Could End System Caching and Cooperation Replace In-Network Caching in CCN?

Haibo Wu wuhaibo@cstnet.cn
Jun Li+ lijun@cnic.cn
Jiang Zhi zhijiang@cstnet.cn
Computer Network Information Center, Chinese Academy of Sciences, Beijing, China, 100190

ABSTRACT

CCN has been witnessed as a promising future Internet architecture. In-network caching has been paid much attention, but there is still no consensus on its usage, due to its non-negligible costs. Meanwhile, massive storage and bandwidth resources of end systems still remain underutilized. To this end, we present an **E**nd **S**ystem **C**aching and **C**ooperation scheme in CCN, called ESCC to realize content distribution of CCN, without using costly in-network caching. ESCC enables fast content distribution through clients caching and sharing contents with each other. Experiments show that ESCC can achieve better performance than the universal caching. It is also quite simple, efficient, robust and has low overhead. ESCC could be a candidate substitute for the costly and unnecessary universal caching.

Categories & Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design

Keywords: CCN, end system, caching, cooperation

1. INTRODUCTION

CCN (Content Centric Networking) [1] has been proposed as a promising alternative to current host-to-host communication paradigm. In-network caching is regarded as its key feature. Overall in-network routers equipped with caches potentially store named contents, which can be used for subsequent requests. In-network caching can reduce bandwidth consumption and content retrieval time by making contents closer to requesters. However, these benefits come at a non-negligible cost. Such universal caching introduces numerous extra operations to routers (e.g., content storage, lookup and replacement) and aggravates routers' overhead. Especially, this poses serious challenges on line-speed forwarding of routers.

So far, there has been no consensus on its feasibility. Current trace-driven studies [2] show that in-network caching is not fundamental necessary. Meanwhile, relevant studies indicate larger and larger contents are being exchanged between nearby end systems [3]. Especially, end systems have spare storages and bandwidth resources, which remain underutilized.

To this end, we propose a simple end system caching and cooperation scheme in CCN, aiming to replace the costly universal in-network caching. Our scheme is named ESCC, and several major advantages are listed as below.

- **Simple**: ESCC is an incrementally deployable scheme. The request processing remains the same as CCN. The only modification is associated with data processing and none of extra components is required.
- **Efficient**: ESCC enables requesters to retrieve contents from nearby neighbors through client cooperation. Data sharing works in a P2P-like fashion.
- **Robust**: ESCC is robust to client failure and unavailable content, because the adaptive forwarding [4] of CCN can detect and correct wrong routes by timeout.
- **Low Overhead**: ESCC does not need to advertise content available information like the literature [5], when routing changes. In essence, the "advertising process" is done through the update of FIB. ESCC has a low overhead.

2. SCHEME DESIGN

The main idea of ESCC is that each client caches contents and shares them with other neighbors. The requester/client equipped with a cache potentially turns into a content provider after a period of time. When data is transmitted back to the requester, some nearby routers record the route trace in their FIBs. Following the route trace, subsequent requests could be redirected to the former requester (now content provider). In this way, ESCC can enable fast access to contents within client's neighborhood.

Specifically, the request message processing remains the same as CCN. The mere modification is related to data processing. When a data message is transmitted back to a requester, some nearby routers record a route trace to the requester in FIB. Following the route trace, subsequent requests from other nearby clients could be redirected to the requester (now content owner). The route trace enables fast access to data within neighborhood.

Now, we give an example. Assume client 1 wants to retrieve a content named "/cinc.cn/ccn.jpg", which is possessed by the server. As shown in Fig. 1, first client 1 issues a request message with a name of "/cinc.cn/ccn.jpg/chunk1". It will traverse a path through A-C-D-E and reach the server. Then the data message returns back in the opposite direction. When the data message travels to a location close to client 1, e.g., router C, a route trace with the name of "/cinc.cn/ccn.jpg/chunk1" toward A will be

+Jun Li is the corresponding author.

SIGCOMM '15, August 17-21, 2015, London, United Kingdom.
ACM 978-1-4503-3542-3/15/08. http://dx.doi.org/10.1145/2785956.2790006

added into C's FIB. Likewise, router A also adds a route trace toward client 1 in its FIB.

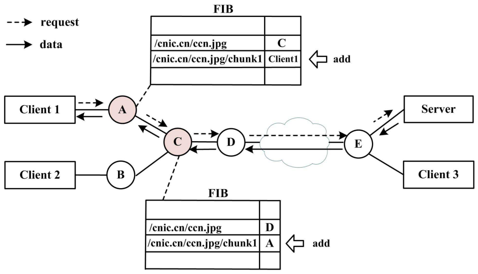

Figure1. Client 1 retrieves content from server

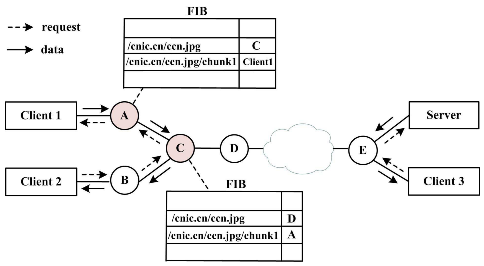

Figure 2. Client 2 retrieves content from client 1

As shown in Fig. 2, when client 2 requests for the same content later, the request will be redirected by router C to client 1 rather than to server. As the longest prefix match is used in FIB, the entry of "/cinc.cn/ccn.jpg/chunk1" has a high priority over the entry of "/cinc.cn/ccn.jpg". Thus, the longest prefix match still fits for ESCC.

Actually, end systems only cooperate with each other in the neighborhood, while FIBs of far routers remain unchanged. Note that FIB of A and C is updated, while E's FIB stays intact. Thus, client 3 can still retrieve content from the nearby server, rather than from the remote client 1.

3. ALGORITHM

The algorithm for data message processing at each router is shown in Table I (request message processing is the same as CCN). If FIB records the route trace for all passing data packets, FIB will expand dramatically. To tackle this issue, first, route trace can be only added to the FIB of routers in short distance away from the requester. Second, route trace can be only generated for popular contents.

TABLE I. ALGORITHM FOR DATA MESSAGE PROCESSING

```
/* r:current router; d:current data; p_d:popularity of d;
   h_r:hop count from r to client; T_p, T_h: popularity/hop threshold */
FOR (each matched PIT entry)
  forward data d via corresponding face to user;
  delete the corresponding PIT entry;
  IF (p_d>T_p)    /* Content selection */
    IF (h_r<T_h)    /* Router selection */
        record the route trace in FIB;
    End IF
  End IF
End FOR
```

4. EVALUATION

To evaluate ESCC, we conduct extensive simulation experiments based on the NDNSim. We utilize the Watts-Strogatz (WS) model [6] to generate the topology, which can capture the characteristics of Internet topology structure. In-network caching and ESCC is compared under the same

TABLE II. PARAMETER SETTING

Parameter	Value	Parameter	Value	Parameter	Value
Warmup time	10s	Cache size	50	Server num	1
Request rate	200req/s	Replacement	LRU	Core router num	66
Content num	50000	Hop threshold	2 hops	Edge router num	33
Parameter α	0.7~1.3	Pop threshold	Top 30%	Client num	33

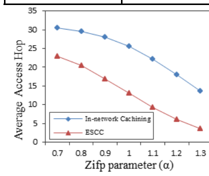

Figure3. Access delay

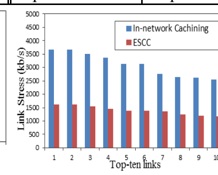

Figure4. Link traffic

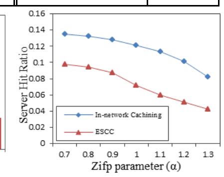

Figure5. Server load

total deployment cost. We assume the storage cost of per MB for in-network caching is 5 times the storage cost for clients, according to literature [7]. Clients' requests follow Zipf distribution with the parameter 0.7<α<1.3.

Experiments show that ESCC (only leave trace in routers two hops from clients), has already achieved better performance than the universal caching, in term of access delay, link traffic and server load reduction (Fig.3-5). Thus, ESCC could be a substitute for the costly universal caching.

5. CONCLUSION

In this paper, we propose a lightweight way to realize content distribution in CCN through client caching and cooperating with each other, without using costly in-network caching. When data is transmitted to a requester, some routers close to the requester leave route traces in their forwarding tables. Following the route traces, the subsequent requests can be redirect to the requester (data owner). Experiments show that ESCC (only leave trace in routers two hops away from clients) has already achieved lighter better performance gain than the universal caching. It is also quite simple, efficient, robust and has low overhead. We argue that ESCC could be a candidate substitute for the costly and unnecessary universal caching.

6. ACKNOWLEDGMENTS

This work was supported by Five Top Priorities of "One-Three-Five" Strategic Planning, CNIC under Grant No. CNIC_PY-1401.

7. REFERENCES

[1] Jacobson V, Smetters D K, Thornton J D, et al. Networking Named Content. In *Proc. CoNEXT*, 2009.

[2] Seyed Kaveh Fayazbakhsh, Yin Lin, Amin Tootoonchian, et al. Less pain, most of the gain: incrementally deployable ICN. In *Proc. SIGCOMM*, 2013.

[3] Haiyong Xie, Yang Y R, Krishnamurthy A, et al. P4P: Provider Portal for Applications. In *Proc. SIGCOMM*, 2008.

[4] Cheng Yi, Alexander Afanasyev, Lan Wang, et al. Adaptive Forwarding in Named Data Networking. *ACM SIGCOMM Computer Communication Review*, 42(3), July, 2012.

[5] Yaogong Wang, Lee K, Venkataraman B, et al. Advertising Cached Contents in the Control Plane: Necessity and Feasibility. In *Proc. INFOCOM Workshop on computer communications*, 2014.

[6] D. J. Watts and S. H. Strogatz. Collective dynamics of smallworld networks. *Nature*, 393(6684), June, 1998.

[7] Perino D, Varvello M. A reality check for content centric networking. In *Proc. SIGCOMM workshop on ICN*, 2011.