# Efficient Content Delivery and Caching Scheme Using Network Coding in CCN

Yan LIU[1,*] and Shun-zheng YU[1]

[1] School of Information Science and Technology

Sun Yat-San University, Guangzhou 510006, PR China

* liuyan_sysu@163.com

**Keywords:** Future Internet, CCN, Network Coding, Content Delivery, Caching.

**Abstract.** Content Centric Networking (CCN) is a new network architecture around content dissemination and retrieval, which treats content as a primitive and routes content by name. Differently from the conventional content dissemination and retrieval of the CCN, we introduce a novel content delivery scheme based on network coding to support multisource transfers and to reduce both the transmission amount and time required for getting the content. A network coding based caching mechanism (NCC) is designed for guaranteeing the cached blocks can be reused. In our scheme, one coded block can satisfy multiple Interests for different chunks from different consumers. A special request-response mechanism is designed to avoid requesting an independent coded block that may occur in a random network coding method. Our simulation results show that the proposed scheme achieves better performance than conventional schemes of CCN in terms of average download time and cache hit rate.

## Introduction

Since the primary uses of the Internet are content distribution and retrieval, the current Internet architecture which was designed for end-to-end communication between hosts is facing unprecedented challenges in many aspects. Therefore, the concept of Information-Centric Networking (ICN) is introduced [1]. The architecture of Content-Centric Networking (CCN) proposed by Van Jacobson et al. in [2] is the most popular one, it treats content as a primitive. In CCN, the content is cached on all nodes along the delivery path and thus has copies cached in multiple nodes. Each node can respond with a cached copy to a tree created by the interests from a set of consumers. If each node caches and responds only a few of blocks of the content, then different blocks of the content may meet on their way to the set of consumers.

Network coding which is originally proposed by R. Ahlswedeet al. in 2000 [3], has attracted a great deal of attention. Differently from the traditional routing of storing and forwarding data messages only, network coding allows intermediate nodes to encode and decode messages. The system with network coding performs much better than the system without network coding in terms of throughput, robustness, security and load balancing [4][5]. Recently, several works have shown that network coding also benefits ICN. The biggest problem for applying network coding into ICN is “How to provide consumer with linearly independent blocks”. Based on the in-network caching mechanism, a coded block may be cached in multiple nodes along its transmission path. The same coded block or linearly dependent blocks that were cached in multiple nodes may be provided to the same consumer in responding its multicast request. Several solutions have been proposed. In some centralized schemes [6][7], center routers are responsible for determining the optimal caching strategy and optimal content routing to provide independent blocks. In some distributed schemes [8][9], the information about coded blocks having collected must be appended in the Interest packet to get independent block. With this information, the router can tell whether it can serve the request or not. In our previous work [10], only original blocks are cached by content router (CR) for responding the future Interest to guarantee the cached blocks reusable.

In this paper, we design a special Interest forwarding strategy for getting blocks simultaneously from multiple nodes. Interests received by a node that are coming from multiple consumers for different chunks of the same content are aggregated to eliminate duplicate and re-separated to forward to different directions. An on-path non-cooperative caching mechanism is designed for guaranteeing the cached blocks can be reused. One coded block can satisfy multiple Interests for different original blocks from different consumers. A chunk-level cache replacement policy is proposed to effectively enhance cache hit ratio.

We experimentally evaluate the proposed scheme by comparing with other three schemes. The simulation results show that our scheme performs much better than those schemes in terms of average download time and cache hit rate.

## System Model

In this section, we introduce a novel model for chunk-level content distribution and network coding based caching strategy over Content-Centric Network. The main idea of our scheme is that by dividing content into multiple blocks and distributing them over the network, a consumer can multicast an Interest to get them simultaneously from multiple nodes. The network coding technique is used for encoding the blocks cached in routers. This will reduce both the transmission amount and time required for getting the content.

### Content Publishing and Requesting

When a large file is generated by node S, it will be divided into h generations, and each generation contains N blocks. The generation size is decided by file size, and N is a constant decided by file type. CCN nodes only combine the blocks in the same generation. Moreover, if a consumer asks for the $k^{th}$ block, it will receive the $k^{th}$ blocks in all generations. In order to state conveniently, we don't explicitly state every block belonging to which generation in the rest of the paper.

We treat N blocks as vectors, $\overline{V_1},...,\overline{V_N}$ , over a finite field $F_q$ . The server creates coefficient vectors for these blocks, $V_1, . . . , V_N$, where $V_i = (0, . . . , 1, . . . )$ , and its elements are zero except that the $i^{th}$ one is 1. We name the $k^{th}$ original block of file *C* as $C_k$ , k=1, 2, …, N.

When a user wants to access a file *C*, it will generate an Interest $(C,U,N)$ for the file, where $U=\{C_1,...,C_N\}$ the set of original block indices, N is the total number of coded blocks the user requires. Before sending the Interest, forwarding interface of each original block is determined by looking up FIB. If more than one interfaces are determined, the Interest $(C,U,N)$ is divided into multiple interests $\{Interest(C,U_i,N_i)\}$, each of which contains a subset $U_i$ of U and is forwarded to one of the interfaces for a subset of the destinations, where $U=\bigcup_i U_i$ and $U_i \cap U_j = null$ .

### Interest Packet Processing

When an Interest $(C, X, n)$ is received from an interface $i$, a longest-match look-up is done on the Content Name of the node. For each chunk $C_k$ in X, the node's content store is first searched for original block, and the original block is sent out of interface $i$ if present, and $X^{'} = X \setminus C_k$. If no such original block exits, the node responds a coded blocks in which the $k^{th}$ one is a nonzero coefficient if present, and $X^{'} = X \setminus C_k$. One coded block only can be used once during the process of responding the same Interest $(C, X, n)$, thus can ensure that the consumer receives linearly independent coded blocks. For instance, coded block $cb_1 = e_1 ob_1 + e_2 ob_2$ can be treated as chunk $C_1$ for responding Interest $(C, \{C_1, C_3\}, 2)$ and chunk $C_2$ for Interest $(C, \{C_2, C_4\}, 2)$. If $X' \neq null$, for each $C_k$, the node checks its PIT, adds interface $i$ into face list if PIT matches, otherwise establishes a new PIT entry for $C_k$, and $X'' = X' \setminus C_k$. If $X'' \neq null$, the request for $(C, X'', n'')$ will be divided into several Interests according to forwarding interface for each chunk before forwarding to next hops.

### Data packet Processing $C_1$

Table 1. An example of PIT

| Content Name | Chunk ID | Faces |
|---|---|---|
| $C$ | $C_1$ | 1 |
| | | 2 |
| | $C_2$ | 3 |
| | | 2 |
| | $C_3$ | 4 |

Data packet processing is relatively simple. When a Data packet, Data $(C, Y, block)$, is received from an interface, a longest-match lookup is done on the Content Name of the node. A PIT match means the Data was requested by this node. Then the Data is added to Content Store if the Data carries original block. If *block* is a coded block, one *block* can consume multiple Interests received from different interfaces. As shown in Table 1, a received coded block $cb_1 = e_1 ob_1 + e_2 ob_2$ can be used for responding Interests for $C_1$ received from interface 1 and 2, and Interest for $C_2$ received from interface 2, but it cannot be used for responding Interest for $C_2$ received from interface 2 at the same time, because Interest $(C, \{C_1, C_2\}, 2)$ received from interface 2 maybe come from same consumer.

### Caching Strategy

When an original block $C_k$ is received, the Content Store is searched for coded block $cb$ generated by chunk $C_k$ and other chunks, $cb$ is replaced by $C_k$ if exists. Otherwise the node caches $C_k$ without any change.

Since memory is limited in nodes, cache replacement policy is necessary. When the node's capacity is exhausted, first, the candidate content $C$ to be replaced is determined by existing content-level cache replacement policy, e.g. LRU. Suppose the cache space for caching content $C$ is $n$ units (one unit for one block), if $n > 1$, the Content Store is searched for coded block $cb$, and $cb$ is replaced by a random linear combination of $cb$ and a randomly selected original block, the selected original block is deleted. If no such coded block exists (i.e. all blocks in content store are original blocks), two randomly selected original blocks are replaced by one random linear combination of them. Hence, when a content is frequently accessed, its storage size in the cache will grow and the blocks in content store are original blocks, whereas if a content has not been accessed for long time, the number of cached original blocks will shrink until zero. Because one coded block can satisfy multiple Interests for different original blocks, our chunk-level cache replacement policy can effectively enhance cache hit ratio.

## Simulation Results

In this section, we study the performance of our system and compare it with other caching schemes : Cache Everything Everywhere (CEE) [2], Leave Copy Down (LCD) [11], Move Copy Down (MCD) [11].

We implement our scheme described in previous sections by using an event-driven network simulator written in Matlab. We use Dijkstra algorithm for routing. All the links have a bandwidth of 1Gbps. We set file size to 1GB and divide the file into 100 chunks, 10 chunks are grouped into a generation, and chunks in the same generation are coded into blocks with network coding. Consumers generate the content request according to Poisson process. We use random linear network coding to created coded blocks to transmit, i.e. choose integers of the coefficients randomly from a finite field.

As Figure 1 shows, the average download time of different caching schemes decreases as the Zipf parameter α increases. With the increasing of number of requests, the requested content will be cached by more nodes, the consumers can get the requested content from the nodes much closer to them. Thus the average download time decreases as the number of requests increases. The reason why NCC scheme always performs better than other schemes is because NCC supports multi-source content delivery.

The cache hit ratio of different caching schemes is shown in Figure 2, which is defined as the ratio of the number of Interests served by the caches to the total number of Interests responded. With the increasing of Zipf parameter α, the content with higher popularity will be cached by more nodes, and then the cache hit ratio increases. With the increasing of number of requests, the users can get the requested content from the nodes much closer to them. The performance of NCC achieves much higher cache hit ratio than other schemes.

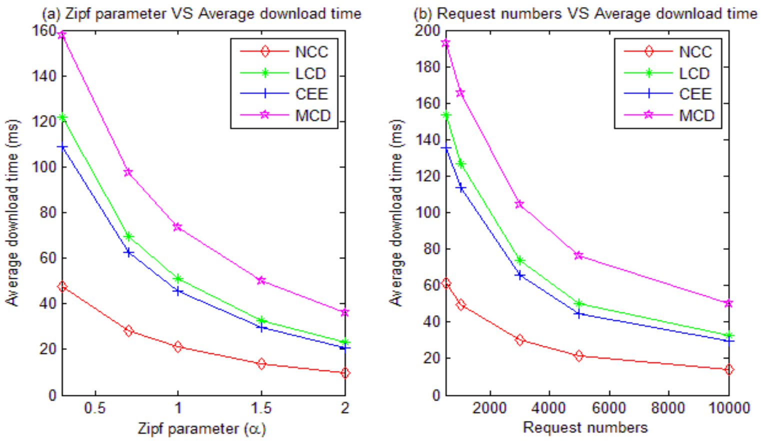

Figure 1. Average Download Time

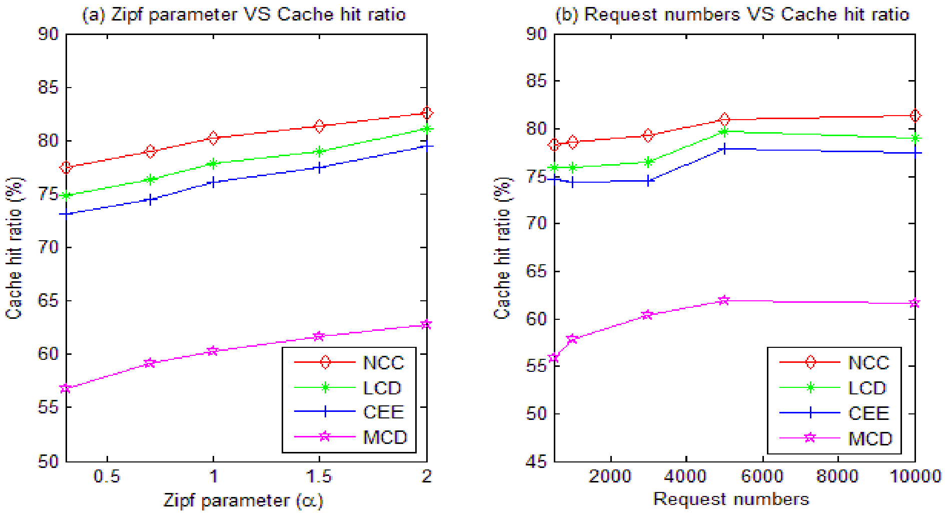

Figure 2. Cache Hit Ratio

## Conclusion

In this paper, we have proposed and presented a novel content delivery scheme, which supports multi-source transfers. The main idea of our scheme is that by dividing a content into multiple blocks and distributing them over the network, a consumer can multicast an Interest to get them simultaneously from multiple nodes. CCN nodes encode the blocks that are cached in content store. The performance of NCC has been investigated over CCN. The results demonstrate that the proposed NCC scheme performs much better than other three schemes in terms of average download time and cache hit rate. Furthermore, our work indicates that there is a potential for significant benefits by deploying network coding over Content Centric Networking.

## Acknowledgement

This work was supported by Key Program of NSFC-Guangdong Joint Funds (Grant No. U0735002), Guangdong Natural Science Foundation (Grant No. 2014A030313637) and Natural Science Foundation of Guangdong Province, China (Grant No. 2014A030313130).

## References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, Communications Magazine, IEEE 50 (7) (2012) 26–36. doi:10.1109/MCOM.2012.6231276.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, R. L. Braynard, Networking named content, in: Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT ’09, ACM, New York, NY, USA, 2009, pp. 1–12. doi:10.1145/1658939.1658941.

URL http://doi.acm.org/10.1145/1658939.1658941

[3] R. Ahlswede, N. Cai, S.-Y. Li, R. Yeung, Network information flow, Information Theory, IEEE Transactions on 46 (4) (2000) 1204–1216. doi:10.1109/18.850663.

[4] P. A. Chou, Y. Wu, K. Jain, Practical network coding (2003).

[5] C. Gkantsidis, P. Rodriguez, Network coding for large scale content distribution, in: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Vol. 4, 2005, pp. 2235–2245 vol. 4. doi:10.1109/INFCOM.2005.1498511.

[6] H. R. Sadjadpour, A new design for information centric networks, in: Information Sciences and Systems (CISS), 2014 48th Annual Conference on, IEEE, 2014, pp. 1–6.

[7] P. Talebifard, H. Nicanfar, V. C. Leung, A content centric approach to energy efficient data dissemination, in: Systems Conference (SysCon), 2013 IEEE International, IEEE, 2013, pp. 873–877.

[8] G. Zhang, Z. Xu, Combing {CCN} with network coding: An architectural perspective, Computer Networks 94 (2016) 219 – 230. doi: http://dx.doi.org/10.1016/j.comnet.2015.11.008.

URL http://www.sciencedirect.com/science/article/pii/S1389128615004144

[9] D. Nguyen, M. Fukushima, K. Sugiyama, A. Tagami, Conat: A network coding-based interest aggregation in content centric networks, in: 2015 IEEE International Conference on Communications (ICC), IEEE, 2015, pp. 5715–5720.

[10] Y. Liu, S.-Z. Yu, Network coding-based multisource content delivery in content centric networking, Journal of Network and Computer Applications 64 (2016) 167 – 175. doi:http://dx.doi.org/10.1016/j.jnca.2016.02.007.

URL http://www.sciencedirect.com/science/article/pii/S1084804516000813

[11] N. Laoutaris, H. Che, I. Stavrakakis, The lcd interconnection of lru caches and its analysis, Performance Evaluation 63 (7) (2006) 609–634.