CodingCache: Multipath-aware CCN Cache with Network Coding

Qinghua Wu[†] [⋆], Zhenyu Li[†], Gaogang Xie[†]
[†]Institute of Computing Technology, Chinese Academy of Sciences, China
[⋆]University of Chinese Academy of Sciences, China
{wuqinghua, zyli, xie}@ict.ac.cn

ABSTRACT

Content Centric Networking (CCN) performance by definition depends on the in-network caching efficiency. We propose *CodingCache* which utilizes network coding and random forwarding to improve caching efficiency under multipath forwarding. Its advantage is that existing caching strategies can be easily incorporated with it for better performance. We evaluate CodingCache by extensive simulation experiments with the China Telecom network topology and a unique dataset consisting of video access logs from the PPTV system. The results demonstrate that compared with the CCN caching strategy, CodingCache improves the cache hit rate by about 60%.

Categories and Subject Descriptors

C2.1 [**Computer-Communication Networks**]: Network Architecture and Design - Distributed Networks

Keywords

CCN, Caching, Multipath Forwarding, Network Coding

1. INTRODUCTION

Content Centric Networking (CCN) [1] performance is largely impacted by cache efficiency. Previous studies like [2] on CCN cache, which only consider the caching strategies, such as caching decisions and replacement policies, could not achieve high performance. A main factor that these works ignore is the potential benefits brought by the available multiple paths from a router node to the requested content chunks.

The adaptive forwarding [3] in CCN does not result in improved caching efficiency when there are multiple paths available[4]. The main reason is that content chunks are cached along individual paths with the same probability.

In this poster, we present *CodingCache*, a multipath-aware CCN cache strategy with network coding. Instead of caching every single content chunk, CodingCache caches a coded block of several content chunks. In order to achieve diverse cached contents in the network, routers recode the linearly independent coded blocks constituted by the same original chunks into a new coded block. CodingCache exploits a random forwarding strategy to uniformly select forwarding interfaces, achieving maximized average cache efficiency. Consumers decode the coded blocks to get the original content chunks. One of the prominent features of CodingCache is that existing caching strategies, such LCD[2], can be easily incorporated with it for better caching performance.

We use the China Telecom network topology [5] and video requests logs collected from PPTV[1]. The results demonstrate that CodingCache improves the caching performance by about 60%, compared with the native CCN caching strategy. The improvement is more notable when the number of available forwarding interfaces becomes larger.

The remainder of this poster is organized as follows. Section 2 presents CodingCache and analyzes its overhead. In section 3 we evaluate CodingCache with extensive experiments. Section 4 concludes our work.

2. CODINGCACHE

In this section, we propose CodingCache (CC) to improve caching efficiency with network coding under multipath forwarding. The main idea of CodingCache is to achieve diverse cached contents in the network by Random Linear Network Coding (RLNC)[6] and recoding.

2.1 Caching Strategy

In CodingCache, content chunks are grouped into segments. The number of chunks in a segment is t, which is a design parameter. Chunks in a segment are coded into one *block* with the RLNC. Consumers get the original chunks in a segment by decoding t linearly independent coded blocks.

The content server codes the chunks of a segment using RLNC. Due to the randomization of RLNC, the coefficient vectors of two blocks corresponding to the same segment are likely to be linearly independent. We say two coded blocks of the same segment are linearly independent if their coefficients are linearly independent. Suppose a consumer u requests for the chunk K of the segment S. The coefficient vectors of all blocks he has obtained corresponding to S are appended in the interest packet.

Upon receiving an interest from a downstream node, the router first checks whether there is a coded block corresponding to S in its cache with linearly independent coeffi-

[1]http://www.pptv.com/

ICN'13, August 12, 2013, Hong Kong, China.
ACM 978-1-4503-2179-2/13/08.

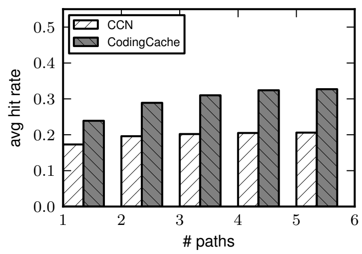

Figure 1: Comparison of cache hit rate in CCN and CodingCache.

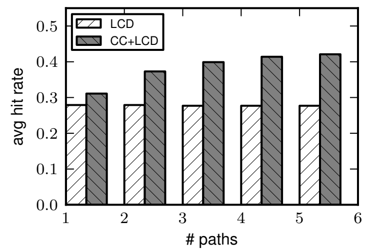

Figure 2: Comparison of cache hit rate by LCD and CodingCache (CC).

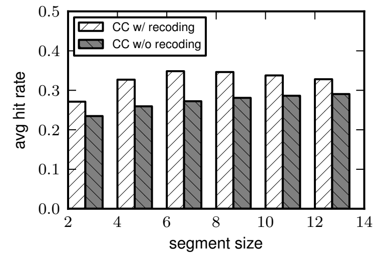

Figure 3: Comparison of hit rate by CodingCache w/ and w/o recoding.

cient vector. If there is, the router responds with the coded block to the downstream node, which means a *cache hit*. Otherwise, it forwards the interest to its upstream node.

On receiving a coded block from its upstream node, the router stores it into its cache according to existing caching decisions, like LCD[2]. A router might cache several coded blocks of a segment, which leaves a smaller space to cache blocks of other segments. We solve this by *recoding* coded blocks of the same segment on intermediate routers. Suppose that a router receives a coded block B_2 of segment S and there already has a block B_1 (linearly independent with B_2) of S in its cache. The router recode them to get a new block $B_3 = c_1B_1 + c_2B_2$, where c_1 and c_2 are randomly nonzero coefficients. Instead of replacing a block to store B_2, we replace B_1 with B_3.

2.2 Forwarding Strategy

CodingCache can simply adopt the adaptive forwarding [3] in CCN as its forwarding strategy. However, we find through Theorem 1 that the overall hit rate is maximized with a strategy forwarding interests uniformly at random. The proof of the theorem is omitted due to space limit.

THEOREM 1. *The cache hit rate is maximized in CodingCache with a forwarding strategy that forwards interests uniformly at random on routers.*

3. PERFORMANCE EVALUATION

In this section, we take various experiments to evaluate CodingCache. The network topology is from the provincial connections of China Telecom collected by [5]. Content requests are generated with a unique dataset collected from PPTV. Video contents are divided into chunks of equal size. A router node is assigned to cache 10,000 chunks. By default, 4 chunks constitute a segment to be coded.

3.1 Experimental Results

Figure 1 compares the native CCN caching strategy (noted as CCN) with CodingCache using the same caching decision, in terms of cache hit rate by varying the average number of paths to the content server. We see that CodingCache outperforms CCN even with only one path. The hit rate in CCN does not vary greatly with the growth of the number of paths. In contrast, the hit rate in CodingCache increases notably as it exploits the benefits of multiple paths by caching linearly independent blocks of individual contents along different paths. The results confirm our previous analysis. In particular, compared with CCN, CodingCache improves the hit rate from 20% to 32%, *i.e.* a 60% improvement rate.

As CodingCache is orthogonal to existing caching strategies focusing on cache decision and replacement, we incorporate LCD [2] with CodingCache. LCD moves the cached chunks one hop closer to the consumer on each request. In Figure 2, the hit rate of LCD does not vary with the increase of the number of paths to the content server. The results confirm the findings in [4]. The hit rate is further improved to 40% after incorporating with CodingCache when there are more than 4 paths available to the content server.

As we argued in Section 2, recoding in intermediate routers facilitates the diversity of cached blocks among routers in the network. Figure 3 illustrates the effect of recoding by varying the number of chunks t coded into one block. The hit rate of CodingCache with recoding increases with the growth of t until t reaches 6, after that the rate decreases slightly. With a larger t, the coded block contains information of more chunks and therefore the block can match more requests. This explains the increase of hit rate when $t \leq 6$. However, a larger t also means that consumers need to retrieve more linearly independent blocks to get the original chunks. The probability of caching a larger number of such blocks in intermediate routers is indeed lower.

4. CONCLUSION

We have detailed the design of CodingCache and evaluated it with real-life topology and request traces. The results have shown the advantage of CodingCache on cache hit rate over other caching strategies. Existing caching strategies can easily incorporate with it for better performance.

5. ACKNOWLEDGEMENTS

This work was supported by National Basic Research Program of China with grant 2012CB315801, by NSFC with grants 61133015 and 61272473, by CNGI with grant CNGI-12-03-007.

6. REFERENCES

[1] V. Jacobson and et. al., “Networking named content,” in *CoNEXT*, 2009.
[2] N. Laoutaris and et. al., “The lcd interconnection of lru caches and its analysis,” *Performance Evaluation*, 2006.
[3] C. Yi and et. al., “Adaptive forwarding in named data networking,” *ACM SIGCOMM CCR*, 2012.
[4] G. Rossini and D. Rossi, “Evaluating ccn multi-path interest forwarding strategies,” *Comp. Comm.*, 2013.
[5] Y. Tian and et. al., “China’s internet: Topology mapping and geolocating,” in *INFOCOM*, 2012.
[6] T. Ho and et. al., “On randomized network coding,” in *Allerton CCC*, 2003.