Virtual IPv6 Connectivity for IPv4-Only Networks
draft-vogt-durand-virtual-ip6-connectivity-00.txt

Abstract

This document proposes Virtual IPv6 Connectivity, a method for deployment in IPv4-only networks to enable communications with the IPv6 Internet. Virtual IPv6 Connectivity uses IP protocol translation to support both incoming traffic from IPv6 to IPv4, and outgoing traffic from IPv4 to IPv6. Communications initiated into the latter direction are facilitated through DNS or DNSSEC proxies. Different than existing IPv4/IPv6 interworking techniques, which are for deployment in IPv6 networks, Virtual IPv6 Connectivity does not require extra, globally unique IPv4 addresses.
# Table of Contents

1. Introduction ................. 3
2. Conceptual Overview ........... 4
3. Virtual IP Addresses .......... 6
4. IP Protocol Translation ........ 7
5. Discovery of Virtual IP Addresses 12
7. Security Considerations ......... 17
8. IANA Considerations ........... 17
9. Acknowledgment ................. 17
10. Normative References .......... 18
Authors’ Addresses ................ 18
Intellectual Property and Copyright Statements 19
1. Introduction

The introduction of IP version 6 on the Internet requires techniques for interworking between IP versions 6 and 4. Such techniques can be classified into two kinds:

- Dual stack: Host and network at one end of a communication support both IP versions.
- IP protocol translation: Packet are converted from one IP version into the other.

Of these, IP protocol translation has the advantage that it is easier to deploy and to operate because it enables hosts and networks to run only a single IP version. It can therefore be expected that, with a growing deployment base of IPv6, an increasing portion of Internet traffic will pass through IP protocol translation.

The responsibility for IPv4/IPv6 interworking will naturally be on the side of IPv6 networks as long as the deployment base of IPv6 is small compared to that of IPv4. The incentives for IPv4 networks to invest into interworking technology will then be low. There is hence an immediate need for IP protocol translators that can be deployed in IPv6 networks, and several proposals [###REFs] have been made. Over time, however, deployment of IP protocol translators in IPv4 networks will become increasingly relevant for two reasons:

- The inavertable exhaustion of IPv4 addresses will make it difficult for IPv6 networks to deploy IP protocol translators. This is because IPv4/IPv6 interworking techniques in general, if for deployment in IPv6 networks, necessarily require extra globally unique IPv4 addresses for external reachability from IPv4 networks.
- The expected growth in IPv6 deployment will make it relevant for IPv4 networks to ensure IPv4/IPv6 interworking without having to rely on IPv6 networks to deploy the necessary technology. Especially IPv4 networks that depend on universal reachability, such as Web sites or application service providers, will see themselves under increasing pressure to invest into IPv4/IPv6 interworking.

Address multiplexing mechanisms that reduce the requirement for globally unique IPv4 addresses -- as typically used in address translators [RFC3022] -- can mitigate, but not eliminate, the problem with IPv4 address exhaustion. IPv4 addresses will be difficult and expensive to acquire even in small quantities once the pool of unallocated IPv4 addresses is exhausted. Furthermore, address
multiplexing mechanisms are not universally applicable. Since they depend on port number rewriting to facilitate de-multiplexing, they cannot handle communications with no or encrypted port numbers. Consequently, there is a need for IP protocol translators that do without extra globally unique IPv4 addresses. Those will necessarily have to be deployed on the side of IPv4 networks.

Virtual IPv6 Connectivity, as proposed in this document, is a technique that uses IP protocol translators on the border of IPv4 networks to enable communications with the IPv6 Internet. Such an IPv4 network can be a provider network or a single customer network. Virtual IPv6 Connectivity supports the initiation of communications in both directions: incoming communications from IPv6 to IPv4, and outgoing communications from IPv4 to IPv6. It uses DNS or DNSSEC proxies to facilitate communications initiated in to the latter direction.

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].

2. Conceptual Overview

Virtual IPv6 Connectivity consists of three components, which together enable hosts in an IPv4-only network to communicate with remote IPv6 correspondent hosts:

- IP protocol translators, which are placed on border links of the IPv4-only network to translate packets exchanged with IPv6 correspondent hosts.

- Virtual IPv4 and IPv6 addresses, which represent, respectively, the IPv6 addresses of remote correspondent hosts to local IPv4 hosts, and the IPv4 addresses of local hosts to remote IPv6 correspondent hosts.

- Provisioning of virtual IPv4 and IPv6 addresses in DNS.

Figure 1 depicts an IPv4-only network that uses Virtual IPv6 connectivity. The network has an IP protocol translator, denoted "xlate" in the figure, on its border link to the IPv6 Internet. The network also has sets of virtual IPv6 and IPv4 addresses, which are taken from the IPv6 address prefix 2001:DB8::/96 and from the IPv4 address prefix 10.0.0.0/8, respectively. The virtual IPv6 addresses represent the regular IPv4 addresses of the IPv4-only network to remote IPv6 correspondent hosts. The virtual IPv4 addresses represent the regular IPv6 addresses of remote correspondent hosts to
hosts in the IPv4-only network.

```
+----------------+                  +----------------+                  +----------------+
+---------------+                  +---------------+                  +---------------+
|                |                  |                |                  |                |
|  virtual IPv6  |                  |  virtual IPv4  |                  |  virtual IPv6  |
|  addresses    |                  |  addresses    |                  |  addresses    |
|  2001:DB8::/96|                  |  10.0.0.0/8   |                  |  2001:DB8::/96|
|                |                  |                |                  |                |
|                |                  |                |                  |                |
|  v4-only network  | xlate | v6 Internet |  v4-only network  | xlate | v6 Internet |  v4-only network  | xlate | v6 Internet |
|                |                  |                |                  |                |                  |                |                  |                |
|                |                  |                |                  |                |                  |                |
|                |                  |                |                  |                |                  |                |
```

Figure 1: Conceptual overview of Virtual IPv6 Connectivity

The pair of a virtual IP address and the represented regular IP address is called a "mapping". Mappings are used in recursive DNS name servers, authoritative DNS name servers, and IP protocol translators operated by an IPv4-only network with Virtual IPv6 Connectivity. They enable IPv4 hosts and IPv6 correspondent hosts to discover IP addresses for each other and communicate. When a local IPv4 host resolves the hostname of a remote IPv6 correspondent host, a recursive DNS name server maps the remote correspondent host’s IPv6 address onto a virtual IPv4 address. When a remote IPv6 correspondent host resolves the hostname of the local IPv4 host, an authoritative DNS name server maps the local host’s IPv4 address onto a virtual IPv6 address. Packets exchanged between an IPv4 host and an IPv6 correspondent host are therefore always destined to a virtual IP address when initially dispatched by the sender. Virtual IP addresses route to an IP protocol translator, which performs three tasks with a received packet that is destined to a virtual IP address:

- It maps the IP source and destination addresses in a packet, replacing the virtual IP destination address with the represented regular IP address, and replacing the regular IP source address with the representing virtual IP address.

- It uses the Stateless IP/ICMP Translation Algorithm [##SIIT] to translate the IP header of the packet into an IP header from the respective other IP version. The new IP header uses the IP source and destination addresses determined in the foregoing mapping.
Virtual IPv6 addresses embed the regular IPv4 addresses they map to. Mappings between virtual IPv6 addresses and regular IPv4 addresses are therefore constant and do not require state in DNS name servers or IP protocol translators. Mappings between virtual IPv4 addresses and regular IPv6 addresses are necessarily dynamic and on-demand given that any set of virtual IPv4 addresses is smaller than the global set of regular IPv6 addresses, for which mappings are potentially needed. Mappings between virtual IPv4 addresses and regular IPv6 addresses therefore require state in IP protocol translators and recursive DNS name servers to keep track of the regular IPv6 addresses that map onto a given virtual IPv4 address.

3. Virtual IP Addresses

For the local hosts in an IPv4-only network to communicate with remote IPv6 correspondent hosts, local hosts and remote correspondent hosts must be able to reach their respective peer with an IP version they support: The local hosts must be able to reach the remote correspondent hosts via an IPv4 address, and the remote correspondent hosts must be able to reach the local hosts via an IPv6 address. Virtual IPv6 Connectivity achieves this reachability across IP versions by means of virtual IPv4 and IPv6 addresses, which respectively map onto the regular IPv6 addresses of remote correspondent hosts and the regular IPv4 addresses of local hosts.

Virtual IPv6 addresses represent local hosts in an IPv4-only network to remote IPv6 correspondent hosts. They are ordinary, global IPv6 addresses, which the IPv4-only network obtains either from its Internet services provider or from an Internet registry. Virtual IPv6 addresses are announced externally in the global routing system, but are not used internally within the IPv4-only network. They map one-to-one onto the regular IPv4 addresses in the IPv4-only network, i.e., there is one unique virtual IPv6 address for each regular IPv4 address. There are no restrictions as to how to realize this one-to-one mapping. A simple, recommended way is to reserve a slash-96 IPv6 address prefix, and to map each regular IPv4 address to the virtual IPv6 address that is defined by attaching the 32 bits of the regular IPv4 address to the 96 bits of the reserved slash-96 IPv6 address prefix.

Virtual IPv4 addresses represent remote IPv6 correspondent hosts to local hosts in an IPv4-only network. They are ordinary IPv4 addresses that map onto the regular IPv6 addresses of the remote correspondent hosts. Virtual IPv4 addresses are announced internally
in the local routing system of an IPv4-only network, but are not used outside the IPv4-only network. Since virtual IPv4 addresses are used only locally within an IPv4-only network, they can be non-global. This makes them re-usable in different IPv4-only networks with Virtual IPv6 Connectivity. As an example, virtual IPv4 addresses could be taken from the private net-10 or 192.168.0.0/16 address spaces [RFC1918], or from a new IPv4 address block specifically allocated for use in IPv4-only networks with Virtual IPv6 Connectivity.

Different than mappings between regular IPv4 addresses and virtual IPv6 addresses, mappings between regular IPv6 addresses and virtual IPv4 addresses cannot be one-to-one because the set of regular IPv6 addresses that potentially need to be mapped is larger than any pool of virtual IPv4 addresses. Mappings between regular IPv6 addresses and virtual IPv4 addresses may instead change dynamically depending on the set of remote IPv6 correspondent hosts that communicate with an IPv4-only network. Mappings between regular IPv6 addresses and virtual IPv4 addresses are therefore created in an on-demand manner, and they have a lifetime after which they may be replaced.

In the example of Figure 1, the IPv4-only network has obtained the slash-96 IPv6 address prefix 2001:DB8::/96 for virtual IPv6 addresses, so any IPv4 address a.b.c.d from the IPv4-only network can be mapped one-to-one to a virtual IPv6 address 2001:DB8::a.b.c.d. The IPv4-only network furthermore uses the private net-10 address space, 10.0.0.0/8, for virtual IPv4 addresses. So for any remote IPv6 correspondent host that communicates with a local host in the IPv4-only network, a virtual IPv4 address is allocated from the slash-8 IPv4 address prefix 10.0.0.0/8, and is temporarily mapped to the remote correspondent host’s regular IPv6 address.

4. IP Protocol Translation

In addition to the mapping between regular IP addresses from one IP version and virtual IP addresses from the respective other IP version, packets exchanged between the local hosts in an IPv4-only network and remote IPv6 correspondent hosts must be converted from IPv4 to IPv6 and vice versa on the border between the IPv4-only network and the IPv6 Internet. For this purpose, IPv4-only networks with Virtual IPv6 Connectivity deploy IP protocol translators on each border link that shall be enabled for communications with remote IPv6 correspondent host. An IP protocol translator is a router with the extra capabilities to map between regular and virtual IP addresses, and to convert packets between IPv4 and IPv6 using the appropriate mapping. Figure 1 illustrates an IPv4-only network with one border link, which is endowed with an IP protocol translator to achieve
Virtual IPv6 connectivity for the IPv4-only network.

Since in general not all packets traversing a border link need to be translated between IPv4 and IPv6, IP protocol translators must have a means to distinguish packets that require translation from packets that can be forwarded without modification. IP protocol translators perform this differentiation based on the packets’ IP destination address: Packets destined to a regular IP address do not require translation, whereas packets destined to a virtual IP address do. This rule is independent of whether packets originate within and leave the IPv4-only network, or whether the packets originate outside and enter the IPv4-only network. For packets that are destined to a regular IP address, an IP protocol translator behaves as an ordinary router. For packets that are destined to a virtual IP address, the IP protocol translator performs three tasks:

- It maps the IP source and destination addresses in a packet, replacing the virtual IP destination address with the represented regular IP address, and replacing the regular IP source address with the representing virtual IP address.

- It uses the Stateless IP/ICMP Translation Algorithm [SIIT] to translate the IP header of the packet into an IP header from the respective other IP version. The new IP header uses the IP source and destination addresses determined in the foregoing mapping.

- It forwards the packet towards the new (regular) IP destination address.

### DESCRIBE WHEN TRANSLATOR CREATES MAPPING? ###

To ensure that packets pass through an IP protocol translator if exchanged between a host in an IPv4-only network and a remote IPv6 correspondent host, the routes towards virtual IPv4 and IPv6 addresses are announced by IP protocol translators themselves: An IP protocol translator announces a route towards virtual IPv6 addresses on its interface towards the IPv6 Internet, and a route towards virtual IPv4 addresses on its interface towards the IPv4-only network. Thus, when a remote IPv6 correspondent host sends a packet to the virtual IPv6 address of a local host in the IPv4-only network, the packet routes to an IP protocol translator, which translates the packet’s virtual IPv6 destination address into the mapped regular IPv4 destination address, and the packet’s regular IPv6 source address into a mapped virtual IPv4 source address. Vice versa, when the local host in the IPv4-only network sends a packet to a remote IPv6 correspondent host, an IP protocol translator translates the packet’s regular IPv4 source address into the mapped virtual IPv6 source address, and the packet’s virtual IPv4 destination address.
into the mapped regular IPv6 destination address. Both local host
and remote correspondent host therefore use only their respective IP
version; the peer’s regular IP address from the other IP version is
transparent to them. Figure 2 illustrates the route announcement for
virtual IP addresses for the IPv4-only network from Figure 1.

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( v4-only network )</td>
<td>============== xlate ==============</td>
</tr>
<tr>
<td>(</td>
<td>&lt;-------- # # #</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

announces route to virtual
IPv6 address prefix 2001:DB8::/96
```

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td># # # #</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

announces route to virtual
IPv4 address prefix 10.0.0.0/8
```

Figure 2: Route announcement for virtual IP addresses

Since the explicit routes towards virtual IP addresses ensure that
packets are routed via an IP protocol translator if they require
translation between IPv4 and IPv6, IPv4-only networks with Virtual
IPv6 Connectivity may deploy IP protocol translators on only some,
but not all of their border links. This can reduce the expenses for
deployment and operation of Virtual IPv6 Connectivity.

Since the communication between a host in an IPv4-only network and a
remote IPv6 correspondent host would break if the mapping between
either host’s regular and virtual IP addresses changed throughout the
course of the communication, IP protocol translators must ensure that
mappings are stable for the duration of the communications in which
they are used. This is trivial for the mappings between regular IPv4
addresses and virtual IPv6 addresses because these mappings are one-
to-one and therefore constant. However, as one-to-one mappings are
not possible between the large set of regular IPv6 addresses and the
necessarily smaller set of virtual IPv4 addresses, IP protocol
translators need state to ensure that these mappings remain constant
during the communications in which they are used. An IP protocol
translator establishes such state in either of two ways, depending on
which way a communication is established:
For communications that are initiated by a local host in an IPv4-only network, mapping state is created at the time the local host retrieves the remote IPv6 correspondent host’s virtual IPv4 address via the DNS. This ensures that the IP protocol translator can map the virtual IPv4 address in the first packet that the local host sends to the remote correspondent host. Mapping state creation is in this case initiated by a recursive DNS name server from the IPv4-only network. This will be explained further in Section 5.

For communications that are initiated by a remote IPv6 correspondent host, the creation of mapping state is postponed to the time the IP protocol translator receives the first packet that the remote correspondent host sends to the local host in the IPv4-only network. It is then the IP protocol translator which initiates mapping state creation. Postponing the creation of mapping state is possible in this case because the remote correspondent host does not use the virtual IPv4 address for which mapping state is created. The postponement then has the advantage that both the creation of mapping state as well as the allocation of a virtual IPv4 address happens only when there is actually a communication using it.

The involvement of both IP protocol translators and recursive DNS name servers in the creation of mapping state and in the allocation of virtual IPv4 addresses calls for coordination between IP protocol translators and recursive DNS name servers: IP protocol translators must have relevant mapping state even if the creation of the state was initiated by a recursive DNS name server; and simultaneous allocation of a given virtual IPv4 address by both an IP protocol translator and a recursive DNS name server must be avoided. How such coordination is achieved is outside the scope of this document. It may be proprietary because the IP protocol translators and recursive DNS name servers are under the same or under related administrations. However, it is recommended that the maintenance of virtual IPv4 addresses is with the IP protocol translator to which they route. This accelerates packet forwarding when the creation of mapping state is initiated by an IP protocol translator, and it hence accounts for the commonly stricter delay constraints on packet forwarding in routers compared to those on DNS lookups.
Figure 3 illustrates the mapping state creation and the translation of packets between IPv4 and IPv6 for a communication that is initiated by a remote IPv6 correspondent host. The packets are exchanged between a local host in an IPv4-only network on the left, and the remote IPv6 correspondent host on the right. The IPv4-only network uses Virtual IPv6 Connectivity, so the packets traverse an IP protocol translator. Continuing from previous examples, virtual IPv6 addresses are taken from the IPv6 address prefix 2001::DB8:ABC::/96, and virtual IPv4 addresses are taken from the private net-10 address space, 10.0.0.0/8. The local host’s regular IPv4 address, a.b.c.d, maps to the virtual IPv6 address 2001:DB8:ABC::a.b.c.d. This mapping is constant. The remote correspondent host’s regular IPv6 address, 2001:DB8:BCD::X:Y:Z, maps to the virtual IPv6 address 10.0.0.0/8. This mapping is temporary and created in an on-demand manner. The dynamic state for the mapping between the regular IPv6 address 2001:DB8:BCD::X:Y:Z and the virtual IPv4 address 10.0.0.0/8 is created when the first packet in the communication reaches the IP protocol translator. Once the state is created, subsequent packets from the same communication can be translated from IPv6 to IPv4 and vice versa, as shown in the
5. Discovery of Virtual IP Addresses

Virtual IPv6 Connectivity requires that the initiator of a new communication obtains a virtual IP address for its peer: An IPv6 correspondent host intending to reach an IPv4 host must obtain a virtual IPv6 address for the IPv4 host; an IPv4 host intending to reach an IPv6 correspondent host must obtain a virtual IPv4 address for the IPv6 correspondent host. Since the prevailing method for destination address discovery is the use of the DNS, Virtual IPv6 Connectivity makes virtual IPv4 and IPv6 addresses available via the DNS. This section explains the configuration and operational extensions for recursive and authoritative DNS name servers in an IPv4-only network that are necessary for this.

5.1. Discovery of Virtual IPv6 Addresses

For an IPv6 correspondent host to obtain via DNS the virtual IPv6 addresses of an IPv4 host in an IPv4-only network with Virtual IPv6 Connectivity, the authoritative DNS name servers of the IPv4-only network must be configured with AAAA records for the IPv4 host’s virtual IPv6 addresses. These AAAA records can be statically provisioned in the authoritative DNS name servers because the mapping between IPv4 addresses and virtual IPv6 addresses does not change. The retrieval of virtual IPv6 addresses via DNS hence does not differ from the retrieval of regular IP addresses.

IPv6 correspondent host

|---------------------------------|----------------------------------------------------------|

Figure 4: Discovery of virtual IPv6 addresses via DNS

Figure 4 shows the resolution of an IPv4 host’s name into a virtual IPv6 address. In this example, the IPv4 host’s name is "host.v4.net". This resolves into the regular IPv4 address a.b.c.d, which in turn maps to the virtual IPv6 address 2001:DB8:ABC::a.b.c.d.
The message exchange in the figure begins where the IPv6 correspondent host queries an authoritative DNS name server from the IPv4-only network for a resource record of type AAAA associated with host name "host.v4.net". The authoritative DNS name server responds with a DNS reply that contains a type-AAAA resource record with the virtual IPv6 address 2001:DB8:ABC::a.b.c.d.

5.2. Discovery of Virtual IPv4 Addresses

IPv4 hosts in an IPv4-only network with Virtual IPv6 Connectivity must be able to obtain a virtual IPv4 address when resolving the host name of a remote IPv6 correspondent host via DNS. Virtual IPv4 addresses are provisioned by a recursive DNS name server in the IPv4-only network when a local IPv4 host queries for an IPv4 address associated with an IPv6 correspondent host’s name. The recursive DNS name server then retrieves resource records of type AAAA associated with the correspondent host’s name, maps the regular IPv6 addresses from those resource records onto virtual IPv4 addresses, and returns the virtual IPv4 addresses in resource records of type A to the requesting IPv4 host.

Different than virtual IPv6 addresses, virtual IPv4 addresses cannot be provisioned directly by the DNS name server that is authoritative for the IPv6 correspondent host. Since virtual IPv4 addresses are mapped to regular IPv6 addresses in dynamic manner, a DNS name server handing out virtual IPv4 addresses must necessarily coordinate with the entity in the IPv4-only network that maintains the virtual IPv4 addresses. Such coordination is infeasible for a correspondent host’s authoritative DNS name server, since this is in general under a different administration than the IPv4-only network.

The operation of a recursive DNS name server in an IPv4-only network with Virtual IPv6 Connectivity more specifically proceeds as follows:

- Upon receiving a DNS query from a local IPv4 host for resource records of type A associated with a correspondent host’s name, the recursive DNS name server first attempts to retrieve regular IPv4 addresses of the correspondent host. It thus seeks to directly obtain the requested resource records of type A. This is the standard operation of a recursive DNS name server.

- If the recursive DNS name server receives a DNS reply with a resource record of type A, it returns the DNS reply unchanged to the resolving IPv4 host. The correspondent host in this case has a regular IPv4 address, and it is not necessary to allocate a virtual IPv4 address for the correspondent host.
o If the recursive DNS name server fails to receive a DNS reply with a resource record of type A, the correspondent host cannot be reached at a regular IPv4 address. The recursive DNS name server in this case attempts to retrieve resource records of type AAAA associated with the correspondent host’s name.

o If the recursive DNS name server receives a DNS reply with a resource record of type AAAA, the correspondent host can be reached at one or more regular IPv6 addresses. These need to be presented to the resolving IPv4 host in form of virtual IPv4 addresses. The recursive DNS name server maps each of the correspondent host’s regular IPv6 addresses onto a separate virtual IPv4 address in coordination with the IP protocol translator to which these virtual IPv4 addresses route. The recursive DNS name server then synthesizes a DNS reply with one resource record of type A for each virtual IPv4 address, and returns this DNS reply to the resolving IPv4 host.

o If the recursive DNS name server fails to receive a DNS reply with resource records of either type A or type AAAA, it returns an error notification to the resolving IPv4 host. The correspondent host is in this case unreachable from the IPv4 host.

Figure 5 illustrates the operation of a recursive DNS name server in an IPv4-only network with Virtual IPv6 Connectivity. In the figure, an IPv4 host on the left requests a recursive DNS name server in the middle to resolve an IPv6 correspondent host’s name into an IPv4 address. The name of the correspondent host is "host.v6.net" in this case, and this is associated with a single resource record of type AAAA for IPv6 address 2001:DB8:BCD::X:Y:Z. The resolving IPv4 host initiates the procedure with a DNS query to the recursive DNS name server, requesting resource records of type A associated with "host.v6.net". The DNS query is not for resource records of type AAAA since only IPv4 addresses are of use for the IPv4 host. The recursive DNS name server first processes the DNS query unchanged, so it tries to retrieve resource records of type A from the DNS name server that is authoritative for the correspondent host, as represented on the right of Figure 5. This fails because the correspondent host does not have an IPv4 address. In a second attempt, the recursive DNS name server queries for resource records of type AAAA. This is successful; the recursive DNS name server receives a resource record with IPv6 address 2001:DB8:BCD::X:Y:Z. The recursive DNS name server then coordinates with an IP protocol translator in the IPv4-only network in order to create a mapping between the regular IPv6 address 2001:DB8:BCD::X:Y:Z and a virtual IPv4 address that routes to this IP protocol translator. In the example of Figure 5, this virtual IPv4 address is 10.u.v.w. The recursive DNS name server finally returns a DNS reply to the
resolving IPv4 host that contains a resource record of type A containing the virtual IPv4 address 10.u.v.w.

<table>
<thead>
<tr>
<th>IPv4 host</th>
<th>recursive DNS name server</th>
<th>DNS name server authoritative for correspondent host</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------------------&gt;</td>
<td>------------------------&gt;</td>
<td>&lt;-----------------------------</td>
</tr>
<tr>
<td>+------------------------+</td>
<td>+------------------------+</td>
<td>+------------------------+</td>
</tr>
<tr>
<td>create mapping state</td>
<td>2001:DB8:BCD::X:Y:Z &lt;-&gt; 10.u.v.w</td>
<td>create mapping state</td>
</tr>
<tr>
<td>+------------------------+</td>
<td>+------------------------+</td>
<td>+------------------------+</td>
</tr>
<tr>
<td>DNS reply: A for host.v6.net is 10.u.v.w</td>
<td>DNS reply: A for host.v6.net is 10.u.v.w</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Discovery of virtual IPv4 addresses via DNS

6. Multi-Homing Support

A technique for networks to increase the performance and reliability of their Internet connectivity is to establish multiple border links to either the same or different providers. This technique is called "multi-homing". To combine multi-homing with Virtual IPv6 Connectivity, an IPv4-only network must ensure that packets exchanged between local IPv4 hosts and remote IPv6 correspondent hosts pass
through an IP protocol translator that is aware of the mappings for the virtual IP addresses used in the packets. This can be achieved in one of the following two ways:

- **Shared virtual IP addresses** — The route to a virtual IPv4 or IPv6 address is announced by all IP protocol translators, and all IP protocol translators can handle the mapping for all virtual IP addresses. This requires synchronization of mapping state across IP protocol translators.

- **Unshared virtual IP addresses** — The route to each virtual IPv4 or IPv6 address is announced by exactly one IP protocol translator, and only that IP protocol translator can handle the mapping for the virtual IP address. It is then sufficient to maintain mapping state for a given virtual IPv4 address in only one IP protocol translator.

Sharing virtual IP addresses across multiple IP protocol translators provides a basis for automated fail-over and load balancing between border links. At the same time, it requires that IP protocol translators map virtual IP addresses in a consistent fashion. This is straightforward for the mappings of virtual IPv6 addresses: Since those mappings are constant, they can be pre-configured into all IP protocol translators. Pre-configuration is not possible for the mappings of virtual IPv4 addresses, however, because those mappings are dynamic. Mappings of virtual IPv4 addresses must instead be synchronized between IP protocol translators.

Synchronization of mapping state for virtual IPv4 addresses may be proactive or on-demand. Proactive synchronization ensures that IP protocol translators have a consistent view of all mappings at any time. This requires immediate mapping updates whenever a new mapping is created. On-demand synchronization enables IP protocol translators to retrieve any missing mapping they may need to process a packet. The mappings of virtual IPv4 addresses may then be centrally maintained in a single database that IP protocol translators can query when needed.

Unshared virtual IP addresses can be used without synchronization between IP protocol translators. It is instead sufficient to ensure that the virtual IPv4 and IPv6 addresses that are used in the mappings for a particular communication belong to partitions from the same IP protocol translator. The disjoint route announcements of the IP protocol translators then ensure that packets exchanged between IPv4 and IPv6 hosts always route to the IP protocol translator that can map the IP addresses in the packets. A disadvantage of partitioned virtual IP addresses is that active communications cannot be switched to a different IP protocol translator for the purpose of
fail-over or load balancing because they are bound to a route via a particular IP protocol translator.

Although active communications cannot be switched between IP protocol translators, communications that are newly established should be able to go via any IP protocol translator of an IPv4-only network. The partitioning of virtual IPv4 addresses does not prevent this: Since the mappings for virtual IPv4 addresses are dynamically created, they can be created with a virtual IPv4 address that routes to a preferred IP protocol translator. However, the mappings for virtual IPv6 addresses cannot reflect a current preference for a specific IP protocol translator because those mappings are constant. So to enable the establishment of a new communication via any IP protocol translator, each regular IPv4 address of a host within the IPv4-only network must be mapped to multiple virtual IPv6 addresses, one for each IP protocol translator.

The decision of whether to use shared vs. partitioned virtual IP addresses may have an impact on whether a multi-homed IPv4 network can use virtual IPv6 addresses that have been allocated to its provider: If the border links of the multi-homed IPv4 network connect to different providers, virtual IPv6 addresses must be provider-independent in order to be shared across IP protocol translators. This may cause additional load for the global routing system because routes towards the virtual IPv6 addresses could not be announced in aggregated manner. It is therefore recommended that multi-homed IPv4 networks with border links to multiple providers use virtual IPv6 addresses that are provider-allocated, and hence by necessity partitioned virtual IP addresses.

7. Security Considerations

No security considerations yet; to be written down.

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgment

Many ideas in Virtual IPv6 Connectivity were originally proposed in 2002 by Alain Durand [draft-durand-ngtrans-nat64-nat46].

This document was generated using the xml2rfc tool.
10. Normative References


Authors’ Addresses

Christian Vogt
Ericsson Research, NomadicLab
Hirsalantie 11
02420 Jorvas
Finland

Email: christian.vogt@ericsson.com

Alain Durand
Comcast
1500 Market Street
Philadelphia, PA 19102
USA

Email: alain_durand@cable.comcast.com
Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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.