Journal of Network and Computer Applications 65 (2016) 36–47

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Name prefix matching using bloom filter pre-searching for content centric network

Jungwon Lee, Miran Shim, Hyesook Lim *

Department of Electronics Engineering, Ewha Womans University, Seoul, Korea

ARTICLE INFO

Article history:
Received 15 August 2015
Received in revised form
9 January 2016
Accepted 16 February 2016
Available online 23 February 2016

Keywords:
Content centric network
Name prefix matching
Bloom filter
Name prefix trie

ABSTRACT

As a new networking paradigm for future Internet, content centric networking (CCN) technology provides a contents-oriented communication infrastructure for the rapidly increasing amount of data traffic. For the successful realization of CCN, it is essential to design an efficient forwarding engine that performs high-speed name lookup. This paper proposes the use of a hashing-based name prefix trie and a Bloom filter. In the proposed approach, an off-chip hash table storing the nodes of the name prefix trie is only accessed when the Bloom filter states that the node under querying exists in the trie. In accessing the node depending on the result of the Bloom filter, we propose two algorithms that have different strategies. The first algorithm accesses the trie node for every positive result of the Bloom filter, while the second algorithm first attempts to determine the longest matching length using Bloom filter queries. Trie nodes are accessed from the possible longest length, and tracked back if there is no match. Simulation results show that the proposed approach can provide the output face of each input name, with a single node access on average and with two node accesses in the worst-case using a reasonable size of a Bloom filter.

1. Introduction

Emerging Internet applications such as social network services largely share image, video, and music files. As large and repeatedly requested contents, this traffic is not efficiently transferred in the current Internet, which has a host-based infrastructure. Content centric network (CCN) is a promising next-generation network designed to solve such issues of the current Internet. The CCN is also known as the information centric network (ICN) or named data network (NDN). While the current Internet uses host-to-host communication based on IP addresses, the CCN performs data communication based on content names (Jacobson et al., 2009; Vasilakos et al., 2015; Bari et al., 2012; Xu et al., 2014; Qiao et al., 2015; Wang et al., 2012; Esteve et al., 2008; Perino and Varvello, 2011). In Jacobson et al. (2009), basic CCN features are implemented and the resilience and the performance of CCN architecture are compared with the host-based communication in terms of file transfer, content distribution, and voice calls.

Instead of the concept of source hosts or destination hosts, the CCN uses the concept of content generator and content consumer. Content generators produce contents, and content consumers receive and consume the contents. Routers perform routing using content names instead of IP addresses. Unlike conventional routers, CCN routers have an additional role of caching, which stores contents temporarily, and sends them to consumers requesting the contents. In this way, CCN consumers can rapidly acquire the desired contents, and the same contents are not repeatedly transferred over a network.

CCN uses two types of packets: *Interest* and *Data*. The *Interest* is broadcasted by a consumer. The *Data* is produced by a content generator and transmitted by any node hearing the *Interest* and having the *Data*. A CCN router has three different tables: Contents Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). The CS is a cache storing *Data* packets. The PIT is used to forward *Data* packets, and the FIB is used to forward *Interest* packets. For wire-speed packet forwarding, it is essential to have efficient lookup algorithms that perform the longest name matching for every incoming *Interest* packet.

A trie is an ordered tree-based data structure, the name of which originates from the word re*trie*val. We differentiate the term *trie* from the *tree* as follows. In a trie structure, all the descendants of a node have a common prefix of the string associated with that node, while this is not always true in a tree structure. A name prefix trie (NPT) has been proposed as an extended trie for a name lookup (Wang et al.,

* Corresponding author. Tel.: +82 2 3277 3403.
E-mail address: hlim@ewha.ac.kr (H. Lim).

http://dx.doi.org/10.1016/j.jnca.2016.02.008
1084-8045/

2011). The NPT provides an intuitive way for the name lookup with the longest name matching, but has an issue in search performance, since an FIB table can have many millions of content names and each content name can be excessively long.

The motivation of this paper is to propose a new approach for the longest name matching used in FIB lookups in CCN routers. The proposed approach is based on the name prefix trie. In order to solve the search performance issue of the NPT, we propose to add an on-chip Bloom filter which is queried before the access to the NPT, which is stored in an off-chip hash table. As a space-efficient probabilistic data structure used to test whether an element is a member of a set, Bloom filters have been popularly applied to network algorithms (Song et al., 2005; Tong et al., 2014; Mun and Lim, 2015; Lim et al., 2014a, 2014b). Since an access to an off-chip memory takes 10–20 times longer than an access to an on-chip memory (Panda et al., 2000), by pre-searching the on-chip Bloom filter, the off-chip hash table storing trie nodes is only accessed when there is a high possibility of a matching entry in our proposed approach. The earlier and shorter version of this paper was presented in (Lim et al., 2015).

The performance of our proposed algorithms is evaluated through simulation. Since the format of CCN names is not yet defined, URL names that have hierarchical characteristics similar to CCN names are used for our simulation (Wang et al., 2011). Memory requirements for creating a Bloom filter and storing the NPT are also evaluated. Using inputs with 3 times the number of stored URLs, the search performance is also evaluated.

This paper is organized as follows. Section 2 describes related works such as the name prefix trie, previous name lookup algorithms, and the Bloom filter theory. Section 3 introduces the building and searching procedures of the proposed algorithms. Section 4 shows the performance evaluation results, and Section 5 concludes the paper.

2. Related works

2.1. Name prefix trie

Each component in a URL is composed of variable-length character strings, and is differentiated by a dot or a slash. For example, the URL of http://www.youtube.com/user/PewDiePie is composed of 4 components: *youtube*, *com*, *user*, and *PewDiePie*. In storing the URL into a name

Table 1
An example of an FIB at CCN routers.

Content name	Output face
http://t-online.de	1
http://youtube.com	2
http://google.com	3
http://facebook.com	4
http://visitKorea.or.kr	5
http://warning.or.kr	6
http://google.com.hk	7
http://empowernetwork.com/25e8w	8
http://youtube.com/user/SkydoesMincraft	9
http://youtube.com/user/PewDiePie	10

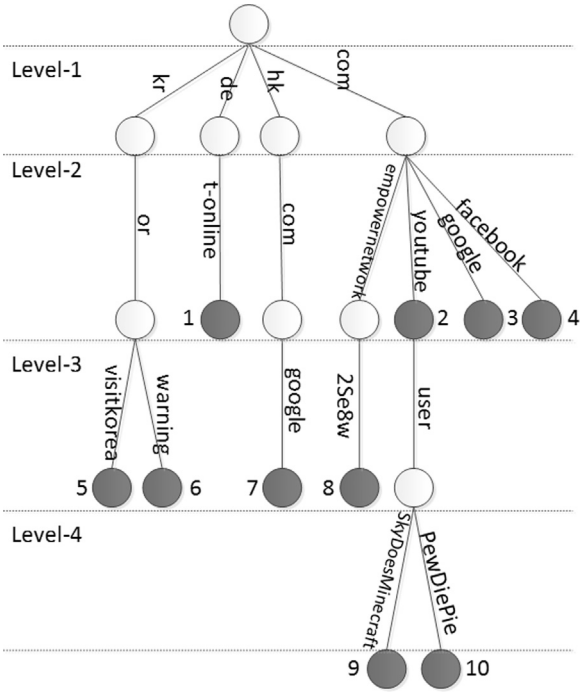

Fig. 1. Name prefix trie.