IPv6 Addressing of IPv6/IPv4 Translators

draft-thaler-behave-translator-addressing-00.txt

Dave Thaler

dthaler@microsoft.com
Terminology

IPv6 addresses assigned to IPv6 hosts:

IPv6 addresses used for IPv4 hosts:

IPv4-\textit{translatable} address (::ffff:0:a.b.c.d in SIIT)

IPv4-\textit{mapped} prefix (::ffff:a.b.c.d in SIIT, sockets)

Any address
Prefix Requirements

IPv4-”Mapped”:
1. With IPv6 network, an IPv4-mapped prefix MUST map to a unique IPv4 network
2. Prefix MUST be short enough to allow more-specifics for multiple translators
3. IPv6 nodes SHOULD be able to distinguish IPv4-mapped vs. native IPv6 addresses

IPv4-”Translatable (stateless only):
4. Prefix MUST be <= 64 bits to work with stateless addrconf

Both:
5. MUST NOT inject into IPv6 network a route for every route in the IPv4 Internet
6. SHOULD be deployable with no code changes to existing hosts
7. SHOULD only require routing on 80 bits or less
IPv6 Internet <-> an IPv4 network

- Requires stateful

- IPv4-”Mapped”:
  - Must be network-specific prefix
An IPv6 network <-> IPv4 Internet

• IPv4-”Mapped”:  
  – Default to a well-known prefix, but be configurable in all devices to be network-specific  
  – WKP MUST NOT be advertised into IPv6 Internet, & SHOULD be filtered  
  – Can’t use ::ffff:<IPv4> for in-network translator without changing hosts, so need new IANA range

• IPv4-“Translatable” (stateless only):  
  – If IPv6 Internet connected, a WKP would require separate translatable vs globally-routable IPv6 prefixes  
  – Simplest to just use network-specific prefix
Address Format Requirements

1. MUST be 1-1 and reversible
2. MUST NOT change meaning of universal/local bit (71st bit) in IPv6 address unless starts with binary 000
3. SHOULD provide space to number multiple translators if “mapped” address
4. SHOULD be checksum-neutral if stateless
5. SHOULD allow v4-dotted decimal in addr (which requires IPv4 in last 32 bits)
6. SHOULD support (optional) IPv4 topology hiding (for “IPv6 Internet”<->”an IPv4 network” mapped addrs)
7. MAY hide IPv4 addr for “helpful” NATs
“Helpful” IPv4 NATs

IPv4-“Mapped” addresses:

A IPv6/IPv4 Translator IPv4 Internet “Helpful” IPv4 NAT B

A

Payload: MPFX::2.2.2.2

Payload: MPFX::192.168.2.2

IPv4-“Translatable” addresses:

A Stateless IPv6/IPv4 “Helpful” IPv4 NAT IPv4 Internet B

A

Payload: TPFX::10.1.1.1

Payload: TPFX::1.1.1.1

TPFX::10.1.1.1 10.1.1.1 1.1.1.1
Address Formats

• SIIT/sockets mapped= ::ffff:<IPv4>
• SIIT translatable= ::ffff:0:<IPv4>
• Zero-pad & embed= PREFIX::<IPv4>
• Compensation-pad & embed= PREFIX::<comp16>:a.b.c.d
• Embed & zero-pad (IVI)= PREFIX:<IPv4>::
  – Unless WKP: prefix length <38 or >70
• Embed & comp-pad= PREFIX:<IPv4>:<comp16>
• Preconfigured table
## Comparison against opt. requirements

<table>
<thead>
<tr>
<th></th>
<th>Route on &lt;= /80 (xlatable)</th>
<th>Multiple xlators</th>
<th>Checksum neutral</th>
<th>Dotted decimal</th>
<th>Topology hiding</th>
<th>Obscure IPv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>::ffff:&lt;IPv4&gt;</td>
<td>No</td>
<td>Using IPv4</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>::ffff:0:&lt;IPv4&gt;</td>
<td>No</td>
<td>n/a</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PFX::&lt;IPv4&gt;</td>
<td>Using PFX</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PFX::&lt;comp&gt;:&lt;IPv4&gt;</td>
<td>Using PFX</td>
<td>Using PFX</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PFX:&lt;IPv4&gt;::</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PFX:&lt;IPv4&gt;::&lt;comp&gt;</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Configured table</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Summary

• Can we narrow down the list?
• Do we need an opt. topology hiding algorithm?
• Get a new IANA WKP for IPv4-mapped default for scenarios:
  – “an IPv6 network” <-> “IPv4 Internet”
  – “an IPv6 network” <-> “an IPv4 network”
• Filter it (like ::ffff:0:0/96) on IPv6 Internet

More open questions:
1. Expand doc to cover encapsulation too?
   – If both, then joint Behave/Softwires doc?
2. Any additional requirements (e.g., for referrals)?
3. Deprecate SIIT IPv4-translatable format?