Relaying HTTP from IPv4 to IPv6
draft-wing-behave-http-46-relay-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 22, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

As IPv6-only hosts become commonplace, it is desirable to still have IPv4 clients access those IPv6-only hosts. Due to the size of the IPv4 and IPv6 address spaces, this has proven difficult using pure IP
address translation. Rather than trying to support all IP protocols, or even all TCP or UDP applications, this document proposes a mechanism for an IPv4 HTTP client to connect to an IPv6-only HTTP server.

Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Table of Contents

1. Introduction .................................................. 3
2. Operation ..................................................... 3
   2.1. Approach 1: HTTP Relay ................................. 3
   2.2. Approach 2: Redirection ............................... 4
       2.2.1. User Experience Considerations ................. 6
3. Security Considerations ......................................... 6
4. IANA Considerations ............................................ 6
5. Acknowledgements ............................................... 6
6. References ........................................................ 7
   6.1. Normative References ..................................... 7
   6.2. Informative References ..................................... 7
Author’s Address .................................................... 7
1. Introduction

One of the scenarios considered by the BEHAVE working group is "the IPv4 Internet to an IPv6 network" (Scenario 2 of [I-D.ietf-behave-v6v4-framework]). A difficulty with this scenario is that one IPv4 address is consumed for every IPv6 server, using a 1:1 mapping. This 1:1 mapping can be economized by temporarily mapping an IPv4 address to an IPv6 address [RFC2766][I-D.perkins-sourceipnat]. However, such economizing can cause some IPv6 hosts to be inaccessible when all of the available IPv4 addresses are temporarily in the process of completing their flow assignments, even though many different IPv6 destinations remain typically available at each IPv4 address via flow assignments that have already been completed.

To avoid this problem, the mechanism described in this paper solves the problem only for HTTP initiated from the IPv4 network towards the IPv6 network, and uses features of HTTP to provide access to the HTTP server.

The mechanism described in this paper requires at least one IPv4 TCP port be routed to the IPv6 host’s HTTP server. This is possible with stateful NAT64 translation (with a configured static mapping), stateless NAT64 translation, and port- mapped stateless NAT64 translation.

2. Operation

Two approaches are proposed. The approaches can be deployed at the same time on the same network, and appear different to the users of typical web browsers.

2.1. Approach 1: HTTP Relay

The simplest method is to perform an HTTP relay function, similar to a reverse HTTP proxy. The relay would listen for incoming HTTP requests, for many FQDNs, on the IPv4 network. Using HTTP’s "host:" header it would relay the HTTP request to the appropriate host on the IPv6 network.

The first few steps, where the in-home PC communicates with the HTTP relay, are out of scope of this paper. This might be done via a customer portal web page, for example.
Figure 1: HTTP Proxy, IPv4 client to IPv6 server

Notes:

- A drawback of this method is that all HTTP traffic goes through the HTTP relay. This approach is constrained by the HTTP relay’s bandwidth, packets per second, and maximum number of sessions.

- The HTTP relay can be operated by anyone on the Internet, including the ISP providing access service.

2.2. Approach 2: Redirection

To avoid the costs of TCP relaying, the service provider can also provide HTTP redirection. The HTTP redirector uses the HTTP "host:" header and its knowledge of the static IPv4/IPv6 mapping in the 6/4 NAT to respond with an HTTP redirect message containing a Location header pointing to the IPv4 address and port that maps to the requested IPv6 host. If the HTTP client indicated it supports HTTP/1.1, a 307 redirect SHOULD be used; for HTTP/1.0, a 302 redirect SHOULD be used.

Compared with the previous approach, only the initial HTTP transaction occurs with the HTTP redirector. Subsequent HTTP traffic is through the 6/4 NAT itself. Because the HTTP redirection can be to a specific port, approximately 64K IPv6 hosts can be served in this manner. However, the redirection changes the web browser’s...
address bar, which can make this approach unappealing; see also Section 2.2.1.

The first few steps, where the in-home PC communicates with the 6/4 NAT to acquire a permanent mapping and communicates this mapping to the HTTP redirector, are out of scope of this paper. Some 6/4 translation mechanisms would not require that step (e.g., port-mapped stateless translation). These steps might be done with a customer portal web page, for example.

---subscriber---

<table>
<thead>
<tr>
<th>in-home HTTP server</th>
<th>6/4</th>
<th>HTTP</th>
<th>DNS</th>
<th>unmodified</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC on port 80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ ---give me a port--&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ &lt;--IP=A, port=B----</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ --Webcam has IP=A, port=B--&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ &lt;--Ok, I will redirect there- ]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---Internet-->

<table>
<thead>
<tr>
<th>in-home</th>
<th>HTTP</th>
<th>DNS</th>
<th>unmodified</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE, Firefox, Safari, Opera</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: HTTP Redirection, IPv4 client to IPv6 server

Notes:

- The most optimal performance is obtained if the HTTP clients support HTTP/1.1 persistent connections and the in-home HTTP server uses relative URIs. In this way, the TCP connection from the HTTP client to HTTP server only hits the HTTP redirector once.

- The HTTP redirector can be operated by anyone on the Internet, including the ISP providing access service.
2.2.1. User Experience Considerations

A drawback of redirection is the URL displayed in the address bar changes. It can reduce the professional appearance of a website to have the URL displayed in the browser re-written to include a port number or a name associated with the ISP’s 6/4 translator, rather than a pretty domain name. It is possible to obscure this redirection from most web browsers using frames.

For example, the following frame redirects the user to a long URI without changing the URI seen on the address bar of most browsers:

```html
<html><head>
<frameset>
</frameset>
</head></html>
```

3. Security Considerations

Logging on the HTTP server works differently if the HTTP connection is relayed Section 2.1, because all connections appear to come from the ISP’s HTTP relay device. The IP address of the original client could be conveyed using the de facto standard X-Forwarded-For [XFF] header.

4. IANA Considerations

This document does not require any IANA action.

5. Acknowledgements

The mechanism described in this draft is based on existing "web redirection" services provided by a number of services on the Internet (e.g., [DynDNS]), extended to describe similar operation with IPv4/IPv6 translators.

Thanks to Patrik Faltstrom and Bob Van Zant for their review comments.

6. References
6.1. Normative References


6.2. Informative References


Author’s Address

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA  95134
USA

Email:  dwing@cisco.com