Computer Networks 57 (2013) 2465–2482

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Performance analysis of in-network caching for content-centric networking

Yusung Kim, Ikjun Yeom *

Department of computer engineering, Sung Kyun Kwan University, Suwon, Republic of Korea

ARTICLE INFO

Article history:
Available online 8 April 2013

Keywords:
Content centric networking
In-network caching
Optimal content assignment
Information-centric networking

ABSTRACT

With the explosion of multimedia content, Internet bandwidth is wasted by repeated downloads of popular content. Recently, Content-Centric Networking (CCN), or the so-called Information-Centric Networking (ICN), has been proposed for efficient content delivery. In this paper, we investigate the performance of in-network caching for Named Data Networking (NDN), which is a promising CCN proposal. First, we examine the inefficiency of LRU (Least Recently Used) which is a basic cache replacement policy in NDN. Then we formulate the optimal content assignment for two in-network caching policies. One is *Single-Path Caching*, which allows a request to be served from routers only along the path between a requester and a content source. The other is *Network-Wide Caching*, which enables a request to be served from any router holding the requested content in a network. For both policies, we use a Mixed Integer Program to optimize the content assignment models by considering the link cost, cache size, and content popularity. We also consider the impact of link capacity and routing issues on the optimal content assignment. Our evaluation and analysis present the performance bounds of in-network caching on NDN in terms of the practical constraints, such as the link cost, link capacity, and cache size.

1. Introduction

The Internet was developed based on a host-to-host conversation model in the 1960s and 1970s. Although the Internet Protocol (IP) has been widely deployed for facilitating ubiquitous interconnectivity, the host-to-host conversation model is no longer effective for handling the explosion of multimedia content [1,2]. The majority of current Internet traffic consists of requests and dissemination of content provided by popular websites such as YouTube and Facebook. Internet bandwidth is wasted by repeated downloads of identical content. To minimize waste and to deliver content more efficiently without changing the current Internet architecture, several application layer solutions such as the Content Distribution Network (CDN) and peer-to-peer systems have been deployed.

Recently, novel network architectures have been proposed for efficient content delivery. In these architectures, the Internet needs a fundamental paradigm shift from a traditional host-to-host conversation model to a content-centric communication model. In this model, it is no longer necessary to connect to a server to obtain content. Instead, a user can directly send a request for content to the network with the content name without considering the original content location.

- In DONA [3], content is published on the network by content sources. There is a resolution infrastructure consisting of resolution handlers. A user sends a request with the flat name of the content to a resolution handler. Resolution handlers route the request by name in

* Corresponding author. Tel.: +82 31 299 4595.
E-mail addresses: yskim525@gmail.com (Y. Kim), ijyeom@gmail.com (I. Yeom).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.11.026

a hierarchical fashion and try to find the copy of the content that is closest to the user. When the request reaches the content source or a node holding a copy, the content is sent back through the reverse path.

- In PSIRP [4], content is published into Rendezvous by the content sources. A user subscribes to the content by a flat name, and the publication and subscription are matched by a rendezvous system. After the matching process, the rendezvous system creates a forwarding identifier and sends it to the content source. The content source starts to send the content with the forwarding identifier to the user. Each PSIRP node routes the content toward the user by using the forwarding identifier.
- In NetInf [5], a hybrid architecture is introduced to support name resolution as well as name-based routing to retrieve the content. Content sources publish their content through a Name Resolution Service (NRS) or by announcing the content in a routing protocol. A user can request content by a flat name from the NRS and the user will then receive a set of available sources. Alternatively, the user can directly send a request with the content name, and it will be forwarded to the source or a node holding a copy of the content.
- In CCN [6], each content has a hierarchical name, and the content is published by a routing protocol. Because of hierarchical naming, name aggregation and longest prefix matching are available in routing. A content request is forwarded toward a content source. If the request reaches a node that is a content source or is a node holding a copy of the content, data is routed back on the reverse path. All nodes on the path store the data, and they can answer subsequent requests for the same content.

In these architectures, storage for caching content is essential. All nodes have storage, and a content request can be satisfied by any node holding a copy of the content in its storage. With this in-network caching mechanism, we can avoid wasting network bandwidth due to the repeated delivery of popular content. From the users' perspective, in addition, we can expect a reduced response time for content by placing the content closer to users.

Among proposals of this type, the Content Centric Network (CCN) [6], now renamed Named Data Networking (NDN) [7], has received wide attention due to its simple and efficient content delivery mechanism. To maximize the benefits of NDN, it is necessary to develop an efficient cache management, scheme since cache size is likely to be very limited in order to contain all of the requested content. In the current literature on NDN [7], a very simple approach, such as Least Recently Used (LRU), is considered for replacement. In this approach, each router stores all of the content that is delivered, and replaces the content according to the LRU order. When the cache is sufficient to maintain a demanded cache hit ratio, this approach can be effective for reducing the bandwidth consumption. When the cache is insufficient, however, this approach frequently involves cache replacement on all routers along the data path. It produces multiple duplicated copies of content in a network, which results in a low cache hit ratio.

In this paper, we investigate the performance of in-network caching on NDN with a cache management scheme. First, we examine the inefficiency of LRU as a cache replacement policy. Then we formulate an optimal content assignment problem for NDN. To maximize the in-network cache hit ratio, it is sufficient to spread content to the network without duplicating. To minimize the average response time, however, a cache hit is insufficient. We have to consider the distance of popular content to be close to the users. Another consideration is routing protocol. If we allow a request to reach a router that is not along the path toward the content source, additional overhead is required in order to determine which router has a copy of the content, and to update the routing information on each router. Hence, we consider two different policies. One is for avoiding additional routing overhead. Content is supposed to be cached only along the path to the sources. We call this policy *Single-Path Caching*. The other policy, called *Network-Wide Caching*, is intended to improve the in-network cache hit ratio with additional routing overhead. Content can be cached at any place in a network. For both policies, we develop optimal content assignment models using a Mixed Integer Program (MIP) as a function of the popularity of the content and the internal and external link costs of a given network. Note that the internal link refers to a link within the network, and an external link refers to a link connected to outside networks, used for requesting content which is not within the network. Using these models, we evaluate and analyze the performance of NDN in terms of the total delivery cost for all requests, the in-network cache hit ratio, and the average response time. We also consider the impacts of the link capacity and routing issues on optimal content assignment. The complexity of MIP is high, and we do not claim here that the presented models are practical. The objective of this paper, rather, is to investigate the performance bounds of NDN with practical constraints, such as the internal and external link costs, cache size, and link capacity.

NDN with in-network caching is a new area, and performance analysis has not been investigated thoroughly. The primary objective of this paper is to provide the performance bounds of NDN. For that, we derive the MIP solution for the optimal content assignment to obtain the best performance with the limited cache size. Even though our MIP solution is theoretical, we believe that it can still be feasible. To implement our solution, we must have information regarding the popularity of all content items. Usually, the popularity of the content is considered to be changing, but recent studies show that the top most popular content is rather long-term [8,9]. Since the amount of long-term popular content is very small compared to the amount of existing content on the Internet, we think that it would be enough to use the long-term popular content in our optimal content assignment in an Autonomous System (AS) network. In Section 6, we present the implementation issues in more detail.

Through this paper, we make the following significant contributions: (a) We examine the limitations of the current approach for cache management in NDN, and address the need for efficient cache management schemes. To find the performance bound of in-network caching, we derive