How to realize CDN Interconnection (CDNI) over OpenFlow?

Dukhyun Chang
Seoul National University
Seoul, Korea
dhchang@mmlab.snu.ac.kr

Junho Suh
Seoul National University
Seoul, Korea
jhsuh@mmlab.snu.ac.kr

Hyogi Jung
Seoul National University
Seoul, Korea
hgjung@mmlab.snu.ac.kr

Ted "Taekyoung" Kwon
Seoul National University
Seoul, Korea
tkkwon@snu.ac.kr

Yanghee Choi
Seoul National University
Seoul, Korea
yhchoi@snu.ac.kr

ABSTRACT

Traffic of content-oriented applications/service is increasingly more dominant. How to support content delivery and distribution efficiently is one of the important issues in future Internet community. In contrast to clean slate approaches in the recent literature, we seek to deliver content efficiently leveraging current networking devices with programmability. In particular, we propose how to deliver content across Internet service providers (ISPs) on top of OpenFlow. We design a preliminary framework to support content delivery network interconnection (CDNI) assuming two ISPs have deployed OpenFlow. There are two central ideas to realize CDNI across ISPs: (i) request and data packets of a content file are forwarded by a private address (assigned to the file) in each ISP, and (ii) the controllers of the two ISPs exchange the signaling information to deliver the content.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network communications*

Keywords

Content-Oriented, Software-Defined Networking, OpenFlow, CDNI

1. INTRODUCTION

While the current Internet is designed based on host-to-host communications, traffic of content-oriented applications and services such as P2P and content delivery networks (CDNs) is increasingly more dominant [1]. How to support content delivery and distribution efficiently is one of the important issues. Recently there have been clean-slate proposals for content-oriented networking (CON) to address this issue [3] [2]. Although the clean slate approaches have many advantages like mobility, security in addition to efficient content delivery, it is not easy for internet service providers (ISPs) to deploy these solutions due to interoperability and backward-compatibility.

Software-defined networking (SDN) emerges as a viable solution to provide programmability and controllability to current IP networking. With SDN, many new protocols and algorithm can be deployed without disrupting the current IP traffic. OpenFlow is a most representative component for SDN solutions, which allows network admins/researchers to take diverse actions in the forwarding plane (of a network switch or a router) [5].

In this paper, we propose an OpenFlow-based framework to support CON across ISPs by leveraging the current OpenFlow functionalities. The key idea is to map a content file to a private address within an ISP. Firstly we explain how the proposed framework performs CON using private addresses in an ISP. Then we extend the framework to realize CDN interconnection (CDNI) [4], which allows "managed and provisioned" content delivery between ISPs. For this, we will show how the framework covers four CDNI functionalities: control, request routing, distribution and logging. We assume that two ISPs have deployed OpenFlow respectively.

2. CON OVER OPENFLOW IN AN ISP

The central idea of supporting CON over OpenFlow is to map a private address to each content file to be delivered. Request packets (i.e. HTTP GET) for the content as well as its data packets will be routed by the private address assigned on-demand by the OpenFlow controller, which emulates route-by-name. In-network caching is also possible, which is also illustrated below.

Fig. 1 illustrates how content is delivered in the proposed framework. First, client1 sends an HTTP GET message to get content C1 (youtube.com/a.avi) to its access router R1 (1). R1 then relays this message to the controller (2), which has a mapping table that keeps track of which content is cached at which router (i.e., a co-located in-network storage). At this moment, C1 is not cached, and hence there is no entry in the mapping table. The controller will then contact the DNS to find out the original server (youtube.com), and assign a private address 10.1.2.3 for C1. The controller

CFI '12, September 11-12, 2012, Seoul, Korea

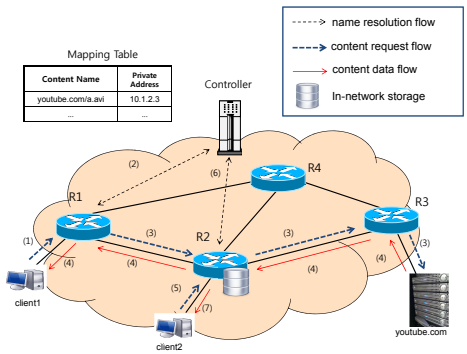

Figure 1: OpenFlow-based signaling framework for efficient contents delivery in an ISP.

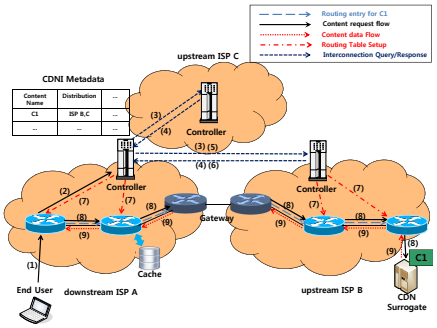

Figure 2: The content is delivered across ISPs over the OpenFlow-based framework.

will send control messages (skipped in Figure 1) to R1, R2, and R3 to set up forward/backward routing entries to forward GET request packets and the corresponding data packets. Then, the HTTP GET requests (whose destination address is 10.1.2.3) will be delivered from R1 to `youtube.com` (3). The server will send data packets (whose source address is 10.1.2.3) back to client1 (4). Suppose R2 decides to cache these packets in its co-located storage, which will be informed to the controller. Later, client2 sends the HTTP GET message to its access router R2 to download the same content (5). R2 asks the controller which node has the content C1 (6). (As R2 performs no DPI, it cannot know this HTTP GET is for C1.) Then the controller requests R2 to deliver the content from R2's own storage, which is then carried out (7).

3. CONTENT DELIVERY BETWEEN ISPS

CDNI has four functionalities: control, request routing, distribution and logging. The control messages are exchanged via the corresponding four interfaces: a control interface, a request routing interface, a CDNI metadata interface, and a logging interface, respectively. In this section, we will show how these four interfaces are realized in the proposed framework.

Fig. 2 illustrates one of the ways in which the framework supports CDNI. The control interface is to initialize/maintain CDNI between ISPs, whise details are not yet defined; we assume the controllers of ISPs (or their proxies) can set up the control interface. When an end user issues a content request (1)(2), the controller in ISP A checks its local mapping table. If no mapping entry, the controller checks whether other ISPs (connected via CDNI) have the content before contacting the DNS to find out the original server. For sake of simplicity, we assume that CDNI metadata is pre-positioned between ISPs over the CDNI metadata interface; that is, a controller knows the content repositories of other ISPs (more precisely, CDN surrogates). Suppose the requested content is stored in ISPs B and C; then the controller of ISP A sends requests to both ISPs B and C whether they can deliver the content, perhaps with the QoS requirement (3). The controllers of ISPs B and C check the location of the content and network traffic status, and reply with how much bit rate can be provided while delivering the content (4). Messages (3) and (4) are exchanged over the request routing interface. Suppose ISP A's controller selects ISP B since it provide the better QoS. A's controller assigns its private address to the content, which is informed to B's controller (5). B's controller also assigns its own private address, which is informed to A's controller (6). In this way, A's controller and B's controller set up a path for content delivery between the end user in ISP A and the CDN surrogate in ISP B (7).

Recall that each controller assigns its own private address to forward request and data packets. ISP B's controller informs its gateway of the two private addresses, so that a request packet with A's private source address should be replaced by B's private source address (8). The converse action will be done at the gateway of ISP A for data packets (9).

Lastly, the logging function of CDNI will record all the information of the delivery: who requested the content, how much of content has been delivered, the throughput of content delivery, security information, and so on.

4. ACKNOWLEDGMENTS

This research was supported by the IT R&D program of KCA(10913-05004: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity) and the ICT at Seoul National University provides research facilities for this study.

5. REFERENCES

[1] Internet study 2008/2009. http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proceedings of ACM CoNEXT*, 2009.

[3] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *Proceedings of ACM SIGCOMM*, 2007.

[4] B. Niven-Jenkins, F. L. Faucheur, and N. Bitar. Content distribution network interconnection (cdni) problem statement, March 2012. draft-ietf-cdni-problem-statement-04.

[5] Openflow switch specification, February 2011. Version 1.1.0 Implemented (Wire Protocol 0x02).