

# Two layers cooperative caching in Content Centric Networks

Kyi Thar, Saeed Ullah and Choong Seon Hong

Department of Computer Engineering, Kyung Hee University,  
1 Seocheon, Giheung, Yongin, Gyeonggi, 446-701 South Korea.  
Email: {kyithar, saeed, cshong}@khu.ac.kr

## Abstract

Content Centric Network (CCN) is designed for content distribution like video distribution. CCN network can reduce the traffic and delay by caching the content data on each router. The original CCN caching policy stores several copies of same contents on the request path for redundancy. However, the cache space is limited for each router. Cooperative caching scheme can eliminate the overlap contents. In this paper, an Autonomous System is clustered as several groups of routers. Each group cooperatively stored the content and eliminated the duplicate content by using consistent hashing.

## 1. Introduction

CCN [1] is one of the proposed architectures for future Internet and designed to cache contents on router's cache. Two types of packet run the CCN network: Interest and Data. Interest is the packet to request data. Data or content data is the reply packet for the request and formed as a sequence of chunks. One Interest can request only one Data chunk. Interest packets are flooded to neighbor routers when the request content is not in current router. Same Data chunks are cached at each router that is on the request path. That kind of caching is good for redundancy but the utilization of the cache space is not efficient.

In order to eliminate the duplicate chunks, hash based caching is used in [3] [4] [5]. In [3], data are stored on the corresponding router by using modulo hashing. In modulo hashing, numbers of buckets are same as the number of routers. When even a router joins or moves from the group, the previous cached data location will change and almost all of the previous cached data cannot be accessed. It is light weigh hashing scheme but it has consistency and load balancing issues. In [4] authors considered network as a one Autonomous System (AS) and edge nodes make decision whether to forward data or store data. Adding a new router can cause the same problem as [3]. In [6], one hop neighbor routers periodically exchanged "cache summary". The greedy heuristic method is used to eliminate the duplicate data chunks.

In this paper, we propose two types of cooperative

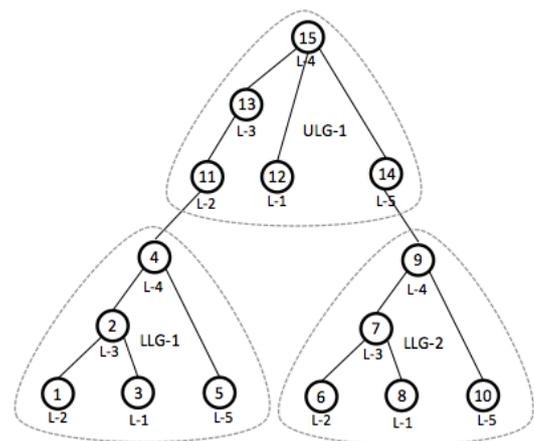


Figure 1: ULG and LLG inside AS

cache decision algorithms to eliminate the duplicate chunks and store the chunks only at the corresponding routers within one group. By using consistent hashing [2], routers can eliminate the duplicate chunks and cache the chunks effectively. Consistent hashing solves the problem of modulo hashing, which is mentioned above. In consistent hashing, numbers of buckets are independent from number of routers. The routers inside one group make a cache decision cooperatively. By doing this way, the group of routers guarantee to store the chunks without duplicate.

## 2. Cooperative Caching Algorithm

This section describes the system model and two types of caching algorithm: Cache Decision for Upper Layer Group (CDULG) and Cache Decision for Lower Layer Group (CDLLG).

---

**Algorithm 1: Cache Decision for ULG**

---

```

1: Input:  $R_k^i$  = CCN Router k
2:  $O$  = Content Object Name
3:  $O \rightarrow \text{hash} = p$ 
4: mapped p with the hash ring map
5: if p is in the range of  $R_C^i$  then
6:   Cache ( $O$ )
7:   Send ( $O$ ) to requested faces
8: else
9:   Send ( $O$ ) to  $R_k^i$ 
10:  Send ( $O$ ) to the Consumer
11: end if

```

---



---

**Algorithm 2: Cache Decision for LLG**

---

```

1: Input:  $R_k^i$  = CCN Router k
2:  $O$  = Content Object Name
3:  $O \rightarrow \text{hash} = p$ 
4: mapped p with the hash ring map
5: if p is in the range of  $R_C^i$  and  $LPC < T$  then
6:   Cache ( $O$ )
7:   Send ( $O$ ) to the Consumer
8: else
9:   Send ( $O$ ) to  $R_k^i$ 
10:  Send ( $O$ ) to the Consumer
11: end if

```

---

**2.1. System Model**

CCN routers are clustered as several groups inside one AS. We assume that the same number of routers are formed as groups. Each router possesses the key range (1, 2...n) and caches the chunks depend on its keys range. If the router possesses key (1, 2, 3), this router will be cached the chunks that hash values are 1, 2, 3. The cache sizes of the routers are heterogeneous. The keys are distributed depending on the cache size of the router. If the cache size of a router is bigger than other, that router will possess more keys than other. We skip the detail explanation about key distribution because of page limitation. Routers inside one group know each other by construction neighbor table and forward Interest and Data directly to the dedicated router.

**2.3. Cache Decision Algorithm**

In this proposal, two types of caching algorithms are used: CDULG and CDLLG. One AS consists of two groups: ULG and LLG. ULG connects with other Autonomous System (AS) or LLG and stores all the chunks corresponding to the keys of each router. LLG connects with users and stores the popular chunks depend on the popularity threshold value and keys. By this way each router can effectively use their cache. In CDA,  $O$  represents the content object name and  $R_k^i$  represents the content router  $k$  in  $i$  group.  $R_C^i$  represents the current router in  $i$  group.  $p$  represents the output of the hash value.

**CDAULG:** Whenever ULG routers receive a chunk, it checks chunk name and hash that name to know the corresponding router. If the hashed value of the chunk is within the key range of the current router, chunks will be cached in ULG. In algorithm 1, current router  $C$

receives the chunk and hashes that chunk name to find out which router needs to keep this chunk. If the hash value  $p$  of the chunk is in the range of the current router  $C$ , that chunk will be cached on current router  $C$ . At the same time, current router forwards that chunk to requested face(s). When the hash value of the chunk is not for current router  $C$ , router  $C$  forwards it to the dedicated router,  $R_k^i$ .

**CDALLG:** Whenever LLG routers receive the Data chunk, routers check two conditions, chunk name and Local Popularity Count (LPC). If the hashed value is within the key range of the current router and the LPC is greater than the threshold, chunks will be cached in LLG. As shown in algorithm 2, router  $C$  receives the data chunk, it hashes that chunk name and find out which router needs to keep this chunk. If the hashed value  $p$  is in the key range of the current router  $C$ , it will check the LPC value of receiving chunks. In here LPC of the chunk is the number of request from consumer and it is measured at the each router. When the LPC is over the threshold  $T$ , the chunk will be cached at current router and forwarded to requested face(s). Otherwise, the chunk will be just passed to the consumer or other routers via requested face(s).

**4. Performance Evaluation**

In this section, we evaluate the performance of our proposed scheme by using ccsim [7], which is a chunk-level simulator. We create the groups that consist of same number of routers. In this simulation one group consists of five routers. Routers have different cache size from (200 to 600). Clients request generated by random manner with zipf distribution. Table 1 shows the parameters used in simulation. We compare our proposed algorithm with NRR1-LCE, NRR-LCE, NRR1-LCD and NRR-LCD. In this

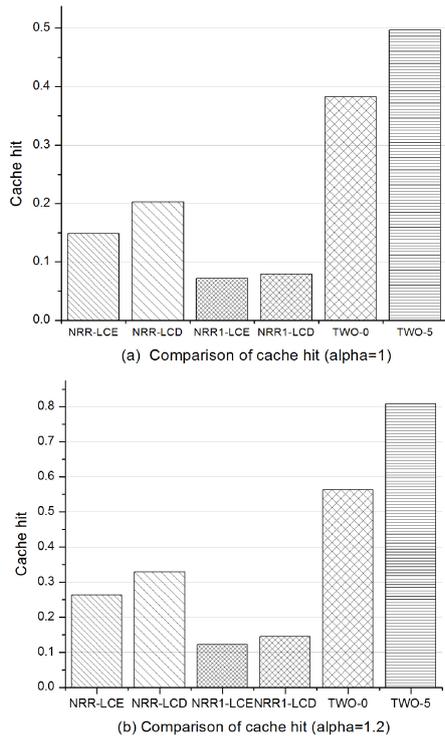


Figure 2 Comparison of cache hit

simulation, NRR1 is the original CCN forwarding algorithm, which flood the interest. NRR forwards the interest by using shortest path to the server. LCE (Leave Copy Everywhere) is the original CCN caching algorithm. LCD (Leave Copy Down) is the caching algorithm, which cache the chunks on the one hop downstream router of cache hit occurs.

We simulated the proposed scheme with two thresholds, 0 and 5. That threshold are only used in CDALLG. If the threshold value is 0, routers of LLG will be processed similar as ULG's routers. When the threshold value is 5, routers of LLG will be cached the chunks when the LPC value is greater than the threshold. Fig. 2 shows the comparison of average cache hit with proposed one and other. In figure 2 (a), the zipf parameter value (alpha) is one and our proposed scheme improves cache hit 49% than NRR-LCE, 37% than NRR-LCD, 72% than NRR1-LCE and 70% than NRR1-LCD. Fig. 2 (b) shows, the alpha value is 1.2 and proposed scheme improves cache hit 44% than NRR-LCE, 35% than NRR-LCD, 70% than NRR1-LCE and 65% than the NRR1-LCD. Our scheme outperform than other scheme in term of cache hit. This is because, one group of router stored the chunks without overlap and Interest can be directly forwarded to the dedicated nodes.

## 5. Conclusion and future work

In this paper we proposed two types of cooperative

caching algorithms. The proposed scheme is outperform than other in term of cache hit. Simulation results confirm that the proposed algorithms can improve the network cache hit than other. As a future work, we will try to deploy our scheme in CCNx [8] environment.

Table 1 Parameters used in the experiments

Parameters	Values
num repos	1
replicas	1
num clients	8
lamda	1
file size	20
alpha	1,1.2
objects	1000
DS	LCE, LCD
RS	LRU
FS	NRR, NRR1
Cache size	200, 200, 400, 500, 600

## 6. Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014(H0301-14-1020)) supervised by the NIPA (National IT Industry Promotion Agency). \*Dr. CS Hong is the corresponding author.

## 7. References

- [1] Jacobson, Van, et al. "Networking named content." Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009.
- [2] Karger, David, et al. "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web." Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. ACM, 1997.
- [3] Li, Zhe, and Gwendal Simon. "Time-shifted tv in content centric networks: the case for cooperative in-network caching." Communications (ICC), 2011 IEEE International Conference on. IEEE, 2011.
- [4] Saino, Lorenzo, Ioannis Psaras, and George Pavlou. "Hash-routing Schemes for Information Centric Networking." (2013).
- [5] Wang, Sen, Jun Bi, and Jianping Wu. "Collaborative caching based on hash-routing for information-centric networking." Proceedings of the ACM SIGCOMM, 2013.
- [6] Wang, Jason Min, Jun Zhang, and Brahim Bensaou. "Intra-AS cooperative caching for content-centric networks." Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking, 2013.
- [7] Chiocchetti, Raffaele, Dario Rossi, and Giuseppe Rossini. "ccnSim: An highly scalable CCN simulator.", IEEE International Conference on Communications (ICC), 2013.
- [8] <http://www.ccnx.org/>