Design and Implementation: the Native Web Browser and Server for Content-Centric Networking

Guoshun Nan, Xiuquan Qiao, Yukai Tu, Wei Tan[§], Lei Guo, Junliang Chen
State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, 100876, Beijing, China
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA [§]
nanguoshun@bupt.edu.cn, qiaoxq@bupt.edu.cn, tuyukai1994@gmail.com,
wtan@us.ibm.com, guolei@upc.edu.cn, chjl@bupt.edu.cn

ABSTRACT

Content-Centric Networking (CCN) has recently emerged as a clean-slate Future Internet architecture which has a completely different communication pattern compared with exiting IP network. Since the World Wide Web has become one of the most popular and important applications on the Internet, how to effectively support the dominant browser and server based web applications is a key to the success of CCN. However, the existing web browsers and servers are mainly designed for the HTTP protocol over TCP/IP networks and cannot directly support CCN-based web applications. Existing research mainly focuses on plug-in or proxy/gateway approaches at client and server sides, and these schemes seriously impact the service performance due to multiple protocol conversions. To address above problems, we designed and implemented a CCN web browser and a CCN web server to natively support CCN protocol. To facilitate the smooth evolution from IP networks to CCN, CCNBrowser and CCNxTomcat also support the HTTP protocol besides the CCN. Experimental results show that CCNBrowser and CCNxTomcat outperform existing implementations. Finally, a real CCN-based web application is deployed on a CCN experimental testbed, which validates the applicability of CCNBrowser and CCNxTomcat.

CCS Concepts

•Information systems → Web applications; •Networks → Network experimentation;

Keywords

Content-Centric Networking, Web browser, Web server

1. INTRODUCTION

In recent years, the World Wide Web (WWW) has become one of the most fundamental Internet applications. Web browsers and servers play a key role in satisfying the needs of a large and growing community of web users. Content-Centric Networking (CCN) [1] has emerged as a clean-slate network architecture to address the problems faced by current IP networks.

Existing web browsers and servers are designed for the hypertext transfer protocol (HTTP) and IP-based Internet. Consequently, the deployment of the existing web applications on CCN is still an open issue due to the lack of interconnections tools. Current research focuses on the gateway for HTTP and CCN protocol conversion [5] or plug-in manner [4] to validate CCN under real HTTP traffic. However, a gateway is not an efficient approach given the additional cost of the protocol conversion. The lack of native support for web applications has greatly hampered the wide adoption of CCN.

To address above issues, we designed and implemented a full-fledged CCN web browser (called CCNBrowser) [2] and a server (called CCNxTomcat) [3]. CCNBrowser was developed based on the open source Webkit, and CCNxTomcat was based on Apache Tomcat. We leverage CCNx developed by PARC laboratory as the CCN prototype. Native CCN functions are integrated into implementations to support interactions among users, servers and routers in CCN. In addition, to facilitate smooth network evolution, our implementations support the content request using both 'ccnx://' and 'http://' URI schemes for HTML, CSS and JavaScript.

2. IMPLEMENTATION

To give a clear description, architectures of CCNBrowser and CCNxTomcat are depicted in Fig. 1 and Fig. 2, respectively, where the components highlight in blue are CCN related.

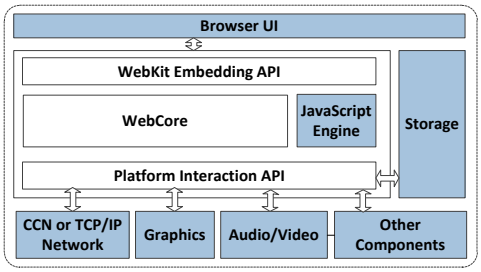

Figure 1: CCNBrowser architecture

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: http://dx.doi.org/10.1145/2785956.2790024

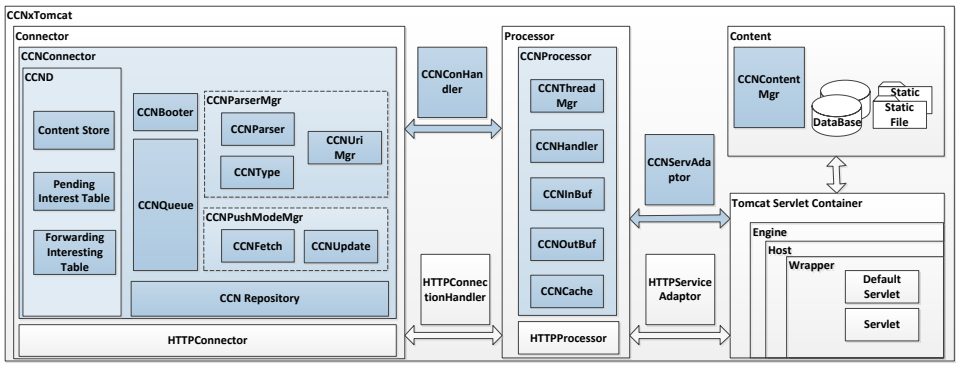

Figure 2: CCNxTomcat architecture

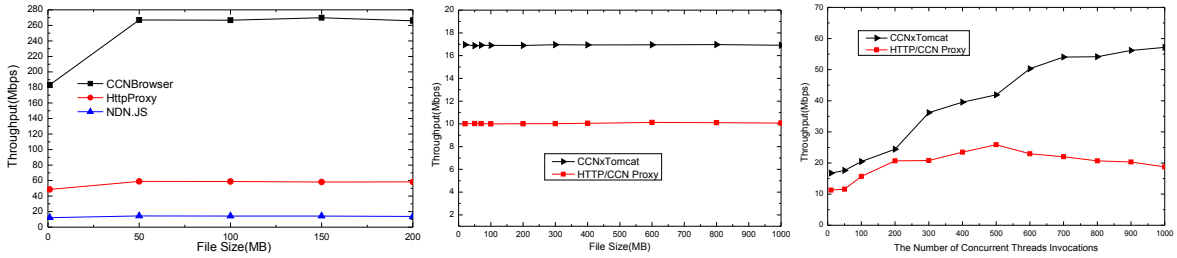

Figure 3: Test results: (a) CCNBrowser vs. NDN.JS and HTTP/CCN Proxy in terms of the single file test. (b) CCNxTomcat vs. HTTP/CCN Proxy in terms of the single file test. (c) CCNxTomcat vs. HTTP/CCN Proxy in terms of the throughput.

3. EXPERIMENT

We conduct experiments on a real testbed to compare the performance of CCNBrowser and CCNxTomcat with existing NDN.JS and HTTP/CCN proxy. The results in Fig. 3 indicate that CCNBrowser exceeds existing NDN.JS and HTTP/CCN gateway implementations by a factor of nineteen and five, in terms of the throughput, and CCNxTomcat outperforms existing HTTP/CCN implementations by 58% in a single request, and by a factor of three under 1000 concurrent requests. We also setup a real CCN-based personalized web site as shown in Fig. 4 for the practical utilization of CCNBrowser and CCNxTomcat.

Figure 4: A real CCN-based personalized web site

4. ACKNOWLEDGMENTS

This work was supported by the National Key Basic Research Program of China (973 Program) under Grant No. 2012CB315802, the National Natural Science Foundation of China under Grants No. 61171102 and No. 61132001, the Prospective Research on Future Networks of Jiangsu Future Networks Innovation Institute under Grant No. BY2013095-4-01.

5. REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. L. Braynard. Networking named content. CoNEXT 09. ACM, 2009.

[2] X. Qiao, G. Nan, Y. Peng, L. Guo, J. Chen, Y. Sun, and J. Chen. Ndnbrowser: An extended web browser for named data networking. *Journal of Network and Computer Applications*, 50:134–147, 2015.

[3] X. Qiao, G. Nan, W. Tan, L. Guo, J. Chen, W. Quan, and Y. Tu. Ccnxtomcat: An extended web server for content-centric networking. *Computer Networks*, 75:276–296, 2014.

[4] W. Shang, J. Thompson, M. Cherkaoui, J. Burkey, and L. Zhang. Ndn. js: a javascript client library for named data networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 2013.

[5] S. Wang, J. Bi, J. Wu, X. Yang, and L. Fan. On adapting http protocol to content centric networking. In *Proceedings of the 7th International Conference on Future Internet Technologies*. ACM, 2012.