CDN-CCN collaboration for in-network content caching

Anselme Ndikumana, Choong Seon Hong*
Department of Computer Science and Engineering, Kyung Hee University, Rep. of Korea
{anselme, cshong}@khu.ac.kr

Abstract

Content Delivery Network (CDN) and Content Centric Networking (CCN) are the promising architectures for downloading contents from the origin server(s) and cache them near to the end-users, where the end-users can get the contents with minimized delay. CDN provider relies on cache servers deployed in different geographical regions. Cache server deployment in a region is based on the end-users' demands and not on the ISP network condition; this may cause service degradation in both CDN provider and ISP. To overcome the above highlighted issue, in this paper, we proposed a new architecture which join CDN and CCN for in-network content caching; where in CCN based on network condition, ISP can cache the contents from CDN server(s) in router (s) available in transmission path between CDN server and end-users. The simulation results show that CDN-CCN collaboration reduces the delay experienced by end-users during the content retrieval.

1. Introduction

In the current Internet, Content Delivery Network (CDN) Provider provides service that Content Publisher or Content Provider (CP) can use to deliver content to the end-users (consumers) [1]. CDN provider deploys cache servers in different ISPs' networks that deliver content to the end-users based on their geographical locations (figure 1). Closer to the end-users the content is placed, the faster the content will be delivered to them. According to Cisco Visual Networking Index published and updated on May 26, 2015, CDN will have 62% of all Internet Traffic by 2019, up from 39% in 2014 [2].

In additional to CDN, since last couple of decades, many research projects such as NSF GENI, EU FIA, etc. focused on defining new Future Internet (FI) architecture. Among the projects, CCN is the one of promising architectures for content sharing and retrieval, where the end-users are interested in content and not where the content is located [3]. In CCN, to get content, end-user sends out a request called Interest, and any node in the network, which has content returns the content to the end-user in reverse path of Interest. The routers on the transmission path can cache returning content, in order to serve similar future requests from other end-users. This embedded router storage on a transmission path can help the ISP to control network traffic crossing his network.

By caching content in many cache servers and/or cache routers, CDN and CCN collaboration solve existing problems in current client-server computing architecture [4] of single point of failure, where one central server/origin server delivers content to all end-users based on requests; when central server fails, end-users cannot get the content.

For efficient content delivery, CDN provider caches content in many cache servers placed in different geographical regions. Cache server deployment in a region is based on end-users' requests from that region, and not on the ISPs' network condition operating in that region; this may cause service degradation in both CDN provider and ISP. In additional to that, due to the growth of demands, CDN provider cannot always have cache server deployed in every region that can satisfy the users' requests [5]. On the other side, in CCN, cache router has small cache size to hold data needed, and when the cache is full, the node needs to remove some content from the cache in order to make a room for new incoming content needs be cached. To overcome the above highlighted issues, in this paper, we proposed a new architecture, which join CDN and CCN for in-network content caching. CDN provider, which has big cache size, authorizes each ISP to download contents from CDN server(s) and cache them in routers available in transmission path between CDN server and end-users.

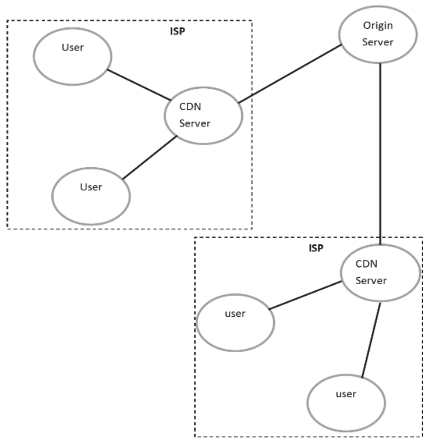

Fig. 1. Content Delivery Network

The rest of the paper is organized as follows, section 2 presents in detail CDN-CCN collaboration for in-network content caching. Section 3 provides an evaluation for proposed architecture, while the section 4 presents our future scope and conclusions.

2. CDN-CCN collaboration for in-network content caching

The objective of CDN-CCN collaboration for in-network content caching is to reduce the delay experienced by end-users in retrieving content. Based on end-users' requests and network condition, ISP can download content from CDN server, and cache it on his router(s) available in transmission path; this result in solving the challenges faced by CDN provider related to the server deployment and content placement. Caching content on the ISP network can help the ISP to do better traffic engineering such as network load balancing.
The collaboration of CDN-CCN helps CDN providers to deliver the content to the users with minimized delay, where content is placed near to the end-users inside the ISPs' routers (figure 2).

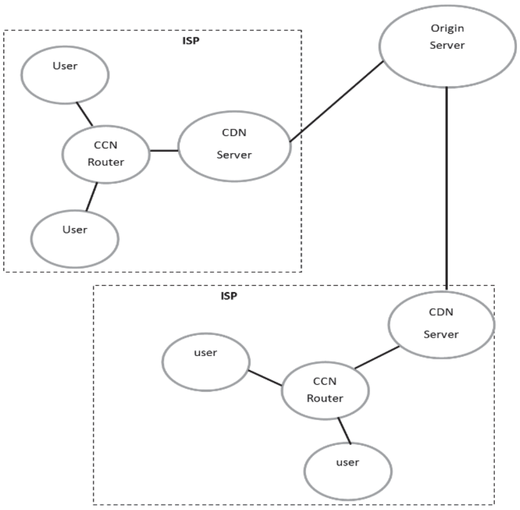

Fig. 2. CDN-CCN collaboration

In our system model presented in figure 3, CDN and CCN work together in delivering content to the end-users. CDN has responsibility to deploy server inside the ISP network based on demand from the ISP; this deployment does not require network information such as ISP network topology, link capacity, traffic crossing the link, etc. from the ISP. On the other side, an ISP based on network condition can cache contents in the routers near to the end-users. The ISP network is equipped with CCN routers for caching and forwarding the contents based on their names.
The main components of our system model are:

- Origin Server: is the main server, where the content is hosted before being duplicated in many CDN servers and cache routers. The origin server is deployed based on an existing IP network, where the IP address is used to locate the origin server.
- Content Controller (CC): Content Controller is responsible for content control, which includes to track the location of the CDN server(s), neighbor routers, and content cached in each neighbor router inside the ISP network. In additional to that, CC is responsible for taking decision on which contents to cache, and where to cache them based on content demands. For a requested content, this is not cached in CCN routers inside the ISP network, CC maps content name to the IP address and forwards the request to the nearest CDN server available in ISP network. In case of miss, CC forwards request to the external CDN server or to the origin server. When requested content reaches CC (returns in the reverse path of the request), CC forwards it to the end-user and CDN server; it decides also whether the content needs to be cached or not, and on which router to cache it.

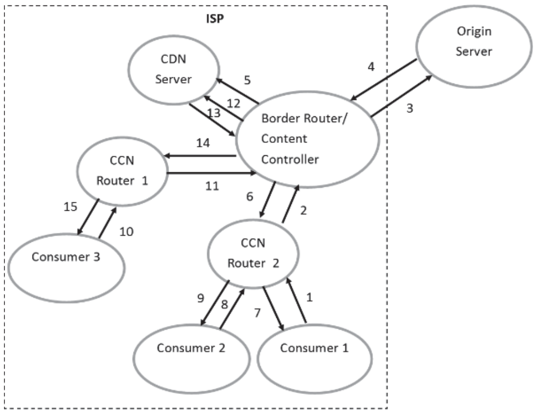

Fig. 3. System Model

- CDN server: CDN server caches the content downloaded from the Origin Server. CDN server is deployed inside the ISP network based on existing IP network, where the IP address is used to locate the origin server or other CDN servers. The communication between Origin Server and CDN server is based on IP network.
- CCN router: It is a router which has storage capacity for caching the content passing through it and forwards requests and contents based on content names. The communication between CCN router and the CC is based on content name. For large network, every CCN router and CC control its neighbor(s) in terms of caching decision through adding incoming face and its demand rate in Pending Interest Table (PIT). The neighbor node with high demand rate than others caches the content.

In figure 3, consumer 1 requests content by sending out an interest packet (step1). When the request reaches CCN router 2, it checks whether content is cached in its Content Store; in case of miss, it forwards request to CC (step 2) without flooding request to other routers. When the request reaches CC, CC checks whether the content is cached in the other router(s) or in CDN server inside the ISP network. In case of miss, CC maps content name with IP address and forwards request to origin server or external CDN server (step3). On returning content (step 4), CC forwards content to the CDN server for caching (step 5). Based on users' demands, CC may decide to cache the content in its neighbor router(s) near to the end users. In step 6, Content Controller orders CCN router 2 to cache the content. After caching the content CCN router 2 forwards the content to consumer 1 (step 7). In the step 8 and 9, when consumer 2 requests the same content, he can be served by CCN router 2, which is nearby to him. When Consumer 3 requests content which is not cached in CCN router 1 (step 10), CCN router 1 forwards request to CC, and when the request arrives at CC (step 11), the CC maps the content name to IP address and forwards request to CDN server, where the content is cached (step 12). CDN server replies with content in the reverse path of the request (step 13,14, 15).

3. Performance evaluation

In this section, we analyze CDN-CCN collaboration in terms of delay experienced by end-users during the content retrieval through simulation. We used the ns-3 based simulator, namely ndnSIM [6].

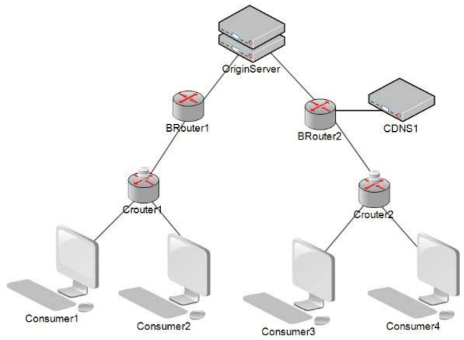

Fig. 4. Network topology

From	To	capacity	metric	delay	queue
CRouter1	Consumer1	10Mbps	1	1ms	100
CRouter1	Consumer2	10Mbps	1	1ms	100
CRouter2	Consumer3	10Mbps	1	1ms	100
CRouter2	Consumer4	10Mbps	1	1ms	100
CRouter1	BRouter1	10Mbps	1	1ms	100
CRouter2	BRouter2	10Mbps	1	1ms	100
BRouter2	CDNS1	10Mbps	1	1ms	100
BRouter1	OriginServer	10Mbps	1	1ms	100
BRouter2	OriginServer	10Mbps	1	1ms	100

Table 1: Topology information

Node	Caching Capacity (Number of chunks)	Sending rate/second	Cache replacement policy
Consumer1	0	100	
Consumer2	0	100	
Consumer3	0	100	
Consumer4	0	100	
CRouter1	10000		FIFO
CRouter2	10000		FIFO
BRouter1	0		
BRouter2	0		
CDNS1	50000		FIFO
OriginServer	50000		FIFO

Table 2. Caching configuration

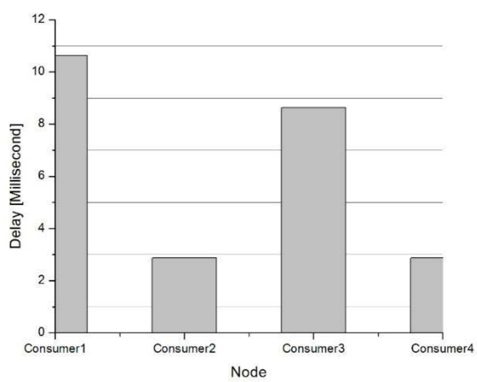

Fig.5. The Delay experienced by end-users

In our network topology and results presented in figure 4 and 5, Consumer1 retrieves content from the Origin Server (client-server) with maximum delay. While Consumer3 gets the same content from CDNS1 (CDN server) with lower delay than Consumer1. After 20 seconds, Consumer2 starts to retrieve content cached in Crouter1 (CCN), while Consumer4 gets the same content from Crouter2 (CDN-CCN). From the simulation results show that CDN-CCN collaboration helps the consumers to get the content with minimized than the existing client server and CDN.

4. Future Scope and Conclusion

In this paper, we proposed a new architecture which joins CDN-CCN for in-network content caching under the ISP network. The simulation results show that CDN-CCN collaboration reduces the delay experienced by end-users during the content retrieval. In the future, we aim to extend our work with more analysis through game theory perspective.

6. Acknowledgement

This work was supported by the Industrial Core Technology Development Program (10049079, Development of Mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) *Dr. CS Hong is the corresponding author.

5. References

[1].Sharma, A. Venkataramani, and R. K. Sitaraman, “Distributing content simplifies isp traffic engineering,” ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 1, pp. 229–242, 2013.

[2].Cisco visual networking index (vni), Forecast and Methodology, 20142019 White Paper update May 26, 2015. Document id:1458683795628678

[3].V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009, pp. 1–12.

[4].S. M. Lewandowski, “Frameworks for component-based client/server computing,” ACM Computing Surveys (CSUR), vol. 30, no. 1, pp. 3–27, 1998.

[5].Frank, Benjamin, et al. "Pushing CDN-ISP collaboration to the limit." ACM SIGCOMM Computer Communication Review 43.3 (2013): 34-44.

[6].S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, “ndnSIM 2.0: A new version of the NDN simulator for NS-3,” NDN, Technical Report NDN-0028, 2015

[7].Afanasyev, Alexander, Ilya Moiseenko, and Lixia Zhang. "ndnSIM: NDN simulator for NS-3." University of California, Los Angeles, Tech. Rep (2012).