Internet Engineering Task Force	E. Vyncke
Internet-Draft	Cisco Systems
Intended status: Informational	September 17, 2008
Expires: March 21, 2009	

TOC

IPv6 Connectivity Check and Redirection by HTTP Servers
draft-vyncke-http-server-64aware-00

Status of this Memo

Abstract

Rather than forcing the client to decide whether IPv4 or IPv6 is more convenient to reach a web server; this document proposes to let the web server check whether there is IPv6 connectivity to the client; then the web server can do a HTTP redirect to the force the client to use IPv6. This is done easily by a script within the server HTML pages and does not require any change in the client applications or configuration. The client still can control whether he/she wants to enabled IPv6.
This draft could be discussed on the Applications Discuss mailing list, https://www.ietf.org/mailman/listinfo/apps-discuss.

Table of Contents

---

1. Introduction

TOC

It is often claimed that web servers are not dual-stack because the IPv6 has poor connectivity. Therefore, little to no web servers are dual-stacks; it is common to find the same content on www.example.com (for IPv4 access) and on ipv6.example.com (for IPv6 access).
The drawback of this setup is that once a user uses www.example.com, then all his/her communication will be over IPv4. This document proposes that the web server MAY run a dynamic and transparent check for the IPv6 connectivity between the client and the server and if there is IPv6 connectivity, then the client MAY be transparently redirected to the IPv6 server, i.e. ipv6.example.com.
The check and the redirect can easily be done by a script within the server HTML pages and MUST NOT require any change in the client applications or configuration. The client still can control whether he/she wants to enabled IPv6.

1.1. Requirements Language

TOC

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 RFC 2119 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) [RFC2119].

---

2. Verifying IPv6 Connectivity and Redirection

TOC

TOC

2.1. How can this be done?

The procedure can be described as:

1. the IPv4 web server has a start page which includes a small transparent image which is located on the ipv6.example.com web server. Note: this kind of image is often used on web sites to count access or as a spacer; it is usually a 1 by 1 pixel image.

2. the HTML SRC tag includes two events: onload() and onerror() which are commonly implemented in browsers.

3. if the client has only access to IPv4: the image will fail to load but the overall aspect of the web page will not be affected as the size of the error is only 1 by 1 pixel. The onerror() event is also triggered which could lead to change the web page content (see the code example (Example of HTML and Ecmascript Code)) or even replace the 1x1 image by another 1x1 image reachable over IPv4.

4. if the client has also access to IPv6 (i.e. it has the IPv6 protocol installed and has a valid IPv6 connectivity), then the image will load and the onload() event will be triggered. The script code associated with the onload() can force an immediate redirect of the client to the IPv6 web content.

All the above is only implemented in the HTTP server and does not require any change in the client even if the actual script is done by the client browser.
The redirect is also mostly transparent to the user.
The IPv6 connectivity works independently whether the client has configured IPv4 or IPv6 as his/her preferred protocol. For instance, it will work with Teredo (Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)," February 2006.) [RFC4380] tunnels even if those are usually configured as less preferable than IPv4.

2.1.1. Example of HTML and Ecmascript Code

TOC

Here is an example in HTML (W3C, "HTML 4.01 Specification," 1999.) [HTML] and in ECMAscript ECMA-262 (ECMA, "ECMAScript Language Specification," 1999.) [ECMA-262] (also known as JavaScript). It demonstrates two things which occur when IPv6 connectivity is detected (see the function IPv6Image()):

1. the appearance of the web page is changed (a text about IPv6 is displayed) and could now differ from the IPv4 only web page; this step is optional;

2. a immediate redirection to http://ipv6.example.com is executed; this is the core step.

Here is the HTML image tag which MUST include the onload() event and MAY include the onerror() event. It MAY be followed by a HTML span element which is used to display the result of the IPv6 connectivity check.

```
<img src="http://ipv6.example.com/1x1pixel.gif"
        width="1" height="1" border="none"
        onerror="NoIPv6Image();" onload="IPv6Image();">
<span id="CheckV6">Checking whether you have IPv6 access... </span>
```

The example below implements the mandatory onload() event handler, IPv6Image(), which redirects to the IPv6 version of the web site and the optional onerror() event handler, NoIPv6Image(), which is triggered when IPv6 connectivity does not exist (in this example, it is used to display a message).

```
<script language="javascript">

// Call back when IPv6 image can load
function IPv6Image() {
        document.getElementById('CheckV6').innerHTML='Good news: ' +
                'you have IPv6 connectivity, redirecting to IPv6' ;
        document.location='http://ipv6.example.com' ;
}

// Call back when IPv6 image cannot load
function NoIPv6Image() {
        document.getElementById('CheckV6').innerHTML='Bad news: ' +
                'you have no IPv6 connectivity.';
} ;

</script>
```

2.1.2. Proof of Concept

TOC

There is a very simple proof of concept of this technique. It is hosted on a web server at the University of Liège in Belgium and can be reached via the following URI:

> http://sigma.hec.be/~evyncke/family/ip.php: the normal dual-stack URI.
>
> http://193.190.125.15/~evyncke/family/ip.php: if you would like to check the technique as if your browser preferred IPv4.

> http://[2001:6a8:2c80:1::15]/~evyncke/family/ip.php: if you would like to check the technique as if your browser preferred IPv6.

2.2. Benefits of this Technique

TOC

The main benefit of this technique is that there is nothing to install or to configure at the client side. Nevertheless, the client has still the possibility to use only IPv4 by configuring his/her protocol stacks.
Another benefit is that there is basically little connectivity risk associated with this procedure, the client is redirect to the IPv6 version of the web server only if a HTTP transaction has been completed successfully over IPv6. This transaction is a normal HTTP transaction with full TCP handshake and HTTP protocol, so, it includes several IPv6 datagrams of varying size. Therefore, if it is successful then the IPv6 connectivity between the client and the server has been proven.
Note: if Path MTU Discovery is a concern, the 1x1 pixel image could changed to a larger one as long as it is kept transparent. As soon as the larger image exceeds 1,500 bytes, Path MTU discovery will need to be successful to display the image and to trigger the intrinsic event 'onload()'.

---

3. Extensions

TOC

This section briefly describes potential extensions of this technique.

3.1. Improving the User's Experience

TOC

By using another script associated to the onload() event, the user experience could be improved:

> The default page can dynamically be updated to reflect the availability of IPv6 connectivity;
>
> A pop-up window can be displayed asking the user whether he/she wants to be redirect to the IPv6 version of the web server;
>
> A HTTP (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.) [RFC2616] can be used by the web server to remember the user's decision.

3.2. Extension to only measure the amount of IPv6 capable client

TOC

Rather then taking the drastic decision of redirecting its client to its IPv6 content, the web server can simply log the result of the connectivity check:

*IPv4 client with no IPv6 connectivity;

*IPv4 client with IPv6 connectivity.

This can be achieved by loading the 1x1 image over IPv6 but the image source is no more a static image file but rather a server script (PHP, Perl, etc.) which is called with the IPv4 address of the client as an argument. Even if the server script execution (used to generate the image) is initiated over IPv6, it can retrieve the original IPv4 address from the HTTP query and log the association between the IPv4 and IPv6 addresses.
The PHP (The PHP Group, “PHP: Hypertext Preprocessor,” .) [PHP] code fragment below shows how it can be done. This fragment is run when the dual-stack client connects to www.example.com; it collects the client IPv4 address with the help of $_SERVER['REMOTE_ADDR'] and passes it as an argument to the 1x1pixel.php server script which is accessed over IPv6.

```
<img
 src="http://ipv6.example.com/1x1pixel.php?<=?$_SERVER['REMOTE_ADDR']?>"
 width="1" height="1" border="none"
 onerror="NoIPv6Image();" onload="IPv6Image();">
```

---

4. Acknowledgements

TOC

This I-D is based on some discussions with Chip Popoviciu.

---

5. IANA Considerations

TOC

This memo includes no request to IANA.

---

6. Security Considerations

TOC

No security issue has been identified.

Note: this technique requires to enable script execution on the client browser; this setting is sometimes deemed less secure than preventing the execution of any script by the browser.

7. Normative References

TOC

[ECMA-262]	ECMA, “ECMAScript Language Specification,” 1999.
[HTML]	W3C, “HTML 4.01 Specification,” 1999.
[PHP]	The PHP Group, “PHP: Hypertext Preprocessor.”
[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999 (TXT, PS, PDF, HTML, XML).
[RFC4380]	Huitema, C., “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs),” RFC 4380, February 2006 (TXT).

Author's Address

TOC

	Eric Vyncke
	Cisco Systems
	De Kleetlaan, 6A
	Diegem, B-1831
	Belgium
Phone:	+32 2 778 4677
Email:	evyncke@cisco.com

Full Copyright Statement

TOC

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.