ to get $c \in C$ from $w \in V$. For both policies, we use the content assignment to minimize the total delivery cost within the link capacity and cache size constraints.

4.2.1. MIP model for single-path caching

We first present the MIP model for the *Single-Path Caching* policy. We assume that there is a regular path (based on the shortest path routing) between an access router and an outside server. With this routing policy, (1) describes our objective function to minimize the total delivery cost for all requests. (2) Ensures the cache size limitation at each node. (3) Denotes that $w \in V$ can deliver $c \in C$ to $v \in V$ only when it has stored the content in its cache where $w \in V$ is on the path from Src (c) to $v \in V$. (4) Assures that every request can be served by at least one node. In (5) we define the variable SP (c,v) to satisfy the minimum cost for the request of $c \in C$ at $v \in V$ while satisfying (2)–(4). By applying (5) to (1), we get the final objective function in (6).

Additionally, we can use the link capacity constraint shown in (7). This ensures that each link usage is limited by LC (l), $\forall l \in L$ for all requests.

$$min\sum_{v\in V}\sum_{c\in C}Cost(c,v,w)\times R(v,c)\times Size(c), \forall w\in PV(v,Src(c)) \tag{1}$$

$$\sum_{c\in C}M(c,v)\times Size(c)\leqslant CS(v), \forall v\in V \tag{2}$$

$$M(w,c)\geqslant Y(c,v,w)\forall c\in C, \forall v,w\in PV(v,Src(c)) \tag{3}$$

$$\sum_{v\in V}Y(c,v,w)=1, \forall c\in C, w\in PV(v,Src(c)) \tag{4}$$

$$SP(c,v)>=Y(c,v,w)\times Cost(c,v,w),\ \forall c\in C, v\in V, w\in PV(v,Src(c))$$

$$min\sum_{v\in V}\sum_{c\in C}SP(c,v)\times R(c,v)\times Size(c) \tag{6}$$

$$\sum_{c\in C}\sum_{l\in PL(v,w)}Y(c,v,w)\times R(v,c)\leqslant LC(l), \forall v\in V, w\in PV(v,Src(c)) \tag{7}$$

4.2.2. MIP model for network-wide caching

The MIP model for *Network-Wide Caching* is similar to that of *Single-Path Caching*. The only difference is a routing part for the content request and response. It allows a request to reach any router in a network if the router holds

[1] Actual network topologies are not like a tree. In a basic NDN, however, interest forwarding and content delivery are assumed to be handled over the shortest path between a user and a content source, and the network can be simplified as a tree. Most related literatures in [20–23] make a similar assumption. Later, we will present how to utilize links that are not on a tree to improve the performance of an NDN.

the corresponding content. With this policy, (8) minimizes the total delivery cost when $w \in V$ receives $c \in C$ from $w \in V$ which can be any router in the network. (9) denotes a cache size limitation, and (10) and (11) ensure that every request can be served by at least one node. In (12) we define the variable NW (c, v) to satisfy the minimum cost for the request of $c \in C$ at $v \in V$ while satisfying (9)–(11). Using (8) and (12), we obtain the final objective function in (13).

Contrary to *Single-Path Caching*, *Network-Wide Caching* could increase intra network traffic because it receives content from not only routers on the shortest path, but from any routers in the network. If the intra-traffic converges on some links, they can be overloaded. Using (7), we can provide optimal content assignment with a link capacity constraint.

$$min\sum_{v\in V}\sum_{c\in C}Cost(c,v,w)\times R(v,c)\times Size(c), \forall w\in V \quad (8)$$

$$\sum_{c\in C}M(c,v)\times Size(c)\leqslant CS(v), \forall v\in V \quad (9)$$

$$M(c,w)\geqslant Y(c,v,w)\forall c\in C, \forall v,w\in V \quad (10)$$

$$\sum_{v\in V}Y(c,v,w)=1, \forall c\in C, w\in V \quad (11)$$

$$NW(c,w) >= Y(c,v,w)\times Cost(c,v,w), \forall c\in C, v,w\in V \quad (12)$$

$$min\sum_{v\in V}\sum_{c\in C}NW(c,v)\times R(c,v)\times Size(c) \quad (13)$$

$$\sum_{c\in C}\sum_{l\in PL(v,w)}Y(c,v,w)\times R(c,v)\leqslant LC(l), \forall v,w\in V \quad (14)$$

In our models, the popularity of content is the main consideration when computing the optimal solution. In our models, $R\ (v,c)$ is the request rate of content $c \in C$ generated at a node $v \in V$ and is calculated by multiplying the request probability of $c \in C$ and the total request rate generated at $v \in V$. We may add additional considerations based on the content in the models. For example, the concept of content weight can be introduced to support the priority of content or differentiation of services. We believe that this is easily applicable to our models as shown in (15), where *Prob* (c) is the request probability of content $c \in C$, *Weight* (c) is the content's weight, and *All_Req* (v) is the request rate at a node $v \in V$.

$$R(c,v)=Porb(c)\times Weight(c)\times All\_Req(v), \forall c\in C, \forall v\in V \quad (15)$$

4.3. Group based MIP model

Finding an optimal solution using MIP is NP-hard. If we consider a link capacity constraint, the problem becomes complex due to the high computation time as the amount of content, and the numbers of nodes, and links increase. We designed an algorithm grouping content to reduce the number of constraints in MIP. In our simulation, there are 10,000 content items in order of popularity. We divide the content items into multiple groups and use the groups in (1)–(14) instead of the individual content item. After solving the optimal content assignment, we assign content to the routers at the group level. In Section 6, we describe the reduction in the computation time and describe the near-optimal performance as compared to the original MIP model.

5. Performance evaluation

5.1. Simulation configuration

We perform a simulation in the arbitrary network topology shown in Fig. 4a. R0 is the root, which is the gateway node in this topology. If content does not exist in this network, then the request is forwarded over R0 to outside the network. R7 $\sim$ R14 are access routers, and requests are issued from the users attached to them. All of the network links are uni-directional, and there are two-way links between two nodes. There is an external link between R0 and the outside networks. The internal one-way latency ranges between 1 and 5 ms at random. The external one-way latency is different for each content. It is randomly set to range from 5 to 150 ms. In a basic NDN, request forwarding and content delivery are assumed to be the shortest path between a user and a content source. The network topology can be simplified as the shortest path tree in Fig. 4b.

We define the link cost for our MIP models. The internal link cost is set to be 1, and the external link cost is set to the average link latency of all content items. To evaluate the impact of the external link cost, we vary the external link cost from 1 to 256. We also evaluate the performance of in-network caching with both unlimited and limited link capacity. In the simulation, each access router receives content requests at the rate of 5 per second with a Poisson distribution, and 10,000 content items are partitioned in K classes in a Zipf popularity distribution where K is 200. The content items of class k are requested with probability $q_k = \frac{c}{k^\alpha},\ k \in K$ where α = 1.15 and $c = 1/\left(\sum_k^K \frac{1}{k^\alpha}\right)$. This configuration means that 80% of requests are issued for the top (most popular) 20% of the content. All routers (R0 $\sim$ R14) have a cache and we vary the cache size from 32 to 1024 where a unit is one content item. The size of content is 10 MBytes which is same to the average size of videos in YouTube [33]. Therefore 1024 of cache size is about to 10 GBytes. We compare three different cache replacement policies. The first policy is LRU, which replaces new content with the least recently used content. The second policy is OPT_SP, which is optimal for *Single-Path Caching*, where each router has content assigned by the MIP solution of (6). The last policy is OPT_NW, which is optimal for *Network-Wide Caching*, where each router keeps content assigned by the MIP solution of (13). We finish each simulation after a million content requests are generated. All of the simulation parameters are summarized in Table 2.

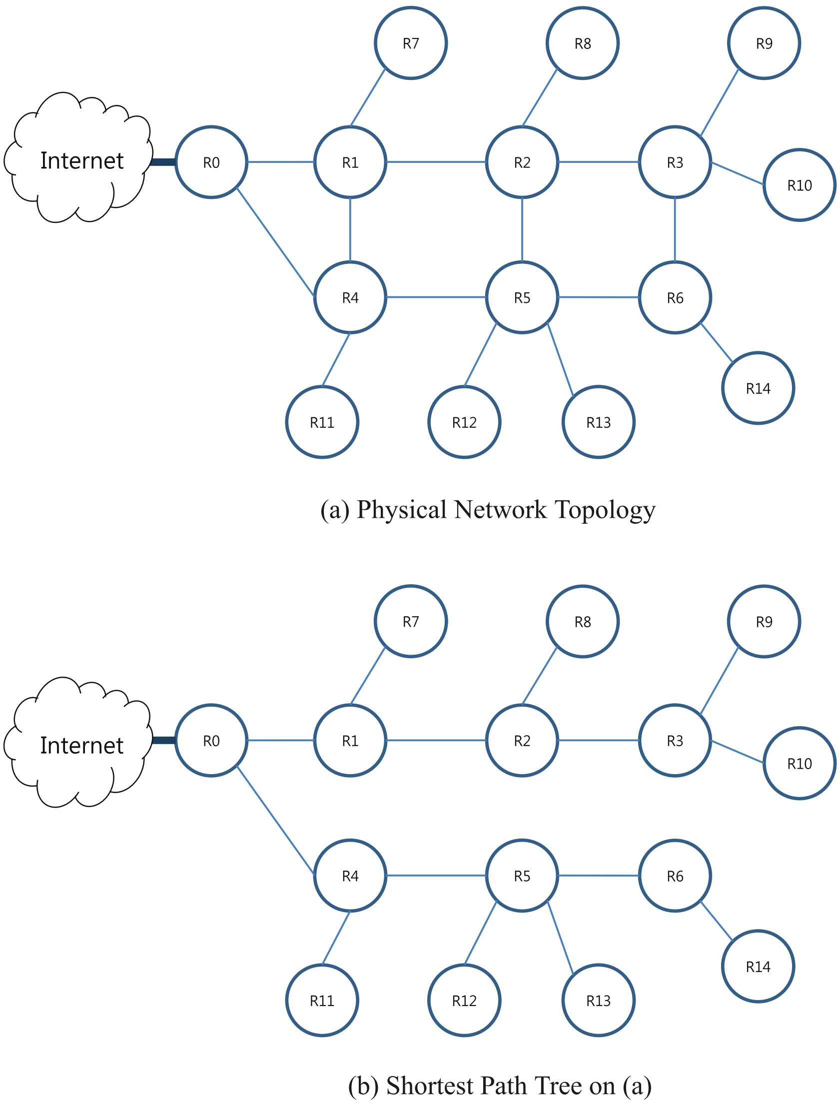


Fig. 4. Arbitrary topology: S represents outside servers and R0 ~ R14 are routers in a network.

Table 2
Simulation parameters.

Parameter	Value
Internal link latency	For each internal link, randomly assigned in 1–5 ms
External link latency	For each content, randomly assigned in 5–150 ms
Internal link cost	1
External link cost	Average external link latency (about 78) (option) Ranging from 1 to 256
Link capacity constraint	Unlimited request rate (option) Minimize the maximum request rate
Cache size	256 ($\times$10 MBytes) (option) Ranging from 32 to 1024
Request rate	Five requests per second at edge router
Number of content items	10,000 in a Zipf distribution
Total number of content requests	A million

5.2. Performance metrics

We introduce four performance metrics: (1) the in-network cache hit ratio; (2) the average response time; (3) the average hop count; and (4) the max link load. The in-network cache hit ratio refers to the number of cache hits over the number of requests in a single AS network. The average response time is the average time that a user waits to receive the requested content after sending a request. The average hop count is the average number of hops from a user to a node holding the requested content. In this evaluation, we count the number of hops within the network, excluding hops over outside networks. The max link load is the highest load among all of the internal link loads during the simulation. The load is defined as the rate of sending content (the number of content deliveries per second) on a link.

5.3. Impact of cache size

Fig. 5 shows the impact of the cache size from 32 to 1024, where other simulation parameters are set by default in Table 2. The in-network cache hit ratio and the average response time of LRU are the worst, and the two metrics of OPT_NW are better than those of OPT_SP for all cache sizes. The average hop count of OPT_SP is the lowest, while that of OPT_NW differs according to the cache size as shown in Fig. 5c. Since OPT_NW allows a request to reach any router in a network, the average hop count may increase. For example, a request generated under R7 may be served by R14 Fig. 4a. However, the average hop count of OPT_NW sharply decreases if the cache size is sufficient (more than 256), because it can find most popular content in the nearby routers. Fig. 5d shows the max link load of each policy. LRU and OPT_SP have the max load at a link between R0 and R1, or a link between R0 and R4. The two bottleneck links occur because content is delivered via R0 based on the shortest path rule. In OPT_NW, the bottleneck links can be different depending on the status of content assignment. Because a lot of content is served by routers that are not on the shortest paths, intra network traffic can have greater volume than the incoming network traffic via R0. When the cache size is sufficient (more than 256), the max link load decreases because the intra network traffic decreases and the average hop count decreases.

In Fig. 6, we take a close look at the average response times for the most popular content. Fig. 6a shows the average response time for the top 10% of content (the most popular 1000 items). The average response time of OPT_NW is the lowest, and it seems to be similar to Fig. 5b. However, we can see a different result in Fig. 6b, which shows the average response time for the top 2.5% of content (the most popular 250 content items). The average response time of LRU is always the highest, because the top 250 content items are not guaranteed to be stored in the shortest path from a user to a content source, and the missing content is retrieved from outside of the network. OPT_SP assigns content to routers based on the MIP solution as in (6). If the aggregate cache size along the shortest path is less than 250, the top 250 content items cannot be served in the path. The average response time of OPT_SP is around 80 ms when

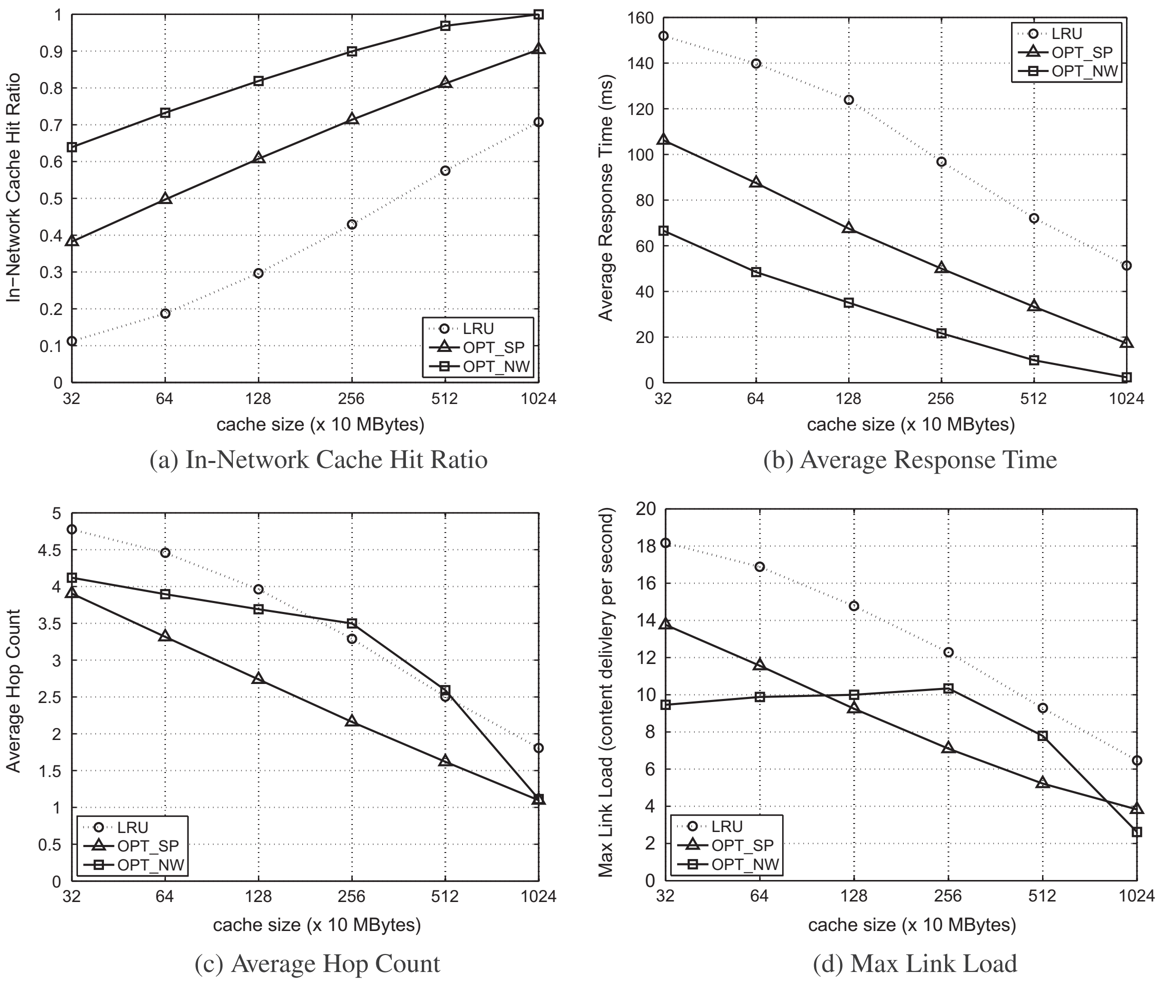
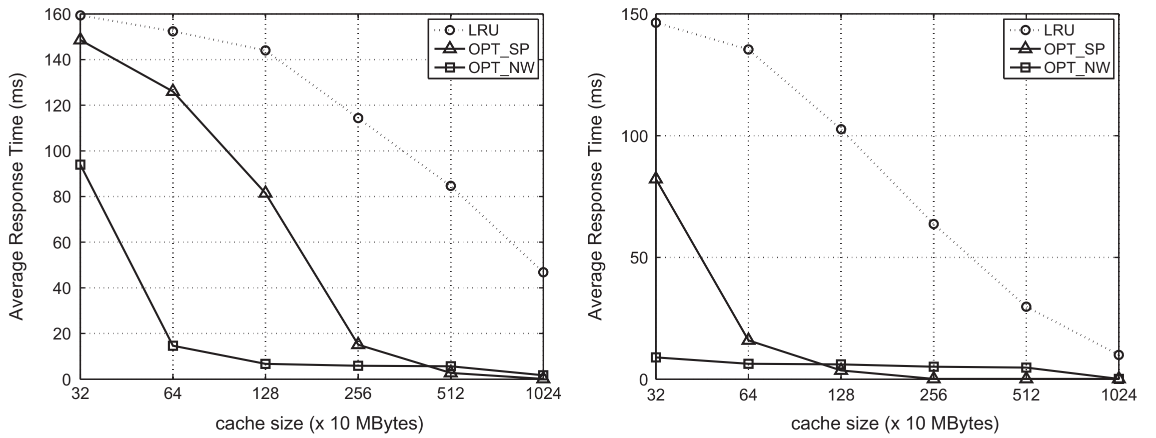


Fig. 5. Impact of cache size ranging from 32 to 1024.



(a) Average Response Time of Top 10% content (b) Average Response Time of Top 2.5% content

Fig. 6. Average response time of top content over cache size ranging from 32 to 1024.

the cache size is 32, while the response time decreases as the cache size increases. OPT_NW allows a request to be served by any router in a network regardless of the shortest path. Even though, the aggregate cache size along the shortest path is less than 250, the top 250 content items can be hit in the network. In Fig. 6b, the average response time of OPT_NW is around 5 ms when the cache size is only 32. However, the average response time of OPT_NW is still 5 ms when the cache size is sufficiently large, such as 512, while that of OPT_SP is close to 0 ms (0.1 ms hit by an access router). This is because OPT_SP tends to assign the most popular content to all of the access routers redundantly, while OPT_NW may assign the most popular content to an intermediate router (or routers) to satisfy the MIP solution (13). Therefore OPT_SP can provide the lowest response time for the most popular content if the cache size is sufficient. However, OPT_NW can provide a lower average response time than OPT_SP for all content requests.

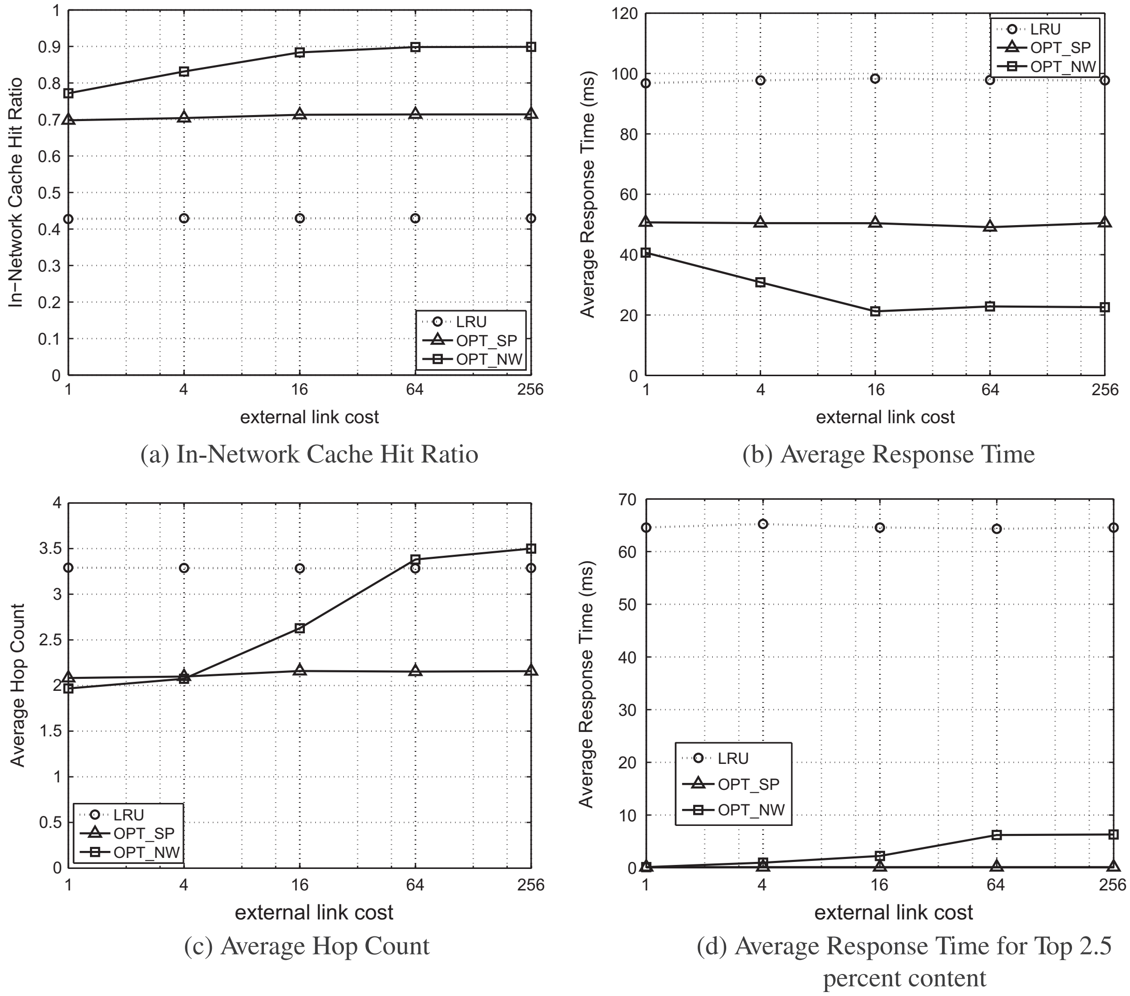
5.4. Impact of external link cost

Fig. 7 shows the impact of an external link cost when the cache size is fixed at 256. We vary the external link cost from 1 to 256. The link cost is used only in MIP models for OPT_SP and OPT_NW. Even though LRU is not impacted by the external link cost, for comparison, we put the LRU together in Fig. 7. In the result, the external link cost does not greatly impact the performance of OPT_SP. On the other hand, the performance of OPT_NW differs according to the external link cost. When the external link cost is set to 1, the performance of OPT_NW is similar to that of OPT_SP, but the performance of OPT_NW improves as the external link cost increases. The in-network cache hit ratio increases by up to 20%, and the average response time decreases by about 50%. However the average hop count of OPT_NW continuously increases as the external link cost increases. In Fig. 7d, we compare the average response time for the 250 most popular content items. The average response time of OPT_SP is close to 0 ms, regardless of the external link cost. However, the average response time of OPT_NW increases up to 8 ms as the external link cost increases. As the external link cost increases, OPT_NW tends to store more diverse content in the network to reduce usages of the external link as well as to minimize the MIP solution (13). Putting more content in the network increases the in-network cache hit ratio by reducing the content redundancy. Conversely, copies of very popular content are moved from access routers to a few intermediate routers, while relatively unpopular content can be stored in the edge side routers. Because of the changing status of content assignment, the average response time and the average hop count increase as the external link cost increases. We analyze these results in Section 6 in detail.

5.5. Impact of link capacity constraint

We evaluate the link capacity constraint introduced in (7) and (14). In this simulation, the link capacity is defined as the rate of content delivery (content deliveries per second). If the link capacity constraint is given, the internal links cannot exceed the constraint. First we solve for the optimal content assignment with unlimited link capacity, and obtain the max link load. Next we set the link capacity constraint to be less than the current max link load, and solve for the optimal content assignment again. This is repeated until the MIP solution is infeasible. From this process, we can obtain the minimum link capacity constraint as well as the optimal content assignment when feasible. We define SP_LCC as the optimal content assignment of *Single-Path Caching* with the minimum link capacity constraint. NW_LCC is also the optimal content assignment of *Network-Wide Caching* with the minimum link capacity constraint.

Fig. 8 shows that the link capacity constraint does not have much impact on the in-network cache hit ratio, the average response time, and the average hop count. However, the max link load of NW_LCC is significantly reduced compared to that of OPT_NW, while that of SP_LCC is similar to that of OPT_SP. In OPT_NW, the most popular content is



(a) In-Network Cache Hit Ratio

(b) Average Response Time

(c) Average Hop Count

(d) Average Response Time for Top 2.5 percent content

Fig. 7. Impact of external link cost ranging from 1 to 256 where cache size is 256.

stored at only a few routers on one side, which results in a high max link load on the links adjacent to the routers. By adopting the link capacity constraint, NW_LCC can distribute the most popular content into several different routers. The max link load of NW_LCC is reduced by at most half to maintain the in-network cache hit ratio, the average response time, and the average hop count.

In Fig. 9, we examine all internal link loads in detail with the link capacity constraint. When the cache size is fixed at 256 and the link capacity constraint is set to the minimum as described above, we plot all internal link loads in both directions. To define the direction, one direction is from a router with a smaller number to another router with a higher number (for example, from R0 to R1) in Fig. 4a. The opposite direction is the reverse of the previous one (for example, from R1 to R0).

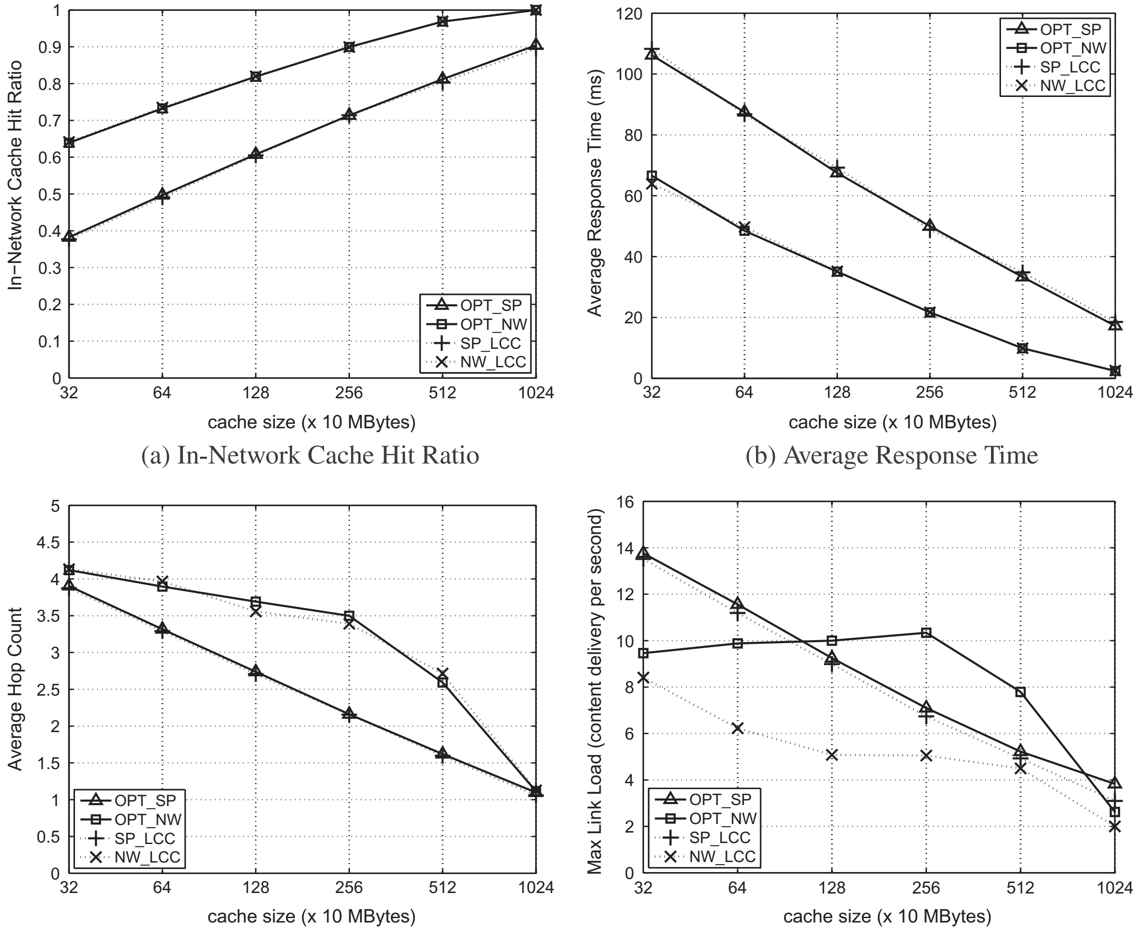
In Fig. 9a, the one direction link loads of SP_LCC are similar to those of OPT_SP, while the max link load of NW_LCC is reduced by half. Other link loads also decrease slightly compared with those of OPT_NW.

Another notable difference between *Single-Path Caching* and *Network-Wide Caching* is existence of traffic in the opposite direction as shown in Fig. 9b. OPT_SP and SP_LCC (including LRU) do not use links in the opposite direction because they only download data from the uplink router on the shortest path. However, OPT_NW and NW_LCC utilize links in both directions to deliver content. In Fig. 9b, Link loads of the first and second links of NW_LCC are smaller than those of OPT_NW, but the loads of the next six links of NW_LCC are higher than those of OPT_NW. This means that NW_LCC distributes link loads by assigning content to fit the link capacity constraint.

5.6. Impact of content popularity distribution

In content popularity distribution,the α parameter plays an important role in determining the performance of in-network caching. Fig. 10a shows the impact of the α parameter. The x-axis is the value of the α ranging from 1 to 2. The y-axis is the request ratio issued for the most popular 20% of the content. For example, 73% of requests are issued for the top 20% of the content, if α is set to 1, and 99% of requests are issued, if α is set to 2.

For all values of α, OPT_SP outperforms LRU but the performance difference decreases as α increases. When α is 2, LRU also has a 90% in-network cache hit ratio. OPT_NW



(a) In-Network Cache Hit Ratio (b) Average Response Time

(c) Average Hop Count (d) Max Link Load

Fig. 8. Impact of link capacity constraint where the cache size ranges from 32 to 1024.

always outperforms OPT_SP in terms of the in-network cache hit ratio and average response time, while the average hop count is slightly higher than that of OPT_SP due to increasing internal network traffic as described in the previous section.

6. Analysis

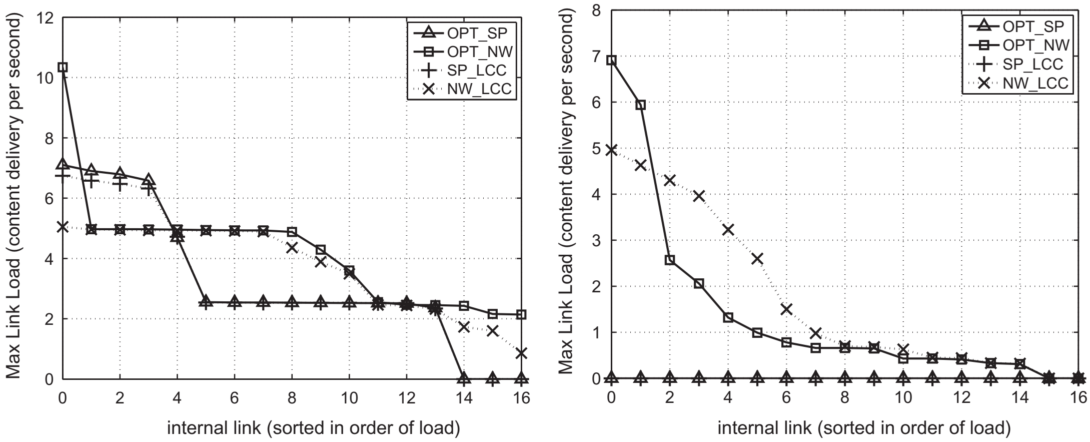
6.1. Impact of content type

Content can be classified as (1) static or dynamic; and (2) short or long-term. In this paper, we introduce the optimal content assignment for in-network caching in a single AS network, instead of ubiquitous caching at all routers on the data path. Our strategically selective caching is appropriate for static and long-term content. Recently, on-demand video streaming services have occupied the majority of Internet traffic [1,2]. These videos will remain unmodified for a period of time. Other static content such as file sharing, email, and software updates is also applicable in our solution. Even though the cache size of each router is extremely small compared to the amount of Internet content, according to the Pareto Principle, caching can improve the hit ratios with the small amount of content. For example, by storing only 10% of long-term popular videos in YouTube, a cache can serve 80% of the requests [8].

Dynamic content such as dynamic web pages needs to be retrieved from a content source, rather than an intermediate router. In real-time services such as VoIP, on-line gaming, and TV/Radio live broadcasting, content is not modified, but is usually short-term. After closing the service sessions, the content will not be requested by other users. Therefore both dynamic and short-term content are not included in our optimal content assignment, but can be handled by a basic LRU policy (storing content at all routers on the data path).

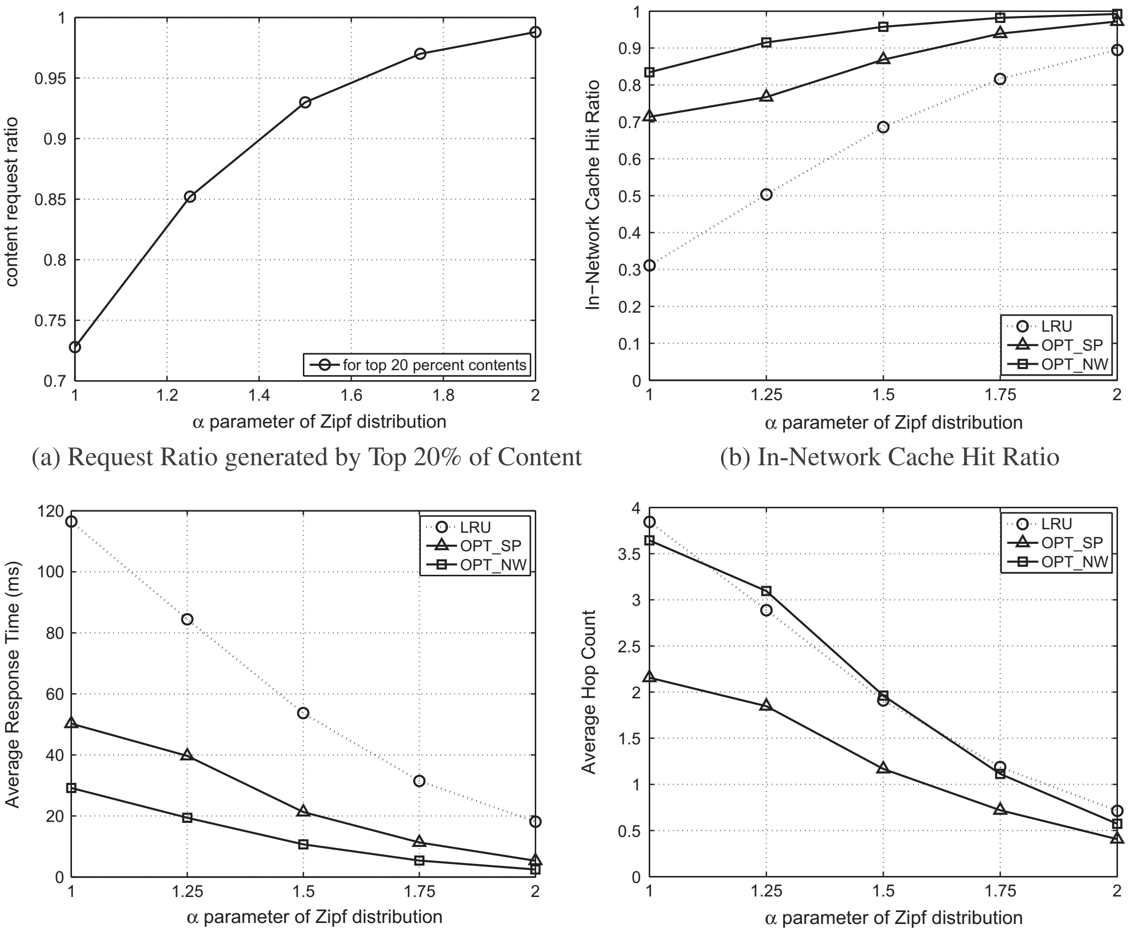
According to content types, we use two different cache management policies. Our optimal content assignment is for long-term popular content, while LRU is used for unpopular, dynamic, and short-term content. To do this, we divide the Content Store into two types. One is the managed Content Store for our optimal content assignment, and the other is the unmanaged Content Store for a basic LRU policy, which stores content in all routers along a data path.

To select long-term popular content for our scheme, we need to know the information of the content, but it is not



(a) Each Link Load in one direction (b) Each Link Load in opposite direction

Fig. 9. Two-way link loads over link capacity constraint where the cache size is 256.

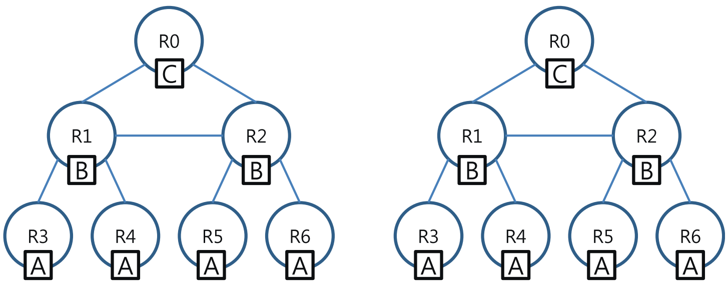


(a) Request Ratio generated by Top 20% of Content (b) In-Network Cache Hit Ratio

(c) Average Response Time (d) Average Hop Count

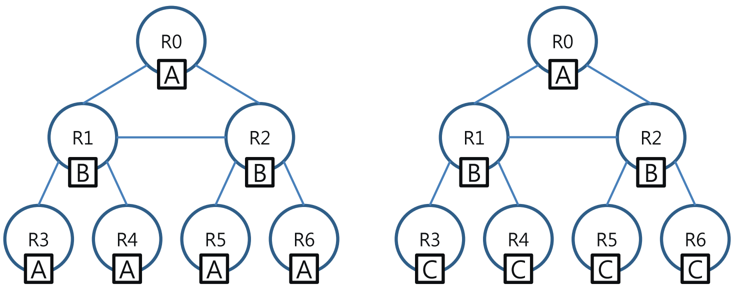
Fig. 10. Impact of content popularity distribution; α ranges from 1 to 2 where the cache size is 256 and the external link cost is 100.

known a priori. To obtain this information, we may use the recent history (e.g. the past 24 h) [9]. Because we consider long-term content, the recent history is still available to the current optimal content assignment. Another is a simple estimation based on the observation of regular programs such as TV series. There is a considerable



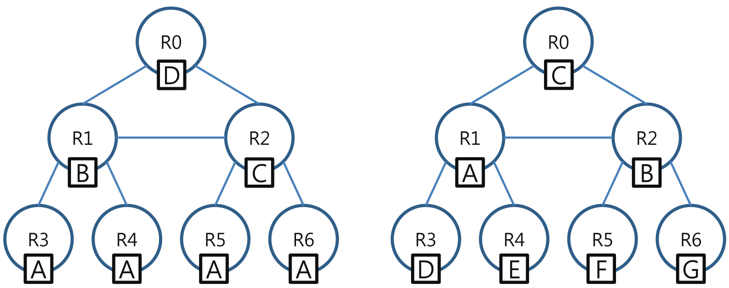
(a) the external link cost is 1 (b) the external link cost is 10

Fig. 11. Single-path caching where cache size is 1 and requests are generated at bottom routers.



(a) the external link cost is 1 (b) the external link cost is 10

Fig. 12. Single-path caching where cache size is 1 and requests are generated at all routers.



(a) the external link cost is 1 (b) the external link cost is 10

Fig. 13. Network-wide caching where cache size is 1 and requests are generated at bottom routers or all routers.

similarity in the request patterns for each episode of the series [9]. In [34,35], there are some studies that estimate user demand, but they are beyond the scope of this paper.

6.2. Single-path caching vs. network-wide caching

In our simulations, we evaluated the performance of the optimal content assignment for both *Single-Path Caching* and *Network-Wide Caching*. The simulation results show that there is a considerable difference between the two policies. To easily understand the difference, we present a small topology with a cache size of 1 in Figs. 11–13. We assume that an external link is connected to a root router R0, and its link cost is set to be 1 or 10, while all internal link costs are fixed at 1. The content is titled *A* to *Z* in order of popularity (where *A* is the most popular).

Fig. 11 shows the caching status of OPT_SP, where all requests are generated at the bottom routers. The external link cost is different on each diagram (1 for left and 10 for right), but the caching status is the same regardless of the external link cost. This is because there is no better solution, even though the external link cost changes.

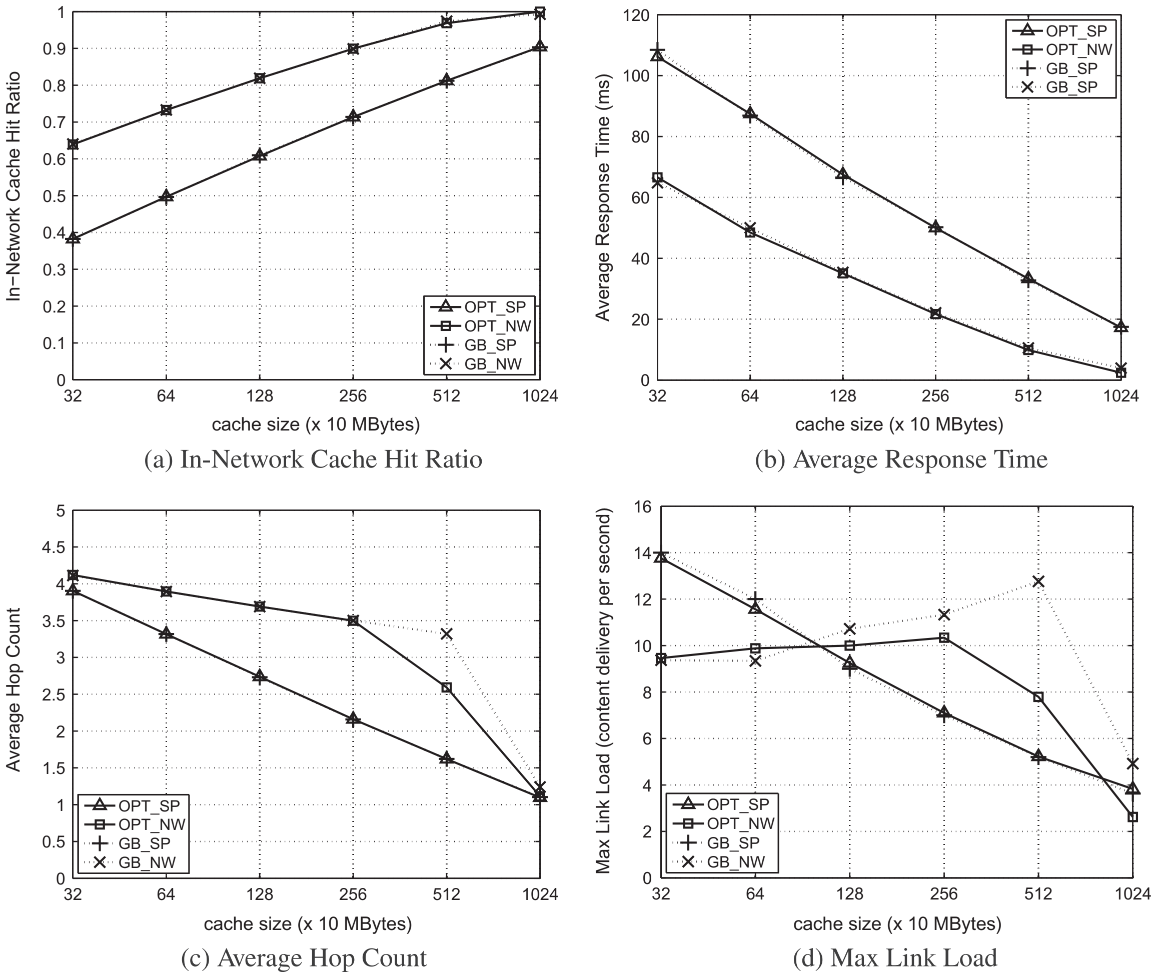
However, if we generate requests at all routers in the topology, the results differ. In Fig. 12a, the external link cost is 1, and most routers try to have the most popular content, *A*, and to obtain other content from outside the network. R1 and R2 store content *B* because they can get content *A* from a nearby router R0. In Fig. 12b, the external link cost is 10, and the bottom routers store content C because they want to reduce the external link usages as much as possible. They obtain content A, B, and C within the network because the delivery cost is less than the cost of obtaining it from outside the network.

OPT_NW allows a request to reach any router in a network if any router has the requested content. In Fig. 13a, where requests are generated at the bottom routers and the external cost is 1, OPT_NW has more content within the network than OPT_SP in Fig. 11a. R3 and R4 obtain content C from R2, even though R2 is not on the shortest path. Of course, R5 and R6 obtain content B from R1. When the external link cost is 10, OPT_NW demonstrates that the most popular content A, B, and C are located in R1, R2, and R0, while relatively unpopular content ($D \sim G$) are stored at the access routers (R3 $\sim$ R6) in Fig. 13b. This is because OPT_NW enables a request to obtain its content from any router in the network. The different caching statuses with OPT_SP results in differences in performance, such as the in-network cache hit ratio, the average response time, the average hop count, and the max link load. In this simple topology, OPT_NW has the same result when the requests are generated in either the bottom routers or all routers. However, in our extensive simulations, we verified that it is also changed in a different network topology.

From this analysis, we determine two facts. The first is that OPT_SP and OPT_NW can be influenced by the external link cost according to the location of the request creation in a topology. The second is that OPT_NW is impacted more strongly than OPT_SP, because OPT_NW allows a request to be served from any router irrespective of the shortest path toward a content source. This results in improving the in-network cache hit ratio, but it increases the internal network traffic.

6.3. Near-optimization by grouping content

It is already known that MIP is NP hard, and our optimization solution has a scalability issue. To reduce computing time, we provide a near-optimal solution by grouping content. We divide content items into multiple groups and solve MIP models with the groups instead of individual content. We then conduct the group-level content assignment. This grouping scheme is effective in reducing computing time. We define GB_SP and GB_NW as the group-based versions of OPT_SP and OPT_NW. First, we



(a) In-Network Cache Hit Ratio

(b) Average Response Time

(c) Average Hop Count

(d) Max Link Load

Fig. 14. Group-based optimization when each group size is the same as the cache size ranges from 32 to 1024.

set the size of each group to the same cache size. Fig. 14 shows that there is no difference in terms of the in-network cache hit ratio and the average response time. However, the average hop count and the max link load are affected as the cache size (which is the same as the group size) increases. Because our MIP model minimizes the total delivery cost, it does not preferentially care about the max link load or average hop count. Even though the total delivery cost is the same (or almost the same), the max link load and the average hop count can be different depending on the content assignment. For example, when the cache size is 512, GB_NW has a much higher max link load than OPT_NW because the most popular content (512 content items, according to the group size) is assigned to one side. It also causes an increase in the average hop count. However, the most popular content is located in a few routers redundantly when the cache size is 1024. Therefore, both the max link load and the average hop count can be reduced.

With a cache size of 512, we can use the link capacity constraint to rearrange the location of popular content. The 512 most popular content items are moved to the middle of the network rather than to its corner. Another alternative solution is to reduce group size (for example, 32 content items in each group). It is easier to distribute popular content items in a network with a smaller group size. We verified that both solutions are as effective in reducing the max link load and the average hop count as OPT_NW. Both solutions, of course, showed that their in-network cache hit ratio and average request response time were almost the same as those of OPT_NW.

Finally, we compare the computation times of our original schemes and grouping schemes. Our original schemes require hundred of seconds to obtain the optimal solution. With the grouping schemes, we obtain a near-optimal solution that is two orders of magnitude faster, as shown in Table 3 when the group size is fixed at 32.

6.4. Routing issue in Network-Wide Caching

For the optimal content assignment of *Network-Wide Caching*, we need to consider a routing scheme. *Network-Wide Caching* needs to route a request to reach any router in a network, if the router has the requested content. Even though we do not claim that our Network-Wide Caching is a practical solution in NDN, we consider the routing scheme based on two simple approaches.

In the first approach, each router broadcasts a request and waits for the nearest cache to respond first. After this process, all of routers on the data path learn the routing information of the content and add a routing table entry for the content to their FIB if needed. We select popular content, that is static and long-term, for optimal assignment. The number of content items depends on the aggregate cache size in an AS network. We assume that each router knows the list of selected content by using a Bloom filter [36]. When a router receives a request, it first checks to see whether the requested content is in the list of selected content. If the content is in the list, the router tries to broadcast it. Otherwise, the longest prefix matching is used as a default. Once the optimal content assignment is computed, the placement of the content will be constant for a certain time. Therefore the request broadcast process is required for the first time after changing the optimal content assignment.

This approach has two scalability issues. One is request flooding over the network, as we mentioned. The other is increasing the number of routing table entries. As the number of content items assigned in the network increases, the size of the routing table increases. The number of additional routing table entries is almost linearly proportional to the cache size when the cache size is relatively small compared to the number of content items in the network, since we need an additional entry for each content item that is not stored in either the local cache or the caches in the routers on the path to the content source. However, as the cache size increases, the probability increases that the content can be cached in the local router or on the path so that, we do not need an additional entry for each content item. In Fig. 15, the average numbers of additional table entries are 170, 339, 672, 1342, 2671, 3810 when the cache size are 32, 64, 128, 256, 512, 1024, respectively.

To mitigate the amount of broadcasting and the increasing size of the routing table, we can use group-level routing by grouping content as described in Section 5.3. Because the content assignment is done by a group-level, we can route the content by the ID of the group to which the content belongs. Fig. 15 shows the average number of routing table entries added at intermediate routers (seven routers

Table 3
MIP computation time.

Cache size	OPT_SP (s)	OPT_NW (s)	GB_SP (s)	GB1_NW (s)
32	17.1	72.63	0.38	1.42
64	25.93	154.07	0.48	1.45
128	18.83	173.1	0.42	1.72
256	26.16	236.26	0.51	2.04
512	26.97	201.23	0.42	1.45
1024	29.16	119.54	0.59	2.43

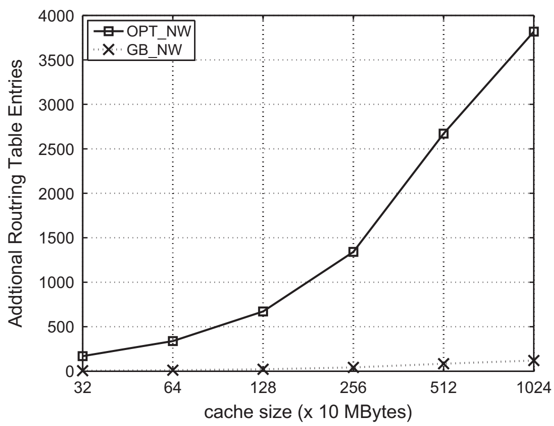


Fig. 15. Average number of routing table entries added by intermediate routers.

in the topology in Fig. 4). OPT_NW has 32 times more routing table entries to be added than GB_NW when each group size is 32 and the cache size ranges from 32 to 1024.

The second approach is advertising the Bloom filter in a distributed fashion. The Bloom filter is a space efficient probabilistic data structure that is efficient at checking whether an element is a member of a set. In [37,38], each router advertises Bloom filters, which represent the content that is stored at its own Content Store. With our assumptions, the Bloom filter needs to be advertised only if the content assignment is newly computed. The size of the Bloom filter depends on the number of content items to be cached in the AS network. Depending on the size of the AS network, a routing algorithm can be a distance-vector routing algorithm [37] or a link state routing algorithm [38].

6.5. Implementation

In our simulation results, we showed that our optimal content assignment is more efficient than a basic LRU policy which stores content in all routers on a data path. However, it is important to decide what content we select and how often to calculate the optimal content assignment. To select the long-term popular content and to calculate their optimal assignments, we need centralized network management that is similar to that in other proposals [17,32]. Each router reports the content request information to a centralized server. The server estimates the whole content popularity and selects the long-term popular content. With the selected content, the server calculates the optimal content assignment. Finally, the server lets each router know which content it is allowed to store in its Content Store.

The interval of calculating the optimal content assignment is a trade-off between accuracy and overhead. The appropriate interval can reflect the changing demands of requesters while reducing management overhead. The popularity of content tends to continue for a long time (from several hours up to several days). We adjust the interval over a relatively long-term period according to the popular content status. With a long interval, the estimation of the content popularity may no longer fit the current demand of requests. For example, the requests of new content can be rapidly increasing. To absorb the sudden changes in requests, we suggest the use of an additional Content Store. This follows an original LRU policy in NDN, and may handle all content that is not selected in our optimal assignment.

For the optimal content assignment of *Network-Wide Caching*, the requested content may not be located on the path from a requester toward an original source. To allow the request to be forwarded to a router holding a copy of the content, we require a new hash table or a Bloom filter to check whether a received interest packet for a piece of content is selected in the optimal content assignment. If the interest packet is for one of the selected content items, the router updates the FIB entry for the content as described in Section 6.4. Otherwise, the interest packet is forwarded by the existing FIB information without any updates.

7. Conclusion

The Internet Protocol (IP) has been widely deployed to facilitate ubiquitous interconnectivity. Unfortunately, the host-to-host conversation model is no longer effective for handling the explosion of multimedia content. The majority of current Internet traffic consists of requests and dissemination of popular content, and Internet bandwidth is wasted by repeated downloads of this content. Recently Content-Centric Networking (CCN) or Information-Centric Networking (ICN) architectures have been proposed for efficient content delivery. In these architectures, the Internet needs a fundamental paradigm shift from a traditional host-to-host conversation model to a content-centric communication model. It is no longer necessary to make a connection to obtain content. Instead, a user can directly send a request for content to the network. These architectures adopt in-network caching to enable requests to be served from nearby routers before arriving at the content source if the content is cached. Therefore, it is important to study an efficient cache management scheme for in-network caching.

In this paper, we investigate the performance of in-network caching for Named Data Networking. First we examine the inefficiency of LRU as a simple cache replacement policy at all routers on a data path. Instead of LRU, we suggest two in-network caching policies for long-term popular content in an AS network. One is *Single-Path Caching*, in which a request is supposed to find its content along the shortest path toward the content source. The other policy, called *Network-Wide Caching*, allows a request to acquire its content at any router in a network, if it is cached. For both policies, we develop optimal content assignment models using a Mixed Integer Program (MIP), which considers the internal and external link costs, cache size, link capacity, and the popularity of the content. From the evaluation, we show improved performance using the optimal content assignment as compared to LRU. Particularly, *Network-Wide Caching* outperforms *Single-Path Caching* in exploiting the whole in-network caches in an AS network. We also provide a nearly optimal scheme by grouping content to mitigate scalability problems in terms of the computation time of MIP solutions and the overhead of routing information updates.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MEST) (No. 2012003355).

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea government (MEST) (No. 2012004958).

References

[1] Cisco, Forecast and Methodology, 2010–2015.
[2] Cisco, Global Mobile Data Traffic Forecast Update, 2010–2015.
[3] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, ACM SIGCOMM Computer Communication Review, vol. 37, ACM, 2007, pp. 181–192.

[4] D. Lagutin, K. Visala, S. Tarkoma, Publish/subscribe for internet: Psirp perspective, Towards the Future Internet Emerging Trends from European Research 4 (2010) 75–84.
[5] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, V. Vercellone, Design considerations for a network of information, in: Proceedings of the 2008 ACM CoNEXT Conference, ACM, 2008, p. 66.
[6] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, R. Braynard, Networking named content, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, ACM, 2009, pp. 1–12.
[7] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named Data Networking (NDN) Project, Tech. Rep., PARC, Tech. Report ndn-0001, 2010.
[8] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, S. Moon, I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, 2007, pp. 1–14.
[9] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, K. Ramakrishnan, Optimal content placement for a large-scale vod system, in: Proceedings of the 2008 ACM CoNEXT Conference, ACM, 2010.
[10] M. Dahlin, R. Wang, T. Anderson, D. Patterson, Cooperative caching: using remote client memory to improve file system performance, in: Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation, USENIX Association, 1994, p. 19.
[11] A. Rowstron, P. Druschel, Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility, ACM SIGOPS Operating Systems Review, vol. 35, ACM, 2001, pp. 188–201.
[12] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, H. Levy, On the scale and performance of cooperative web proxy caching, ACM SIGOPS Operating Systems Review 33 (5) (1999) 16–31.
[13] E. Nygren, R. Sitaraman, J. Sun, The akamai network: a platform for high-performance internet applications, ACM SIGOPS Operating Systems Review 44 (3) (2010) 2–19.
[14] K. Park, V. Pai, Scale and performance in the coblitz large-file distribution service, in: Proceedings of the 3rd Conference on Networked Systems Design & Implementation, 2006, pp. 3–3.
[15] M. Freedman, Experiences with coralcdn: a five-year operational view, in: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, 2010. 7–7.
[16] A. Anand, A. Gupta, A. Akella, S. Seshan, S. Shenker, Packet caches on routers: the implications of universal redundant traffic elimination, ACM SIGCOMM Computer Communication Review, vol. 38, ACM, 2008, pp. 219–230.
[17] A. Anand, V. Sekar, A. Akella, Smartre: an architecture for coordinated network-wide redundancy elimination, ACM SIGCOMM Computer Communication Review, vol. 39, ACM, 2009, pp. 87–98.
[18] G. Tyson, S. Kaune, S. Miles, Y. El-khatib, A. Mauthe, A. Taweel, A trace-driven analysis of caching in content-centric networks, in: Proc. 21st Intl. Conference on Computer Communication Networks (ICCCN), IEEE, 2012.
[19] T. Rautio, O. Mmmel, J. Mkel, Multiaccess netinf: a prototype and simulations, in: TRIDENTCOM'10, 2010, pp. 605–608.
[20] I. Psaras, R. Clegg, R. Landa, W. Chai, G. Pavlou, Modelling and evaluation of ccn-caching trees, Networking 2011 (2011) 78–91.
[21] G. Carofiglio, M. Gallo, L. Muscariello, D. Perino, Modeling data transfer in content-centric networking, in: Proceedings of the 23rd International Teletraffic Congress, ITCP, 2011, pp. 111–118.
[22] L. Muscariello, G. Carofiglio, M. Gallo, Bandwidth and storage sharing performance in information centric networking, in: Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, ACM, 2011, pp. 26–31.
[23] S. Arianfar, P. Nikander, J. Ott, Packet-Level Caching for Information-Centric Networking, Finnish ICT-SHOK Future Internet Project, Tech. Rep.
[24] K. Katsaros, G. Xylomenos, G. Polyzos, Multicache: an overlay architecture for information-centric networking, Computer Networks 55 (4) (2011) 936–947.
[25] D. Rossi, D. Rossini, Caching Performance of Content Centric Networks Under Multi-Path Routing (and more), Tech. Rep., Technical Report, Telecom ParisTech (2011).
[26] D. Rossini, D. Rossi, A Dive into the Caching Performance of Content Centric Networking, Tech. Rep., Technical Report, Telecom ParisTech (2011).
[27] G.R. Dario Rossi, On sizing ccn content stores by exploiting topological information, in: Proceedings of IEEE Infocom Workshop on Emerging Design Choices in Name-Oriented Networking, IEEE, 2012.
[28] C. Fricker, P. Robert, J. Roberts, N. Sbihi, Impact of traffic mix on caching performance in a content-centric network, in: Proceedings of IEEE Infocom Workshop on Emerging Design Choices in Name-Oriented Networking, IEEE, 2012.
[29] K. Cho, M. Lee, K. Park, T.T. Kwon, Y. Choi, S. Pack, Wave: popularity-based and collaborative in-network caching for content-oriented networks, in: Proceedings of IEEE Infocom Workshop on Emerging Design Choices in Name-Oriented Networking, IEEE, 2012.
[30] W. Chai, D. He, I. Psaras, G. Pavlou, Cache Less for More in Information-Centric Networks, Networking (2012).
[31] S. Borst, V. Gupta, A. Walid, Distributed caching algorithms for content distribution networks, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010.
[32] S. Wang, J. Bi, Z. Li, X. Yang, J. Wu, Caching in information-centric networking, in: Proceedings of AsiaFI 2011 Summer School Program, AsiaFI, 2011.
[33] P. Gill, M. Arlitt, Z. Li, A. Mahanti, Youtube traffic characterization: a view from the edge, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, 2007, pp. 15–28.
[34] S. Asur, B. Huberman, Predicting the Future with Social Media, Arxiv preprint arXiv:1003.5699.
[35] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 426–434.
[36] B. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.
[37] M. Lee, K. Cho, K. Park, T. Kwon, Y. Choi, Scan: scalable content routing for content-aware networking, in: Communications (ICC), 2011 IEEE International Conference on, IEEE, 2011, pp. 1–5.
[38] Y. Wang, K. Lee, B. Venkataraman, R. Shamanna, I. Rhee, S. Yang, Advertising cached contents in the control plane: necessity and feasibility, in: Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on, IEEE, 2012, pp. 286–291.

Yusung Kim received his B.S degree in Computer Science from Kyungpook National University, Daegu, South Korea in February 2002, and the M.S. and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in February 2004 and February 2008 respectively. He had been a senior engineer in Samsung Electronics from 2008 to 2010. Currently, he is a research professor of Computer Engineering Department at Sungkyunkwan University. His research interests include congestion control, network optimization, and information-centric networking.

Ikjun Yeom received his B.S degree in Electronic Engineering from Yonsei University, Seoul, Korea in February 1995, and the M.S. and Ph.D degrees in Computer Engineering from Texas A&M University in August 1998 and May 2001, respectively. He had been a assistant professor in Computer Science Division at KAIST from January 2002 to August 2008. Currently, he is a associative professor of Computer Engineering department at Sungkyunkwan University. His research interests include QoS, TCP and mobile ad hoc networking.