IETF Work on network to application signaling

And many others: Pierre Pfister, Tommy Pauly, David Schnazi, Wenqin, Lorenzo, Eric, Mikael, Ian, Veronika, Marcus, ....

Eric Vyncke, evyncke@cisco.com, @evyncke
Distinguished Engineer, Paris Innovation & Research Lab
May 2018
This session is about technologies being drafted at the IETF and still under development...

Comments will be welcome 😊
Problem statement #1: Selecting Among Several IPv6 Addresses
Short Introduction to IPv6

- Only $2^{32}$ addresses in IPv4 => shortage even with NAT & CGN
- IPv6 specified 1997 (!), updated by RFC 8200
  - Larger 128-bit addresses
  - Unchanged datalink layer: WiFi, 5G, Ethernet, ...
  - Mostly transparent for transport and application layers: TCP, HTTP, FTP, ...
  - Neighbour Discovery Protocol (NDP) new layer-2 protocol for address allocation (stateless DHCP), address resolution (ARP)
Neighbor Discovery Protocol: Router Advertisement

Router Advertisements contains:
- 64-bit prefix to be used by hosts (with 64-bit random) to form IPv6 address
- Data-link layer address of the router
- Miscellaneous options: MTU, DHCPv6 use, DNS servers, ...

1. Router Solicitation (RS):
   • Data = Query: please send RA

2. Router Advertisement:
   • Data= options, prefix, DNS servers, ...
IPv6 Is Here to Stay

https://www.vyneke.org/ipv6status
### IPv6 For Mobile

- 3GPP PDP Contexts
  - IPv6
  - IPv4-IPV6
  - IPv4
- IETF has RFC 6459
- 3GPP relies on RA
  - Only one /64 prefix

<table>
<thead>
<tr>
<th>Rank</th>
<th>Participating Network</th>
<th>ASN(s)</th>
<th>IPv6 deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Comcast</td>
<td>7015, 7016, 7725, 7922, 11025, 13367, 13385, 20214, 21508, 22258, 22909, 33287, 33489, 33490, 33491, 33650, 33651, 33652, 33653, 33654, 33655, 33656, 33657, 33659, 33660, 33661, 33662, 33664, 33665, 33666, 33667, 33668, 36732, 36733</td>
<td>65.33%</td>
</tr>
<tr>
<td>2</td>
<td>KDDI</td>
<td>2516</td>
<td>41.97%</td>
</tr>
<tr>
<td>3</td>
<td>RELIANCE JIO INFOCOMM LTD</td>
<td>55836, 64049</td>
<td>87.53%</td>
</tr>
<tr>
<td>4</td>
<td>SoftBank</td>
<td>17676</td>
<td>34.17%</td>
</tr>
<tr>
<td>5</td>
<td>ATT</td>
<td>6389, 7018, 7132</td>
<td>64.71%</td>
</tr>
<tr>
<td>6</td>
<td>Charter Communications</td>
<td>7843, 10796, 11351, 11426, 11427, 12271, 20001, 20115, 33363</td>
<td>31.41%</td>
</tr>
<tr>
<td>7</td>
<td>Verizon Wireless</td>
<td>6167, 22394</td>
<td>84.04%</td>
</tr>
<tr>
<td>8</td>
<td>T-Mobile USA</td>
<td>21928</td>
<td>93.05%</td>
</tr>
<tr>
<td>9</td>
<td>Deutsche Telekom AG</td>
<td>3320</td>
<td>53.45%</td>
</tr>
<tr>
<td>10</td>
<td>British Sky Broadcasting</td>
<td>5607</td>
<td>83.68%</td>
</tr>
<tr>
<td>11</td>
<td>Vivo</td>
<td>10429, 11419, 18881, 19182, 26599, 27699</td>
<td>40.33%</td>
</tr>
<tr>
<td>12</td>
<td>Liberty Global</td>
<td>5089, 6830, 20825, 29562</td>
<td>16.35%</td>
</tr>
<tr>
<td>13</td>
<td>Orange Business Services</td>
<td>3215</td>
<td>37.19%</td>
</tr>
<tr>
<td>14</td>
<td>Rogers Communications</td>
<td>812, 20453</td>
<td>49.43%</td>
</tr>
<tr>
<td>15</td>
<td>SKTelecom</td>
<td>9644</td>
<td>31.31%</td>
</tr>
<tr>
<td>16</td>
<td>Cox Communications</td>
<td>22773</td>
<td>48.12%</td>
</tr>
<tr>
<td>17</td>
<td>AT&amp;T Wireless</td>
<td>20057</td>
<td>56.79%</td>
</tr>
</tbody>
</table>

http://www.worldipv6launch.org/measurements/
Hosts and networks are multi-homed

Just a few examples…

Add 5G slices?

Phone Connection Sharing
Addressing in Multi-Homed Networks in IPv6

• Assign Provider Assigned (PA) addresses to hosts.
  • Native to IPv6 hosts (RFC4861, ...)
  • HNCP for home networks (RFC7788)
  • draft-ietf-rtgwg-enterprise-pa-multihoming for corporate networks.

• Teach the hosts to pick and use multiple addresses.
  • IPv6 source address selection (RFC6724)
  • Multi-Path TCP (RFC6824), SCTP, QUIC, ...

• Give the host meaningful information about the addresses.
Multihoming and CDNs

- Name lookups for resources stored on CDNs give different answers depending on the network connection
- Host on homenet may look up name using resolver from provider A, then connect to CDN using provider B
- This will generate support requests
- What to do?

Ted Lemon, Homenet WG, IETF-99
Multihoming problem illustrated

Which source does the client use?

From Marcus Kean, Microsoft IT, at V6OPS IETF-99
Provisioning the host

- How can the host discover all network prefixes and services?
- At the network and application layers
1. Identify Provisioning Domains (PvDs)

[ RFC7556 ] Provisioning Domains (PvDs) are consistent sets of network properties that can be implicit, or advertised explicitly.

Differentiate provisioning domains by using FQDN identifiers.

2. Extend PvD with additional information

For the applications
Step 1: Identify PvDs

With the PvD ID Router Advertisement Option

- At most one occurrence in each RA.
- PvD ID is an FQDN associated with options included in the PvD option.
- H bit to indicate Additional Information is available with HTTPS.
- L bit to indicate the PvD has legacy DHCP on the link.
- A bit to indicate that another RA header is included in the container
- Seq. number used for push-based refresh.
Step 1b: Identifying PvD (Cont.)

- Information in an RA without PvD ID is linked to an implicit PvD (identified by interface & link-local address of router)

- DHCPv6 information MUST be associated to a PvD ID received on the same interface from the same link-local address

- L-bit can be used to indicate the associated DHCPv4 server

**IPv6 hosts (read iOS, Android, Windows, Linux, ...) can receive PVD even in an IPv4-only network**
Step 2: Get the PvD Additional Application Data

When the H bit is set:

```
GET https://<pvd-id>/well-known/pvd
```

Using network configuration (source address, default route, DNS, etc…) associated with the received PvD.
Step 2: Get the PvD Additional Data

```json
{
    "name": "Foo Wireless",
    "expires": "2018-07-26T06:00:00Z",
    "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
    "dnsZones": ["example.com","sub.example.com"]
}
```

Some other examples (see also [https://smart.mpvd.io/.well-known/pvd](https://smart.mpvd.io/.well-known/pvd)):

noInternet : true,
metered : true,
captivePortalURL : "https://captive.org/foo.html"
Problem Statement #2: Captive Portals

capport Working Group
Flow Examples
Status Quo

Host

With Captivity
- Send Probe
- User Portal Interaction
- User Requests

Without Captivity
- Send Probe
- User Requests

Network

- DHCP/RA
- Redirect to Captive Portal
- Portal Complete
- Pass Requests

Discovery Interaction
Flow Examples

Status Quo

Host

Network

With Captivity

Send Probe

DHCP/RA

Requirements:

- Requires probe with or without captivity
- No updates of captive changes
- Relies on redirects of what look like user requests
- No support for non-browser clients

Without Captivity

Send Probe

Pass Probe

User Requests

Pass Requests
Captive Portals...

• Current working: HTTP(S) redirection
  • Not working with HSTS and normal browser
  • Or rely on OS detection via [http://captive.example.com/hotspot-detect.html](http://captive.example.com/hotspot-detect.html)
  • Not easy for users when having multiple providers on a single portal (Boingo, Ipass, ...)

• PvD
  • One PvD per provider
  • Each PvD additional data has the provider name, optionally walled garden information and the URL for the captive portal (working with HSTS)
Flow Examples

PvD

Host

<table>
<thead>
<tr>
<th>Without Captivity (with PvD URI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request PvD URI</td>
</tr>
<tr>
<td>Send PvD JSON</td>
</tr>
</tbody>
</table>

Network

<table>
<thead>
<tr>
<th>DHCP/RA + PvD ID &amp; URI</th>
</tr>
</thead>
</table>

User Requests

Pass Requests
Flow Examples

PvD

Host

Network

Without Captivity (with PvD URI)

Request PvD URI

Send PvD JSON

DHCP/RA + PvD ID & URI

✔️ No captive probe needed

✔️ Other network properties are conveyed through PvD
Flow Examples

PvD

Host

Network

With Captivity as Built-In Interaction

- Request PvD URI
- System Portal Interaction

- DHCP/RA + PvD ID & URI
- Send PvD JSON

Portal Complete

User Requests

Pass Requests

Discovery
Interaction
## Flow Examples

### PvD

<table>
<thead>
<tr>
<th>Host</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>With Captivity as Built-In Interaction</td>
<td></td>
</tr>
<tr>
<td>Request PvD URI</td>
<td>DHCP/RA + PvD ID &amp; URI</td>
</tr>
<tr>
<td>System Portal Interaction</td>
<td>Send PvD JSON</td>
</tr>
<tr>
<td>User Requests</td>
<td>Portal Complete</td>
</tr>
<tr>
<td></td>
<td>Pass Requests</td>
</tr>
</tbody>
</table>
PvD Status and Next Steps
Implementation status

Linux - [https://github.com/IPv6-mPvD](https://github.com/IPv6-mPvD)

- **pvdd**: user-space daemon managing PvD IDs and additional data
- **Linux Kernel** patch for RA processing
- **iproute** tool patch to display PvD IDs
- **Wireshark** dissector
- **RADVD** and **ODHCPD** sending PvD ID
A New, Evolutive API and Transport-Layer Architecture for the Internet:
https://www.neat-project.org/

European H-2020 project
10 partners (Cisco, Mozilla, EMC, Celerway…)

Integration to NEAT code: https://github.com/NEAT-project/neat/pull/80

Asking the user to choose with relevant criteria and simple UI

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Time</th>
<th>Data</th>
<th>Cost</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTE (Orange)</td>
<td>2 min</td>
<td>0.5 GB</td>
<td>$0</td>
<td>3%</td>
</tr>
<tr>
<td>VPN over LTE (Orange)</td>
<td>6 min</td>
<td>0.5 GB</td>
<td>$0</td>
<td>4%</td>
</tr>
<tr>
<td>Wi-Fi (Oslo Hotell Wi-Fi)</td>
<td>11 min</td>
<td></td>
<td></td>
<td>1%</td>
</tr>
</tbody>
</table>
Extending PvD Keys for Applications?

- Extension mechanism is via a IANA registry
- What could be signaled to the applications?
  - Optimized for VoD video?
  - Fake WiFi (actually a MiFi router) detection?
  - Announcing a free but walled garden WiFi (entertainment, IoT, ...)?
  - Properties of each 5G slice?
  - ...


Privacy and Security

- Can Pvd ID be spoofed?
- Confidentiality of additional information?
Spoofing the PvD ID

- Can an hostile party send rogue PvD, pretending to be example.org while they are hacker.org?
- No signature in the RA option (SeND not used)

The draft has mitigation mechanism based on TLS, X.509 certificates, ....
Confidentiality of PvD Additional Information

- The well-known URL https://pvd-name.example.org/.well-known/pvd could contain some sensitive data (bandwidth, recursive DNS servers, ...)

- This well-known URL is guessable ;-)

- How to provide confidentiality?

  1) do not put anything which is really confidential
  2) the HTTPS server should reject connections originated from prefixes not belonging to example.org
Host Privacy with Additional Information

• Each host will fetch the additional information on connection
• The HTTPS server will know the IP address of all clients and that the client is connecting...
  • Some privacy issues esp. if using EUI-64 or stable address

• Host can change to another IP address after fetching the file
• HTTPS belongs to the network operator (same as RADIUS, DHCP, ...)
• Anyway, it has more privacy than [http://captive.example.com/hotspot-detect.html](http://captive.example.com/hotspot-detect.html) which belongs to another global operator
So, **PvD with additional information are not THAT bad**

But we all know that nothing is never 100% secure!

And, in current standards/deployments hosts have to trust the first level of access (switch, WiFi AP, router)
This session was about technologies being drafted at the IETF and still under development...

Comments are welcome 😊
Conclusion

- Multi-homing in IPv6 is vastly different than in IPv4
- Several addresses per interface
- Several interfaces per host in 2018
- Host must select the right bundle of DNS, address, next hop
- Implementations exist
- Huge momentum at IETF
Back-up Slides
Layer-2 Adjacent Attacker

WiFi hotspot, ....

RA-guard

PvD=good.com
Attackers are First Hop Router and PvD "Server"

PvD=good.com
Flag=H
PIO=2001:db8:bad::/64

H-flag is required
X.509 certificate is wrong
=> Do not trust
Attacker is the First Hop Router

H-flag is required
PIO not covered by "Prefixes"
=> Do not trust
Attacker is the First Hop Router with NPTv6

PvD=good.com
Flag=H
PIO=2001:db8:beef::/64

My PvD are in 2001:db8:beef:: but this TLS client is in 2001:db8:bad::

=> Do not trust

NPT
2001:db9:beef:: ↔
2001:db8:bad::

H-flag is required
But cannot connect to the PvD server
=> Drop HTTPS request
Attacker Has a Foothold in “Good” PvD

IPv6 tunnel over foo

All appears good to host and PvD server... PvD approach does not help in this case
But, it requires a foothold in good PvD