# Caching Framework in Content Delivery Networks Based on Matching Theory

Chuan Pham, Kyithar, Tra Le Thi Huong and Choong Seon Hong
Department of Computer Engineering, Kyung Hee University
E-mail: {pchuan, kyithar, huong_tra25, cshong}@khu.ac.kr

**Abstract**: Caching contents in content delivery networks (CDNs) has many attentions in recent years to reduce the network traffic and network delay. In this work, we propose a cooperating caching model which supports adaptive bit-rate streaming in content delivery networks. A matching algorithm is applied to maximize the total user satisfaction. With the rapid convergence, our proposed algorithm can be used in the practical model in terms of the large amount of contents in the network.

## 1. Introduction

According to the Cisco VNI report [1], videos will be soon a dominant traffic type in the Internet. In their mention, the video traffic will be accounting for 69% of the Internet traffic in 2017 where nearly a million minutes of video will be transferred. Ideally, the users can obtain the contents at the closest router instead of downloading from the original sources. Therefore, to improve performance of content delivery to the end users, all the contents should be cached in each storage node in the network. However, with a huge of contents in the Internet network, the storage capacity of each node cannot satisfy all caching requests. Hence, many current works usually cache only the most popular contents. If a node receives a request that is not in the caching storage, the content can be fetched from the other nodes in the network. Although this is a simple solution, it offers high traffic inside the network. In [2], the authors proposed the cooperative caching in content delivery networks (CDNs) and illustrated that it can reduce amount of traffic transferred.

On the other hands, to reduce the overhead of traffic, adaptive bit-rate (ABR) streaming also becomes an efficient video streaming technique in the Internet nowadays. It can utilize a huge of the network resource. With ABR, each content is encoded into different representations corresponding to different playback rates. However, a few works [3, 4] model specifically for ABR streaming in CDN network. In [4], the authors modeled the caching problem as an optimization problem, but their work remained some weakness points. For example, the constraint is not flexible since each video content has a limited storage capacity. They also did not take into account the cooperative caching.

Our work addresses the cooperative caching problem. The cache nodes cooperate to leverage caching capability of each other. The amount of contents stored on each node depends not only on the storage capacity, but also on the demands of the contents and the link costs. Based on the matching theory, we propose an algorithm to solve the optimization problem that maximizes the number of concurrent requests in the network

## 2. The problem formulation

We consider a CDN network with N cache nodes providing the caching service to the users, and the network is managed by a single administrative domain. The system is studied for one time period *T*. All nodes in the network collaborate in caching to reduce the total link cost. When a user requests content m at node i, depending on the network condition, the $k^{th}$ representation of content m can be stored in node i, i.e., $x_{ki}^m = 1$, or fetched from node $j$, i.e., $x_{kj}^m = 0, y_{kj}^m = 1$. We assume that all the contents are assumed to be cached in the CDN cluster. The model of cooperative caching in CDN is shown in Figure 1.

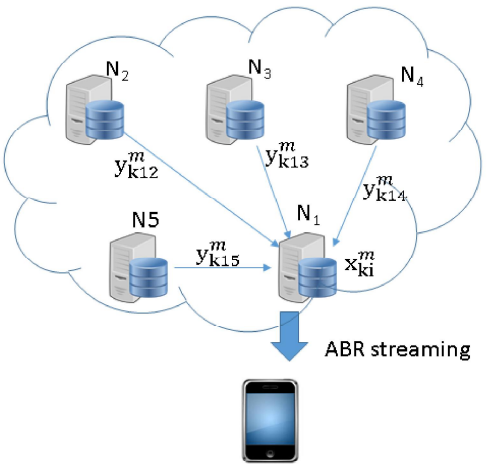


Figure 1: The cooperative caching in a CDN. To serve resquests on content m at node 1, node 1 can receive the amount of content from other nodes, if it does not cache the content.

Table 1: Main notations

| **Parameters**: | |
|---|---|
| M | Set of contents. |
| N | The set of storage node in the network. |
| $B_i$ | Storage capacity of node *i*. |
| K | Maximum number of representations of a content. |
| $U_k$ | User utility when receiving one unit of content at $k^{th}$ representation. |

| | |
|---|---|
| $s_m^k$ | Size of content m at $k^{th}$ representation. |
| $d_i^m$ | Demand for content *m* at node *i*. |
| $c_{ij}$ | Cost for one unit of data from node *j* to node *i*. |
| **Variables**: | |
| $x_{ki}^m$ | Indicating node *i* has content *m* at $k^{th}$ streaming rate. |
| $y_{kij}^m$ | Indicating content *m* at $k^{th}$ streaming rate on node *j* serves the demand on node *i*. |

With the notations in Table 1, we consider the total network cost for serving demand as follows

$$\sum_{m\in M}\sum_{i\in N}\sum_{k\in K} s_k^m d_i^m \left(\sum_{j\in N} c_{ij} y_{kij}^m\right).$$

On the other hand, when a user requests content m at node *i*, the user satisfies with the $k^{th}$ representation with $U_k$. Therefore, we consider the utility when obtaining $k^{th}$ representation of the content m from node *i* is $U_k s_k^m d_i^m (x_{ki}^m + \sum_{j\in N, j\neq i} y_{kij}^m)$. So, the total utility is

$$\sum_{m\in M}\sum_{i\in N}\sum_{k\in K} U_k s_k^m d_i^m (x_{ki}^m + \sum_{j\in N, j\neq i} y_{kij}^m)$$

The main purpose is to maximize the network performance (i.e., the total utility). The optimization problem can be formulated as follows.

$$\text{Max.} \sum_{m\in M}\sum_{i\in N}\sum_{k\in K} s_k^m d_i^m (U_k x_{ki}^m + \sum_{j\in N, j\neq i}(U_k - c_{ij}) y_{kij}^m) \quad (1)$$

$$\text{st.} \sum_m \sum_k s_k^m x_{ki}^m \leq B_i, \forall i \in N, \quad (2)$$

$$y_{kij}^m \leq x_{kj}^m, \forall k \in K, i, j \in N, i \neq j, m \in M, \quad (3)$$

$$\sum_{k\in K}(x_{ki}^m + \sum_{j\in N, j\neq i} y_{kij}^m) = 1, \forall i \in N, m \in M \quad (4)$$

$$x_{ki}^m, y_{kij}^m = \{0,1\}, \forall k \in K, i, j \in N, i \neq j, m \in M \quad (5)$$

Constraint (2) is the storage capacity constraint of each node. Constraint (3) means that the amount of content m from node j serving the request at node i must be less than the amount of content m stored at node j. Constraint (4) guarantees that the amount of content m stored at node i and the total received amount of m for all representations must be equal to the whole content in order to serve the request (from the assumption that all the contents are cached inside the network).

This problem is an integer linear program which is solved by a high complexity algorithm. In the next section, we advocate a matching approach to solve this NP-hard problem.

## 3. Stable matching in the cooperative caching model

We propose a matching algorithm for the cooperative caching model where the representations and nodes are relatively stable. Resource management in CDN can be naturally cast as a stable matching problem. Broadly, it can be modeled as a many-to-one matching: college admission problem [6] where one node can enroll multiple representations but one representation can only be cached to one node.

### 3.1. Background

A college admissions problem is a four-tuble $< C, I, q, \succ >$ where *C* is a finite set of colleges, *I* is finite a set of student, $q = (q_c)_{c\in C}$ is a vector of college capacities and $\succ = (\succ_l)_{l\in C\cup I}$ is a list of preferences.

***Definition 1****: A matching for college admissions is a correspondence* $\mu: C \cup I$ *such that:*

*1.* $\mu(c) \subseteq I$ *such that* $|\mu(c)| \leq q_c$ *for all* $c \in C$,

*2.* $\mu(i) \subseteq C$ *such that* $|\mu(i) \leq 1|$ *for all* $i \in I$, *and*

*3.* $i \in \mu(c)$ *if and only if* $\mu(i) = \{c\}$ *for all* $c \in C$ *and* $i \in I$.

***Definition 2****: A matching* $\mu$ *is blocked by a college* $c \in C$ *if there exists* $i \in \mu(c)$ *such that* $\emptyset \succ_c i$. *Vice versa, a matching* $\mu$ *is blocked by a student* $i \in \mu(c)$ *such that* $\emptyset \succ_i \mu(i)$.

*Definition 3: A matching* $\mu$ *is blocked by a pair* $(c, i) \in C \times I$ *if*

*1.* $c \succ_i \mu(i)$, *and*

*2. a) either there exists* $j \in \mu(c)$ *such that* $\{i\} \succ_c \{j\}$, *or*

*b)* $|\mu(c) < q_c|$ *and* $\{i\} \succ_c \emptyset$.

***Definition 3****: A matching is stable if it is not blocked by any agent or pair.*

### 3.2. Stable matching in cooperative caching model

The optimization in section 2 reflects a policy goal that maximizes the total utility. However, the computation is quite expensive due to their combinatorial and the large scale of the contents. In this paper, we consider each node in the CDN network as a college with a storage capacity $B_i$, and each representation of content m becomes as a student.

To design the preference list for nodes *i* and the representation *k* of content *m*, we consider the popularity of contents in time slot t at each node. For example, given 2 contents A and B at time slot t, each content has two representations such as base-line and high representation. Based on the popularity of the content in history, node 1 can rank each representation ($\text{node1}: A_{high} \succ B_{high} \succ B_{base} \succ A_{base}$. It implies that node 1's first choice of caching is $A_{high}$, second choice is $B_{high}$).

On the side of the content, the preference list of each representation is based on the total requests from node *i*. For example, the number of requests for content A with high performance comes to node 1 is greater than node 2, so $N_1 \succ_{A_{high}} N_2$.

### 3.3. Stable caching algorithm

Based on the algorithm college-proposing deferred acceptance algorithm (Gale and Shapley 1962), we can devise an algorithm that produce a weak stable solution.

*Step 1:* Each node *i* proposes to its top $B_i$ acceptable representations. Each representation rejects any unacceptable proposals, if more than one acceptable proposal is received, she "hold" the most-preferred and rejects the rest

In general, at

*Step k:* any node *i* who was rejected at step *k-1* by any representation proposes to its most-preferred $B_i$ acceptable representation who have not yet rejected it. Each representation

"holds" her most-preferred acceptable offer to date and rejects the rest.
The algorithm terminates when there are no more rejections. Each representation is matched with the node she has been holding in the last step. However, with the limitation storage capacity of nodes, some representations reject all nodes from their preferred lists. It means that these rejected representations cannot cache.

***Example 1***: *There are two nodes $N_1$, $N_2$ with capacity respectively $B_1$=3, $B_2$=1.5, and three representations $R_1$, $R_2$ (have respective size 1, 1.5, 2) of the requested content m. The preferences are as follows*

$$N_1: R_1 \succ R_2 \succ R_3, N_2: R_1 \succ R_2$$
$$R_1: N_1 \succ N_2, R_2: N_2 \succ N_1, R_3: N_1 \succ \emptyset$$

*The result of matching algorithm is as follows* $N_1: R_1, R_3; N_2: R_2$.

## 4. Simulation results

We now consider the recommended representation set of Apple's HTTP live streaming [7]. In our simulation, we use three levels of representation (low, medium and high performance), which are chosen from base-line, additional main and additional high profile. The size of representation is calculated as follows

$$\mathrm{s} = \mathrm{r} \times \frac{\mathrm{T}}{8 * 1024}\ MB,$$

where $r$ is the corresponding bit-rate (in Kbps) of the representation and $T$=600 is the video playback time in seconds.

We compare our algorithm with the linear programming (LP) method with the simulation from 3 to 10 nodes. Our result nearly obtains the optimal value. Especially, when the number of nodes increases, our results is closer to the optimal curve, as depicted in the Figure.2.

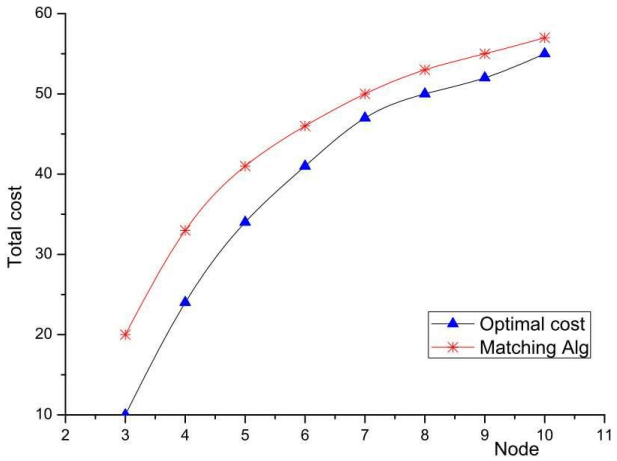


Figure 2: The comparison of matching and LP method

About the convergence, we simulate a cluster network including 10000 contents stored on 10 cache nodes. Each node has 100GB of storage capacity. Our method can converge after 97 minutes, meanwhile the LP runs with more than 10hours.

## 5. Conclusion

In this paper, we propose a matching algorithm for the cooperative caching in CDN which supports adaptive bit-rate streaming. The performance and the bandwidth cost of the network nearly obtain the optimized. The simulation demonstrates the rapid convergence of our algorithm that can be applied in the practical model of the CDN network.

## Acknowledgement

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2015.

*Dr. CS Hong is the corresponding author.

## References

[1] Cisco. Cisco visual networking index: Forecast and methodology, 2012-2017. 2013.
[2] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In Proceedings of the IEEE Conference on Computer Communications, INFOCOM'10, pages 1 {9, march 2010.
[3] H. Ahlehagh and S. Dey. Adaptive bit rate capable video caching and scheduling. In Proceeding of IEEE Wireless Communications and Networking Conference, WCNC'2013, pages 1357{1362, April 2013.
[4] W. Zhang, Y. Wen, Z. Chen, and A. Khisti. Qoe-driven cache management for http adaptive bit rate streaming over wireless networks. IEEE Transactions on Multimedia, 15(6):1431{1445, Oct 2013.
[4] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In Proceedings of the Second Annual ACM Conference on Multimedia Systems, MMSys '11, pages 157{168, New York, NY, USA, 2011. ACM.
[5] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In Proceedings of the IEEE Conference on Computer Communications, INFOCOM'10, pages 1 {9, march 2010.
[6] Ehlers, L. (2006), Respecting Priorities when Assigning Students to Schools, mimeo.
[7] Apple. Using http live streaming. [Online]. Available: http://goo.gl/fJIwC.