LETTER

Adaptive Server and Path Switching for Content Delivery Networks

Hiroyuki NISHIMUTA[†a)], *Student Member*, Daiki NOBAYASHI[††], *and* Takeshi IKENAGA[††], *Members*

SUMMARY The communications quality of content delivery networks (CDNs), which are geographically distributed networks that have been optimized for content delivery, deteriorates when interflow congestion conditions are severe. Herein, we propose an adaptive server and path switching scheme that is based on the estimated acquisition throughput of each path. We also provide simulation results that show our proposed method can provide higher throughput performance levels than existing methods.

key words: *CDN, route control, server selection, quality of service*

1. Introduction

Due to increases in the number of providers providing large-capacity content, multimedia products such as videos, images, and music account for most of the traffic flowing over the Internet. Because of this, content delivery networks (CDNs) that can guarantee efficient load distribution and communication quality for such products are now in widespread use around the world. Indeed, Cisco Systems Inc. [1] now predicts that the percentage of CDN traffic entering the Internet will increase from 52% in 2016 to 71% by 2021.

In the CDN paradigm, the server that holds the original content is not directly involved with its delivery. Instead, CDN service providers deploy CDN servers that meet the user requests to Internet service providers (ISPs) [2]. Therefore, generated flow start points are determined by the CDN server selection method used by the CDN provider. This method selects the server that distributes content to users from among the available CDN servers based on their server loads and cache hit ratios, which means that the method used to select the CDN server affects the user's quality level and the network traffic volume [3].

With these points in mind, [4], [5] have reported an optimal CDN server selection method. CDNs are typically operated by CDN providers (such as Akamai) that are independent of ISPs. However, some content providers (such as Google and enterprises with large-scale networks such as Tier-1 ISPs) are increasingly building CDNs on their own; including multi-CDNs that use either their own CDNs or a combination of their own CDNs and those of other providers [6]–[8]. In addition, a method with which ISPs and CDN providers cooperate to simultaneously both control routing and CDN server selection has been proposed [9], while a separately proposed method uses OpenFlow [10] to construct a software-defined network (SDN), and then dynamically performs both route control and CDN server selection [9], [11]–[14]. Since using an SDN makes it possible to collect network information and control routing in units of real-time flow, more advanced routing control and CDN server selection can be expected in the future [15]. However, network conditions are in a constant state of flux, and mid-transmission traffic congestion often occurs on communication paths, which causes communication quality levels to deteriorate. Because of this, several previous studies have investigated methods aimed at reducing the degradation of communication quality levels. However, in cases where there are no available paths that can improve communication quality, the use of dynamic routing alone is insufficient for coping with degraded communication quality [16], [17].

In this paper, we propose a server and path switching method that can be used in conjunction with dynamic routing to reduce communication quality degradation by switching CDN servers.

2. Related Work

Several previous studies have reported attempts to reduce mid-communication quality degradation. Currently, most video streaming services use a mechanism called adaptive streaming, which dynamically changes the bit rate of video content according to network conditions [18]. In Jiang et al., communication quality degradation at the application level was handled by improving the algorithm for selecting the bit rate of video content [19].

However, since application-level responses to network congestion do not fundamentally improve the communication quality levels, various methods that can be used to switch the flow path during communications have been proposed [16], [17]. For example, Long et al. defined the OpenFlow switch reception amount as a link load and proposed a method that can switch the route to the path with the largest available bandwidth when the link load in a network exceeds a certain value [16]. However, since the instantaneous value of the link load is the value that determines the path switching, problems occur when this value fluctuates dynamically and significantly at short intervals, thus causing additional path switching.

Manuscript received March 25, 2020.
Manuscript revised July 11, 2020.
Manuscript publicized August 13, 2020.

[†]The author is with Graduate School of Engineering, Kyushu Institute of Technology, Kitakyushu-shi, 804–8550 Japan.
[††]The authors are with Faculty of Engineering, Kyushu Institute of Technology , Kitakyushu-shi, 804–8550 Japan.
a) E-mail: nishimuta.hiroyuki171@mail.kyutech.jp

DOI: 10.1587/transinf.2020EDL8045

Therefore, in Lan et al. [17], a method was proposed with which the moving average of the link load is set as the value that determines the path switching, and a certain period is set after which the path will not change again. However, in the existing methods for switching the flow path mid-communication (depending on the network topology or the congestion status of the communication path), there is no certainty that any of the available paths can improve the communication quality. As a result, the related work described above could not always cope with mid-communication quality degradation.

3. Adaptive Server and Path Switching Scheme

In this paper, we propose a server and path switching method that can resolve the problems described in the previous section and thus improve user communication quality levels. In this method, if the quality degradation problem can be handled just by path switching between the current client and server, only path switching is performed. Here, it should be noted that this study assumes the use of a single-domain network in which CDNs are composed of OpenFlow-based SDNs. Thus, the clients, the OpenFlow Switch (OFS), and the CDN servers are centrally controlled by a single OpenFlow Controller (OFC), and all CDN servers have the same content. The OFS measures the number of flows on the links connected to it and the reception amount of its interface. The OFC obtains this information from the OFS in the network it controls at a constant cycle.

3.1 Flow of the Proposal Method

The flow of the proposed method is as follows.

Step 1 The client checks the quality of the flow it is receiving.
Step 2 The OFC calculates the estimated acquisition throughput of all paths between the client and the server.
Step 3 Based on the estimated acquisition throughput of the path calculated in Step 2, the OFC searches for a path that can improve the communication quality between the current client and server.
Step 4 If there is a path that can improve the quality in Step 3, the OFC switches communication to that path.
Step 5 If no path is found that can improve the communication quality in Step 3, the OFC searches for a server that can improve the communication quality and switches communication to that server if one is located.

3.2 Determining Communication Quality Degradation at the Client

In this method, in order to measure the quality of all flows, each client measures the throughput of all the flows that it is receiving for an arbitrary period T. The moving average of the flow throughput F_i, which is measured by the client as an indicator of the communication quality, is changed to $S_i(t)$, as shown in (1).

$$S_i(t) = Y_i(t) * (1 - \alpha) + S_i(t-1) * \alpha \tag{1}$$

Here, the flow into the client is denoted by F_i ($i = 0, \dots, n$), the current time is denoted by t, the throughput of the flow F_i at time t is denoted by $Y_i(t)$, and the smoothing coefficient is denoted by α. The maximum S_i calculated from the start of the communication flow F_i to the current time t is defined as $\max S_i$. If the state in which $S_i(t)$ becomes less than a fixed ratio (2) of $\max S_i$ continues for a fixed period (γ), the client recognizes that the quality of the F_i communication has deteriorated and notifies the OFC.

$$S_i(t) < \max S_i * \beta \tag{2}$$

where β is any coefficient that satisfies $0 \leqq \beta \leqq 1$.

3.3 Calculation of Estimated Acquisition Throughput

When the OFC receives a notification from the client, it searches the paths between the client and the server for a path that can improve quality based on the estimated acquisition throughput of the available paths. To calculate the estimated acquisition throughput of a path, the estimated acquisition throughput $X_{j,k}$ of the links $L_{j,k}$ ($k = 0, \dots, m$) belonging to the path P_j ($j = 0, \dots, n$) is obtained by two methods. In the first method, the estimated acquisition throughput $X_{flow,j,k}$ is obtained based on the number of flows on the link $L_{j,k}$, which is expressed by (3).

$$X_{flow,j,k} = \frac{Capacity\ of\ L_{j,k}}{1 + Number\ of\ Flows\ in\ the\ L_{j,k}} \tag{3}$$

The remaining bandwidth $X_{residual,j,k}$ is obtained based on the received amount of each OFC interface, which is expressed by (4).

$$X_{residual,j,k} = Capacity\ of\ L_{j,k}\ \text{- OFS}\ received \tag{4}$$

The estimated acquisition throughput $X_{j,k}$ of the links $L_{j,k}$ is obtained from (5).

$$X_{j,k} = \max(X_{flow,j,k}, X_{residual,j,k}) \tag{5}$$

Ultimately, the OFC calculates the estimated acquisition throughput of the path between the client and the server based on the estimated acquisition throughput of each link. It also calculates the estimated acquisition throughput Z_j of the path P_j as the minimum of the estimated acquisition throughput $\{X_{j,0}, X_{j,1}, \dots, X_{j,k}\}$ of all the links comprising the P_j via (6).

$$Z_j = \min(X_{j,0}, X_{j,1}, \dots, X_{j,k}) \tag{6}$$

3.4 Path Switching

The OFC searches for a path that can improve the communication quality from the calculated estimated acquisition throughput of each path. The path condition that can improve communication quality is given by (7).

$$Z_j > Z_{current} \wedge Z_j > \max S_i * \beta \tag{7}$$

Here, $Z_{current}$ indicates the estimated acquisition throughout for the path currently being used by the flow being switched. The condition in (7) indicates that the selected path has an estimated acquisition throughput that is greater than the current path, and is capable of obtaining a throughput greater than the threshold for judging communication quality degradation. However, if there are no paths that can improve the communication quality, the OFC starts searching for a server with a path that can fulfill that role.

3.5 Server Switching

If path switching alone cannot reduce mid-communication quality degradation between the client and server, the OFC switches to a server that satisfies the conditions in (8).

$$l' + u < l \tag{8}$$

Here, l is the residual communication time of the current ongoing communication, and l' is the estimated residual communication time that would result after server switching (which can be obtained from the estimated acquisition throughput between the server and client after server switching), and u is the time required for server switching by the OFC. The procedure for switching servers is as follows.

Step 5-1 Excluding the current destination server, the OFC selects the server with the smallest number of OFSs from the client as a candidate server.

Step 5-2 The estimated acquisition throughput of the path between the candidate server and the client is calculated, and if the (8) condition is satisfied, the candidate server is selected as the switching destination server.

Step 5-3 If no server satisfies the condition in Step 5-2, the candidate server is released, and Step 5-1 and 5-2 are repeated.

Step 5-4 If none of the servers meet the conditions, the client maintains its current state.

After selecting the server to be switched, the OFC needs to update the flow table of OFSs and synchronize the flow information between the servers before and after switching. These are the costs associated with switching servers. However, a problem with this method is that a difference occurs between the traffic information held by the OFC and the latest information in the network, and that the servers and paths at switching destinations are concentrated in one part of the network. In order to suppress this problem, network information held by the OFC is updated every time a path or a server is switched. In this way, we aim to improve the communication quality of clients by flexibly switching servers and paths according to the network situation.

4. Simulation Environment

This section describes the simulation used to evaluate the performance of our proposed method. We evaluated this method using the Network Simulator-3 (NS-3) simulator [20].

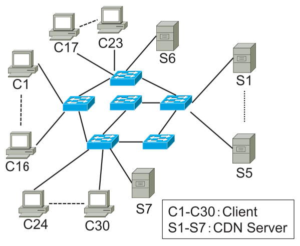


Fig. 1 Simulation topology.

Table 1 Simulation setting.

Element	Value
Link transfer delay	1 ms
Link bandwidth	100 Mb/s
Simulation times	400 s
Amount of content data	500 MB
Number of attempts	3
γ	5 s

4.1 Simulation Model

The simulation topology is shown in Fig. 1. Here, it can be seen that the configuration between OFSs is similar to that of the [11] simulation topologies. In Fig. 1, C1–C30 represents 30 clients, and S1–S7 represents seven CDN servers. Each terminal is connected via an OFS. The simulation settings are shown in Table 1. In this method, the smoothing coefficient α of the moving average is set to 0.8. This is because, by assigning weights to past values, frequent switching of servers and paths is suppressed. We also set $\beta = 0.6$ according to the result of [21].

4.2 Simulation Scenario

In the simulation, after a random amount of time elapses from the start, the client starts to communicate with the CDN server that has the smallest number of OFSs to pass through. The duration of each flow was set at 100 s, and the generation interval of the flow in each client was set based on the exponential distribution with an average from 100 s to 200 s. At this time, a parameter Load related to the generation interval of the flow is set. The relation between Load and the generation interval is given by (9).

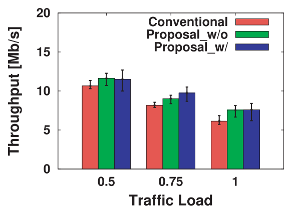
$$Load = \frac{Duration}{Generation\ Interval} \tag{9}$$

All flows that occur are 500 MB file transfers via the Transmission Control Protocol (TCP). The results were obtained when the OFC traffic information update period (cycle) was set to 1, 5, and 10 s, the time required for server switching (Overhead) was set to 0 and 5 s, and the ratio (Load) of the duration to the generation interval of the flow was set to 0.5, 0.75, and 1.0. For comparison purposes, we evaluated the throughput performance of the flows and the number of server and path switches using our proposal method (Proposal_w/), a method that does not switch servers and paths (Conventional), and a method that does not update information held by OFC after switching (Proposal_w/o).

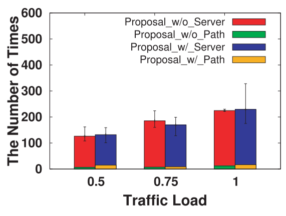
5. Simulation Results

This section verifies the effectiveness of the server and path switching method based on the evaluation items described in the previous section.

The error-bars in Figs. 2–4 represent the maximum and minimum value. Figures 2 (a) and (b) show the results for traffic load changes when Cycle = 10 s and Overhead = 5 s. Since the traffic volume in the network increases as the load increases, the throughput decreases and the switching frequency increases in each of the methods tested. In Fig. 2 (b), it can be confirmed that there is a significant difference between the maximum and minimum values when attention is paid to the switching times of the proposed method at the point when Load = 1.0. This is because unnecessary switching frequently occurs depending on the traffic patterns. From these results, we confirmed that the proposed method could operate normally and improve communication quality even if the traffic load changed.

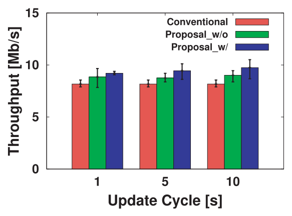


(a) Average throughput.

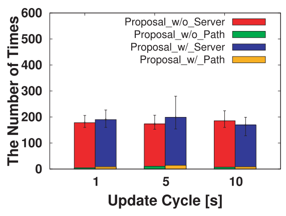


(b) Switching times.

Fig. 2 Load impact (Cycle = 10 s, Overhead = 5 s).

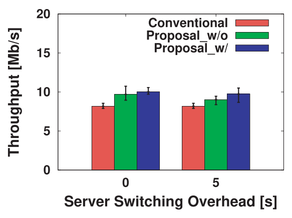


(a) Average throughput.

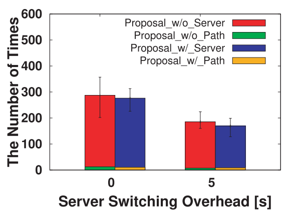


(b) Switching times.

Fig. 3 Update cycle impact (Load = 0.75, Overhead = 5 s).



(a) Average throughput.



(b) Switching times.

Fig. 4 Overhead impact (Load = 0.75, Cycle = 10 s).

Figures 3 (a) and (b) show the results of changes in the update cycle when Load = 0.75 and Overhead = 5 s are set. As can be seen in the figure, we confirmed that the load imposed by traffic information collection could be reduced by setting a longer update period because no significant effects were observed in the proposed method's throughput and switching times, even if the update period was changed. We also found that the proposed method could improve its throughput compared with other methods, and that its switching times were equivalent to Proposal_w/o. Taken together, these results show that a longer traffic renewal period can be set by OFC when the proposed method is used.

Figures 4 (a) and (b) show the results of changes in the time required for server switching when Load = 0.75 and Cycle = 10 s are set. Here, it can be seen that when the time required for server switching was changed, the throughput of the proposed method decreased. However, the proposed method still showed better throughput performance than the existing methods. From this result, we confirmed that our proposed method could improve communication quality when the time required for server switching was about 5 s. From the results of switching times, the number of server switching is more dominant than path switching. In our simulation topology in this letter, because there are few paths between the client and the server, it was difficult to discuss the priority and the necessity of path switching. On the other hand, when we assuming about topologies, such as Fat-tree structure used in the data center, there are many paths between the client and the server. Therefore, our proposed method's path switching will have the potential for improvement of the communication quality in the whole network.

We need to discuss the priority and necessity of path switching. Still, it is challenging only to discuss the simulation results this time, so evaluation by simulation using other topologies is our future work.

6. Conclusion

In this study, we focused on traffic increases due to content distribution using CDNs and proposed a method that avoids bottleneck links between clients and servers by flexibly switching servers and paths when communication degradation is encountered. Our proposed method, which was constructed in a CDN on an OpenFlow-based SDN, calculates the estimated acquisition throughput of the path between the client and the server in response to flow quality degradation, and then performs server and path switching that can improve that quality. In the future, we will examine the performance evaluation in other topologies and the switching method while simultaneously considering the effect of the switching operation on other flows.

References

[1] Cisco, "Cisco Visual Networking Index:, "

https://www.cisco.com/c/ja_jp/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[2] A.M.K. Pathan, et al., “A taxonomy and survey of content delivery networks,” Grid Comput. Distrib. Syst. Lab, Univ. Melbourne, VIC, Australia, Tech. Rep. GRIDS-TR-2007-4, Feb. 2007.

[3] N. Kamiyama, Y. Takahashi, K. Ishibashi, K. Shiomoto, T. Otoshi, Y. Ohsita, and M. Murata, “Optimizing cache location and route in content delivery service using model prediction control,” IEICE Tech. Rep., vol.114, no.477, NS2014-237, pp.349–354, March 2015.

[4] C. Liu, R.K. Sitaraman, and D. Towsley, “Go-with-the-winner: Performance based client-side server selection,” 2016 IFIP Networking Conference (IFIP Networking) and Workshops, pp.404–412, May 2016.

[5] F. Chen, R.K. Sitaraman, and M. Torres, “End-user mapping: Next generation request routing for content delivery,” Proc. 2015 ACM Conference on SIGCOMM, SIGCOMM ’15, pp.167–181, 2015.

[6] R.K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain, “Overlay networks: An Akamai perspective,” Advanced Content Delivery, Streaming, and Cloud Services, pp.305–328, Oct. 2014.

[7] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat, “Taking the edge off with espresso: Scale, reliability and programmability for global Internet peering,” Proc. Conference of the ACM Special Interest Group on Data Communication, SIGCOM ’17, pp.432–445, Aug. 2017.

[8] V.K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.-L. Zhang, M. Varvello, and M. Steiner, “Measurement study of Netflix, Hulu, and a tale of three CDNs,” IEEE/ACM Trans. Netw., vol.23, no.6, pp.1984–1997, Dec. 2015.

[9] M. Wichtlhuber, R. Reinecke, and D. Hausheer, “An SDN-based CDN/ISP collaboration architecture for managing high-volume flows,” IEEE Tran. Netw. Service Manag., vol.12, no.1, pp.48–60, March, 2015.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling innovation in campus networks, ” SIGCOM Comput. Commun. Rev., vol.38, no.2, pp.69–74, Nov. 2009.

[11] F. Qin, Z. Zhao, and H. Zhang, “Optimizing routing and server selection in intelligent SDN-based CDN, ”Wireless Communications & Signal Processing (WCSP), pp.1–5, Oct. 2016.

[12] C. Chen-xiao and X. Ya-bin, “Research on load balance method in SDN,” International Journal of Grid and Distributed Computing, vol.9, no.1, pp.25–36, Sept. 2016.

[13] J. Li, X. Chang, Y. Ren, Z. Zhang, and G. Wang, “An effective path load balancing method in SDN,” 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp.527–533, Sept. 2014.

[14] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” NSDI ’10 Proc. 7th USENIX Conference on Networked Systems Design and Implementation, vol.10, no.8, pp.89–92, April 2010.

[15] H. Xu, Z. Yu, X.-Y. Li, C. Qian, L. Huang, and T. Jung, “Real-time update with joint optimization of route selection and update scheduling for SDNs,” 2016 IEEE 24th International Conference on Network Protocols (ICNP), pp.1–10, Nov. 2016.

[16] H. Long, Y. Shen, M. Guo, and F. Tang, “LABERIO: Dynamic load-balancing routing in openflow-enabled networks,” 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp.290–297, March 2013.

[17] Y.-L. Lan, K. Wang, and Y.-H. Hsu, “Dynamic load-balanced path optimization in SDN-based data center networks,” 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), pp.1–6, July 2016.

[18] S.J. Kesaven and J. Jayakumar, “Dynamic adaptive streaming over HTTP (DASH)-comprehensive study and rate adaptation performance analysis,” International Journal of Soft Computing, vol.10, no.4, pp.261–273, Jan. 2015.

[19] J. Jiang, V. Sekar, and H. Zhang, “Improving fairness efficiency, and stability in HTTP-based adaptive video streaming with Festive,” IEEE/ACM Trans. Netw., vol.22, no.1, pp.326–340, Feb. 2014.

[20] nsnam, “ns-3, ” https://www.nsnam.org

[21] H. Nishimuta, D. Nobayashi, and T. Ikenaga, “Adaptive server and path switching scheme for content delivery network,” 2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), pp.1–6, Aug. 2019.