Abstract

Current approaches can not solve the scenario that the users from IPv4 Internet to access IPv6-only content. When IPv6 content are becoming more and more popular, it is important to ensure that IPv6-only content can be reachable from legacy IPv4-only clients via some IPv4-only network. This document proposes two approaches for IPv4-only users to access IPv6-only content. It is designed to cover the Scenario 2 in [RFC6144].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 9, 2014.

Copyright Notice

Copyright (c) 2013 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents.
carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
2. Terminology .................................................. 3
3. The NAT46 translator for IPv4 Internet to access IPv6 network ................................................. 4
   3.1. IPv6-converted Address Selection ......................... 5
4. Approach 1: DNS46-based solution ................................. 6
5. Approach 2: Redirect-based Solution ............................... 7
6. The Dynamic IPv6-converted Address Issue ....................... 10
   6.1. DNS cache problem ........................................... 10
6.2. Mapping entry’s lifetime problem before session finished ................................................. 10
   6.3. Address Reusability Problem ................................. 11
7. IANA Considerations ............................................ 11
8. Acknowledgements .............................................. 11
9. References ..................................................... 12
   9.1. Normative References ...................................... 12
   9.2. Informative References ..................................... 12
Authors’ Addresses .................................................. 12
1. Introduction

In [RFC6144], Scenario 2 is an important use case. Not only could servers move directly to IPv6 without trudging through a difficult transition period, but they could do so without risk of losing connectivity with the IPv4-only Internet.

Existing solutions have not solved this scenario well. NAT-PT [RFC2766] can be used in this scenario, but it requires a tightly coupled DNS Application Level Gateway (ALG) in the translator, and have been deprecated by the IETF [RFC4966]. The stateless translation solution [RFC6219] can work too, but since each IPv6 server will consume one IPv4 public address, it is not suitable to deploy in situation that operators are running out of IPv4 address. [RFC6156] can be used for IPv4 client to communicate with IPv6 client. But this requires the IPv4 client and IPv6 client to implement a TURN client. Therefore, it is not suitable for C-S (Client-Server) and B-S (Browser-Server) mode.

[I-D.rfvlb-behave-v6-content-for-v4-clients] can work for IPv4-only user to access IPv6 content. But since it uses private IPv4 address to mapping the IPv6 server, it can only be used for IPv4 network to reach IPv6 network.

This document is designed for IPv4 Internet to reach IPv6 network. There are several requirements in this design:

1. Considering IPv4 address has been a scarce resource, the amount of public IPv4 addresses consumed by the translator should be less than that the number of IPv6 servers in the IPv6 network.

2. It should not require extra modifications on the server, e.g. by using a dynamic port number, implementing TURN client, etc.

In this document, we propose two approaches for this scenario. The first one requires modification on the DNS server, but introduces no application-layer functionality. The second one can use the current DNS infrastructure, but would require to implement a redirect server.

2. 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].

Terminology defined in [RFC6144] is used extensively in this document. Besides, this document uses the following terminologies:
IPv6-converted addresses: IPv4 addresses used to represent IPv6 nodes in an IPv4 Internet. They have an explicit mapping relationship to IPv6 addresses.

NAT46: a stateful IPv4/IPv6 translation functionality. It is consistent with IP/ICMP translation [RFC6145], and can also support IPv6-converted address selection and binding table maintenance.

3. The NAT46 translator for IPv4 Internet to access IPv6 network

The NAT46 solution is used for IPv4 clients in IPv4 Internet to reach IPv6 servers (depicted in Figure 1).

Mapping Table in NAT46

<table>
<thead>
<tr>
<th>No.</th>
<th>Client IPv4 Addr</th>
<th>IPv6-converted Server Addr</th>
<th>IPv4-converted Client Addr</th>
<th>Server IPv6 Addr</th>
<th>Lifetime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23.0.0.1</td>
<td>210.0.0.2</td>
<td>2001:c68::23.0.0.1</td>
<td>2001:c61::1</td>
<td>256</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Figure 1: Overall solution for IPv4 Internet to IPv6 network

In order to achieve the translation initiated from IPv4 side, two addresses need to be determined by NAT46 translator. The first one is the IPv6-converted address of the IPv6 server, which is selected from the IPv4 address pool configured in NAT46. The second one is the IPv4-converted address for the IPv4 client, which can be synthesized using the stateless approach defined in [RFC6052].

In our approach, since no extra modifications (e.g. changing the port number, etc.) can be taken to existing sever, only source address and destination address can be used as an index in the mapping table. Therefore, the mapping table includes client’s IPv4 address, server’s
IPv6-converted address, client’s IPv4-converted address, server’s IPv6 address and the lifetime (as depicted in Figure 1). For upstream traffic from the IPv4 client, the source address and destination address is the client’s IPv4 address and the server’s IPv6-converted address, while for downstream traffic of the IPv6 server, the source address and destination address is the Server’s IPv6 address and the client’s IPv4-converted address respectively.

3.1. IPv6-converted Address Selection

To achieve translation from IPv4 Internet to IPv6 network, the client’s IPv4 address and server’s IPv6 address has been known in advance. Therefore, the key point is to determine the IPv6-converted address for the IPv6 server.

Considering IPv4 address has been a scarce resource, the following rule is defined to select IPv6-converted address from the IPv4 address pool:

If NAT46 has already created a mapping entry for the IPv4 client, a new IPv6-converted address for which the client is not being used in the current mapping table MUST be selected from the IPv4 address pool. In other words, different servers can share one IPv6-converted address by different clients.

If there is still no mapping for a certain IPv4 client in the mapping table, a previous used IPv4 address can be selected for this client.

Example: if there have been two mapping entries in the mapping table.

<table>
<thead>
<tr>
<th>No.</th>
<th>Client IPv4 Addr</th>
<th>IPv6-converted Address</th>
<th>IPv4-converted Address</th>
<th>Server IPv6 Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23.0.0.1</td>
<td>210.0.0.1</td>
<td>2001:c68::23.0.0.1</td>
<td>2001:c61::1</td>
</tr>
<tr>
<td>2</td>
<td>23.0.0.2</td>
<td>210.0.0.2</td>
<td>2001:c68::23.0.0.2</td>
<td>2001:c61::2</td>
</tr>
</tbody>
</table>

Figure 2: Example of Existing Table

Then user3 arrives with the source address 23.0.0.3, the IPv6-converted address can reuse an existing IPv6-converted address in the mapping table, e.g. 210.0.0.2. However, if user2 arrives again with the same source address 23.0.0.2 to visit a new IPv6 server, a different IPv6-converted address other than 210.0.0.2 in the above example has to be selected from the address pool (e.g. 210.0.0.1 or Xie, et al. Expires January 9, 2014 [Page 5]
With this rule, the number of public IPv4 addresses consumed by the translator equals to the number of the destination addresses that one client is connecting at the same time (or within a certain period). Since the number of destination addresses consumed by one client in a short time is usually limited, this can greatly reduce the number of public IPv4 addresses that needs to be configured in the address pool.

Besides, since each mapping entry has a lifetime, one can further control the amount of consumed address by adjusting the mapping lifetime. See more detailed discussion on dynamic IPv6-converted address issue in section 7.

4. Approach 1: DNS46-based solution

This approach is designed for situation when DNS46 functionality is able to deploy in the network. It is independent of translated protocol. For applications without DNS process can not be solved by this approach. The overall solution is depicted in Figure 3.

Figure 3: DNS46-based approach

It consists of several functionalities:

1. NAT46: This functionality achieves the translation between IPv4 packet and IPv6 packet. It is consistent with [RFC6145]. Besides, it should also support IPv6-converted address selection (as defined in section 4.)

2. DNS46: A DNS translator that translates AAAA record to A record. It should also have an interface with NAT46 to retrieve IPv6-converted on demand.

The workflow of this approach is as follows:
1. An IPv4 client initiates a DNS query for A record (e.g. ipv6.example.com).

2. DNS46 receives the query. If there is no existing A record, it initiates an AAAA query for ipv6.example.com.

3. On receiving the IPv6 address for this domain name, it initiates a request to get IPv6-converted address from NAT46. The specific protocol for the request is out of scope.

4. NAT46 selects IPv6-converted address from its IPv4 address pool using IPv6-converted address selection mechanism (see section 3.1). It creates the mapping table to be used in subsequent IPv4 traffic translation.

5. NAT46 returns the IPv6-converted address to DNS46 and DNS46 in turn returns A record to the IPv4 client.

6. IPv4 client sends IPv4 traffic with the returned IPv6-converted address as the destination address. The traffic will be routed to NAT46, and NAT46 translates the IPv4 packet to IPv6 packet according to [RFC6145].

5. Approach 2: Redirect-based Solution

This approach is designed for situation when DNS server is not be
able to be upgraded to support DNS46 functionality. This approach can only be used for HTTP-based application. The overall solution is depicted in Figure 4.

![Figure 5: Redirect-based Solution](image)

It consists of several functionalities:

1. **NAT46**: This functionality is the same as the first approach.

2. **Redirect Server**: A Redirect server is used to redirect traffic to a different IPv6-converted address.

The workflow of this approach is as follows:
**Figure 6: Workflow of Proxy-lite Approach**

1. An IPv4 client initiates a DNS query for A record (e.g. ipv6.example.com).

2. In DNS server, the address of the redirect server is configured as the A record for ipv6.example.com and returns to the IPv4 client.

3. The IPv4 client sends HTTP GET request. The domain name (e.g. ipv6.example.com) is carried in HOST field.

4. The redirect server interprets the domain name, and sends the request to get IPv6-converted address to NAT46 (carrying the address of IPv4 client). The specific protocol for the request is now out of scope.

5. NAT46 selects IPv6-converted address from its IPv4 address pool using IPv6-converted address selection mechanism (see section 3.1). It creates the mapping table to be used in subsequent IPv4 traffic translation.

6. NAT46 returns the IPv6-converted address to redirect server and the redirect server in turn returns IPv6-converted address in HTTP
redirect packet with HTTP error "302 not found".

7. IPv4 client replaces the destination IPv4 address with the returned IPv6-converted address. The IPv4 traffic is routed to the NAT46 and the NAT46 translates the IPv4 packet to IPv6 packet according to [RFC6145].

6. The Dynamic IPv6-converted Address Issue

The major feature in the above two approaches is that the IPv6-converted address for destination IPv6 server is now dynamically determined by NAT46. In this way, IPv4 address sharing can been achieved, however, it will also bring some issues. In this section, some issues raised by the dynamic IPv6-converted address are analyzed and some guidelines for implementation are proposed.

6.1. DNS cache problem

In practice, DNS cache is widely used in local DNS server and end-host’s operating system. This might cause problem when the mapping entry has expired in NAT46 but the records still exist in DNS cache or end-host. For example, when a user has finished visiting a website (v61.example.com) and the lifetime of the mapping entry has expired. Then the same user re-opens the same website later. He might not initiate further DNS requests if the operating system still stores the previous A record which is no long effective anymore.

Possible solution:

This problem can be solved by carrying the DNS TTL value equals to the lifetime of the mapping entry in the NAT46. DNS46 SHOULD also return the TTL value to the user so that the operating system will not cache the DNS record when DNS TTL has expired.

6.2. Mapping entry’s lifetime problem before session finished

In NAT46, each mapping entry has a corresponding lifetime, which means this mapping will be deleted after the lifetime has expired. However, when there is no DNS requests for subsequent traffic, the mapping can not been created dynamically without knowing the destination IPv6 address for each session. Therefore, unfinished sessions can not be translated successfully.

Possible solution:

To solve this problem, the status of the mapping entry can be further introduced, including active and inactive. If there is traffic for
this mapping within a certain period, the mapping’s status should keep in active and the lifetime of the mapping entry should be prolonged before it expires. Only inactive mapping entries for which there is no traffic in a certain period can be deleted when the lifetime expires.

6.3. Address Reusability Problem

In NAT46, the IPv4 address can be released when the mapping has been deleted. But if a user has finished surfing a website (which means the mapping has been deleted in NAT46) and turns to a new website, but does not close the previous webpage, the released IPv4 address might be assigned to the new website in NAT46. When the user turns back to the previous webpage again after a long time, there might be no DNS requests as the application thought it has ready got the destination address before. But this time, this address will lead to a new site.

Possible solutions:

The problem can be alleviated if NAT46 will select a new IPv4 address rather than a newly released one for the same user. But it will still happen eventually when the address pool has been exhausted.

In reality, this problem is not severe since the user will not get the correct content with a wrong request. For example, in HTTP GET request, a wrong server will not response correctly if it does not have the content as the client is asking for. On receiving an error report, the user can would refresh the website manually and so the DNS process will be re-initiated. Actually, in current situation, a user will normally refresh a website that has not been visited for a long time.

7. IANA Considerations

No requirement on IANA.

8. Acknowledgements

The authors would like to thank the following individuals who have participated in the drafting, review, and discussion of this memo: TO BE COMPLETED

9. References
9.1. Normative References


9.2. Informative References
Authors’ Addresses

Chongfeng Xie
China Telecom
P.R.China

Phone: 86 10 58552116
Email: xiechf@ctbri.com.cn

Qiong Sun
China Telecom
P.R.China

Phone: 86 10 58552936
Email: sungiong@ctbri.com.cn

Qi He
China Telecom
P.R.China

Phone: 86 10 58552332
Email: heqi@ctbri.com.cn

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone:
Email: cathy.zhou@huawei.com