Computer Networks 91 (2015) 341–355

Contents lists available at ScienceDirect

Computer Networks

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

A bottleneck-free tree-based name resolution system for Information-Centric Networking

Wassef Louati [a,*], Walid Ben-Ameur [b], Djamal Zeghlache [b]

[a] ReDCAD, National School of Engineering of Sfax, University of Sfax, Tunisia
[b] Samovar, UMR 5157, CNRS Telecom SudParis, Evry, France

ARTICLE INFO

Article history:
Received 6 March 2015
Revised 27 July 2015
Accepted 27 August 2015
Available online 8 September 2015

Keywords:
Information-Centric Networking
Name resolution
Tree architecture
Multiple hash functions

ABSTRACT

In this paper we propose Ftree, a tree-based name resolution system in which name-records (name-to-locator mappings) are stored only at leaf level. Ftree addresses scalability and bottlenecks encountered at the root level in tree-based name resolution systems and answers some of the requirements for ICN (Information-Centric Networking). Given an object name, Ftree generates a set of keys using multiple hash functions. The name-record performs a key-based routing in a prefix tree and reaches a leaf node, becoming then its resolver. Using the same multiple hash functions, an object requester can select the key that corresponds to the nearest leaf resolver. A requester of a data object can have access to the nearest replica when resolvers hold them in addition to name-records. Moreover, and unlike existing name resolution systems, the root and high level nodes are rarely reached. An analytical study has been conducted to estimate the cost and the load induced by the lookup and registration processes. Through this study, the optimum replication factor has been computed to minimize the network traffic. Experimental evaluations of Ftree are reported, with a comparison with the MDHT, DONA and DNS-like resolution systems. Ftree outperforms these systems in terms of lookup overhead and root node loading.

1. Introduction

The Future Internet will be characterized by a massive data exchange, in which users and applications describe the content that they are looking for, not its location. The current Internet has a rigid host-centric scheme focusing on communications between hosts without paying enough attention to locate the best place containing the required information.

Existing solutions, such as Content Distribution Networks (CDNs) and peer-to-peer (P2P) networks accelerate data delivery but these overlay solutions are unaware of the underlying network topology. To ensure optimal performance, Information-Centric Networking (ICN) has emerged as a new concept for the architecture of the Future Internet, by putting the information at the center of the networking model.[1]

Existing ICN architectures rely on location-independent naming to support data movement, replication, and security. The huge number of named data objects requires a scalable name resolution system (NRS) to translate the object IDs into network addresses. Data are then accessed through these addresses.

An NRS architecture can be a tree, a Distributed Hash Table (DHT),[2] or hybrid. Tree-based NRS adapt to the underlying network topology (e.g. DONA [11], CURLING [12]). The

* Corresponding author. Tel.: +216 52175451.
E-mail address: wassef.louati@redcad.org, louatiw@gmail.com (W. Louati).

[1] There are several good papers discussing the Future Internet area (e.g. [1,2,3,4]) and treating ICN architectures (e.g. [5,6,7]). ICN is currently the subject of wide research in the community of network designers [8].

[2] DHTs, such as Chord [9] and Pastry [10], provide a distributed lookup service similar to a hash table. Each node in a DHT network has a unique identifier (nodeID) and is responsible for some (key, value) pairs which have

http://dx.doi.org/10.1016/j.comnet.2015.08.024
1389-1286/

tree node hierarchy follows generally inter-AS (Autonomous System) relationships to minimize hop stretch. However the root performing global name resolution can become a bottleneck. To alleviate this bottleneck, some proposals build a DHT-based system at the root level, such as CODI [13] and MDHT [14]. Other proposals delegate the root resolution task to an external Cloud platform (e.g. [15]). Instead of increasing the computation power and processing of the root, our proposal is to avoid as much as possible this root level by placing resolvers closer to their requesters.

In this paper, we propose a new tree-based name resolution system called Ftree. The proposed system enables fast name resolution by using multiple hash functions. For each object, Ftree distributes several name-to-locator mappings (name-records) in the Internet resolvers. An object requester can find the name-record in the nearest resolver. Ftree is a prefix tree that given an object name, generates a set of key using multiple hash functions. The name-record performs a key-based routing in the tree until reaching leaf nodes acting as resolvers. Using the same multiple hash functions, an object requester can select the key that corresponds to the nearest leaf resolver, reachable in few hops. Ftree is a bottleneck-free system since root nodes are rarely visited by name lookups. By adding object replicas to the name-records, Ftree brings replicas near to their requesters, an essential requirement for information-centric networking.

Scalability of the Ftree architecture has been evaluated using theoretical analysis. The analysis proved that the lookup cost decreases with an increasing number of hash functions. This avoids root node congestion as well. The load at each level has been also analyzed. The use of multiple hash functions *migrates* and *distributes* the load from the root node to internal nodes. An optimal sizing of the Ftree architecture in terms of the number of hash functions and the tree depth is also addressed in the paper.

The performance evaluation shows that Ftree outperforms two popular tree-based name resolution systems (DONA and DNS), in terms of lookup[3] load at the expense of a larger registration load. Nevertheless, the registration frequency is generally low comparatively to lookup requests, especially for popular objects. Compared to MDHT that MDHT performs a nearly constant and low lookup load in all hierarchy levels. Ftree can reach this low load by the use of several hash functions, and can outperform MDHT with more functions.

To address the large registration load induced by multiple hashing, both object's popularity and its mobility (if there is any) are considered. The registration and lookup processes become dynamic since an approximated optimal number of hash functions is used taking into account the current registration and lookups rates. We show that the number of generated name-records should be proportional to the object popularity. When the object is a mobile device, this number is reduced in order to minimize the traffic of locator updates. These results have been proved by analytical and experimental evaluations.

This paper is organized as follows. Section 2 reviews and discusses related work. A description of the Ftree system is presented in Section 3. Section 4 provides an analysis of Ftree. Section 5 describes the Ftree enhancement proposal (dynamic Ftree). An experimental study with further comparisons with other systems is proposed in Section 6.

2. Related work

There is currently a strong interest worldwide in enabling networking of information via tree-based name resolution systems, since they minimize hop stretch.[4] DNS [16] is the first tree-based name resolution system. It is clear that DNS rigid structure does not support large numbers of named objects. The Data Oriented Network Architecture (DONA) [11] defines a large namespace using flat and self-certifying names. DONA suffers from bottlenecks at root and high levels, since name lookups generally reach the root level to be resolved. Bloom filter-based flat name resolution system (B-NRS) [17] has a similar structure as DONA, and uses bloom filters to accelerate the forwarding process at tree nodes. However, B-NRS cannot prevent the arrival of massive lookups to the root node. DHT-NRS [18] designs a publish/subscribe NRS based on H-Pastry, a hierarchical version of the Pastry DHT [10]. Since its Rendezvous points are object ID-dependant, most subscriptions reach the root level.

To distribute the resolution task, some recent work have introduced the use of Distributed Hash Tables (DHTs). MDHT [14] solves the root bottleneck problem by replacing the nodes by rings of nodes (DHTs) at all levels. The registration messages of MDHT are propagated up to the root ring. CODI [13] integrates the tree and DHT structures with a two-tier model where DHT is defined at the root level. These ring-based approaches cannot adapt perfectly to the underlying network topology. Neighbors of nodes in the ring can be physically scattered and this would increase hop stretch. Tree-based architectures, such as our proposed one, can be easily mapped to the underlying network, accelerating thus the lookup process, as well as the forwarding process, when the same tree system is used for packets forwarding.

In the cited existing systems, the name-records go up the tree from the registering node to the root node/ring. Our proposed system registers the name-records only at the leaf level, so that the upper levels just forward the records. In addition, the root level in the cited approaches, is massively solicited by name lookups. Ftree avoids going through the root nodes since a request can generally be resolved by a resolver in the same subtree as the requester (see Fig. 1).

Ftree preserves the network hierarchy since the logical proximity coincides with the physical one. This reduces the lookup path length and avoids crossing administrative domain boundaries.

Beside their actual human-readable naming service, name resolution systems are acquiring new roles as they will be deployed by emerging ID/locator split architectures, such as LISP [19] and HIP [20]. In this context, name resolution

closest keys to its nodeID. Generally, nodes are virtually organized according to the order growing of nodes identifiers on a ring.

[3] In the rest of the paper, the terms 'lookup' and 'resolution' are used interchangeably (the lookup of name locators or replicas).

[4] In the context of name resolution systems, we define the hop stretch as the ratio of the number of hops of the lookup path to the corresponding shortest path in the tree.