Available online at www.sciencedirect.com

ScienceDirect

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

Survey

A survey of identifier–locator split addressing architectures

Miika Komu[a,b], *Mohit Sethi*[a,b,*], *Nicklas Beijar*[a]

[a] *NomadicLab, Ericsson Research, Finland*
[b] *Aalto University, Finland*

ARTICLE INFO

Article history:
Received 22 July 2013
Received in revised form
13 April 2015
Accepted 21 April 2015
Published online 18 May 2015

Keywords:
TCP/IP stack
Internet architecture
Namespace
Addressing models
Mobility
Multihoming
Renumbering
Internet transparency
identifier–locator split

ABSTRACT

The TCP/IP architecture of the Internet was originally designed around the contemporary restrictions of large computers that were difficult to move around. However, electronics followed Moore's law, resulting in cheaper and smaller electronics for consumers, and portable devices, such as laptops and cellular phones, became pervasive. Consequently, the original restriction on static hosts was no longer true even though is still present in the design of the TCP/IP networking stack. The TCP/IP stack remains still constrained by its original design, which was effectively a design compromise to make the addressing model simpler. As TCP connections are created based on the same addresses used by the underlying network layer, the connections break when the address changes or is removed. Thus, the TCP/IP architecture is challenged in the temporal dimension of addressing as it was designed to assume stable addresses. This is not only problematic from the viewpoint of initial connectivity but also critical in sustaining of active data flows. In this paper, we first outline the challenges related to the inflexible nature of the TCP/IP architecture resulting from the fact that the same namespace is shared between the transport and network layers. We then discuss existing solutions for these challenges that arise from the transient nature of addresses in the TCP/IP architecture. Finally, we perform a qualitative analysis of the solutions discussed in the paper.

Contents

* *Corresponding author at:* NomadicLab, Ericsson Research, Finland.
E-mail addresses: miika.komu@ericsson.com (M. Komu), mohit.sethi@aalto.fi (M. Sethi), nicklas.beijar@ericsson.com (N. Beijar).

http://dx.doi.org/10.1016/j.cosrev.2015.04.002

1. Introduction

The basis of the TCP/IP architecture and Sockets API is founded on the assumption of stable or persistent addresses because hosts were immobile in the original Internet. Paradoxically, addresses are nowadays non-persistent, especially due to the advancements in modern, mobile end-user equipment and dynamic network environments. Initially, IP addresses were supposed to only be used at the network layer, but then TCP just reused the addresses as its connection identifiers [1].

While the reuse of IP addresses at transport and network layers offers relief from address management issues, it is effectively a layer violation that results in undesired dependencies between the layers. An IP address is tied to the local network topology and effectively defines "where" the host is located, whereas a transport-layer identifier defines "who" the connection end-point is [2]. Consequently, the transport layer becomes dependent on the location of the end-host and its data flows are interrupted when the end-host changes its point of attachment to the network.

The problem is further aggravated by applications that should be using application-layer identifiers (defining "what"), such as FQDN-based identifiers, but instead reuse the underlying IP addresses.[1] The reasons are historical; the Sockets API, the de-facto low-level programming interface for network applications, was designed before DNS and is therefore heavily encumbered with the use of IP addresses [4]. To further aggravate the problem, applications have also few means of discovering when IP addresses are stale because the Sockets API does not attach any lifetime to the data structures associated with IP addresses [1].

As TCP/IP and the Sockets API are universally deployed and adopted, changing their fundamental nature is economically challenging. To fix the misalignment between applications excepting persistent addresses and networking stack offering ephemeral addresses, various "workarounds" to fix TCP/IP stack have emerged, with varying degrees of backward compatibility. However, many of the solutions tackle only a single problem emerging from non-persistent addressing, and it is not always guaranteed that such band-aid solutions interoperate with each other seamlessly and efficiently.

The rest of the paper is organized as follows: Section 2 gives an overview of the challenges associated with addressing in the current TCP/IP architecture. The addressing architectures that are proposed in response to these challenges are described in Section 3. Section 4 provides a qualitative analysis of the presented architectures. Lastly, Section 5 concludes the paper with a discussion.

[1] Due to the coupled role of addresses, Fully Qualified Domain Names (FQDNs) could be considered as the new "who", and Universal Resource Locators (URLs) as the new "where" [3] due to the pervasiveness of the web.

2. Challenges

In this section, we look at the challenges related to the transient nature of addresses in the TCP/IP architecture from