Inter-Domain Popularity-aware Video Caching in Future Internet Architectures

(invited paper)

Zhi Liu[†], Mianxiong Dong[⋆], Bo Gu[†], Cheng Zhang[†], Yusheng Ji[‡], Yoshiaki Tanaka[†]
[†] Waseda University, Tokyo, Japan 169-8555
[⋆] Muroran Institute of Technology, Hokkaido, Japan 050-8585
[‡] National Institute of Informatics, Tokyo, Japan 101-8430
Email: {liuzhi@aoni.waseda.jp, mx.dong@csse.muroran-it.ac.jp,
{bo.gu, cheng.zhang}@akane.waseda.jp,kei@nii.ac.jp, ytanaka@waseda.jp}

Abstract—**Current TCP/IP based network is suffering from the usage of IP especially in the era of *Internet of things* (IoT). Recently *Content Centric Network* (CCN) is proposed as an alternative of the future network architecture. In CCN, data itself, which is authenticated and secured, is a name and can be directly requested at the network level instead of using IP and *Domain Name System* (DNS). Another difference between CCN and traditional networks is that the routers in CCN have the caching abilities. Then the end users can obtain the data from routers instead of from the remote server if the content has been stored in the router. Hence the overall network performance can be improved by reducing the required transmission hops and the advantage of the CCN caching has been shown in literature. In this paper, we design a new caching policy for the popularity-aware video caching in CCN to handle the 'redundancy' problem in the existing schemes, where the same content may be stored multiple times along the road from server to users, thus leading to a significant performance degradation. Simulations are conducted and we could observe that the proposed scheme performs better comparing with the existing caching policies.**

I. Introduction

Content Centric Network (CCN) [1] (which is similar with *Information Centric Network (ICN)* [2], *named data networking (NDN)* [3], *data oriented architecture* [4], etc.) was proposed by Van Jacobosen, as an alternative of the current TCP/IP-based network. Comparing with the traditional networks, CCN focuses on 'what' instead of 'where', i.e. content itself is more important than where the content is (where is represented using IP address in the traditional TCP/IP networks). Content is a primitive in CCN, data itself is a name and can be directly requested at the network level, which means IP is not necessary and there is no more DNS. Anybody with the data can answer the data request. The data itself is authenticated and secured instead of securing connections it traverses.

Another difference between CCN and the traditional networks is that CCN's routers have the caching abilities. If the requested data has been stored in the routers, the end users can obtain the data from the routers directly instead of from the remote server. This greatly reduces the total transmission hops (transmission time). The inherent problem is what should be cached in the router and how to replace the cached content when a new content will be cached in a full cache. The caching decision policy [5, 6] decides whether the new arriving video content should be stored in a particular router, and the caching replacement (such as *Least Recently Used* (LRU)) policy decides what content will be moved out when new content is decided to be cached in the caching-size-limited full router. The caching policy greatly affects the overall network performance, hence it is an important issue for CCN.

On the other hand, video plays a great important role, where it dominates the current network traffic and this trend will last at least for a couple of years. According to *Cisco Visual Networking Index (VNI)*[1], IP video traffic will be 79% of all consumer Internet traffic in 2018, up from 66% in 2013. There are various schemes for the video transmission over the traditional wireless networks such as [7, 8] to improve users' received video quality by jointly considering how to encode the source video, how to transmit the encoded video and how to perform the error recovery if necessary. But there are not much work about video transmission in CCN, hence how to provide high-quality video service while at the same time reduce the network burden becomes an important issue to be investigated for the future network architecture. One character of video is that each video has a associated popularity, which indicates the average demanded times among all the requested video. Zipf [9] is commonly

[0] This work is partially supported by JSPS KAKENHI under Grant 15K21599, 26730056 and JSPS A3 Foresight Program.

[1] Cisco Visual Networking Index: Forecast and Methodology, 2013 2018

QSHINE 2015, August 19-20, Taipei, Taiwan

DOI 10.4108/eai.19-8-2015.2260888

applied to help calculate the video popularity. Given some video are requested more often compared with the rest video, caching the more popular video content in the routers close to the users will help reduce the total network transmission cost. How to cache the video given the video's different popularity values has recently been studied [10,11]. A recent study [5] addressed the video popularity-aware video caching in CCN and provides better performance. But all these related works in literature do not handle the 'redundancy' problem well, where maybe the same content will be stored in multiple routers along the path from server to users, thus leading to the caching performance degradation.

In this paper, we propose a novel caching policy for the popularity-aware video streaming in CCN, where the network is with known topology. We design a new caching policy considering the sizes of the routers and the video' popularity, where the new designed caching policy can help solve the 'redundancy' problem. Extensive simulations are conducted to verify the proposed scheme's performance. The experimental results show that the proposed caching policy can outperform the state-of-the-art caching schemes greatly in typical network scenarios.

The rest of the paper is organized as follows: Section II discusses the state-of-the-art works about CCN and CCN caching. The system model and the video popularity model are introduced in Section III. How we design the new caching probability is shown in Section IV. The simulation results and the conclusion are shown in Section V and Section VI, respectively.

II. Related works

This section introduces the CCN and the CCN caching. The differences between the existing related schemes and the proposed caching policy are explained at the end of this section.

CCN is similar with ICN [2], NDN [3], *data oriented architecture* [4][2]. CCN emphasizes the content by making the content directly addressable and routable, and the communication between the endpoints are based on named data instead of IP addresses. This can help solve the problems raised by the IP addresses, and could be an alternative of the future network structure especially in the IoT era. Caching is an critical issue in CCN given that the routers, which could significantly affect the overall network performance. If the requested content has been stored in the router, the user can retrieve the demanded content directly from the router instead of from the remote server, i.e. the transmission hops (transmission time) are reduced. Therefore caching can help reduce the network traffic.

There are various caching strategies proposed in literature, such as [10–14]. Specifically, [12] proposed an implicit coordinate chunk caching location and searching scheme in CCN hierarchical infrastructure. [13] discussed the popularity-driven coordinated caching by formulating this into a optimization problem. But the computation complexity is too large, then the authors also proposed an online algorithm. [14] also handles the 'redundancy' problem, but the authors take the globe routers into consideration to generate the weight for caching. A recent related paper is [5], where the video that are popular are scheduled to be placed close to end users to reduce the expected *round-trip time*(RTT) and improve the server-hit rate comparing with state-of-the-art schemes. But his work has the 'redundancy' problem as introduced in Section I, hence the system performance is affected.

Different from these works, this paper targets the popularity-aware video caching in CCN with fixed topology, and designs a new caching policy based on the routers' caching size and the video popularity. Also the proposed scheme handles the caching 'redundancy' problem, which leads to superior performance over the competing schemes.

III. System Overview

This section overviews the CCN system and the Zipf model used in this paper to model the video popularity.

A. CCN overview

As an initial study of the video caching in CCN, we use a simple CCN network to introduce the CCN and explain the principles of the proposed scheme.

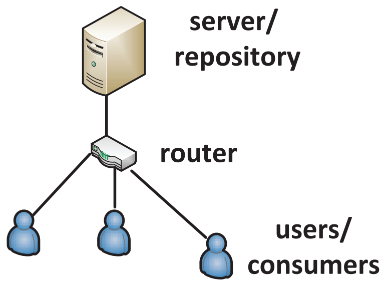


Fig. 1. Illustration of the CCN with single node model.

Fig. 1 is an illustration of the CCN. It is composed by a server/repository, a router and many users/consumers. In CCN, routers can buffer data and content store (CS) of routers plays a role of a buffer memory. The caching in CCN involves 1) *caching decision* and 2) *cache replacement*. The caching decision mainly helps decide where to cache the 'new' data chunks if they are not in the cache currently. We can also say the video caching decision helps decide whether or not to cache a video content when it

[2] Please note that the proposed caching policy also works in other ICN architectures, where the key ideas of these network architectures are more or less the same.

arrives at a router, where it is not cached. When a new data chunk is decided to be stored in a fully occupied CS, some data chunk will be moved out. Cache replacement scheme discusses which existing data chunk should be replaced. In Fig. 1, if a data chunk requested by the user is not stored in the routers' CS, server will send the data chunk to user. When the data chunk forwarded from the server arrives at the router, the router will decide whether to cache the chunks or not according to the caching and replacement decision policy. If the decision output a 'yes' but the CS' memory space is fully occupied, the new chunks will replace the CS's existing data chunks in accordance with the cache replacement scheme. For instance, if the *least recently used* (LRU)[3] is the chosen cache replacement scheme, the fully occupied CS will replace the least recently used chunk with the newly arrived chunk. The caching decision policy and cache replacement scheme affects the caching efficiency and the overall network performance. This paper targets the caching decision policy in a router for alternative future network architecture CCN.

B. Video popularity

Different video contents have different popularity values, where the video i's popularity is defined as the ratio of the number of video i's requests to the number of total video requests. Since the popular content will be subscribed more than the other video contents, video contents that are more popular should be placed closer to users to reduce the network cost and shorten the RTT. In this paper, *Zipf* popularity distribution is adopted. We assume the total number of video contents is N and k_i is ith video item's popularity rank. Smaller rank instance number indicates higher popularity. Then the Zipf's law predicts that out of a population of N video elements, the frequency of element i with rank k_i is as follows:

$$f_{i,k_i,s,N} = \frac{1/k_i^s}{\sum_{n=1}^{n=N} 1/n^s} \qquad (1)$$

Where s denotes the value of the exponent characterizing the distribution and is referred to the skewness of popularity. Fig. 2 shows the popularity distribution with different s. The x-axis is the popularity rank and y-axis is the popularity value. We can observe that large s leads to highly right-skewed histogram, representing high diversity among video contents in term of popularity. On the other hand, the small s leads to a flat-skewed histogram, and represents less video popularity diversity.

To better show the Zipf popularity distribution, we also calculate the *cumulative distribution function* (CDF) of the popularity distribution as shown in Fig. 3. From the figure, we can see that different s leads to different popularity CDF distributions. More importantly, it can be noticed that the first several most popular video can satisfy a large proportion of the total requests. For example, the 5 most popular video when $s = 1.9$ serve 87.2% video requests. Then by placing the popular video close to users, the network cost could be greatly reduced since the transmission hops needed are greatly reduced. This motivates designing caching decision policies to help improve the network video distribution.

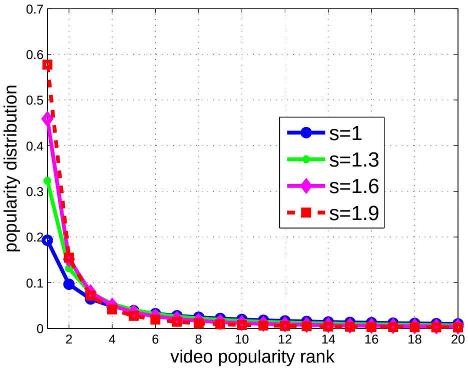


Fig. 2. Zipf popularity distribution for the 20 most popular video contents with different s when N=100.

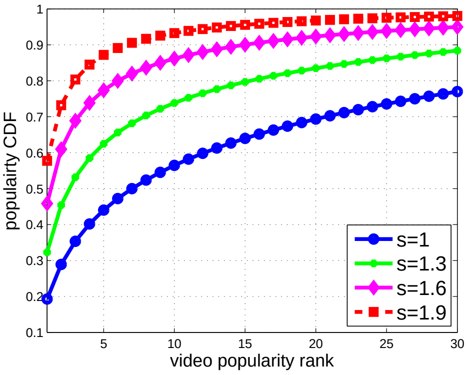


Fig. 3. Zipf popularity CDF for the 30 most popular video contents with different s when N=100.

Next we show how we utilize the video popularity and the routers' caching sizes to design the caching decision policies.

IV. Caching decision policy

This section introduces the objective and the designed caching policies.

A. Objective

This paper dedicates to improve the popularity-aware video content dissemination in CCN with known topology. By caching the video contents efficiently, RTT of each video request could be reduced. Recall that Zipf popularity model indicates that the several most popular video contents could satisfy a large proportion of the

[3] This paper mainly discusses the video caching policy, and we use the LRU as the default cache replacement policy.

video requests. This motives us to take the video popularity into consideration when designing the caching decision policy. Caching decision policy decides which video should be placed in which router or when a video content arrives at a router where it is not cached, the router should cache it or not. The objective of this paper is to design an efficient caching policy for the popularity-aware video distribution over CCN.

B. Caching Decision Policy

There are multiple routers along the path from the server to users possibly. The routers are labeled with different levels according to their distances from the users. Figure 4 and figure 5 show the *Cascade network topology* and the *Binary Tree topology*, respectively, and these two topology models are also used in the simulation section. In Fig. 4, there are five routers which are denoted using level instance numbers from number 1 to number 5. Level 1 router is the most closest to users, and level 5 is the furthest from the users. Fig. 5 illustrates the binary tree topology network structure with 3 levels. The same as the cascade network topology, level 1 router is the most nearest to users and the level 5 is the furthest. ith level router' CS has a cache size x_i chunks. And we assume the video i's size is δ_i chunks with its popularity to be $f_{k_i,i,s,N}$.

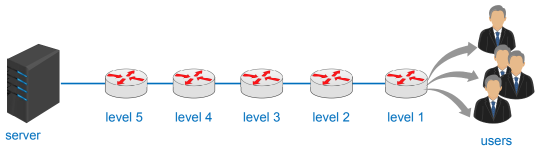


Fig. 4. Illustration of the Cascade topology with five levels. Level 1 router is the one most close to users and level 5 router is the one most close to the server.

Before discussing the caching policy, a router index number $I_{i,j}$ is introduced. Since router j has limited cache size, i.e. x_j chunks, it can only store limited number of video contents. But the several most popular video serves a large proportion of the total video requests. Following this principle, the most popular content should be allocated to the routers that are the closest to users. We first calculate the index number $I_{i,j}$ to indict whether video i should be placed at router j or not considering the video popularity. The method to calculate the $I_{i,j}$ is as follows: We assume along the path from user to server, there are M level routers, denoted as $1, 2, ..., M$. And we can use the following equation to calculate the $I_{i,j}$;

$$I_{i,j} = \begin{cases} 1 & : if(\delta_i + \sum\limits_{\forall p \in N, k(p) < k(i)} \delta_p) > \sum\limits_{q=1}^{j-1} x_q \& \sum\limits_{\forall p \in N, k(p) \le k(i)} \delta_p \le \sum\limits_{q=1}^{j} x_q \\ 0 & : o.w. \end{cases}$$

From this equation, we could find that $I_{i,j} = 0$ or $I_{i,j} = 1$. The $I_{i,j} = 1$ stands for that the video i should be placed at the jth level router according to its popularity rank k_i and the caching abilities of the routers. The caching abilities are determined by the cache size and the video size. The general idea is that the most popular video will be placed in the router nearest to the user. If the router is not empty, the second most popular video will be placed also, otherwise, the video will be placed in the next level router. Then by comparing the cache sizes up to level j and the video sizes whose popularity instance number is less than $k(i)$, we can know whether the video i should be placed in router j or not.

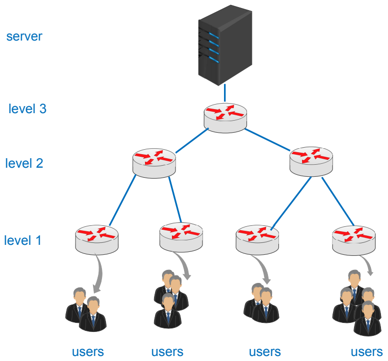


Fig. 5. Illustration of the Binary tree topology with three levels. Level 1 router is the one most close to users and level 3 router is the one most close to the server.

We use $p_{i,j}$ to denote whether video i should be cached in router j, and the caching probability is designed as follows:

$$p_{i,j} = \begin{cases} 1 & : if I_{i,j} = 1 \\ 0 & : if I_{i,j} = 0. \end{cases}$$

This means only when the video should be placed in the corresponding router, the video will be cached. Since if $I_{i,j} = 0$, $p_{i,j} = 0$, i.e. the video will not be cached in jth level router. Hence the 'redundancy' problem in the current existed schemes could be solved. The disadvantage is that the system needs time to cache the video content, and the time needed is discussed in the simulation section.

C. Practical issues

To implement this caching decision scheme, the routers need to know video's rank table and then decide whether they should cache the video or not. But please note that the communication between the routers is not that costly and the table is not that big. Exchanging this information periodically is realistic.

On the other hand, the popularity is based on the requesting times. The video requests come from the end users and if the requests can not be satisfied at router i, the video request will be forwarded to router at level $i + 1$. Therefore level 1 router can know all the request information, and the most popular video contents can be cached. Level 2 routers can know the most popular video contents except the video that has been cached by router 1, since the corresponding requests will not be

TABLE I
Detailed network parameters

Parameters	Values
total request rate: λ	120 content items/s
# of different content items: M	2× 10^4 items
average content size: σ	690 chunks (6.9 MB)
cache size of node 1: X(1)	2× 10^5 chunks (2GB)
# of content classes: K	100 classes

forwarded to router level 2. Similarly, level i router can have the necessary information for the decision making.

Also this paper requires that the known network topology should have routers with clear levels. But this is actually reasonable if we take a look at the last miles of the network.

V. Simulation

We conduct the simulations to verify the proposed scheme's performance by comparing with the state-of-the-art competing schemes. This section introduces the simulation setup and simulation results.

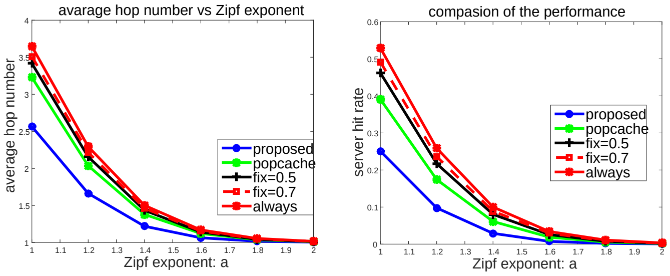
A. Simulation setup

To evaluate the proposed scheme's performance, two different network topology models, i.e. cascading model and binary tree model as shown in Fig. 4 and Fig. 5 are applied. Both networks are with five levels. Each video content's delivery time is composed of the transmission time, propagation delay and queueing delay. Since the propagation delay and the queueing delay is very small comparing with the video transmission time, hence these two terms are assumed to be zero. The transmission time can be denoted using the number of hops the video has traveled. Then we use the *number of hops the video* has traveled and *server hit rates* (the portion that the requested video is retrieved from the server) as the evaluation metrics.

The parameters used to simulate the network are shown in TABLE I, where we follow [5] and design this small-scale simulation. Given the channel bandwidth (10Gbps) is much larger than the source rate of the requested video content (average total size is 120*6.9 MB), the requests are assumed to be finished within one second in this paper. To simplify the implementation, we assume the video is identical but with different popularity ranks. *Kendo*[4] is used and encoded using JSVM [5]. The resolution of kendo is 1024×768 with 32 fps frequency and resulted distortion 39.2 dB. The GOP size is set as 16 frames.

B. simulation results

To show the performance of the proposed scheme, we compare our scheme with *Popcache* [5], a scheme dedicated for video caching with good performance in term of reducing the RTT and improving the server-hit rate, *always* and *fix=x*. *always* means when a new video content arrives at the router, it will be stored. *fix=x* means, when a new video content arrives at a router, it has the probability x to be cached. We could find that *always* is an extreme case of *fix=x*, where $x = 1$. The caching probability of the *Popcache* is obtained from reference [5] directly. LRU is the replacement policy adopted in this simulation. We count the first 7000s as the *learning* period where the cache will be updated following the caching policy. The results shown are the average from 7001 to 10000.



(a) average transmission hops

(b) server hit rates

Fig. 6. Performance evaluation of the CCN with Cascade network topology.

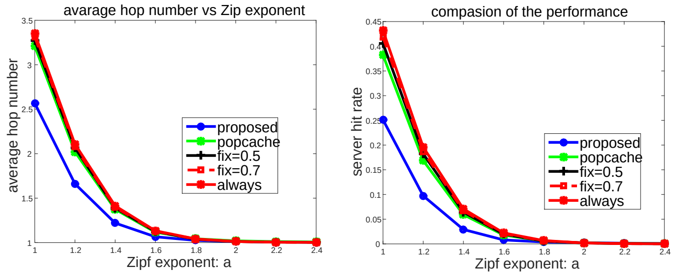
Fig. 6 and Fig. 7 shows the performances in term of the average transmission hops needed and the server hit rates in the cascade topology and the binary tree topology, respectively. From Fig. 6 (a), we can observe that the *always* performs the worst and *fix=x* is better than *always*. *popcache* is better than *always* and *fix=x* since it considers the video popularity. But all these schemes have the 'redundancy' problem, where the same content maybe stored multiple times along the router from server to users. The proposed scheme is much better in term of the transmission hops needed.

The Fig. 6 (b) talks about the server hit rate, which indicates the percentage that the requested video needs to be sent from the server. We can see from the results, that the *always* performs the worst and the *fix=x* is much comparing with *always*. The *popcache* is much better comparing with *always* and *fix=x*. The proposed scheme performs better comparing with *popcache* due to the caching redundancy reduction, which leads to the caching of more contents.

Fig. 7 (a) and (b) show the *average transmission hops needed* and the *server hit rates* in the scenario, with the binary tree topology. We could observe similar performance as shown in the Cascade topology (Fig. 6), where the proposed scheme greatly reduces the transmission

[4] http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

[5] In this paper, we use the identical video to present all the possible video content for simplicity, but the proposed scheme is also applicable where there are different video.

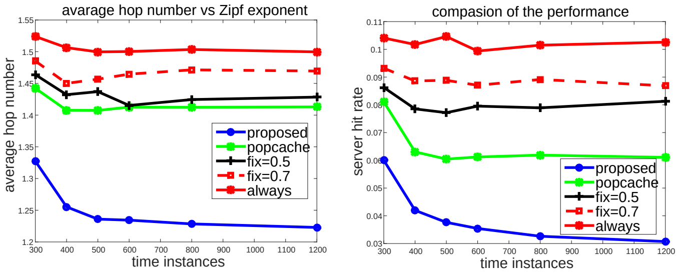


(a) average transmission hops

(b) server hit rates

Fig. 7. Performance evaluation of the CCN with Binary tree network topology.

hops needed and the server hit rates. But the gain becomes much larger comparing with the competing schemes, since the routers that are shared by multiple routers have larger chance to cache the popular video. The reason is that the popular content requested in the different routers are all needed to be via the 'shared' router, and the competing schemes have the 'redundancy' problem.



(a) average transmission hops

(b) server hit rates

Fig. 8. Performance evaluation of the caching policy at different time instances.

We also investigate the performance of the proposed scheme at different time instances, as shown in Fig. 8. This is to show the time needed from the beginning until when the system becomes stable. In this figure, the topology used is the cascade with 5 levels and s is set to be 1.4. The x-axis is the total running time of the proposed scheme, and the number of running time used to calculate the corresponding y-axis value is from $x-300$ to x. For example, with x=500, the system running 500s and the results shown are calculated based on the data from 200s to 500s. Before the $x - 300$, the data is used to update the routers' cache. From the figures, we can observe that the proposed scheme's performance will become stable as the system runs. For the server hit rate, the performance will become quite stable as the simulation started.

VI. Conclusion

Caching is one fundamental issue in CCN. In this paper, we design a new caching policy for the popularity-aware video caching in CCN, where the CCN is with known topology. The caching policy basically calculates where the video content should be allocated according to its popularity and the sizes of the routers. Hence the video will not be redundantly stored along the path from server to users. Simulations are conducted and the performance of the proposed scheme has been shown by comparing with the existed schemes.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 181–192.

[5] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urushidani, and S. Yamada, "Popcache: Cache more or less based on content popularity for information-centric networking," in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, Oct 2013, pp. 236–243.

[6] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou, "Modelling and evaluation of ccn-caching trees," in *NETWORKING 2011*. Springer, 2011, pp. 78–91.

[7] Z. Liu, G. Cheung, and Y. Ji, "Optimizing distributed source coding for interactive multiview video streaming over lossy networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 10, pp. 1781–1794, 2013.

[8] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, "Multiple description coding and recovery of free viewpoint video for wireless multi-path streaming," *J. Sel. Topics Signal Processing*, vol. 9, no. 1, pp. 151–164, 2015.

[9] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and 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.

[10] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 111–118.

[11] K. Suksomboon, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe Nakamura Motonori, M. Aoki, S. Urushidani, and S. Yamada, "On incentive-based inter-domain caching for content delivery in future internet architectures," in *Proceedings of the Asian Internet Engineeering Conference*. ACM, 2012, pp. 1–8.

[12] Y. Li, T. Lin, H. Tang, and P. Sun, "A chunk caching location and searching scheme in content centric networking," in *IEEE ICC*. IEEE, 2012, pp. 2655–2659.

[13] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong, "Popularity-driven coordinated caching in named data networking," in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*. ACM, 2012, pp. 15–26.

[14] H. Wu, J. Li, T. Pan, and B. Liu, "A novel caching scheme for the backbone of named data networking," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3634–3638.