International Journal of Control and Automation
Vol.7, No.6 (2014), pp.231-244
http://dx.doi.org/10.14257/ijca.2014.7.6.23

Enhance Performance of Content Delivery Network Using Provider Oriented Hierarchical Corporative Proposal

Tuyet Mai Thi Anh, Hung Phan Huu, Tien Nguyen and Chul Soo Kim
Department of Computer & Telecommunication, Inje University, Busan, South Korea

{maithianhtuyet, hungphan88, thanhtien501}@gmail.com, charles@inje.ac.kr

Abstract

Content Delivery Network (CDN) has become increasingly popular over the past decade with the expectation to provide a better performance and a more scalable, reliable and available content delivery. This can be achieved by delivering the content from caches which are located closer to the end-users than the original server. However, the performance of CDN system is affected by the collaboration among CDN servers. This paper proposes a collaborative caching among CDN servers known as provider oriented hierarchical cooperative content outsourcing (POHC). To highlight the advantages of this propose, a mathematic analytical model is given to compare with the non-cooperative pull-based content outsourcing (Non-CPB). OMNeT++ software is used for simulation in order to verify the result of the analytical model. The consequence from the analytical model matched well with the result from the simulative experiment in term of packet loss, better reliability and less bandwidth consumption versus the Non-CPB mode. After collecting relevant status of all CDN servers in online network, this paper uses DBSPT-Circular-PO algorithm to build POHC with root, which is the content provider.

Keywords: *Content Delivery Network, hierarchical cooperative content outsourcing, content outsourcing.*

1. Introduction

Nowadays, user demand is increasing faster than the telecom infrastructure development. It is especially true when the explosion of smart network devices from various kinds of technology allows users to connect to the internet easily. In addition, the exponential growth in processor performance and storage capacity is changing our view of computing. Our focus has shifted away from centralized, hand-choreographed systems to global-scale, and distributive self-organizing complexes that are composed of thousands or millions of elements. Thus, Content Delivery Network (CDN) has developed and quickly become an important and necessary element for Network Infrastructure Providers, who promise to provide high quality of services.

The basic idea of CDNs is to push the content closer to the users in order to decrease the downloading time and to avoid network congestion, which in turn obtaining an efficient digital content delivery. The CDN infrastructure composes of three main components: a set of edge servers (also known as surrogates), the request-routing infrastructure and the distribution infrastructure. In general, the distribution infrastructure is a caching network including multiple CDN cache servers which are deployed at different locations. These CDN cache servers connect among themselves in a certain topology and cooperate together to resolve

ISSN: 2005-4297 IJCA

requests sent from clients. CDN can efficiently increase concurrent users because it delivers faster content to end-users, and it also reduces network traffic through caching or replicating content over some mirrored CDN servers placed at various locations strategically. Thus, this system achieves better availability and reliability, higher performance and fault tolerance.

All advantages of CDN are affected by the distributed content location as well as caching strategies, which includes placements and collaboration among caching servers. If caching strategies are not designed well enough, the network will be likely to delay because of the cache miss. In the conventional CDN, the collaboration among caching servers is determined by three content outsourcing modes: Cooperative push-based [3], Non-cooperative pull-based and Cooperative pull-based. The cooperative push-based approach is based on the pre-fetching of content to the surrogates. Content is pushed to the surrogates from the content provider, and surrogates cooperate themselves to reduce replication and update cost. In the Non-cooperative pull-based, client requests are directed to their closest surrogates. If there is a cache miss, surrogates pull content from the content provider. With this function structure, the closest cache server serves as a standalone to process the request; thus, this mode is simple and easy to deploy.

However, if there are large numbers of requests directing to the closest caching server, the standalone will not be able to handle all of them because of its resource limitations such as CPU speed, memory, bandwidth, *etc.* When the numbers of requested objects increase, memory swapping occurs because the old objects are replaced by the new ones which also lead to delay and jittering. In addition, it is not useful to have the information from other cache servers instead of from the content provider. The cooperative pull-based differs from the non-cooperative approach in the sense that surrogates cooperate with one another to get the requested content in case of the cache miss through the Internet Cache Protocol (ICP) [4] and Hypertext Caching Protocol (HTCP) [5]. However, if different users of various geographical locations request the same objects, and a cache server does not have the requested objects, it then has to forward this request to another unknown cache server (origin server or cache servers), thus, makes CDN system complicated and unpredictable. Moreover, different requests from one client may be forwarded to different cache servers, and this causes jitter and latency.

Because of the presence of the above issues, this paper proposes an approach that takes advantage of Non-cooperative pull-based (Non-CPB) and Cooperative pull-based named Provider oriented hierarchical cooperative content outsourcing (POHC). Briefly, there are many different POHCs that optimize for different content providers in the same CDN network. Each POHC guarantees the balance in distributing traffic among CDNs as well as avoiding the choke point for the content provider. This paper considers the strength and reliability in constructing POHC. The mathematical analysis and simulation results show that the proposed method achieves better than the Non-CPB in terms of packet loss, less total bandwidth consumption and creating a more reliable CDN system.

The rest of the paper is organized as followed: In Section 2, the paper gives a literature review including history and some related research approaches. Section 3, which is the main focus of this paper, gives an overview of the proposal in general operation and deployment. The proposal of POHC is a robust balance of the shortest spanning tree with an outbound degree constraint. The root of this tree is the content provider, and the “costs” between the two different vertexes are the abstract values combined of network bandwidth and hop-count. In the next part, assuming that POHC is fully achieved, this paper gives a performance comparison between POHC and Non-CPB by the mathematical analysis as well as simulation using OMNeT++. Finally, Section 5 concludes the paper with final remarks as well as possible issues from this proposal of POHC.

2. Related Works

Data caching has been recognized as an effective way in reducing network traffic, alleviating content provider load, increasing availability and reliability. The original form of CDN is a caching system in which the primary function of a cache is to act as a fast intermediate storage between clients and servers. In the beginning, the caching system was used for file transfer in order to reduce the bandwidth consumption [6]. According to this concept, there were several proposed researches such as harvest system project [7] and cooperative caching in wireless ad hoc [8]. Most of these studies proposed hierarchical and cooperative method.

After the appearance of CDN, caching system is divided into two branches: transparent caching and CDN. These two branches help in reducing network bandwidth as well as improving the quality of experiment; they are different in business model, however. The CDN is a formalized business relationship with the content provider while the transparent caching deployed by service providers without the content owners and publishers' consent. Transparent caching intercepts client requests and redirects users to cached content. Therefore, a number of cache sharing protocols were proposed, which provided mechanisms to reduce the communication cost among caches, *i.e.*, Summary cache [9], Cache digests [10], Cache Array Routing Protocol (CARP) [11], Internet Cache Protocol (ICP), Hypertext Caching protocol (HTCP) etc. The development of several caching protocols enables CDN system to communicate efficiently.

The caching strategies in CDN are determined by content outsourcing models: Cooperative push-based [3], Non-cooperative pull-based and Cooperative pull-based. In this phrase, there are several researches on cooperative caching in CDN. These researches focus on the benefits of cooperative caching. Prior studies in these contexts include simulation experiments [12], prototypes [13], and analytical results and algorithms [14]. Most of the above-mentioned works have focused on minimizing access latency. In fact, the main rationale for CDN system is that bandwidth is assumed abundant, so minimizing the bandwidth consumption or investment cost ([1, 2]) is a far more relevant objective than reducing the delay. In this work, the paper tries to build a complete hierarchical caching system that content provider orients in order to reduce the total bandwidth consumption and the network congestion regarding to the packet loss ratio.

2. Methodology

3.1. An Overview of the POHC and the Deployment Consideration

The POHC is a series of CDN servers which are arranged hierarchically in a tree-like structure. It helps to prevent top-level CDN servers from becoming choke points. Lower level CDN servers receive and handle these requests directly. User requests travel from a given CDN servers towards the content provider until the requested object is found. The requested object is then sent back via the reversed path, leaving a copy of the requested object in each intermediate CDN server. Whether it traverses or not is dependable on the content placement policy. Figure 1 is an illustration for POHC. First, a client sends a request for objects (1). Through the request routing mechanism, the CDN system will choose a CDN server for serving this client (3). This CDN server could be the server that is the closest to the requested client or an available server. Then the chosen CDN server returns the results to the requested client if it contains the requested object (6); otherwise, it then forwards to the direct parent and waits for the response (5).

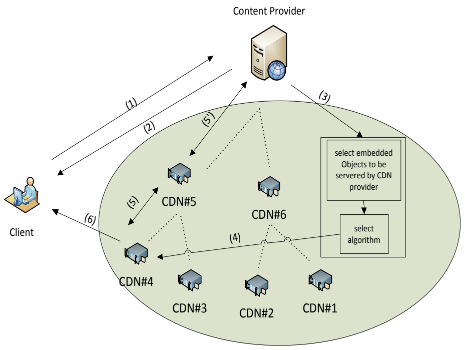


Figure 1. POHC Operation

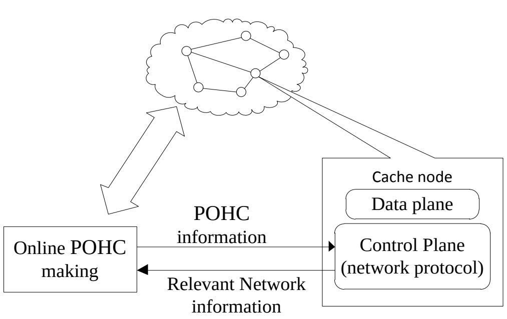
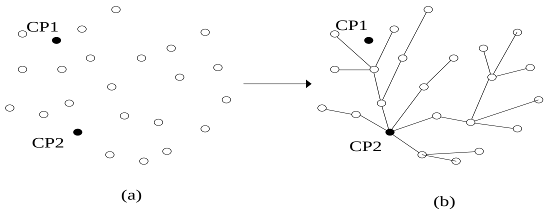

Cache node
Data plane
Control Plane (network protocol)
Online POHC making
POHC information
Relevant Network information

Figure 2. Online POHC Making for Automatic CDN System

For deployment scenario, in an existing CDN infrastructure, a surrogate continuously monitors the surrounding CDN network conditions; collects relevant information and online algorithm topology making are executed in order to construct topologies for different content providers. If there is any change in topology, the center will notice and control the changing. Figure 2 shows the general proposal operation of hierarchical CDN topology as described. Figure 3 (a, b) is an illustration of the content provider oriented CDN topology. Within the same CDN network, the content providers CP1 and CP2 will have different POHC.



(a)

(b)

Figure 3. (a) A CDN Network Infrastructure with Two Content Providers CP1 and CP2, (b) An Optimal POHC for Content Provider CP2

The collecting information includes hop count, bandwidth among CDN servers and CDN servers vs. content providers. By collecting information, the method is able to keep the network tuning to a better operation point to fit the current conditions. Thus, a dynamic and automatic network control can be achieved. The online POHC making then uses the collected information to construct the Tree-like topology as Figure 3(b), in which the minimum total cost subjects to some resources are constrained. In order to guarantee the robustness and reliability for this topology, the cost here is an abstract value that combines hop count and bandwidth capacity. Details of the tree construction and cost will be discussed in Section 3.2 “Online POHC making discussion”.

3.2. Online POHC Formation Discussion

In order to maximize the utilization of a CDN, cooperatives among CDN servers need to consider the limitation of CDN servers in the aspect of computational power, bandwidth and cache size. They also need to guarantee a smooth playback/presentation of the material for subscribers because all customers want to have good quality services in terms of speed and delay time. In addition, network traffic should be distributed among CDN servers equally. From these reasons, this paper considers several properties from the POHC as followed: First, each CDN in POHC should have a bounded out-degree, which is the maximum number of direct child CDN servers. This is because an individual CDN server may have limited resources. Second, the total travelling cost is minimal. Third, the POHC level, which is the number of CDN servers from the content provider to the deepest CDN server, is minimal.

From above consideration of POHC, this study defines a bounded out-degree constrained minimum cost spanning tree (DMC-SPT), where distance is the abstract values that are combined of network bandwidth and the hop-count as followed:

Definition: Given a weighted, undirected, complete graph G = (V, E), where all distances are not negative, and a designated source point root $\in$ *V, find a DMC-SPT T of V which:*

- *Is rooted at root*
- *Has out-degree* $\leq$ *B where B is the defined maximum out-degree threshold*
- *Minimizes where is the shortest-path distance in the tree T*
- *Minimizes the maximum depth in the tree.*

Under these constraints, the investigation seeks to build a multicast tree that does not go over the degree-bound constraints and also provides minimum stretch in relation to the hop count and bandwidth for connecting the end systems. Recent works propose approximate solutions to construct the tree for this problem ([15, 16, 17 and 18]) with different solutions including OMNI [17], MDDBST [18], KLS algorithm [15], and DBSPT [16]. In the paper [16], the author presents an empirical analysis comparing among algorithms in terms of compactness, cost mean and maximum depth. Based on this comparison, the DBSPT-Circular-PO delivers the best result with a compact outbound degree trees formation on different network topologies.

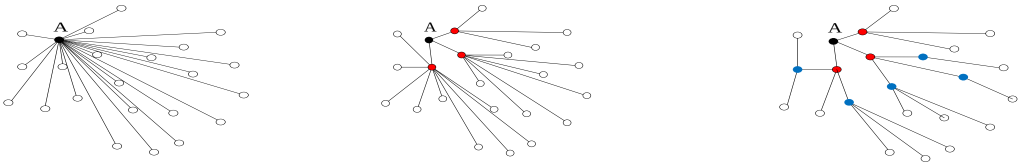


Figure 4. (a) step 1 - the Shortest Path Tree from the Root is calculated.

Figure 4. (b) Level 1 (the root) is adjusted to obey the out-degree bound.

Figure 4. (c) Nodes in level 2 (children of the root) are cleared from the excessive load.

Firstly, the DBSPT-Circular-PO algorithm calculates the shortest path from each node to the root without considering the out-bound degree. Figure 4(a) is an illustration for this step, and A is the root node. Only B nodes (red nodes in Figure 4(b)), which have the shortest paths to the root in the shortest list, are kept. The shortest path on all other remained nodes (blue nodes in Figure 4(c)) are then modified using the heuristic whereby each node must be connected to a parent that is closer to the root than itself, that is if a's parent is b, then $d(b,root) \leq d(a,root)$. The process will examine level by level until there is no more node that has out-degree larger than B. Figure 4(b) and Figure 4(c) are illustrations for this step. After completing the second step, the tree may be unbalanced. The algorithm provides the final step called post optimization. During the running process, post optimization can compact the tree only when there is an improvement, forcing the tree to be at minimum possible height as determined by the out-degree bound and number of vertices in the input graph ([$\log_B n$]). The tree is compacted by examining itself in breadth-first search order, starting with the root. Any vertex encountered that has an out-degree smaller than B is brought up to an out-degree of B by promoting a sub-tree. If the node being examined is at level i, sub-trees rooted in levels $\geq$ i + 2 are then considered potential children. The promotion which causes the smallest increase to average-cost is the transfer enacted in the current tree. This procedure guarantees a final tree of minimum height at the expense of an increased average-cost.

4. Performance Comparison

In this part, a comparison between Non-CPB (Figure 5) and POHC (Figure 6) is analyzed. The comparison includes the mathematical analysis and simulation by using OMNeT++ [20] in the assumption that all caches are equal. M/M/l/K queuing model [19] is used with requests arrive according to a Poisson process with rate λ and the average service rate μ. Because the M/M/1/K queuing model is used, a cache can handle at the most K requests at a time. A request will be blocked if the maximum number of K has been reached. The probability of blocking is denoted as PB and calculated as followed:

$$P_B = P[\mathrm{N} = \mathrm{K}] = \frac{(1-\rho)\rho^K}{(1-\rho^{K+1})} \qquad 1)$$

where is the offered traffic, $p = \lambda / \mu$.

4.1. Packet Loss

Consider two different networks: Non-CPB and POHC (as illustrated in Figure 5 and Figure 6) with M nodes including the root, request arrival rate λ_i at each node ($1 \leq i \leq M-1$) (or request arrival rate λ_{ij} at each node ($1 \leq i \leq B, 1 \leq j \leq N$) in case of POHC); cache hit ratio α at each node with the assumption that all CDN servers have the same capacity. The following part gives the formulation for the number of packet loss in content provider and in the whole network.

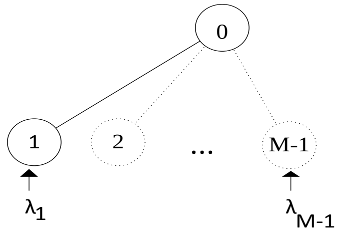


Figure 5. The Non-CPB Includes M-1 Nodes Connects Directly to the Content Provider (root)

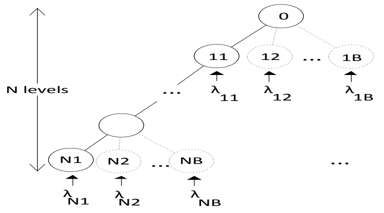


Figure 6. The POHC Includes M-1 Nodes with N levels, each Level has B Nodes

4.1.1. In the Non-CPB

Number of packets dropped D_{c0} at content provider can be defined as:

$$D_{c0} = R_{c0} P_B \qquad 2)$$

where R_{c0} and P_B denote the total request rate to the content provider and the probability of blocking request as defined in (1), respectively. R_{c0} can be calculated as follows:

$$R_{c0} = \sum_{i=1}^{M-1} (1-\alpha)\lambda_i . \tag{3}$$

From Equations (2) and (3), D_{c0} can be rewritten as:

$$D_{c0} = P_B \sum_{i=1}^{M-1} (1-\alpha)\lambda_i . \tag{4}$$

Packet loss in the whole CDN network D_c (without content provider):

$$D_c = \sum_{i=1}^{M-1} \lambda_i \, \mathrm{P}_B = \mathrm{P}_B \sum_{i=1}^{M-1} \lambda_i . \tag{5}$$

4.1.2. In the POHC: The relation between out-degree B, the total number of nodes M (including content provider) and the number of levels N can be described as:

$$N = \frac{\log[\mathrm{M(B{-}1)}+1]}{\log B} . \tag{6}$$

When receiving request from direct clients, the node j at level i then forwards this request to its parent if cache miss happens. Number of requests from this node to the content provider which originate from its connected clients R_{ij} can be calculated as:

$$R_{ij} = \lambda_{ij} \prod_{q=1}^{i} (1-\alpha) = (1-\alpha)^i \lambda_{ij} \tag{7}$$

where λ_{ij} is the request arrival rate from clients that connect directly to this node. From Equation (7) with number of nodes at level i, the total request from level i to the root R_i can be as followed:

$$R_i = \sum_{j=1}^{B^i} R_{ij} = \sum_{j=1}^{B^i} (1-\alpha)^i \lambda_{ij} . \tag{8}$$

Combining Equations (6), (7) and (8), the total request to root R_{h0} can be expressed as:

$$R_{h0} = \sum_{i=1}^{\frac{\log[\mathrm{M(B-1)}+1]}{\log\mathrm{B}}} \; \sum_{j=1}^{B^i} (1-\alpha)^i \lambda_{ij} . \tag{9}$$

Number of dropped packets D_{h0} at content provider:

$$D_{h0} = R_{h0} P_B . \tag{10}$$

At a non-leaf node j of level i, the arrival requests could be from clients that connect directly to this node or from the child nodes, so the number of packet loss D_{ij} can be calculated as:

$$D_{ij} = P_B\lambda_{ij} + P_B\lambda_{ij}(1-\alpha) + ... + P_B\lambda_{ij}(1-\alpha)^{i-1} = P_B\lambda_{ij}\sum_{k=1}^{i}(1-\alpha)^{i-k} . \tag{11}$$

Combining Equation (11) and packet loss rate, the total number of packet loss at level i can be expressed by:

$$D_i = P_B \sum_{j=1}^{B^i} \lambda_{ij} \sum_{k=1}^{i} (1-\alpha)^{i-k} . \tag{12}$$

Because the network has N level identified in Equation (6) and combine with Equation (12), the total number of packet loss D_h can be calculated by:

$$D_h = \sum_{i=1}^{\frac{\log[M(B-1)+1]}{\log T}} \left[P_B \sum_{j=1}^{B^i} \lambda_{ij} \sum_{k=1}^{i} (1-\alpha)^{i-k} \right]. \tag{13}$$

Figure 7 and Figure 8, which are plotted from Equations (4) and (10) with the arrival requests followed by the Poisson distribution, show the packet loss comparison at content provider in two networks. In the Figure 7, POHC can achieve approximate 6 times better than the Non-CPB regarding to the packet loss. The Figure 8 illustrates the number of packet loss in POHC is also better than in Non-CPB when request arrival rate is increased.

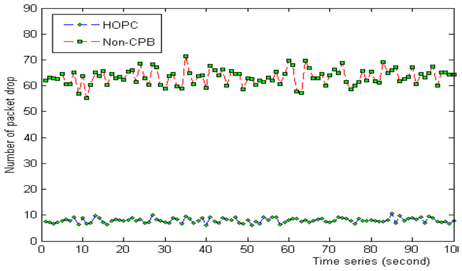


Figure 7. Compare the packet loss at the content provider in POHC and Non-CPB (network include 40 nodes, T=3) (compare D_{c0} and D_{h0})

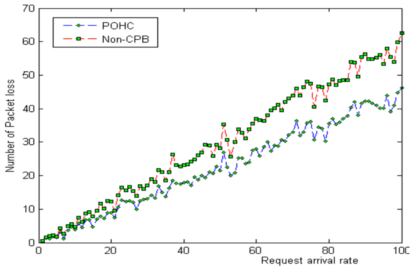


Figure 8. Compare the packet loss at the content provider in 2 cases: POHC and Non-CPB with increasing request arrival rate (network include 40 nodes, T=3) (compare D_{c0} and D_{h0})

Also from Equations (5) and (13), we can compare the total packet loss in the whole network between POHC and Non-CPB as illustrated in Figure 9 and Figure 10. These figures show that the POHC works more stable than the Non-CPB. As mentioned in section 3.3, reliability is a very important factor in distributed computing systems. These figures also show that the POHC is also slightly better than the Non-CPB regarding to the packet loss.

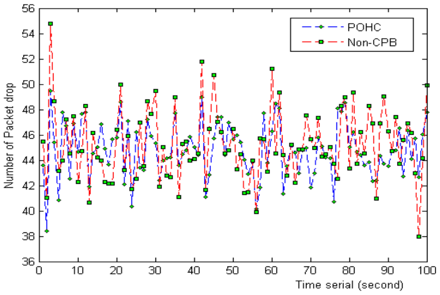


Figure 9. Compare the Packet Loss in the Whole Network in 2 cases: POHC and Non-CPB (network includes 40 nodes, T=3) (compare D_c and D_h)

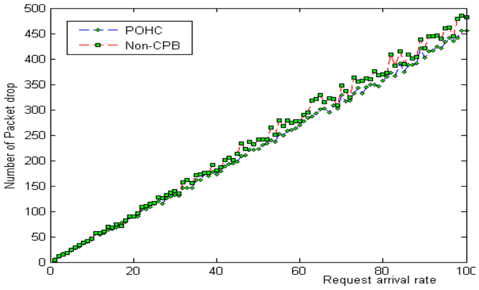


Figure 10. Compare the Packet Loss in the Whole Network in 2 cases: POHC and Non-CPB (network includes 40 nodes, T=3) (compare D_c and D_h)

4.2. Bandwidth Consumption

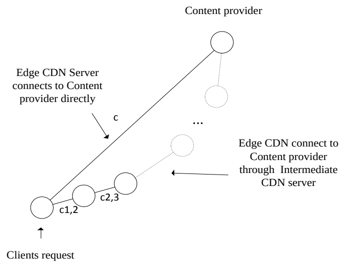

Content provider
Edge CDN Server connects to Content provider directly
c
...
Edge CDN connect to Content provider through Intermediate CDN server
c1,2
c2,3
Clients request

Figure 11. A Consideration between Non-CPB and POHC

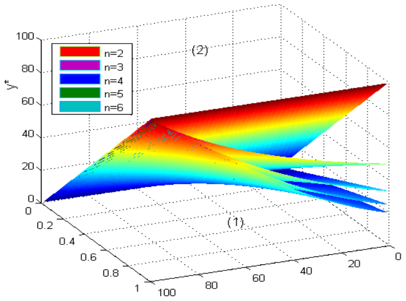


Figure 12. Plot of Equation (19) with $N \in [2,6]$ **and** $\alpha \in (0,1)$

A model simulating the operation of POHC and Non-CPB modes is considered and illustrated in Figure 11. First the left line of side is the Non-CPB mode in which the Edge CDN server connects to content provider directly. Second the right line of side is the POHC mode in which the edge CDN server connects to the content provider through a number of intermediate CDN servers.

Because cache hits at the edge of CDN server, the total bandwidth cost $B_{Non-CPB}$ can be calculated as followed in the non-CBP system:

$$B_{Non-CPB} = (1-\alpha_1)sc \tag{14}$$

where s, c and α_1 are the total requests from clients to an edge CDN server, the bandwidth is consumed when a unit of data is transferred from the edge of CDN server to the content provider, and cache then hits at the edge of the CDN respectively. However, the total bandwidth cost B_{HOPC} in the POHC is different because there are several intermediate CDN servers instead:

$$B_{HOPC} = (1-\alpha_1)c_{1,2}s + (1-\alpha_1)(1-\alpha_2)c_{2,3}s + ... = \sum_{i=2}^{N} sc_{i-1,\mathrm{i}} \prod_{j=1}^{i-1} (1-\alpha_j) \tag{15}$$

where α_i , $c_{i,j}$ ($j = i+1 = 2,\ 3,..,\ N$) are cache hit ratio at CDN server level i and the bandwidth is consumed when a unit of data is transferred between CDN server at level i and CDN Server at level j, respectively.

Without a loss of generality, we can assume $c_{i,j} = c'$ for all $j = i+1 = 2,\ 3,..,\ N$ and $\alpha_i = \alpha$ for all $i = 2,\ 3,..,\ N$ then (14) and (15) can be rewritten as:

$$B_{Non-CPB} = (1-\alpha)sc. \tag{16}$$

$$B_{HOPC} = \sum_{i=2}^{N} sc' \prod_{j=1}^{i-1} (1-\alpha) = \mathrm{sc'} \frac{1-\alpha-(1-\alpha)^N}{\alpha} \tag{17}$$

where $N \geq 2$.

Then, the comparison of total bandwidth consumption between the Non-CPB and POHC is analyzed by dividing two Equations (16) and (17), and the following result is obtained:

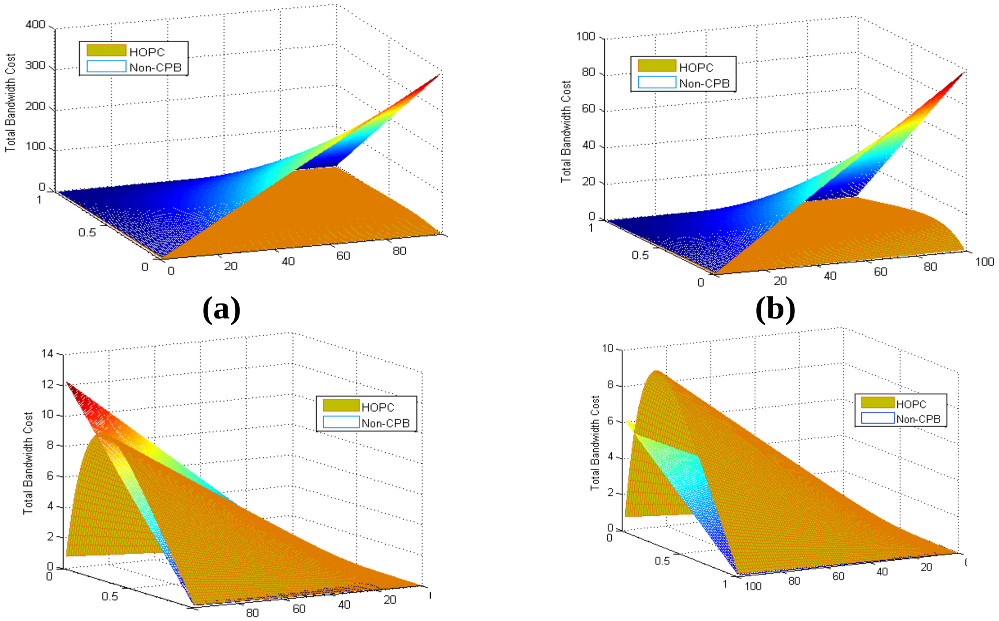
$$\frac{B_{HOPC}}{B_{Non-CPB}} = \frac{c'}{c}\left[\frac{1-(1-\alpha)^{N-1}}{\alpha}\right]. \tag{18}$$

From Equation (18), for $B_{HOPC} < B_{Non-c}PB$, the relationship between c and c' should be as followed:

$$c' < \mathrm{y}^*(\mathrm{c},\alpha) = \frac{\alpha c}{1-(1-\alpha)^{N-1}}. \tag{19}$$

The $\mathrm{y}^*(\mathrm{c},\alpha)$ divides three-dimensional space into two parts, and the Equation (19) indicates that the total bandwidth cost in the POHC system is only better than the Non-CPB system's when c' belongs to the part (1) as illustrated in Figure 12.

The Figure 13 observes the total bandwidth cost in POHC and Non-CPB modes with different combination of c' and c based on Equations (16) and (17). Figure 13(a) with $c' = 0.25c$ and (b) with $c' = c$ show that the total bandwidth cost in POHC is always smaller than the Non-CPB's with any value of the cache hit ratio $\alpha \in (0,1)$. However, when $c' = 8c$ and $c' = 16c$ as illustrated in Figure 13(c) and (d), the total bandwidth cost comparison between POHC and Non-CPB then also depends on the cache hit ratio.



(a)

(b)

(c)

(d)

Figure 13. An Illustration of the Total Bandwidth Cost Comparison between Non-CPB and POHC with n=6, $\alpha \in (0,1)$ in four cases: (a): c'=0.25c,(b): c'=c, (c): c'=8c and (d) c'=16c

4.3. Simulation with OMNeT++

In order to evaluate the proposal, the paper also conducts a case study with MCI topology (as Figure 14) by using OMNeT++ [23], a component-based C++ simulation framework for building network simulators. Based on the MCI topology, the paper builds a simple CDN system by modifying the CDNSim project [21] with some several attributes as described in Table 1.

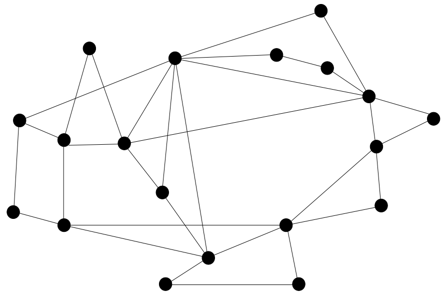


Figure 14. MCI Topology

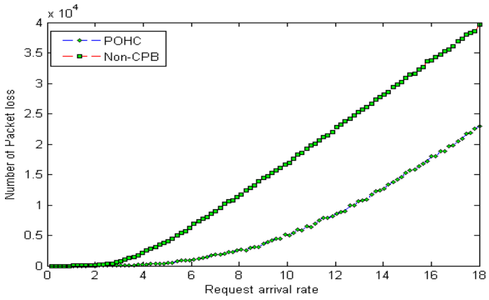


Figure 15. Compare the Packet Loss in the MCI Topology in 2 cases: POHC and Non-CPB

Table 1. Simulation Parameters

Client	+Requests are generated followed by a Poisson process.
CDN servers	+Use FIFO size-limited queue for receiving requests from clients/child nodes. If FIFO queue is full, requests will be dropped. FIFO queue Services time for queue followed by the exponential distribution. +User Least Recently Used (LRU) algorithm for storing objects with limited capacity.
Routers	+Use FIFO size-limited queue for receiving requests from clients/other routers. If FIFO queue is full, requests will be dropped. FIFO queue Services time for queue followed by the exponential distribution.
Conten t provider	+Use FIFO size-limited queue for receiving requests from clients/child nodes. If FIFO queue is full, requests will be dropped.

The simulation results are shown in Figure 15, Figure 16 and Figure 17. In Figure 15, when the request arrival rate increases, the number of packet loss in POHC is then always less than the Non-CPB’s. This result reinforces the conclusion in Section 4.1. Figure 16 and Figure 17 show that by applying the proposal, the throughput can be significantly reduced. When the request arrival rate increases, the change in throughput consumption in POHC is almost zero; this is contrast with the Non-CPB.

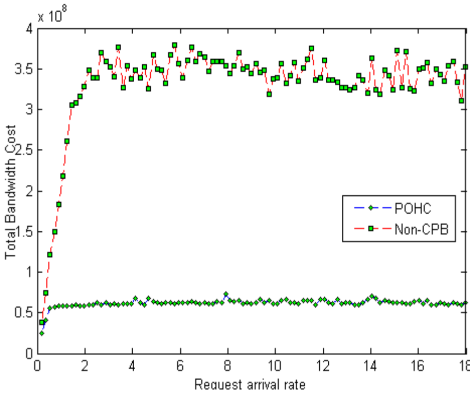


Figure 16. Compare the throughput at Content Provider in 2 cases: POHC and Non-CPB

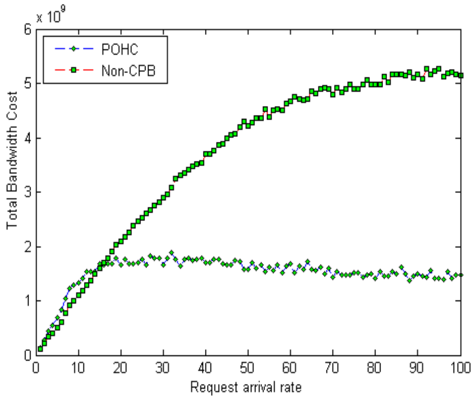


Figure 17. Compare the throughput in whole Network in 2 cases: POHC and Non-CPB

5. Conclusion

Cooperating among caches in CDN system is the main focus of this paper. First, with the increasing of network traffic, the CDN system is needed, and a brief overview of the most relevant state-of-art caching in CDN is demonstrated. Then, a POHC that takes advantage of both Non-cooperatives pull-based and cooperative pull-based is proposed. DBSPT-Circular-PO algorithm is used for constructing the POHC in CDN with a consideration in robustness. By means of mathematical analysis and simulation, the paper analyzes and compares the proposed method to the Non-CPB system. The mathematical analysis shows that the proposed method not only achieves better in content provider than the Non-CPB's but also improves reliability for CDN system regarding to packet loss. In addition, the POHC can significantly reduce the total bandwidth consumption if it is designed well in comparing to the Non-CPB.

However, this proposed method is just a starting point for POHC in CDN. In order to deploy a hierarchical caching in a real CDN system, there are many challenges related to the proposal such as the effect of population on the performance of POHC in CDN system and POHC protocol. Furthermore, in order to build a more reliable CDN system, the combination of bandwidth, hop count and population are needed for further investigate. In addition, Internet Cache Protocol (ICP) [6] does not support hierarchical cooperative caching, so improvement of this protocol might be needed.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2012R1A1A4A01005915).

References

[1] J. Kim and J. K. Choi, "Segmented Content Delivery Scheme for Minimizing Server Cost in CDN", Ubiquitous and Future Networks (ICUFN), **(2012)** July, pp. 462-463.

[2] D. Lee, J. H. Mo and J. Park, "ISP vs. ISP+CDN: Can ISPs in Duopoly Profit by Introducing CDN Services?", SIGMETRICS Performance Evaluation Review, **(2012)**, pp. 46-48.

[3] J. Kangasharju, J. Roberts and K. W. Ross, “Object Replication Strategies in Content Distribution Networks”, Computer Communications, vol. 25, no. 4, **(2002)** March, pp. 367-383.

[4] D. Wessels and K. Claffy, “Internet Cache Protocol (ICP) version 2”, IETF RFC 2186, **(1997)** September, http://www.ietf.org/rfc/rfc2186.txt.

[5] P. Vixie and D. Wessels, “Hyper Text Caching Protocol (HTCP/0.0)”, IETF RFC 2756, **(2000)** January, http://www.ietf.org/rfc/rfc2756.txt.

[6] P. B. Danzig, R. S. Hall and M. F. Schwartz, “A case for caching file objects inside internetworks”, SIGCOMM Computer Communication Review, vol. 23, **(1993)**, pp. 239-248.

[7] Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz and K. J. Worrell, “A hierarchical internet object cache”, Proceedings of the Annual Technical Conference on USENIX, **(1996)**, pp. 13.

[8] L. Yin and G. Cao, “Supporting cooperative caching in adhoc networks”, IEEE Transactions on Mobile Computing, vol. 5, **(2006)** January, pp. 77-89.

[9] L. Fan, P. Cao, J. Almeida and A. Z. Broder, “Summary cache: a scalable wide-area web cache sharing protocol”, ACM SIGCOMM Computer Communication Review, vol. 28, **(1998)**, pp. 254–265.

[10] Rousskov and D. Wessels, “Cache digests”, Computer Networks and ISDN Systems, vol. 30, **(1998)**, pp. 22-23.

[11] V. Valloppillil and K. W. Ross, “Cache Array Routing Protocol v1.0”, IETF Internet Draft, **(1998)** February, http://tools.ietf.org/id/draft-vinod-carp-v1-03.txt.

[12] M. D. Dahlin, R. Y. Wang, T. E. Anderson and D. A. Patterson, “Cooperative caching: using remote client memory to improve file system performance”, OSDI '94 Proceedings of the 1st USENIX conference on Operating Systems Design and Implementation, no. 19, **(1994)**.

[13] G. M. Voelker, E. J. Anderson, T. Kimbrel, M. J. Feeley, J. S. Chase, A. R. Karlin and H. M. Levy, “Implementing cooperative prefetching and caching in a globally-managed memory system”, ACM SIGMETRICS ’98, vol. 26, is.1, **(1998)** June, pp. 33-43.

[14] I. D. Baev, R. Rajaraman and C. Swamy, “Approximation Algorithms for Data Placement Problems”, SIAM Journal on Computing, vol. 38, **(2008)**, pp. 1411-1429.

[15] J. Kanemann, A. Levin and A. Sinha, “Approximating the degree-bounded minimum diameter spanning tree problem”, Algorithmica, vol. 41, **(2003)**, pp. 117–129.

[16] M. T. Helmick and F. S. Annexstein, “Depth-Latency Tradeoffs in Multicast Tree Algorithms”, The 21st International Conference on Advanced Networking and Applications, **(2007)** May, pp. 555-564.

[17] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee and S. Khuller, “Construction of an efficient overlay multi-cast infrastructure for real-time applications”, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, **(2003)** April, pp. 1521-1531.

[18] S.Y. Shi, J. S. Turner and M. Waldvogel, “Dimensioning server access bandwidth and multicast routing in overlay networks”, The 11th international workshop on Network and operating systems support for digital audio and video, **(2001)**, pp. 83-91.

[19] C. Jianhua, A. Mikael, N. Christian and K. Maria, “Web Server Performance Modeling Using an M/G/l/K*PS Queue”, The 10th International Conference on Telecommunications, vol. 2, **(2003)** March, pp. 1501-1506.

[20] Varga, “OMNeT++ Discrete Event Simulation System”, The 15th European Simulation Multiconference (ESM '01), **(2001)** June, pp. 319-324.

[21] K. Stamos and G. Pallis, “CDNsim: A simulation tool for content distribution networks”, Journal ACM Transactions on Modeling and Computer, vol. 20, issue 2, no. 10, **(2010)** April.

Authors

Tuyet Mai Thi Anh

She received a bachelor’s degree in Information Technology from Da Nang University, Vietnam. She is now pursuing her Master of Science in Information and Communication Engineering at Inje University, South Korea. Her research group is Laboratory of Telecommunication Network. Her current research interests include computer network, content delivery network.

Hung Phan Huu

He was received the Diploma in Computer Science from the HCM University of Science in 2010. Currently, he is a master student in Cnc Laboratory, Inje University, Korea. His research interests are on Ethernet, Content Delivery Network, Energy Saving, and Anomaly Detection based on User-behavior.

Tien Nguyen

He was received the Diploma in Computer Science from the HCM University of Science in 2010. His BS thesis was on Windows Azure and Cloud Computing. Currently, he is a master student in Cnc Laboratory, Inje University, Korea. His research interests are on Congestion Control, Quality of Services and content delivery network..

Chul-Soo Kim

He is a professor in the School of Computer Engineering at Inje University, South Korea. He received Ph.D. from the Pusan National University, South Korea and worked for ETRI (Electronics and Telecommunication research Institute) from 1985 - 2000 as a senior researcher for developing TDX exchange. Aside from the involvement in various domestic and international projects, his primary research interests include network protocols, traffic management, OAM issue, and NGN charging. He is a member of ITU-T SG3, SG11, SG13 and a Rapporteur of ATM Lite from 1998 to 2002, and CEO in WIZNET from 2000 to 2001. He was the chairperson if BcN Reference Model in South Korea, and a Rapporteur of ITU-T SG3 NGN Charging. From 2008 to 2012, he worked for Ministry of Knowledge and Economics in South Korea as BcN Program Director.