Computer Communications 34 (2011) 1685–1695

Contents lists available at ScienceDirect

Computer Communications

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

Implementation and evaluation of the Shim6 protocol in the Linux kernel

S. Barré [a,*], J. Ronan [b], O. Bonaventure [a]

[a] Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium
[b] Telecommunications Software & Systems Group, Waterford Institute of Technology, Cork Road, Waterford, Ireland

ARTICLE INFO

Article history:
Received 11 May 2010
Received in revised form 28 December 2010
Accepted 12 March 2011
Available online 21 March 2011

Keywords:
Shim6
IPv6
Multihoming

ABSTRACT

In the changing landscape of the todays Internet, several solutions are under investigation to allow efficient, flexible and scalable multihoming. One of the proposals is `shim6`, a host-based multihoming solution based on the use of multiple IPv6 addresses on each host. In this work, we first describe the main features of this protocol, then we explain our implementation of `shim6`, along with the associated security mechanisms in the Linux kernel and, finally, we evaluate its performance. In particular, we analyse the performance impact of the security mechanisms used by `shim6` and the impact of `shim6` on the performance of end-host systems, especially heavily loaded servers. We conclude by discussing the remaining open issues for a widespread deployment of host-based multihoming techniques such as `shim6`.

1. Introduction

The current IPv4 Internet is facing several challenges. Firstly, the IP version 4 (IPv4) address space is limited and the latest projections[1] indicate that during the year 2011 all IPv4 addresses will have been allocated. Secondly, operators and researchers are becoming more and more concerned about the limits on the scalability of the current Internet architecture [30].

For a number of years, several groups have tried to address these problems. Within the IETF, the work on the development of a replacement for IPv4 started more than 15 years ago with the work on IP next generation. This initiated the development of IP version 6 (IPv6) that was expected to replace IP version 4 before the beginning of this century. Today, IPv6 is now supported by most host and server operating systems. However deployment by network operators is still limited but appears to have been growing recently [24]. We can thus expect that IPv6 will gain more and more importance over the next few years.

On the other hand, the Internet Architecture Board (IAB) has identified several limitations of the current Internet architecture [30]. The first problem is the scalability of the interdomain routing system. This is reflected by the growth of the BGP routing tables and also the growth in the number of messages processed by BGP routers. This routing scalability issue is caused by several main factors. An initial contributor is multihoming, i.e. when an IP network is attached to several Internet Service Providers that need to advertise the corresponding prefix to the global Internet. Another contributor to the growth of the BGP routing tables are the various BGP-based traffic engineering techniques used by network operators to control the flow of their Internet traffic [20,40]. Finally, the allocation of IP addresses also contributes to the BGP growth. In the early days of the Internet, IP address blocks were allocated on a first-come first-served basis. This led to a huge consumption of address blocks that are almost impossible to aggregate. Since the introduction of Classless Interdomain Routing (CIDR), IP address blocks are allocated by Regional Registries (RIRs). There are two types of address block allocations: Provider Independent (PI) and Provider Aggregatable (PA). In the early days, PI address blocks were reserved for Internet Service Providers and customer networks could not obtain such address blocks directly from the RIRs. This allocation policy assumed that customer networks would be single homed and that they would renumber their network each time they change provider. These assumptions do not hold anymore and many enterprise networks insist on obtaining PI address blocks, which contributes to the growth of the BGP routing tables [29].

The second problem is the overloading of IP address semantics. IP addresses are used for two different purposes: identifiers and locators. In its identifier role, an IP address, combined with a port number, identifies an endpoint of a transport flow. In its locator role, an IP address identifies the paths to reach a host via one of its interfaces through a network.

The large IPv6 address space offers several opportunities to solve these problems differently than with IPv4. Several years ago, after evaluating many alternatives [17,23], the IETF chartered the `shim6` working group to develop a host-based IPv6 multihoming solution [36]. The `shim6` specifications are now ready and, in

* Corresponding author. Tel.: +32 1 0479103.
E-mail addresses: sebastien.barre@uclouvain.be (S. Barré), jronan@tssg.org (J. Ronan), olivier.bonaventure@uclouvain.be (O. Bonaventure).

[1] See e.g. http://www.potaroo.net/tools/ipv4/index.html.

0140-3664/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2011.03.005

this paper, we report our experience with one of the first complete and publicly available implementations of this IPv6 multihoming technique in the Linux kernel.

This paper is organised as follows. First, as shim6 is not yet widely known, we describe its key features and benefits in Section 2. This is followed by a description of the architecture of our LinShim6 implementation in Section 3 and an evaluation of the performance of several of its key mechanisms. We conclude by reflecting on the evolution of host-based multihoming techniques based on our experience with shim6 in Section 4 and a discussion of related work in Section 5.

2. Shim6 host-based IPv6 multihoming

Before delving into the details of shim6, consider that there are at least two scenarios that can provide multihoming. The first type is when a single host has two or more IPv6 addresses from two or more layer-2 interfaces connected to separate networks. This can be the case of a laptop having both WiFi and 3G Internet interfaces, or servers having multiple Ethernet interfaces. In these cases, the multihomed host would like to either be able to efficiently use both interfaces simultaneously or use a primary interface, with automatic redirection of all packets over another interface upon failure of the primary one.

The second type of multihoming occurs when a campus, corporate or ISP network is attached to two different service providers. In such a network, each host gets an address from each service provider, and is accessible over both. A host in such a multihomed network can select, for itself, the provider to use for a given flow, through appropriate selection of the source address. Shim6 was designed with the latter form of multihoming in mind but also supports the former.

Today, in the IPv4 Internet, when a network is multihomed, it receives one IPv4 address range, and uses BGP to advertise its IPv4 prefix to its upstream providers which, in turn, advertise the network to the global Internet. This contributes to the growth of the BGP routing tables. If a link between the multihomed network and one of its providers fails, BGP re-converges, to ensure that the multihomed network remains reachable via its other providers. However, a network relying on shim6 for its multihoming behaves differently. The main difference from IPv4 multihoming is that each shim6 host has several IPv6 addresses, one from each of its providers or one on each of its interfaces. This is illustrated in Fig. 1. The corporate network shown at the bottom of the figure is attached to ISP1 and ISP2. Each ISP has allocated a prefix to the corporate network. Each shim6 host has one IPv6 address inside each of these subnets. From a BGP routing table viewpoint, the main advantage of shim6 host-based multihoming is that AS1 and AS2 only need to advertise their global/32 IPv6 prefix and not the more specific prefixes allocated to their customers. However, this also implies that if the link between the corporate network and ISP1 fails, BGP will not announce the failure to the global Internet. This problem is solved in shim6 by using a new failure detection and recovery mechanism, the REAP protocol [4], that allows shim6 hosts to detect a failure and switch traffic to an available working path.

In the following subsections, first we describe the shim6 architecture, then explain how shim6 solves the security issues and finally describe the REAP protocol.

2.1. Shim6

A shim6 host has several IPv6 addresses. All these addresses are locators, i.e. they identify where a network interface is located within the global routing context. For example, in Fig. 1, a packet

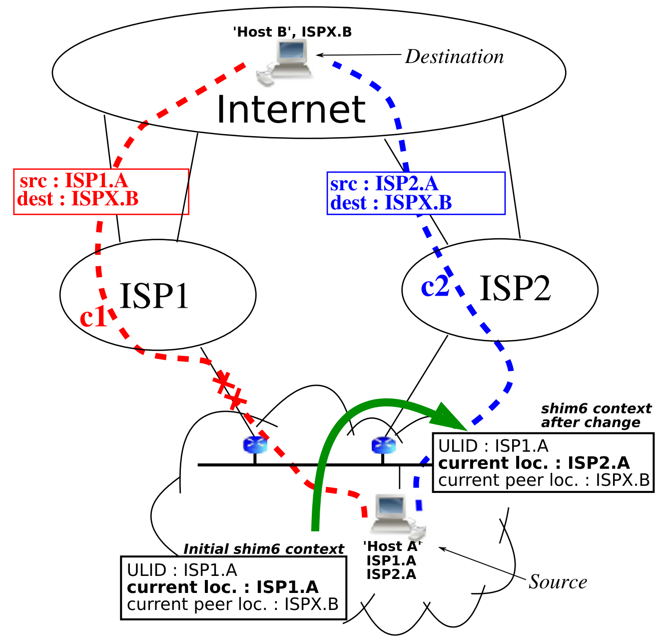

Fig. 1. Basic operation of a shim6 host.

whose destination is ISP1.A will be delivered via ISP1. On the other hand, a packet whose destination is ISP2.A will be delivered via ISP2. As current best practice [9] recommends that ISPs verify the source address of packets received from their customers: a packet produced by host A that contains ISP1.A as its source address must always be sent via ISP1. Such a packet will never be forwarded by ISP2.

When an application on host A contacts an application on host B using an upper-layer protocol (ULP), the default address selected [18] by host A is determined to be the upper-layer identifier (ULID) to identify the transport flows between the hosts. Conceptually, the shim6 sublayer belongs to the network layer and the locators are attached to the lower part of the network layer while the identifier is attached to the upper part of the network layer (Fig. 2).

The main purpose of shim6 is to preserve established flows in spite of network failures, while operating transparently to upper-layer protocols such as TCP or UDP. This is illustrated in Fig. 1. Host A has established a flow between ULID ISP1.A and destination ISPX.B. In addition to its ULID, host A also has the ISP2.A locator. Upon failure of the path between ISP1.A and ISPX.B, host A will use shim6 to switch its flow on the ISP2.A → ISPX.B path. For

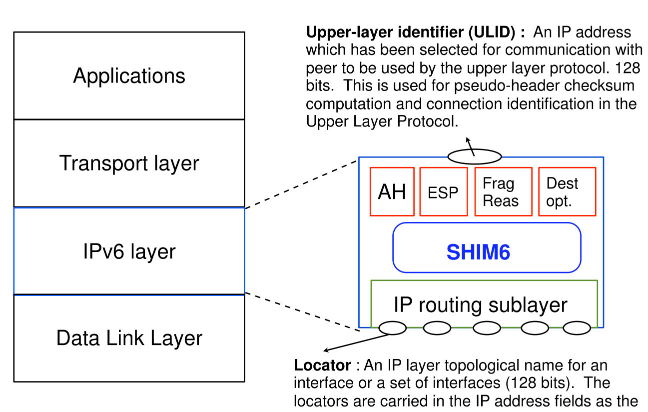

Fig. 2. Networking stack with shim6.