intarea
Internet-Draft
Intended status: Informational
Expires: January 1, 2018

B. Bruneau
Ecole Polytechnique
P. Pfister
Cisco
D. Schinazi
T. Pauly
Apple
E. Vyncke
Cisco

June 30, 2017

Discovering Provisioning Domain Names and Data
draft-bruneau-intarea-provisioning-domains-01

Abstract

An increasing number of hosts and networks are connected to the Internet through multiple interfaces, some of which may provide multiple ways to access the internet by the mean of multiple IPv6 prefix configurations.

This document describes a way for hosts to retrieve additional information about their network access characteristics. The set of configuration items required to access the Internet is called a Provisioning Domain (PvD) and is identified by a Fully Qualified Domain Name (FQDN). This identifier, retrieved using a new Router Advertisement (RA) option, is associated with the set of information included within the RA and may later be used to retrieve additional information associated to the PvD by the mean of an HTTP request.

Copyright Notice

Table of Contents

1. Introduction

It has become very common in modern networks that hosts have Internet or more specific network access through different networking interfaces, tunnels, or next-hop routers. The concept of Provisioning Domain (PvD) was defined in [RFC7556] as a set of network configuration information which can be used by hosts in order to access the network.

In this document, PvDs are identified by Fully Qualified Domain Names called PvD IDs, which are included in Router Advertisements [RFC4861] as a new option and are used to identify correlated sets of network configuration data.

Devices connected to the Internet through multiple interfaces would typically be provisioned with one PvD per interface, but it is worth noting that multiple PvDs with different PvD IDs could be provisioned on any host interface, as well as noting that the same PvD ID could be used on different interfaces in order to inform the host that both PvDs, on different interfaces, ultimately provide identical services.

This document also introduces a way for hosts to retrieve additional information related to a specific PvD by the mean of an HTTP-over-TLS query using an URI derived from the PvD ID. The retrieved JSON object contains additional network information that would typically be considered unfit, or too large, to be directly included in the Router Advertisements. This information can be used by the networking stack, the applications, or even be partially displayed to the users (e.g., by displaying a localized network service name).

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In addition, this document uses the following terminology:

PvD: A Provisioning Domain, a set of network configuration information; for more information, see [RFC7556].

PvD ID: A Fully Qualified Domain Name (FQDN) used to identify a PvD.

Explicit PvD: A PvD uniquely identified with a PvD ID.

Implicit PvD: A PvD associated with a set of configuration information that, in the absence of a PvD ID, is associated with the advertising router.

3. Provisioning Domain Identification using Router Advertisements

Each provisioning domain is identified by a PvD ID. The PvD ID is a Fully Qualified Domain Name (FQDN) which MUST belong to the network operator in order to avoid ambiguity. The same PvD ID MAY be used in several access networks if the set of configuration information is identical (e.g. in all home networks subscribed to the same service).

3.1. PvD ID Option for Router Advertisements

This document introduces a new Router Advertisement (RA) option called the PvD ID Router Advertisement Option, used to convey the FQDN identifying a given PvD.

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |  Seq  |H|L|     Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Lifetime                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PvD ID FQDN                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PvD ID Router Advertisements Option format

Type : (8 bits) To be defined by IANA.

Length : (8 bits) The length of the option (including the Type and Length fields) in units of 8 octets.

Seq : (4 bits) Sequence number for the PvD Additional Information, as described in Section 5.

H-flag : (1 bit) Whether some PvD Additional Information is made available through HTTP over TLS, as described in Section 5.

L-flag : (1 bit) Whether the router is also providing IPv4 access using DHCPv4 (see Section 3.3.2).

Reserved : (10 bits) Reserved for later use. It MUST be set to zero by the sender and ignored by the receiver.

Lifetime : (32 bits) The length of time in seconds (relative to the time the packet is sent) that the PvD ID option is valid. A value of all one bits (0xffffffff) represents infinity.

PvD ID FQDN : The FQDN used as PvD ID in DNS binary format [RFC1035], padded until the next 8 octets boundary. All the bytes after the first empty DNS label MUST be set to zero by the sender and MUST be ignored by the receiver. The DNS name compression technique described in [RFC1035] MUST NOT be used.

Routers MUST NOT include more than one PvD ID Router Advertisement Option in each RA. In case multiple PvD ID options are found in a given RA, hosts MUST ignore all but the first PvD ID option.

Note: The existence and/or size of the sequence number is subject to discussion. The validity of a PvD Additional Information object is included in the object itself, but this only allows for 'pull based' updates, whereas the RA options usually provide 'push based' updates.

Note: including the lifetime option is congruent with the lifetime option of the other RA option. On the other hand, do we need it ?

3.2. Router Behavior

A router MAY insert at most one PvD ID Option in its RAs. The included PvD ID is associated with all the other options included in the same RA (e.g., Prefix Information [RFC4861], Recursive DNS Server [RFC6106], Routing Information [RFC4191], Captive Portal [RFC7710] options).

In order to provide multiple independent PvDs, a router MUST send multiple RAs using different source link-local addresses (LLA) (as proposed in [I-D.bowbakova-rtgwg-enterprise-pa-multihoming]), each of which MAY include a PvD ID option. In such cases, routers MAY originate the different RAs using the same datalink layer address.

If the router is actually a VRRP instance [RFC5798], then the procedure is identical except that the virtual datalink layer address is used as well as virtual IPv6 addresses.

3.3. Host Behavior

RAs are used to configure IPv6 hosts. When a host receives a RA message including a PvD ID Option with a non-zero lifetime, it MUST associate all the configuration options included in the RA (e.g., Prefix Information [RFC4861], Recursive DNS Server [RFC6106], Routing Information [RFC4191], Captive Portal [RFC7710] options) with the PvD ID).

If the received RA does not include a PvD ID Option, or whenever the included PvD ID Option has a lifetime set to zero, the host MUST associate the options included in the RA with an implicit PvD. This implicit PvD is identified by the link-local address of the router sending the RA and the interface on which the RA was received.

This document does not update the way Router Advertisement options are processed. But in addition to the option processing defined in other documents, hosts implementing this specification MUST associate each created or updated object (e.g. address, default route, more specific route, DNS server list) with the PvD associated with the received RA as well as the interface and link-local address of the router which last updated the object.

Each configuration object has a PvD validity timer set to the PvD ID option lifetime whenever an RA that updates the object is received. If the received RA does not include a PvD ID option, or whenever the PvD ID option lifetime is zero, the associated objects are immediately associated with an implicit PvD, as mentioned in the previous paragraph. Similarly, whenever an object's PvD timer reaches zero, the object is immediately associated with an implicit PvD identified by the link-local address and interface of the router which last updated the object.

While resolving names, executing the default address selection algorithm [RFC6724] or executing the default router selection algorithm ([RFC2461], [RFC4191] and [RFC8028]), hosts MAY consider only the configuration associated with an arbitrary set of PvDs.

For example, a host MAY associate a given process with a specific PvD, or a specific set of PvDs, while associating another process with another PvD. A PvD-aware application might also be able to select, on a per-connection basis, which PvDs should be used for a given connection. And particularly constrained devices such as small battery operated devices (e.g. IoT), or devices with limited CPU or memory resources may purposefully use a single PvD while ignoring some received RAs containing different PvD IDs.

The way an application expresses its desire to use a given PvD, or a set of PvDs, or the way this selection is enforced, is out of the scope of this document. Useful insights about these considerations can be found in [I-D.kline-mif-mpvd-api-reqs].

3.3.1. DHCPv6 configuration association

When a host retrieves configuration information from DHCPv6, the configured elements MUST be associated with the explicit or implicit PvD of the RA received on the same interface with the O-flag set

[RFC4861]. If multiple RAs with the O-flag set and associated with different PvDs are received on the same interface, the link-local address of the DHCPv6 server MAY be compared with the routers' link-local addresses in order to disambiguate. If the disambiguation is impossible, then the DHCPv6 configuration information MUST be associated with an implicit PvD.

This process requires hosts to keep track of received RAs, associated PvD IDs, and routers link-local addresses.

3.3.2. DHCPv4 configuration association

When a host retrieves configuration from DHCPv4 on an interface, the configured elements MUST be associated with the explicit PvD, received on the same interface, whose PVD ID Options L-flag is set. If multiple different PvDs are found, the datalink layer address used by the DHCPv4 server/relay MAY be compared with the datalink layer address used by on-link advertising routers in order to disambiguate. If no RA associated with a PvD ID option with the L-flag set is found, or if the disambiguation fails, the IPv4 configuration information MUST be associated with an implicit PvD.

3.3.3. Interconnection Sharing by the Host

The situation when a host becomes also a router by acting as a router or ND proxy on a different interface (such as WiFi) to share the connectivity of another interface (such as cellular), also known as "tethering" is TBD but it is expected that the one or several PvD associated to the shared interface will be also advertised to the clients.

4. Provisioning Domain Identification using IKEv2

RFC 7296 defines Internet Key Exchange version 2 (IKEv2) which includes in sections 2.19 and 3.15 a Configuration Payload (CP) which may be used by an IPsec client to request configuration items (e.g., addresses, recursive DNS servers). IKEv2 also negatiates traffic selectors which represent the 5-tuple(s) to be protected by the Security Association (SA) negotiated by IKEv2. This set of information is another PvD and may include INTERNAL_IP6_ADDRESS, INTERNAL_IP6_LINK, INTERNAL_IP6_SUBNET (to be used to route traffic to this subent), INTERNAL_IP6_DNS, INTERNAL_DNS_DOMAIN.

The PvD ID Configuration option for IKEv2 has the following structure (similar to the RA option):

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|  Attribute Type             |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Seq  |H|L|     Reserved                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Lifetime                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          PvD ID FQDN                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

PvD ID IKEv2 Configuration Payload Attributes format

R-bit Reserved (1 bit) - Defined in IKEv2 RFC [RFC7296].

Attribute Type (15 bits) tbd by IANA PVD_ID.

Length Length (2 octets, unsigned integer) - Length of PvD ID FQDN + 2.

Seq Sequence number (4 bits) for the PvD Additional Information, as described in Section 5.

H-flag (1 bit) Whether some PvD Additional Information is made available through HTTP over TLS, as described in Section 5.

L-flag (1 bit) must be set to 0 if the Configuration Payload contains only IPv6 information, else it must be set to 1.

Reserved (10 bits) Reserved for later use. It MUST be set to zero by the sender and ignored by the receiver.

PvD ID FQDN The FQDN used as PvD ID in DNS binary format [RFC1035], padded until the next 8 octets boundary. All the bytes after the first empty DNS label MUST be set to zero by the sender and MUST be ignored by the receiver. The DNS name compression technique described in [RFC1035] MUST NOT be used.

5. Provisioning Domain Additional Information

Once a new PvD ID is discovered, it may be used to retrieve additional information about the characteristics of the provided connectivity. This set of information is called PvD Additional Information, and is encoded as a JSON object [RFC7159].

The purpose of this additional set of information is to securely provide additional information to hosts about the connectivity that is provided using a given interface and source address pair. It typically includes data that would be considered too large, or not critical enough, to be provided with an RA option. The information contained in this object MAY be used by the operating system, network libraries, applications, or users, in order to decide which set of PvDs should be used for which connection, as described in Section 3.3.

5.1. Retrieving the PvD Additional Information

When the H-flag of the PvD ID Option is set, hosts MAY attempt to retrieve the PvD Additional Information associated with a given PvD by performing an HTTP over TLS [RFC2818] GET query to https://<PvD-ID>/pvd.json [RFC5785]. On the other hand, hosts MUST NOT do so whenever the H-flag is not set.

Note: Should the PvD AI retrieval be a MAY or a SHOULD ? Could the object contain critical data, or should it only contain informational data ?

Note that the DNS name resolution of <PvD-ID> as well as the actual query MUST be performed in the PvD context associated to the PvD ID. In other words, the name resolution, source address selection, as well as the next-hop router selection MUST be performed while using exclusively the set of configuration information attached with the PvD, as defined in Section 3.3.

If the HTTP status of the answer is greater than or equal to 400 the host MUST abandon and consider that there is no additional PvD information. If the HTTP status of the answer is between 300 included and 399 included it MUST follow the redirection(s). If the HTTP status of the answer is between 200 included and 299 included the host MAY get a file containing a single JSON object. When a JSON object could not be retrieved, an error message SHOULD be logged and/ or displayed in a rate-limited fashion.

After retrieval of the PvD Additional Information, hosts MUST watch the PvD ID Seq field for change. In case a different value than the one in the RA Seq field is observed, or whenever the validity time included in the object is expired, hosts MUST either perform a new query and retrieve a new version of the object, or deprecate the object and stop using it.

Hosts retrieving a new PvD Additional Information object MUST check for the presence and validity of the mandatory fields Section 5.3. A retrieved object including an outdated expiration time or missing a

mandatory element, MUST be ignored. In order to avoid traffic spikes toward the server hosting the PvD Additional Information when an object expires, a host which last retrieved an object at a time A, including a validity time B, SHOULD renew the object at a uniformly random time in the interval [(B-A)/2,A].

In order to prevent PvD spoofing by malicious or misconfigured routers, PvD Additional Information object includes a set of IPv6 prefixes which MUST be checked against all the Prefix Information Options advertised in the Router Advertisement. If any of the prefixes included in the Prefix Information Options is not in the set of prefixes contained in the additional information, the PvD (the one included in the RA and in the additional information) MUST be considered unsafe and MUST NOT be used. While this does not prevent a malicious network provider, it can be used to detect misconfiguration.

The server serving the JSON files SHOULD also check whether the client address is part of the prefixes listed into the additional information and SHOULD return a 403 response code if there is no match. The server MAY also use the client address to select the right JSON object to be returned.

Note: In a similar way, a DNS server names list could be included in the PvD AI in order to avoid rogue APs from inserting a different DNS resolver. But this would also prevent CPEs from advertising themselves as default DNS (which is usually done). SPs which really want to use CPEs as DNS, for caching reasons, might find 'creative' ways to go around this rule.

5.2. Providing the PvD Additional Information

Whenever the H-flag is set in the PvD RA Option, a valid PvD Additional Information object MUST be made available to all hosts receiving the RA. In particular, when a captive portal is present, hosts MUST still be allowed to access the object, even before logging into the captive portal.

Routers MAY increment the PVD ID Sequence number (Seq) in order to inform host that a new PvD Additional Information object is available and should be retrieved.

5.3. PvD Additional Information Format

The PvD Additional Information is a JSON object.

The following array presents the mandatory keys which MUST be included in the object:

JSON key	Description	Type	Example
name	Human-readable service name	UTF-8 string	"Awesome Wifi"
expires	Date after which this object is not valid	ISO 8601	"2017-07-23T06:00:00Z"
prefixes	Array of IPv6 prefixes valid for this PVD	Array of strings	["2001:db8:1::/48", "2001:db8:4::/48"]

A retrieved object which does not include a valid string associated with the "name" key at the root of the object, or a valid date associated with the "expiration" key, also at the root of the object, MUST be ignored. In such cases, an error message SHOULD be logged and/or displayed in a rate-limited fashion.

The following table presents some optional keys which MAY be included in the object.

JSON key	Description	Type	Example
localizedName	Localized user-visible service name, language can be selected based on the HTTP Accept-Language header in the request.	UTF-8 string	"Wifi Genial"
noInternet	No Internet, set when the PvD only provides restricted access to a set of services.	boolean	true
characteristics	Connectivity characteristics	JSON object	See Section 5.3.1
metered	metered, when the access volume is limited.	boolean	false

It is worth noting that the JSON format allows for extensions. Whenever an unknown key is encountered, it MUST be ignored along with its associated elements, unless specified otherwise in future updating documents.

5.3.1. Connectivity Characteristics Information

The following set of keys can be used to signal certain characteristics of the connection towards the PvD.

They should reflect characteristics of the overall access technology which is not limited to the link the host is connected to, but rather a combination of the link technology, CPE upstream connectivity, and further quality of service considerations.

JSON key	Description	Type	Example
maxThroughput	Maximum achievable throughput	object({down(int), up(int)}) in kb/s	{"down": 10000, "up": 5000}
minLatency	Minimum achievable latency	object({down(int), up(int)}) in ms	{"down": 10, "up": 20}
rl	Maximum achievable reliability	object({down(int), up(int)}) in losses every 1000 packets	{"down": 0.1, "up": 1}

5.3.2. Private Extensions

JSON keys starting with "x-" are reserved for private use and can be utilized to provide information that is specific to vendor, user or enterprise. It is RECOMMENDED to use one of the patterns "x-FQDN-KEY" or "x-PEN-KEY" where FQDN is a fully qualified domain name or PEN is a private enterprise number [PEN] under control of the author of the extension to avoid collisions.

5.3.3. Example

Here are two examples based on the keys defined in this section.

```
{
  "name": "Foo Wireless",
  "localizedName": "Foo-France Wifi",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes" : ["2001:db8:1::/48", "2001:db8:4::/48"],
  "characteristics": {
   "maxThroughput": { "down":200000, "up": 50000 },
   "minLatency": { "down": 0.1, "up": 1 }
  }
}
```

```
{
  "name": "Bar 4G",
  "localizedName": "Bar US 4G",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "metered": true,
  "characteristics": {
    "maxThroughput": { "down":80000, "up": 20000 }
  }
}
```


6. Security Considerations

Although some solutions such as IPSec or SEND [RFC3971] can be used in order to secure the IPv6 Neighbor Discovery Protocol, actual deployments largely rely on link layer or physical layer security mechanisms (e.g. 802.1x [IEEE8021X]).

This specification does not improve the Neighbor Discovery Protocol security model, but extends the purely link-local configuration retrieval mechanisms with HTTP-over-TLS communications.

During the exchange, the server authenticity is verified by the mean of a certificate, validated based on the FQDN found in the Router Advertisement (e.g. using a list of pre-installed CA certificates, or DNSSec [RFC4035] with DNS Based Authentication of Named Entities [RFC6698]). This authentication creates a secure binding between the information provided by the trusted Router Advertisement, and the HTTP server.

The IPv6 prefixes list included in the PvD Additional Information JSON object is used to validate that the prefixes included in the Router Advertisements are really part of the PvD.

Note: The previous point does not protect against attacker performing NAT66. It would if we mandate the source-address validation on the server side, but not protect against tunnels. In other words, address based security will not protect against rogue APs, but may prevent the use of NAT66.

For privacy reasons, it is desirable that the PvD Additional Information object may only be retrieved by the hosts using the given PvD. Host identity SHOULD be validated based on the client address that is used during the HTTP query.

7. IANA Considerations

IANA is kindly requested to allocate a new IPv6 Neighbor Discovery option number for the PvD ID Router Advertisement option and a new IKEv2 Configuration Payload Attribute Types for PVD_ID.

TBD: JSON keys will need a new register.

8. Acknowledgements

Many thanks to M. Stenberg and S. Barth for their earlier work: [I-D.stenberg-mif-mpvd-dns].

Thanks also to Ray Bellis, Lorenzo Colitti, Thierry Danis, Marcus Keane, Erik Kline, Jen Lenkova, Mark Townsley and James Woodyatt for useful and interesting discussions.

Finally, many thanks to Thierry Danis for his implementation work: [github].

9. References

9.1. Normative references

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2461] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <http://www.rfc-editor.org/info/rfc2818>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

9.2. Informative references

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <http://www.rfc-editor.org/info/rfc4035>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <http://www.rfc-editor.org/info/rfc5798>.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <http://www.rfc-editor.org/info/rfc6698>.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <http://www.rfc-editor.org/info/rfc7556>.

[RFC7710] Kumari, W., Gudmundsson, O., Ebersman, P., and S. Sheng, "Captive-Portal Identification Using DHCP or Router Advertisements (RAs)", RFC 7710, DOI 10.17487/RFC7710, December 2015, <http://www.rfc-editor.org/info/rfc7710>.

[RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <http://www.rfc-editor.org/info/rfc8028>.

[I-D.bowbakova-rtgwg-enterprise-pa-multihoming]
Baker, F., Bowers, C., and J. Linkova, "Enterprise Multihoming using Provider-Assigned Addresses without Network Prefix Translation: Requirements and Solution", draft-bowbakova-rtgwg-enterprise-pa-multihoming-01 (work in progress), October 2016.

[I-D.stenberg-mif-mpvd-dns]
Stenberg, M. and S. Barth, "Multiple Provisioning Domains using Domain Name System", draft-stenberg-mif-mpvd-dns-00 (work in progress), October 2015.

[I-D.kline-mif-mpvd-api-reqs]
Kline, E., "Multiple Provisioning Domains API Requirements", draft-kline-mif-mpvd-api-reqs-00 (work in progress), November 2015.

[PEN] IANA, "Private Enterprise Numbers", <https://www.iana.org/assignments/enterprise-numbers>.

[IEEE8021X]
IEEE, "IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Std", .

[github] Cisco, "IPv6-mPvD github repository", <https://github.com/IPv6-mPvD>.

Appendix A. Changelog

Note to RFC Editors: Remove this section before publication.

A.1. Version 00

Initial version of the draft. Edited by Basile Bruneau + Eric Vyncke and based on Basile's work.

A.2. Version 01

Major rewrite intended to focus on the the retained solution based on corridors, online, and WG discussions. Edited by Pierre Pfister. The following list only includes major changes.

PvD ID is an FQDN retrieved using a single RA option. This option contains a sequence number for push-based updates, a new H-flag, and a L-flag in order to link the PvD with the IPv4 DHCP server.

A lifetime is included in the PvD ID option.

Detailed Hosts and Routers specifications.

Additional Information is retrieved using HTTP-over-TLS when the PvD ID Option H-flag is set. Retrieving the object is optional.

The PvD Additional Information object includes a validity date.

DNS-based approach is removed as well as the DNS-based encoding of the PvD Additional Information.

Major cut in the list of proposed JSON keys. This document may be extended later if need be.

Monetary discussion is moved to the appendix.

Clarification about the 'prefixes' contained in the additional information.

Clarification about the processing of DHCPv6.

Appendix B. Connection monetary cost

NOTE: This section is included as a request for comment on the potential use and syntax.

The billing of a connection can be done in a lot of different ways. The user can have a global traffic threshold per month, after which his throughput is limited, or after which he/she pays each megabyte. He/she can also have an unlimited access to some websites, or an unlimited access during the weekends.

An option is to split the bill in elementary billings, which have conditions (a start date, an end date, a destination IP address...). The global billing is an ordered list of elementary billings. To know the cost of a transmission, the host goes through the list, and the first elementary billing whose the conditions are fulfilled gives the cost. If no elementary billing conditions match the request, the host MUST make no assumption about the cost.

B.1. Conditions

Here are the potential conditions for an elementary billing. All conditions MUST be fulfill.

Key	Description	Type	JSON Example
beginDate	Date before which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
endDate	Date after which the billing is not valid	ISO 8601	"1977-04-22T06:00:00Z"
domains	FQDNs whose the billing is limited	array(string)	["deezer.com","spotify.com"]
prefixes4	IPv4 prefixes whose the billing is limited	array(string)	["78.40.123.182/32","78.40.123.183/32"]
prefixes6	IPv6 prefixes whose the billing is limited	array(string)	["2a00:1450:4007:80e::200e/64"]

B.2. Price

Here are the different possibilities for the cost of an elementary billing. A missing key means "all/unlimited/unrestricted". If the elementary billing selected has a trafficRemaining of 0 kb, then it means that the user has no access to the network. Actually, if the last elementary billing has a trafficRemaining parameter, it means that when the user will reach the threshold, he/she will not have access to the network anymore.

Key	Description	Type	JSON Example
pricePerGb	The price per Gigabit	float (currency per Gb)	2
currency	The currency used	ISO 4217	"EUR"
throughputMax	The maximum achievable throughput	float (kb/s)	100000
trafficRemaining	The traffic remaining	float (kB)	12000000

B.3. Examples

Example for a user with 20 GB per month for 40 EUR, then reach a threshold, and with unlimited data during weekends and to example.com:

```
[
  {
    "domains": ["example.com"]
  },
  {
    "prefixes4": ["78.40.123.182/32","78.40.123.183/32"]
  },
  {
    "beginDate": "2016-07-16T00:00:00Z",
    "endDate": "2016-07-17T23:59:59Z",
  },
  {
    "beginDate": "2016-06-20T00:00:00Z",
    "endDate": "2016-07-19T23:59:59Z",
    "trafficRemaining": 12000000
  },
  {
    "throughputMax": 100000
  }
]
```

If the host tries to download data from example.com, the conditions of the first elementary billing are fulfilled, so the host takes this elementary billing, finds no cost indication in it and so deduces that it is totally free. If the host tries to exchange data with foobar.com and the date is 2016-07-14T19:00:00Z, the conditions of the first, second and third elementary billing are not fulfilled.

But the conditions of the fourth are. So the host takes this elementary billing and sees that there is a threshold, 12 GB are remaining.

Another example for a user abroad, who has 3 GB per year abroad, and then pay each MB:


```
[
  {
    "beginDate": "2016-02-10T00:00:00Z",
    "endDate": "2017-02-09T23:59:59Z",
    "trafficRemaining": 3000000
  },
  {
    "pricePerGb": 30,
    "currency": "EUR"
  }
]
```


Authors' Addresses

Basile Bruneau
Ecole Polytechnique
Vannes 56000
France

Email: basile.bruneau@polytechnique.edu

Pierre Pfister
Cisco
11 Rue Camille Desmoulins
Issy-les-Moulineaux 92130
France

Email: ppfister@cisco.com

David Schinazi
Apple

Email: dschinazi@apple.com

Tommy Pauly
Apple

Email: tpauly@apple.com

Eric Vyncke
Cisco
De Kleetlaan, 6
Diegem 1831
Belgium

Email: evyncke@cisco.com