Discovering Provisioning Domain Names and Data

draft-ietf-intarea-provisioning-domains-02

P. Pfister, E. Vyncke, T. Pauly, D. Schinazi, W. Shao

IETF-102, Montréal, July 2018
Hosts and networks are multi-homed

Just a few examples...

- Corp. ISP1
- Corp. ISP2
- Mobile SP
- Corp. VPN
- Wifi
- Wire
- Phone Connection Sharing
Multi-Homing, the legacy way...

Diagram:
- Corp. ISP1
  - Public Address 1
- Corp. ISP2
  - Public Address 2
- NAT
  - Private Addresses
- Mobile SP
- Phone Connection Sharing
- Routing and/or DPI
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-07 for corp. networks.

- Teach the hosts to pick and use multiple addresses.
  • IPv6 source address selection (RFC6724)
  • draft-linkova-6man-default-addr-selection-update
  • draft-ietf-v6ops-conditional-ras-05
  • Multi-Path TCP (RFC6824)
Multihoming problem illustrated

From Marcus Kean, Microsoft IT, at V6OPS IETF-99
Bundling IP address & DNS resolver

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
The purpose of this draft is to:

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. Give PvD Additional Information.

   Name, characteristics, captive portal, etc...
Step 1: PvD ID

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type    | Length | H|L|R | Reserved | Delay |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sequence Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

... PvD ID FQDN ...

... +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
... | Padding |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

... Router Advertisement message header ...

... (Only present when R-flag is set) ...

... +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
... | Options ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```
Step 1: Identify PvDs

- At most one occurrence in each RA.
- PvD ID is an FQDN associated with options in the RA.
- Implicit PvDs (without option) identified by RA source address and interface.
- L bit to indicate the PvD has DHCPv4 on the link.
- H bit to indicate Additional Information is available with HTTPS.
- R bit to indicate that another RA header is included
- Seq. number used for push-based refresh.
- Delay is for exponential backoff when refreshing
PvD ID Example

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1

<table>
<thead>
<tr>
<th>Type: 21</th>
<th>Length: 12</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequence Number</td>
<td>7</td>
<td>e</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>a</td>
<td>m</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td></td>
<td>l</td>
<td>e</td>
<td>3</td>
<td>o</td>
<td></td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>g</td>
<td>0</td>
<td>0 (padding)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 (padding)</td>
<td>0 (padding)</td>
<td>0 (padding)</td>
<td>0 (padding)</td>
<td></td>
</tr>
</tbody>
</table>
|          | RDNSS option (RFC 6106) length: 5 ...
|          | Prefix Information Option (RFC 4861) length: 4 ...
```
PvD ID Example

<table>
<thead>
<tr>
<th>Type: 21</th>
<th>Length: 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td></td>
</tr>
</tbody>
</table>

Sequence Number | 7 | e |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>a</td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>e</td>
<td>3</td>
</tr>
<tr>
<td>r</td>
<td>g</td>
<td>0</td>
</tr>
</tbody>
</table>

0 (padding) | 0 (padding) | 0 (padding) | 0 (padding) |

RDNSS option (RFC 6106) length: 5

Prefix Information Option (RFC 4861) length: 4
Step 2: Get the PvD Additional 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

HTTP/TLS

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

```
{
  "name": "Foo Wireless",
  "expires": "2017-07-23T06:00:00Z",
}
```

Some other examples (see also [https://smart.mpvd.io/.well-known/pvd](https://smart.mpvd.io/.well-known/pvd)) as well as draft-pfister-capport-pvd-00

captive-api : "https://captive.org/api"
<table>
<thead>
<tr>
<th></th>
<th>Option Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>IP Address/Prefix Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>18</td>
<td>New Router Prefix Information Option</td>
<td>[RFC4068]</td>
</tr>
<tr>
<td>19</td>
<td>Link-layer Address Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>20</td>
<td>Neighbour Advertisement Acknowledgment Option</td>
<td>[RFC5568]</td>
</tr>
<tr>
<td>21</td>
<td><strong>PvD ID Router Advertisement Option (reclaimable in future)</strong></td>
<td>[draft-ietf-intarea-provisioning-domains]</td>
</tr>
<tr>
<td>22</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>MAP Option</td>
<td>[RFC4140]</td>
</tr>
<tr>
<td>24</td>
<td>Route Information Option</td>
<td>[RFC4191]</td>
</tr>
<tr>
<td>25</td>
<td>Recursive DNS Server Option</td>
<td>[RFC5006] [RFC8106]</td>
</tr>
<tr>
<td>26</td>
<td>RA Flags Extension Option</td>
<td>[RFC5175]</td>
</tr>
<tr>
<td>27</td>
<td>Handover Key Request Option</td>
<td>[RFC5269]</td>
</tr>
<tr>
<td>28</td>
<td>Handover Key Reply Option</td>
<td>[RFC5269]</td>
</tr>
</tbody>
</table>
Implementation status

Linux - [https://github.com/IPv6-mPvD](https://github.com/IPv6-mPvD)
- pvdd: A Daemon to manage 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