A Step toward Practical Deployment of the Content-Centric Networking Architecture

Yaogong Wang
N.C. State University
ywang15@ncsu.edu

Kyunghan Lee
N.C. State University
klee8@ncsu.edu

Balakrishna Venkataraman
N.C. State University
bvenkat4@ncsu.edu

Ravi Lingappa Shamanna
N.C. State University
rlshaman@ncsu.edu

Injong Rhee
N.C. State University
rhee@ncsu.edu

ABSTRACT

Content-Centric Networking (CCN) is a promising candidate for the future Internet architecture. Its pivot is to change the host-centric model of the current Internet to a content-centric model. Under the new model, anyone can cache contents and serve them to anyone else. Thus, user latency, redundant traffic and server load are all reduced. Although CCN is widely acknowledged in the research community, practical implementation and deployment of the idea are still scarce. In this paper, we propose a practical instantiation of the CCN architecture. In contrast to the clean-slate approach, we leverage existing infrastructures to build a working system at minimum cost. We believe that our scheme reaps most of the benefits of CCN without an overhaul of the Internet and can serve as an initial step toward the deployment of a whole new CCN-based Internet architecture.

Categories and Subject Descriptors

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

General Terms

Design, Performance

1. INTRODUCTION

Despite its remarkable success, the current Internet is facing plenty of challenges in areas like efficiency, availability, security and mobility. To tackle such issues, a number of new Internet architectures have been proposed among which CCN [2] is a promising candidate. The pivot of CCN is to change the basic abstraction of the Internet from "end-to-end delivery of packets to named host" to "free dissemination of named contents". Under the new abstraction, a user no longer has to visit a specific server to fetch a content. It may come from anywhere as long as the content itself is integral.

Although the idea of CCN is widely acknowledged, practical implementation and deployment of the CCN architecture are still scarce. Questions remain to be answered include but are not limited to:

- How do CCN routers decide what to cache?
- How can we distribute the information on "who has what" in a scalable manner?
- How can CCN routers quickly make forwarding decisions?

The goal of this project is to instantiate the CCN idea into a practically deployable system by answering the questions above. Instead of a clean-slate approach, we leverage existing Internet infrastructures to minimize the cost incurred in building such a system. We believe that our work gives us an opportunity to explore the practical issues of CCN and can serve as an initial step toward the deployment of the CCN architecture.

2. PROPOSED ARCHITECTURE

We assume the existence of IP and DNS. We also assume that the content name includes the publisher's name. Figure 1 gives an overview of our scheme.

2.1 Cache Commitment and Replacement

Caching in CCN is different from traditional Web caching in that cache replacement cannot happen at any time. Instead, we need each cache to commit on how long it will cache a content and depending on the committed duration, we decide how far it should be advertised. This cache commitment alleviates the temporal scalability issue of CCN (i.e., update frequency). We will address the spacial scalability issue (i.e., routing table size) in Section 2.2.

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

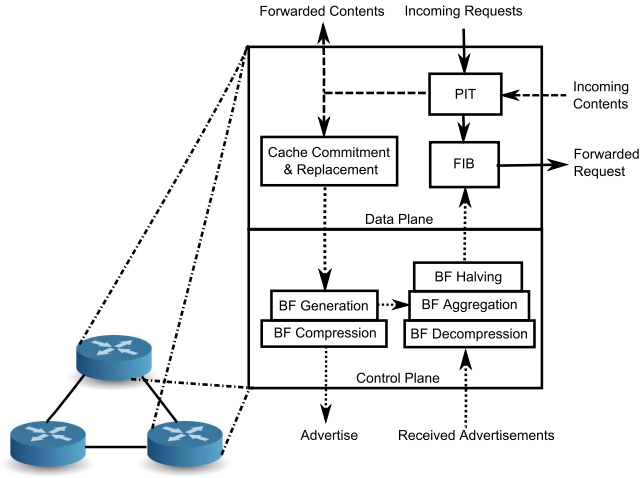

Figure 1: Architectural Overview

We design a hierarchical cache of three levels. The lowest is a transient cache where replacement may occur at any time. Contents in this cache are used locally and are not advertised. The next level is a durable cache where cache duration is at minute scale and will be advertised to routers within a few hops. The highest is a semi-permanent cache where cache duration is at hour scale and will be advertised all over the Internet. All contents start in the transient cache. Depending on their popularity, they are promoted or demoted. Popular contents will be promoted to higher levels and cached longer. They are also advertised farther and can be utilized by more routers.

2.2 Bloom Filter Generation and Aggregation

We need to inform other routers of the contents we have cached. Due to the huge number of contents in the Internet, we are unable to advertise a list of names of the cached contents. Instead, we generate Bloom filters (BF [1]) for them. The size of the Bloom filter is the most important design parameter in our scheme. Since this parameter is global to the whole Internet, we intend to set it to a huge value. If so, it may be too large to be transmitted or loaded into the RAM. We propose two methods to mitigate this issue. For transmission, we apply standard compression techniques as they are very good at processing large sparse arrays. For local look-up, we can shrink the size of a Bloom filter by taking the union of its first half and second half. This can be done recursively and look-up can still be carried out on the shrunk filter. In case the predefined size is still not large enough, we have two choices: 1) we can simply insert all the contents into the Bloom filter even if it leads to high false positive. This false positive will be taken into consideration when the routers make forwarding decisions (discussed in Section 2.3). 2) Alternatively, if the fill ratio (ratio of 1's) of a Bloom filter reaches certain threshold, we can simply stop insertion. This may lead to "false negative" since existent contents are not reflected by the Bloom filter but it does not affect the correctness of our scheme.

The Bloom filters are advertised in a way similar to link-state routing protocols as opposed to the distance-vector-like exchange in [3]. Upon receiving advertisements, a router aggregates Bloom filters by taking their union if they arrive at the same interface, share identical next-hop and are the same distance away. The generated Forwarding Information Base (FIB) is indexed by the (interface, next-hop, hop count) tuple and sorted by hop count in ascending order. After this aggregation, we have a table whose size is bound by the number of interfaces a router has, multiplied by the number of next-hop routers connected to each interface and further multiplied by the diameter of the Internet. We expect that number to be in the order of thousands.

2.3 Forwarding

When the FIB is ready, request forwarding is done by a linear search of the FIB (Linear search is acceptable since the FIB is relatively small). Since the entries are already sorted by hop count, the first match is the nearest cache who has the requested content and we forward the request to it. In case no match is found throughout FIB, we do a DNS resolution on the publisher's name and forward the request toward the original server via IP. We also adopt the Pending Interest Table (PIT) mechanism proposed in [2] so that content data simply follow the path of its request and are not routed.

Note that Bloom filters have false positives. When we find a match in FIB, we should check the false positive probability of the corresponding Bloom filter. Only if the product of the hop count and this probability is below a certain threshold will we terminate the search and forward the request. The intuition is that, if the cache is nearby, we can afford to take some risks since the penalty is small in case of false positive. If the cache is far away, we need to be more conservative.

3. CONCLUSIONS AND FUTURE WORK

Due to space limitation, many details are missing in this two-page abstract. We will deliver more details and some performance evaluation results during the poster session.

4. REFERENCES

[1] B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM*, 13(7):422–426, July 1970.

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

[3] M. Lee, K. Cho, K. Park, T. Kwon, and Y. Choi. Scan: Scalable content routing for content-aware networking. In *IEEE ICC*, June 2011.