CANA: One Step From IP Networking Toward Content Networking

Junho Suh, Donghyun Jang, Ted "Taekyoung" Kwon, Yanghee Choi
School of Computer Science and Engineering
Seoul National University
Seoul, Korea
{jhsuh, dhjang}@mmlab.snu.ac.kr, {tkkwon, yhchoi}@snu.ac.kr

1. INTRODUCTION

The host-to-host communications model cannot efficiently accommodate the current Internet usage, which focuses on content distribution/retrieval than the use of host resources. To bridge the gap, we propose a new content-aware networking architecture (CANA) which is characterized by (i) content identifier (e.g. URL) is added into the TCP/IP protocol header (e.g. IP option header), so that network entities such as routers are aware of the content being requested/delivered, and (ii) the role of an IP address is reduced; it indicates not the endpoint, but a transit point. For each transit point, there should be a mapping entry (as similar to NAT) that connects two adjacent ones of an end-to-end path. In this way, CANA can deliver the contents efficiently to hosts even with private addresses while inherently supporting multicast and mobility. Furthermore, as CANA does not modify the existing TCP/IP header, it can be incrementally deployed with legacy routers, in contrast to the prior content networking proposals [1, 2].

2. CONTENT-AWARE NETWORKING ARCHITECTURE (CANA)

2.1 Design

How can we incorporate the content awareness into the network with keeping the TCP/IP header fields intact to support legacy routers? In CANA, we first use the IP option header to add a content identifier (CID) enabling a network entity to interpret for figuring out the final destination of the corresponding content. We may optionally add other parameters such as bit rate. Such content-related information may help other layers deliver the packets of the particular content more appropriately.

We extend an access router to be aware of the content context by its CID (i.e., URI), and rename it as a content-aware agent (CAA). The reason of renaming is that a CAA solicits a content on behalf of its host. Note that a CAA has a public IP address on its egress link. To receive a content file, the host should first send a content request message containing a CID to its CAA. Then the CAA will vicariously download the content from the corresponding server and relay the content to the host. We can also substantiate this mechanism in a backward-compatible manner. That is, the host sends an HTTP GET message, and the CAA performs the deep packet inspection (DPI) to interpret the CID.

If an intermediate router is modified to interpret a CID (in the IP option header), it is called a content-aware router (CAR). Note that a CAR will be incrementally deployed and there will be a mix of legacy routers and CARs in an ISP's network. CAAs and CARs are collectively called content-aware intermediaries; similarly, end-hosts and content-aware intermediaries are content-aware network entities. Each content-aware intermediary may have a cache to store cacheable contents. For instance, if the content request message encounters a CAR that currently stores the corresponding content, the content will be delivered from the CAR to the host.

A content is published by its publisher, residing under its CAA so a publisher has its domain name. Note that its CAA's public IP address is registered in the resolution system, such as DNS, for the publisher's domain name. In this way, a content request message can be forwarded toward the publisher's CAA. Note that a CAA of a soliciting host will send the name resolution query on behalf of the host. Then the resolution system will reply with the public IP address of the publisher's CAA.

2.2 Mobility

ACM CoNEXT Student Workshop, December 6, 2011, Tokyo, Japan.

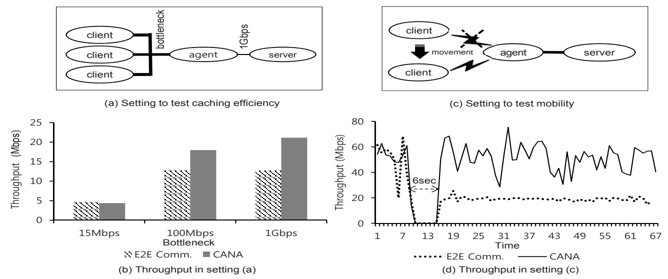

Figure 1: Caching enhances the content delivery efficiency substantially. The throughput of CANA outperforms that of end-to-end model as shown in (b). Caching also affects the throughput after the host switches to a new IP address as shown in (d).

Let us now discuss how CANA can live with host mobility. As the serving CAA is not changed in an L2 handoff, we focus on L3 handoff here. In an L3 handoff, after the setup of the network level connectivity with the new CAA, the host will send a content request message[1]. This time the host requests only the remaining part of the content. The content request message sent by the host will be received by the new CAA, and then will be forwarded toward the CAA of the publisher. If the content request message encounters a CAR that currently stores the corresponding content, the host may receive the content faster than endpoint-based mobility solutions (e.g. Mobile IPv4 and transport level mobility solutions).

3. EVALUATION

We develop the prototype of CANA on NetFPGA platforms to evaluate its performance of content retrieval efficiency and mobility support. The implementation is located on the top of TCP/IP layer adopted from lightweight TCP/IP stack [4] in the SCONE framework of NetFPGA. To highlight the advantage of content-aware networking, the requested content is already cached in the content aware intermediary before experiments.

3.1 Content Retrieval Efficiency

To evaluate content retrieval efficiency, we carry out the experiment in which content retrieval efficiency in CANA is compared with end-to-end communication model. To do end, we configure our testbed as shown in Figure 1(a), in which three clients (or hosts) simultaneously request the same 300MB file from the server by establishing TCP sessions, and the bottleneck link between the content-aware node (in the middle) and the clients, is varied from 15Mbps to 1Gbps. We plot the average download throughput of the three clients as the performance metric in Figure 1(b). In CANA, the content requests from the clients are directly handled by the content-aware node since it stores the content. In contrast, in the end-to-end model (i.e. the current TCP/IP model) all requests are handled by the server only. Even in the case of the 1Gbps link setting, the total throughput cannot exceed approximately 60Mbps. The reason is that lightweight TCP [4] in the NetFPGA platform does not support TCP window size scaling [5] for high speed bandwidth.

[1]Like the range requests in HTTP, the host can request only the remaining part of the content that has been downloaded.

3.2 Mobility Support

To demonstrate how well CANA handles the host mobility, we configure a client that has two interfaces connected to two different content-aware nodes as shown in Figure 1(c). In this scenario, the network level connectivity of the client is switched between the two content-aware nodes at 8 seconds. After the handoff emulation, the host establishes the new connectivity (which takes almost 6 seconds) and then requests the remaining part of the content. Due to the caching effect and the content awareness, the throughput of CANA performes better than the end-to-end model as shown in Figure 1(d).

4. CONCLUSION

In this paper, we propose a content-aware networking architecture (CANA) for content retrieval efficiency, mobility, and multicast. We carry out experiments on NetFPGA testbeds to demonstrate the performance gain of CANA over IP networking.

5. ACKNOWLEDGEMENT

This research was supported by the IT R&D program of KCA (11-913-05-002: Fundamental Research on In-network Caching and Routing for Named-data Networking) and MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2011-(C1090-1111-0004)).

6. REFERENCES

[1] Teemu Koponen et al., "A Data-Oriented (and Beyond) Network Architecture," ACM SIGCOMM 2007.
[2] Van Jacobson et al., "Networking named content," ACM CONEXT 2009.
[3] Alex Snoeren and Hari Balakrishnan, "An end-to-end approach to host mobility," ACM MobiCom 2000.
[4] Lightweight TCP, http://savannah.nongnu.org/projects/lwip/.
[5] RFC 1323, http://www.faqs.org/rfcs/rfc1323.html.