Cache Capacity-aware CCN: Selective Caching and Cache-aware Routing

Sung-Won Lee, Dabin Kim, Young-Bae Ko
Graduate School of Information and Communication
Ajou University
Suwon, Republic of Korea
{skyline, dabin912, youngko}@ajou.ac.kr

Jae-Hoon Kim, Myeong-Wuk Jang
Communication and Networking Group
Samsung Advanced Institute of Technology
Yongin, Republic of Korea
{jaehoonk, myeong.jang}@samsung.com

Abstract—Content-centric networking (CCN) is a new networking paradigm to resolve the data traffic explosion problem of the Internet caused by rapid increase in file sharing and video streaming traffic. Networks with CCN avoid delivery of the same contents on one link as many times as they are requested, as contents can be stored and transferred by the cache of CCN routers. Two major features that are currently considered in CCN are in-network caching and content-aware routing. Even though both aspects are important, there is little work on the comprehensive interaction between them. In this paper, we propose the cache capacity-aware CCN which consists of selective caching and cache-aware routing methods that interacts with each other to encompass cache management and cache-aware request forwarding. The main motivation of the proposed scheme is to utilize the network caches evenly and redirect a content request based on the past forwarding of the desired contents (or chunks). To enable this function, we utilize a cache capacity metric which is collected on forwarded content requests and reflected in the content replication with popularity information from content server. We evaluate the proposed scheme against existing cache replication algorithms and show that it leads to better utilization of network caches with significant performance improvements.

Keywords—cache capacity, cache-aware, selective caching, content-centric networking, information-centric networking

I. INTRODUCTION

Content-Centric Networking (CCN) [1,2] is a new, redesigned communication networking architecture of the Internet to replace traditional host-centric communication with content-centric communication. CCN proposes an architecture which is centered on the content itself, without regard to where it is physically located. The CCN architecture includes an in-network caching and a content-based routing as main features where every content has its own name as an identifier.

The content-based routing mechanism delivers content request messages to routers holding the contents. The content request message, called the Interest packet in CCN, visits intermediate routers along the path toward the designated server. The content message, called the Data packet, is returned along the reverse path of the Interest. The content may be provided by the content server or by the cache storage in an intermediate router and can be transparently replicated in an intermediate router. The in-network caching mostly aims to keep frequently or recently used content items as long as possible because they are expected to be requested by other routers in the near future. Consequently, the in-network caching can respond to content request instead of a distant server and reduce bandwidth usage, latency, and workload on the original content server.

Existing cache replication proposals in CCN do not effectively consider actual cache capacity in a network-wide point of view. Most of these operate with only local information to determine content replication. Therefore, the caches near content server are required to handle more amounts of new incoming contents than others, causing cache pollution problem which evicts more popular content than newly cached content. In the aspect of cache-aware routing, it is mostly required to utilize a special control packet which shares information about cached contents near routers. However, it can impose a burden on networks according to the exchange period and sharing range of control packet, which directly affect the accuracy of contained information over time.

To alleviate these problems, we propose a cache capacity-aware CCN which is composed of a selective caching and a cache-aware routing algorithm, reducing network/server load while improving caching performance. Cache capacity is generally defined as the maximal size of a caching pool. However, in other researches, it also covers a meaning of available cache capacity which is a supportable caching space because only cache size cannot explain actual caching workload of a node. In other words, the available cache capacity can figure out differentiation of caching load among routers with the same cache size. Therefore, the proposed scheme estimates the available cache capacity exploited from observing recent cache consumption. Utilizing the metric, a router with the highest cache capacity on the path is dedicated to cache the content, while there is a possibility of near routers’ caching content according to content popularity and their cache capacity. Then, cache-aware routing keeps the forwarding direction of the content to redirect the following Interest for that content because the cache is usually closer than content of server.

As a contribution, the proposed scheme operates without a separate control plane protocol but simply piggybacks the necessary information on packet. Therefore, all nodes only require minimal additional information and computing...
overhead. As a result, the proposed scheme balances the caching load and combines this with providing track of where recent copies were directed.

II. RELATED WORK

Several traditional approaches for cache replication are revised as chunk-based algorithms and applied to CCN. The simplest method among them is the Leave Copy Everywhere (LCE) approach, which makes content replicas into every intermediate CCN routers. However, despite of the simplicity, it generates excessive cache replacements and incurs severe cache pollution. The Leave Copy Down (LCD) approach replicates the content (or the chunk) only on the next CCN router from Data generator which is either the content server or cache [3]. Although LCD is regarded as being simple yet efficient mechanism, it still shows high level of redundant cache replications on the routers near the content server.

Psaras et al. [4] proposed the ProbCache where the received Data is cached in the CCN router if the result value of an approximate function is higher than the pre-defined threshold value. The approximate function considers the caching capacity of the CCN router and the distance from the content server. Even though the ProbCache utilizes the available cache capacity, cache capacity of each router is not handled individually because it is accumulated with others for remaining path and combined with distance parameter. Cho et al. [5] proposed the WAVE which considers a chunk as one of subsequent data segments belonging to one content. At first request for a content, WAVE caches only the first chunk and then more chunks will be stored exponentially according to the gradual increase of the request frequency for the content. However, this cache management does not consider the cache capacity, still leading to high level of cache replications near the content server. Recently, Yanjua et al. [6] proposed the coordinating in-network caching algorithm to provision routers’ storage capability and network performance as an optimal solution. However, it requires a coordinator node to collect the required information from every router.

In aspect of content-aware routing, there is no generalized protocol so far. Jacobson et al. [1] and Zhang et al. [2] suggest methodology about how to route. The guided-diffusion flooding model is suggested for pre-topology phase and the proactive route management is introduced with periodical announcement of name prefix(es) from content server at least. However, both routing mechanisms have not been detailed as well as their metric. Thus, almost researches on cache management assume pre-constructed routing table or IP-overlay architecture which provides route toward content server at least. To provide enhanced content-aware routing, there are some approaches to get cache-awareness for utilizing cached contents not only within the path but also near the path toward the content server. The simplest method of doing this is to advertise a list of cached contents. However, this is not recommended due to huge overhead and inefficiency. The scalable content routing for content-aware networking (SCAN) [7] aims to alleviate the scalability problem. SCAN exchanges the information of the cached contents and their route information utilizing bloom filter. This algorithm can reduce the name space efficiently according to hash function with space-efficiency. However, problems such as false positive and false negative can occur, which lead to route failures and detouring routes. Consequently, if SCAN stands alone without IP-overlay architecture, it causes large control overhead and unstable communication. Therefore, SCAN cannot be directly utilized as a normal cache-aware routing for CCN.

III. PROBLEM STATEMENT

The main idea of this paper is to evenly utilize in-network caches for maximizing available cache capacity in the whole network. It intends to make efficient caching across the network, alleviating the cache pollution effect which gets worse in a router with weighted caching burden. Mostly, routers near a content server and on the core network tend to handle more Interest and Data forwarding and thus have higher possibility of caching more contents than the others. Consequently, some contents that are better to be resided in the cache cannot help being evicted with short live time, even if they are in parts of popular contents. In other words, cache capacity in the routers is limited to wait for making use of them again due to short caching time, while the others on the opposite side take an undeserved time to distinct popular contents in the cache.

To figure out the weighted caching time and its problem, we evaluate average content residence time and average cache hit ratio via a simulation study. The Internet topology is constructed with 100 routers based on a transit-stub model by Georgia Tech Internet Topology Model (GT-ITM) [8] and the LCD is applied, where the alpha value of Zipf’s distribution varies from 0.8 to 1.5. More detailed parameters for simulation are the same with Table 1 in section V. Fig. 1 presents the average content residence time in the cache per router which is estimated as the average time lag between contents insertion to and eviction from the cache. The red rectangle indicates routers in the transit domain and purple dotted circle, node 15 at the undermost is a content server. They show the least average content residence time due to the weighted caching burden and suffer from the lowest average cache hit ratio as shown in Fig. 2, where they are presented in a range of 1–3 hop from a content server. On the contrary, the routers inside the stub domain relatively have long content residence time and higher

![Figure 1. Average Content Residence Time per Router (alpha = 1.0)](image1.png)

![Figure 2. Average Cache Hit Ratio per Hop Distance from Content Server](image2.png)
cache hit ratio on average. In case of green circles in Fig. 1, they are the routers that connect the transit and stub domain in the 3 or 4 hop distance, showing a medium level of the content residence time. However, their performance is still limited in terms of the average cache hit ratio as shown in Fig. 2. Therefore, even if all the routers have the same physical cache capacity, the actual cache capacity is varied according to the cache location.

From above simulation results, the routers far away from the content server cache popular contents with high probability. By utilizing near those cached contents, the cache-aware routing can enhance caching effect. Furthermore, when our aim which is to alleviate weighted caching load is achieved, more amount of popular contents will be distributed across the network. Therefore, a proper solution for cache-aware routing will gain significant performance enhancements.

IV. CACHE CAPACITY-AWARE CCN

The proposed cache capacity-aware CCN (CC-CCN) consists of three parts: cache capacity estimation, selective caching, and cache-aware routing. Firstly, Cache Capacity Value (CCV) which is a key metric of proposed scheme is passively estimated in each router and the highest value on the path is recorded on the forwarded Interest message. The content is selectively cached according to cache capacity and content popularity while being forwarded and only highly popular content has more opportunity to be cached in intermediate routers. In the selective caching, the routers within the selective caching range create temporal Forwarding Information Base (FIB) entry indicating outgoing face to the router with the highest CCV. With utilizing the temporal FIB entry, the cache-aware routing is performed without additional control overhead, providing high route accuracy for cached contents by effective expiration timer.

A. Cache Capacity Estimation

The cache capacity is a metric to present the cache utilization and there are several ways to express its availability. The cache utilization can be diverse according to cache size, cache location and cache management technology. In addition, even if every router physically has the same cache size, the actual cache utilization will be different according to other factors. The most straightforward way is to use the measured raw values, such as how much a cache is vacant or how many contents a router currently caches. Even though these raw values can be useful, it cannot be a fair metric when we compare the value of a router with those of other routers. For a fairer and more accurate assessment, normalization techniques are needed to manipulate the estimated cache capacity.

For this purpose, we evaluate the cache capacity as an inverse function to the amount of recent cache consumption. In addition, physical cache size of each router is also reflected in the cache capacity because there may be heterogeneous cache sizes. As a result, the cache capacity value (CCV) of router \( r \) is calculated as shown in equation (1),

\[
CCV(r) = \frac{c}{L(r)} \times \text{CacheSize}(r).
\]

where the availability of caching contents has the positive relationship with physical cache size (\( \text{CacheSize} \)) and inverse relationship with caching load (\( L \)) which is an accumulative amount of caching content during a certain period. For instance, within the \( n \)-th period of estimation, a router utilizes caching load (\( L \)) measured in \( (n\text{-}1) \)-th period, while accumulating current caching load for \( (n\text{+}1) \)-th period. The parameter \( c \) is a compensation value for the case that a huge value of caching load and \( \text{CacheSize} \) makes a distortion of CCV among routers. Therefore, in all routers’ cache size are assumed to be homogeneous, it is set to 1 by default.

Fig. 3 shows an example scenario of how the CCV is calculated and the highest CCV is inserted in the forwarded Interest message. In case of Node 1, it has cache size of 500 and caches the contents whose total size amounts to 200. Therefore, the CCV of Node 1 is derived as 2.5 when the \( c \) is 1 as a default and the other routers get their own CCV in the same way. Assume that a client generates an Interest to request for some content in the content server. The Interest maintains a field for recording the \( CCV_{\text{highest}} \) on the path. Initially at the client having no cache, this field is set to zero. Now when Node 1 receives an Interest from the client, it will compare its own CCV (which is assumed 2.5 in the figure) to the value of the corresponding field in the receiving Interest (zero in this example), and make an update as its own CCV is higher. Similarly on receiving the Interest from Node 1, Node 2 is also required to update the receiving CCV (i.e., 2.5) by its own value of 4, which is again assumed to be higher. Note that, in Fig. 3, this highest value of CCV keeps the same until the forwarded Interest eventually reaches to the targeting content server. This is simple because the rest of the nodes on the path are assumed to have the smaller values of CCV, compared to that in the receiving Interest.

In discovering the highest CCV, the router having the highest CCV can be identified using the Network Distance Value (NDV) within the Interest. This value of NDV is reset to one by any node who decides to update the CCV field; otherwise, it is incremented by one in each hop. As a result, the content holder which is either the content server or an intermediate router recognizes the highest CCV on the path and the hop distance with the relevant node from the receiving Interest. That is, in Fig. 3, the content server is aware that there is a router node with the highest CCV 4 at 4 hop distance.

B. Selective Caching

The selective caching focuses on cache replication, more specifically on how and where to cache the content while forwarding Data. The proposed scheme has two decision
Figure 4. Example of Selective Caching and Cache-aware Routing

The second decision is hence needed to make a threshold for multiple replications of more popular contents depending on the cache capacity of intermediate nodes. Even though each CCN router has no way of recognizing the content popularity, the content server can presume the order of popularity (defined as a rank) in the Zipf’s distribution) via exploiting past Interest processing. The amount of request counts according to content popularity is exponentially decreased as the rank $r$ of content is declined in series. Therefore, the order of content popularity is normalized with a logarithmic function as an inverse of exponential function and the normalized value is treated as a weight parameter to estimate the threshold. The logarithmic function for normalization is shown in the equation (2).

$$w_r = \frac{\log r}{\log N_{total}} = \log N_{total}^r$$

Here, the popularity of the rank $r$ is normalized by the total number of contents ($N_{total}$) in the content server and consequently any content with higher popularity gets a lower weight value to be replicated with higher possibility.

When a content server replies a Data packet back to the requesting client, the weight value ($w_r$) for the contained content is enveloped in the Data as well as the NDV value and the highest CCV ($CCV_{highest}$) retrieved from the received Interest. Upon receiving Data, the intermediate router $i$ calculates the CCV threshold ($CCV_{th}$) for the content with the enveloped $w_r$ and $CCV_{highest}$ as shown in the equation (3).

$$CCV_{th} = CCV_{highest} \times w_r$$

Now with this CCV threshold, the router compares it with its own CCV and decides to cache the content if its own CCV is larger. Otherwise, the router forwards the Data without caching (i.e., no replication). Note that the content server provides the $CCV_{highest}$ and $w_r$ separately, instead of pre-calculated $CCV_{th}$. The reason is that the $CCV_{highest}$ is differently estimated with time and path of the forwarded Interest, whereas the $w_r$ tends to be constant. In other words, each CCN router cannot recognize the content popularity and cope with the newly discovered $CCV_{highest}$. Therefore, when content is cached, the weight value $w_r$ provided by the content server, is stored along with the content. When cache hit occurs, Data will include the $CCV_{highest}$ retrieved from received Interest and the $w_r$ from the cache.

As shown in Fig. 2, the network load near the proximity of the network core tends to increase. In order to distribute such an unbalanced network load, the area for making a selective caching is restricted with a half location between the node with the highest CCV and the content holder. This is expressed as a dashed rectangle in Fig. 4. To create the caching area, whenever a router forwards Data, the NDV value which is initially acquired from the relevant Interest is decreased by one. The selective caching is performed only if the remaining NDV value is in between [0, NDV/2]. If the content is popular, then all routers in the dotted rectangle may cache it. Otherwise, only Node 2 will cache it. For instance, assuming that there are total 1,000 contents in the content server, Node 3 can cache the content whose rank is above 75 because the $w_r$ must be below 0.625. In case of content in 100th rank, Node 3 cannot to cache the content due to the CCVth of about 2.7. Over selective caching in limited areas, routers that are distant from the content server will cache more contents when routers near the server have high caching load. In turn when routers near the edge suffer higher caching load, routers near the server will instead try to cache more contents. This oscillation makes the proposed scheme evenly utilize in-network caches.

C. Cache-aware Routing

Most of the existing cache-aware routing methods tend to make its functional boundary into two categories. The first is to share a list of the cached content periodically and the second is to flood the Interest until the desired content is found. Both approaches may result in considerable amount of network load.

In the proposed cache-aware routing, the router within the range of [0, NDV/2] creates a temporal FIB entry with the exact content name and an outgoing face toward the router with the highest CCV if the router does not cache the content, while forwarding Data with the selective caching. The reason is that at least one router in the forwarded direction will cache the content. Then, the Interest will be re-routed according to the temporal FIB entry if the router receives the Interest for the same content. For example, Node 3 in Fig. 4 will redirect the Interest for the content ‘/Content_server/a.txt’ through face 2.

The proposed cache-aware routing approach must manage the expiration time of temporal FIB entry efficiently. As our aim is to distribute the caching load fairly and utilize the cached content, the average residence time of a content in a router is expected to be similar to all the other routers. Therefore, the expiration time is set by locally estimated average content residence time which will be quite accurate to point out the cached content. However, the route looping can occur in case the average content residence time is different to the actual caching time. To solve this problem, the redirected Interest gets a redirection flag set as TRUE and the router receiving redirected Interest judges the possibility of route looping when it has neither temporal FIB entry nor requested content. In this case, the router sets the redirect flag of the Interest as ERROR and forwards it according to the normal FIB mechanism. The other routers receiving this Interest try to
eliminate the corresponding temporal FIB entry.

V. PERFORMANCE EVALUATIONS

A. CCN Implementation

We have developed our CCN simulator based on OPNET [9]). To create various scenarios, we also developed the CCN network scenario generator which has several functionalities such as the topology generator, hierarchical/flat naming, homogeneous/heterogeneous cache sizing, etc. These functions are applicable to the OPNET simulator in xml format. The CCN Simulator provides various cache management schemes that are served in CCNx[10] or ccnSim[11]. The traffic generator provides three kinds of traffic models, which are Zipf’s distribution, MZipf’s distribution, and flash crowd model based on [12]. To consider the layering of the forwarding engine, there are two types of node models which are IP-overlay router model and MAC-overlay router, and for this adaptation, interface modules are served like the link adaptor in the CCNx. We implement a simple routing protocol to discover the shortest path and Interest will be directly forwarded toward content server.

<table>
<thead>
<tr>
<th>Table 1. Simulation Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Category</strong></td>
</tr>
<tr>
<td>Topology</td>
</tr>
<tr>
<td>Interest Arrival rate</td>
</tr>
<tr>
<td>Number of routers</td>
</tr>
<tr>
<td>Number of contents</td>
</tr>
<tr>
<td>Zipf’s distribution</td>
</tr>
<tr>
<td>Average cache size</td>
</tr>
<tr>
<td>Cache policy</td>
</tr>
</tbody>
</table>

B. Simulation Environments

With the CCN simulator, we utilize the Internet topology generated by GT-ITM which includes 12 stub domains, 1 transit domain, and consequently 100 CCN routers in total. It is converted into the CCN network scenario by CCN network scenario generator with the environment parameters as shown in Table 1. We use totally 10,000 contents following the Zipf’s distribution popularity whose alpha value is set to 0.8 and 1.0 considering the large popular content portion [12]. Traffic generator schedules 10 Interests every second by the Interest arrival rate. Cache is assigned with a size of 1% per router which is a ratio to the total content amount. In other words, the 1% in our scenario corresponds to a volume of 100 contents. The proposed scheme is compared with LCE, LCD, MCD, Prob., and ProbCache which operate with a simple shortest path routing towards a content server.

C. Simulation Results

The proposed scheme has a priority to make the cache capacity of each network cache as even as possible. The distribution per router about caching capacity is compared in terms of the average content residence time in Fig. 5. In the proposed scheme, the average content residence time per router was measured to be in a similar level. However, the LCD suffers extremely low average content residence time near the content server as highlighted in the zoomed spot, while the proposed scheme shows an even result. Therefore, popular contents in the LCD are hardly replicated across the network due to severe cache pollution near the content server. However, the result of the proposed scheme has converged upon a low level compared to the stub routers by LCD. To verify reasonability for the result, we evaluate the content reusability, which is a value of how many times the cached contents were hit among the total number of cached contents as shown in Fig. 6. The proposed scheme maintains much higher content reusability than the LCD because our selective caching makes high diversity of cached contents and the content-aware routing efficiently exploits them. In other words, cache capacity in stub routers is not fully exploited in case of the LCD and equable average content residence time of our CC-CN is enough to enhance overall caching performance.
Caching the desired content, resulting in reduction of hop distance that the Data or Interest have to traverse. This can effectively increase the average cache hit ratio, while decreasing the server load. However, the network load is just slightly increased as the detour from Data redirection can cause minor increase in the network load.

Nevertheless, if we consider the tendency that the highest network load is exposed near a content server, the detour by the proposal is intuitively expected to lessen the level of the highest network load. To evaluate this effect, we measure the link stress, which is an average value of the network load within the top 10 percent. As shown in Fig. 10, the proposed scheme’s link stress is alleviated about 30 ~ 31% and about 46 ~ 65%, compared to LCD and LCE, respectively. There was no occurrence of detecting the route loop, proving that our looping prevention method worked properly and did not severely affect the link stress.

VI. CONCLUSION

Content-centric networking utilizes in-network caching and content-aware routing to provide better services on the Internet. In the proposed cache capacity-aware CCN, the selective caching utilizes individual cache capacity and content popularity to fairly distribute caching load. The cache-aware routing provides accurate Interest direction without any control messages by utilizing the average content residence time. The performance evaluation indicates that the proposed scheme utilizes the network caches evenly and achieves the highest performance improvements on average cache hit ratio, server and network load, as well as link stress. Also, it is proved that the cache contents are highly reused. In the future, we will consider more challenging environments such as more limited cache size or Video-on-Demand (VoD) services with requirements of Quality-of-Service (QoS).

REFERENCES


