HTTP-CCN Gateway: Adapting HTTP Protocol to Content Centric Network

Zhaogeng Li, Jun Bi*, Sen Wang
Institute for Network Sciences and Cyberspace, Tsinghua University
Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList)
Beijing, China
li-zg07@mails.tsinghua.edu.cn, junbi@tsinghua.edu.cn, wangsen@netarchlab.tsinghua.edu.cn

***Abstract*—CCN is one of the future Internet architecture. However, the lack of real traffic becomes an obstacle to advanced CCN researches. HTTP, as an content-oriented application-layer protocol working over current Internet, is similar to CCN in many aspects. In this demo, we try to convert HTTP traffic into CCN traffic with HTTP-CCN gateway. Although HTTP is not equivalent to CCN, we can design carefully to make the conversion correct in most situations. The gateway will introduce real traffic into CCN testbed to support CCN researches. Our demonstration will show how to use this gateway.**

***Keywords*—CCN, HTTP, Gateway**

I. Introduction

As content dissemination is becoming the primary application of Internet, Information-Centric Network (ICN) is considered as one of the future Internet architecture. Content Centric Network (CCN [1]), among many ICN proposals, draws more and more researchers' attention. Unlike traditional IP network, each piece of content transmitted in CCN is identified with a URL-like name. CCN follows receiver-driven pattern, which is also known as the term "PULL". There are two types of packets: Interest (request) and Data (response). To get data, a user needs to send an Interest carrying the correct name before a Data containing the corresponding content is returned along the reverse path of this Interest. Any CCN node can cache passing Data to meet the demand of subsequent Interest with the same name.

HTTP, which has become one of the predominant protocols for current Internet, can be considered content-oriented. In fact, we can find many similarities between HTTP and CCN. For example, HTTP also follows receiver-driven pattern. A client should send an HTTP request with a URL to get HTTP response back. Therefore, we can try to convert HTTP request to CCN Interest, and convert HTTP response to CCN Data. However, HTTP is not equivalent to CCN, because HTTP has to understand more semantics since it is an application-layer protocol. As a result, the conversion from HTTP to CCN will drop some semantic details.

We note most research works on CCN employ simulated traffic. The lack of real traffic becomes one obstacle to in-depth research on CCN. To cope with this problem, we try to convert HTTP traffic into CCN traffic. Although the conversion is not perfect, the converted traffic is still useful. In the macroscopic view, the converted traffic can be considered as real traffic and used for studying advanced CCN topics, such as content caching. In this demonstration, we design and implement an HTTP-CCN gateway to achieve this goal.

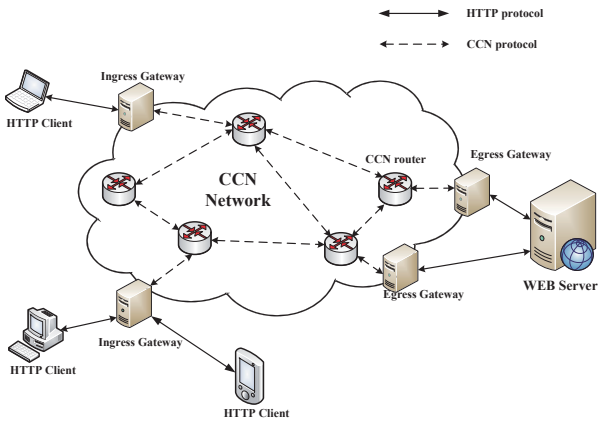

Fig. 1. Overview of HTTP-CCN gateway.

II. Implementation

A. Overview

HTTP-CCN gateway works on the edge of CCN testbed. Fig. 1 shows the system overview. The gateway has two different identities: ingress gateway and egress gateway. Ingress gateway receives HTTP requests from HTTP clients (e.g. browsers), converts them to CCN Interest packets and sends them in CCN testbed. Egress gateway receives the CCN Interest packets, converts them to HTTP requests, and sends the requests to HTTP servers. When egress gateway gets HTTP responses from HTTP servers, it converts them to CCN Data packets, and returns them in CCN testbed. When ingress gateway gets CCN Data packets, it converts them to HTTP responses, and returns the responses to HTTP Clients.

To implement HTTP-CCN gateway, we extended an open source and light-weight WAP and SMS gateway named *Kannel* [2]. The HTTP proxy function in *Kannel* is used for communicating with HTTP clients and HTTP servers. By setting ingress

*Jun Bi is the corresponding author

978-1-4799-1270-4/13/$31.00

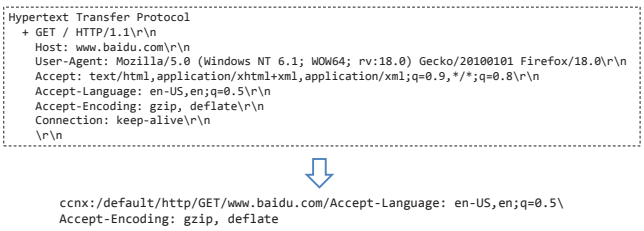

Fig. 2. An example of CCN name generation. Some fields in HTTP header become components of CCN names.

gateway as an HTTP proxy, HTTP requests will be directed to the CCN testbed. The CCN testbed is built on *CCNx* [3], an open source implementation of CCN. We implement HTTP-CCN gateway as an application of *CCNx*.

B. Protocol Conversion

The most important part of HTTP-CCN gateway is protocol conversion. As mentioned in section 1, HTTP is not equivalent to CCN. As an application-layer protocol, HTTP have to understand more semantics. For example, there are many header fields in HTTP request, and each field represents a description of the data. However, if we convert HTTP requests to CCN Interest packets, we can only put the header fields in CCN names.

Fig. 2 gives an example of name generation. In addition to the URL, some header fields become components (any string between two slashes is called a component) of name. For example, *Accept-language* and *Accept-encoding* describe the content language and encoding. We should add the two fields into name. But in the most situations, we can ignore the other fields (*User-agent*, *Accept*, *Connection*).

Besides the HTTP header, there are many other differences between CCN and HTTP we have to handle:

- Content Caching: Caching is considered as one of the most important features of CCN. And HTTP also support content caching. However, HTTP specification for caching behavior is pretty complicated. We have to calculate the freshness of the content according to different HTTP header fields (e.g. *Pragma*, *Cache-control*, *Expires*).
- *POST* Method: *GET* and *POST* are the most common HTTP request methods. In Fig. 2, the component *GET* in the name indicates the method of the request. For *POST* method, there is some data in the body of HTTP request. Our egress gateway should send an Interest to ingress gateway to get the data.
- Content Segmentation: Generally, the HTTP response should be segmented to many small chunks at egress gateway according to the upper limit of Data size. The segmentation often begins before the whole response is received, in order to avoid too high latency for user when the response is too large.

One can refer to [4] to get more details about the protocol conversion.

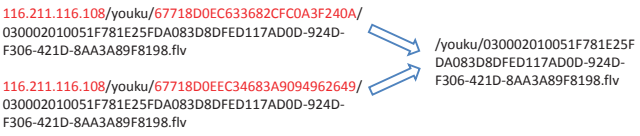

Fig. 3. URL modification for *Youku* video. In order to take advantages of cache, we should ignore the IP address of CDN node and random string in original URL.

III. Demonstration

In this demonstration, our HTTP-CCN gateways will serve as HTTP proxies. The deployed gateways are connected with CCN testbed. We are planing to show the following two things:

- Our gateway can work correctly in most situations. We will present the result of accessing the website with the gateway, and the log information of the gateways and CCN testbed, which can also help people to understand the protocol conversion.
- Our gateway can accelerate content access, especially for video watching and file downloading, with the help of cache in CCN testbed. To evaluate the performance, we will compare the time for downloading a video/file for the first time by someone and the second time by someone else.

Especially, we will show how to cache online video in our CCN testbed. Generally, users need to use different URLs to get the same video in different time. We should generate CCN names with modified (non-original) URLs if we want to identify the same video with the same name (Fig. 3). There is no common solution to this problem because different online video providers generate URLs in different ways, so that we have to put different patches on different online video providers. In this demonstration, we will show our patch on *Youku*, one of the biggest online video providers in China.

We hope the demonstration will show that HTTP-CCN gateway improves user experience, especially for downloading files or watching videos. As a result, more and more users will choose to use this gateway, so that we will get more real CCN traffic to support advanced CCN researches.

Acknowledgement

Supported by the National High-tech R&D Program (“863” Program) of China(No.2013AA010605), the National Science Foundation of China (No.61073172), and Tsinghua-Huawei joint research project. Our thanks to the assistance from Huawei Research Center.

References

[1] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.
[2] Kannel. http://www.kannel.org/.
[3] CCNx. http://www.ccnx.org/.
[4] Sen Wang, Jun Bi, Jianping Wu, Xu Yang, and Lingyuan Fan. On adapting http protocol to content centric networking. In *Proceedings of the 7th International Conference on Future Internet Technologies*, pages 1–6. ACM, 2012.