Network Working Group   P. Pfister
Internet-Draft   E. Vyncke
Intended status: Standards Track   Cisco
Expires: April 2, 2020   T. Pauly
                                                         Apple Inc.
                                                         D. Schinazi
                                                         Google LLC
                                                          W. Shao
                                                            Cisco
                                                 September 30, 2019

                 Discovering Provisioning Domain Names and Data
                   draft-ietf-intarea-provisioning-domains-07

Abstract

   Provisioning Domains (PvDs) are defined as consistent sets of network
   configuration information. This allows hosts to manage connections
   to multiple networks and interfaces simultaneously, such as when a
   home router provides connectivity through both a broadband and
   cellular network provider.

   This document defines a mechanism for explicitly identifying PvDs
   through a Router Advertisement (RA) option. This RA option announces
   a PvD identifier, which hosts can compare to differentiate between
   PvDs. The option can directly carry some information about a PvD and
   can optionally point to additional PvD information that can be
   retrieved using HTTP over TLS.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on April 2, 2020.

Copyright Notice

Table of Contents

1. Introduction

   Provisioning Domains (PvDs) are defined in [RFC7556] as consistent
   sets of network configuration information. This information includes
   properties that are traditionally associated with a single networking
   interface, such as source addresses, DNS configuration, proxy
   configuration, and gateway addresses.

   Clients that are aware of PvDs can take advantage of multiple network
   interfaces simultaneously. This enables using two PvDs in parallel
   for separate connections or for multi-path transports.

   While most PvDs today are discovered implicitly (such as by receiving
   information via Router Advertisements from a router on a network that
   a client host directly connects to), [RFC7556] also defines the
   notion of Explicit PvDs. IPsec Virtual Private Networks are
   considered Explicit PvDs, but Explicit PvDs can also be discovered
   via the local network router. Discovering Explicit PvDs allows two
   key advancements in managing multiple PvDs:

   1. The ability to discover and use multiple PvDs on a single
       interface, such as when a local router can provide connectivity
       to two different Internet Service Providers.

   2. The ability to associate additional informations about PvDs to
       describe the properties of the network.

   While [RFC7556] defines the concept of Explicit PvDs, it does not
   define the mechanism for discovering multiple Explicit PvDs on a
   single network and their additional information.

   This document specifies a way to identify PvDs with Fully Qualified
   Domain Names (FQDN), called PvD IDs. Those identifiers are
   advertised in a new Router Advertisement (RA) [RFC4861] option called
   the PvD ID Router Advertisement option which, when present,
   associates the PvD ID with all the information present in the Router
   Advertisement as well as any configuration object, such as addresses,
   deriving from it. The PVD ID Router Advertisement option may also
   contain a set of other RA options. Since such options are only
   considered by hosts implementing this specification, network
   operators may configure hosts that are 'PvD-aware' with PvDs that are
   ignored by other hosts.

Since PvD IDs are used to identify different ways to access the internet, multiple PvDs (with different PvD IDs) can be provisioned on a single host interface. Similarly, the same PvD ID could be used on different interfaces of a host in order to inform that those PvDs ultimately provide equivalent services.

This document also introduces a mechanism for hosts to retrieve optional additional information related to a specific PvD by means of an HTTP over TLS query using an URI derived from the PvD ID. The retrieved JSON object contains additional information that would typically be considered too large to be directly included in the Router Advertisement, but might be considered useful to the applications, or even sometimes users, when choosing which PvD should be used.

For example, if Alice has both a cellular network provider and a broadband provider in her home, her PvD-aware devices and applications would be aware of both available uplinks. These applications could fail-over between these networks, or run connections over both (potentially using multi-path transports). Applications could also select specific uplinks based on the properties of the network; for example, if the cellular network provides free high-quality video streaming, a video-streaming application could select that network while most of the other traffic on Alice's device uses the broadband provider.

1.1. Specification of Requirements

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

This document uses the following terminology:

Provisioning Domain (PvD): 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. For more information, see [RFC7556].

Implicit PvD: A PvD that, in the absence of a PvD ID, is identified
   by the host interface to which it is attached and the address of
   the advertising router. See also [RFC7556].

PvD-aware host: A host that supports the association of network
   configuration information into PvDs and the use of these PvDs as
   described in this document. Also named PvD-aware node in
   [RFC7556].

3. Provisioning Domain Identification using Router Advertisements

Explicit PvDs are 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 naming collisions. The same PvD ID MAY be
used in several access networks when they ultimately provide
identical services (e.g., in all home networks subscribed to the same
service); else, the PvD ID MUST be different to follow Section 2.4 of
   [RFC7556].

3.1. PvD ID Option for Router Advertisements

This document introduces a Router Advertisement (RA) option called
PvD Option. It is used to convey the FQDN identifying a given PvD
(see Figure 1, bind the PvD ID with configuration information
received over DHCPv4 (see Section 3.4.2), enable the use of HTTP over
TLS to retrieve the PvD Additional Information JSON object (see
Section 4), as well as contain any other RA options which would
otherwise be valid in the RA.

```
 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    |H|L|R|      Reserved   | Delay |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Sequence Number         |                         ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                          ...
...                          PvD ID FQDN                   ...
...               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
...            |                 Padding                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                          ...
...            Router Advertisement message header        ...
...               (Only present when R-flag is set)       ...
...                                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

           Figure 1: PvD ID Router Advertisements Option Format

Type: (8 bits) Set to 21.

Length: (8 bits) The length of the option in units of 8 octets,
   including the Type and Length fields, the Router Advertisement
   message header, if any, as well as the RA options that are
   included within the PvD Option.

H-flag: (1 bit) 'HTTP' flag stating whether some PvD Additional
   Information is made available through HTTP over TLS, as described
   in Section 4.

L-flag: (1 bit) 'Legacy' flag stating whether the router is also
   providing IPv4 information using DHCPv4 (see Section 3.4.2).

R-flag: (1 bit) 'Router Advertisement' flag stating whether the PvD
   Option is followed (right after padding to the next 64 bits
   boundary) by a Router Advertisement message header (See section
   4.2 of [RFC4861]).

Delay: (4 bits) Unsigned integer used to delay HTTP GET queries from
   hosts by a randomized backoff (see Section 4.1).

Reserved: (13 bits) Reserved for later use. It MUST be set to zero
   by the sender and ignored by the receiver.

Sequence Number: (16 bits) Sequence number for the PvD Additional
   Information, as described in Section 4.

   PvD ID FQDN: The FQDN used as PvD ID encoded in DNS format, as
      described in Section 3.1 of [RFC1035]. Domain names compression
      described in Section 4.1.4 of [RFC1035] MUST NOT be used.

   Padding: Zero or more padding octets to the next 8 octet boundary
      (see Section 4.6 of [RFC4861]). It MUST be set to zero by the
      sender, and ignored by the receiver.

   RA message header: (16 octets) When the R-flag is set, a full Router
      Advertisement message header as specified in [RFC4861]. The
      sender MUST set the 'Type' to 134, the value for "Router
      Advertisement", and set the 'Code' to 0. Receivers MUST ignore
      both of these fields. The 'Checksum' MUST be set to 0 by the
      sender; non-zero checksums MUST be ignored by the receiver. All
      other fields are to be set and parsed as specified in [RFC4861] or
      any updating documents.

   Options: Zero or more RA options that would otherwise be valid as
      part of the Router Advertisement main body, but are instead
      included in the PvD Option such as to be ignored by hosts that are
      not PvD-aware.

   Here is an example of a PvD Option with "example.org" as the PvD ID
   FQDN and including both an RDNSS option and a prefix information
   option. It has a Sequence Number of 123, and indicates the presence
   of additional information that is expected to be fetched with a delay
   factor of 5.

```
 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: 21      | Length: 12    |1|0|0|    Reserved    |Delay:5|
+---------------+-----------------------------+---------------+
|       Seq number: 123         |       7       |      e        |
+---------------+-----------------------------+---------------+
|       x       |       a       |      m        |      p        |
+---------------------------------------------------------------+
|       l       |       e       |      3        |      o        |
+---------------------------------------------------------------+
|       r       |       g       |      0        |   0 (padding) |
+---------------------------------------------------------------+
|   0 (padding) |   0 (padding) |   0 (padding) |   0 (padding) |
+---------------+---------------+---------------+---------------+
|   RDNSS option (RFC 6106) length: 5                        ...
...                                                          ...
...                                                            |
+---------------------------------------------------------------+
| Prefix Information Option (RFC 4861) length: 4             ...
...                                                            |
...                                                            |
+---------------------------------------------------------------+
```

Figure 2

3.2. Router Behavior

A router MAY send RAs containing one PvD Option, but MUST NOT include
more than one PvD Option in each RA. The PvD Option MUST NOT contain
further PvD Options.

The PvD Option MAY contain zero, one, or more RA options which would
otherwise be valid as part of the same RA. Such options are
processed by PvD-aware hosts, while ignored by other hosts per
section 4.2 of [RFC4861].

In order to provide multiple different PvDs, a router MUST send
multiple RAs. If more than one different Implicit PvDs are
advertised, the RAs MUST be sent from different link-local source
addresses. Explicit PvDs MAY share link-local source addresses with
an Implicit PvD and any number of other Explicit PvDs.

In other words, different Explicit PvDs MAY be advertised with RAs
using the same link-local source address; but different Implicit
PvDs, advertised by different RAs, MUST use different link-local
addresses because these Implicit PvDs are identified by the source
addresses of the RAs.

As specified in [RFC4861], when the set of options causes the size of
an advertisement to exceed the link MTU, multiple router
advertisements can be sent, each containing a subset of the options.
In such cases, the PvD Option header (i.e., all fields except the
'Options' field) MUST be repeated in all the transmitted RAs. The
options within the 'Options' field, MAY be transmitted only once,
included in one of the transmitted PvD Options.

3.3. Non-PvD-aware Host Behavior

As the PvD Option has a new option code, non-PvD-aware hosts will
simply ignore the PvD Option and all the options it contains (see
section 4.2 of [RFC4861]. This ensure the backward compatibility
required in Section 3.3 of [RFC7556]. This behavior allows for a
mixed-mode network with a mix of PvD-aware and non-PvD-aware hosts
coexist.

3.4. PvD-aware Host Behavior

Hosts MUST associate received RAs and included configuration
information (e.g., Router Valid Lifetime, Prefix Information
[RFC4861], Recursive DNS Server [RFC8106], Routing Information
[RFC4191] options) with the Explicit PvD identified by the first PvD
Option present in the received RA, if any, or with the Implicit PvD
identified by the host interface and the source address of the
received RA otherwise.

In case multiple PvD Options are found in a given RA, hosts MUST
ignore all but the first PvD Option.

If a host receives PvD Options flags that it does not recognize
(currently in the Reserved field), it MUST ignore these flags.

Similarly, hosts MUST associate all network configuration objects
(e.g., default routers, addresses, more specific routes, DNS
Recursive Resolvers) with the PvD associated with the RA which last
updated the object. For example, addresses that are generated using
a received Prefix Information option (PIO) are associated with the
PvD of the last received RA which included the given PIO.

PvD IDs MUST be compared in a case-insensitive manner as defined by
[RFC4343]. For example, "pvd.example.com." or "PvD.Example.coM."
would refer to the same PvD.

While resolving names, executing the default address selection
algorithm [RFC6724] or executing the default router selection
algorithm when forwarding packets ([RFC4861], [RFC4191] and

[RFC8028]), hosts and applications 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. In
particular, 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].

3.4.1. DHCPv6 configuration association

When a host retrieves stateless configuration elements using DHCPv6
(e.g., DNS recursive resolvers or DNS domain search lists [RFC3646]),
they MUST be associated with all the explicit and implicit PvDs
received on the same interface and contained in a RA with the O-flag
set [RFC4861].

When a host retrieves stateful assignments using DHCPv6, such
assignments MUST be associated with the received PvD which was
received with RAs with the M-flag set and including a matching PIO.
A PIO is considered to match a DHCPv6 assignment when the IPv6 prefix
from the PIO includes the assignment from DHCPv6. For example, if a
PvD's associated PIO defines the prefix 2001:db8:cafe::/64, a DHCPv6
IA_NA message that assigns the address 2001:db8:cafe::1234:4567 would
be considered to match.

In cases where an address would be assigned by DHCPv6 and no matching
PvD could be found, hosts MAY associate the assigned address with any
implicit PvD received on the same interface or to multiple of
implicit PvD received on the same interface. This is intended to
resolve backward compatibility issues with rare deployments choosing
to assign addresses with DHCPv6 while not sending any matching PIO.

3.4.2. DHCPv4 configuration association

Associating DHCPv4 [RFC2131] configuration elements with Explicit
PvDs allows hosts to treat a set of IPv4 and IPv6 configurations as a
single PvD with shared properties. For example, consider a router
that provides two different uplinks. One could be a broadband
network that has data rate and streaming properties described in PvD

additional information and that provides both IPv4 and IPv6 network
access. The other could be a cellular network that provides only
IPv6 network access, and uses NAT64 [RFC6146]. The broadband network
can be represented by an Explicit PvD that points to the additional
information, and also marks association with DHCPv4 information. The
cellular network can be represented by a different Explicit PvD that
is not associated with DHCPv4.

When a PvD-aware host retrieves configuration elements from DHCPv4,
the information is associated either with a single Explicit PvD on
that interface, or else with all Implicit PvDs on the same interface.

An Explicit PvD indicates its association with DHCPv4 information by
setting the L-flag in the PvD RA Option. If there is exactly one
Explicit PvD that sets this flag, hosts MUST associate the DHCPv4
information with that PvD. Multiple Explicit PvDs on the same
interface marking this flag is a misconfiguration, and hosts SHOULD
NOT associate the DHCPv4 information with any Explicit PvD in this
case.

If no single Explicit PvD claims association with DHCPv4, the
configuration elements coming from DHCPv4 MUST be associated with the
all Implicit PvDs identified by the interface on which the DHCPv4
transaction happened. This maintains existing host behavior.

3.4.3. Connection Sharing by the Host

The situation when a host shares connectivity from an upstream
interface (e.g. cellular) to a downstream interface (e.g. Wi-Fi) is
known as 'tethering'. Techniques such as ND-proxy [RFC4389], 64share
[RFC7278] or prefix delegation (e.g. using DHCPv6-PD [RFC8415]) may
be used for that purpose.

Whenever the RAs received from the upstream interface contain a PVD
RA option, hosts that are sharing connectivity SHOULD include a PVD
option within the RAs sent downstream with:

o The same PVD-ID FQDN

o The same H-bit, Delay and Sequence Number values

o The L bit set whenever the host is sharing IPv4 connectivity
   received from the same upstream interface

o The bits from the Reserved field set to 0

The values of the R-bit, Router Advertisement message header and
Options field depend on whether the connectivity should be shared

only with PvD-aware hosts or not (see Section 3.2). In particular,
all options received within the upstream PvD Option and included in
the downstream RA SHOULD be included in the downstream PvD Option.

3.4.4. Usage of DNS Servers

PvD-aware hosts can be provisioned with recursive DNS servers via RA
options passed within an Explicit PvD, via RA options associated with
an Implicit PvD, via DHCPv6 or DHCPv4, or from some other
provisioning mechanism that creates an Implicit PvD (such as a VPN).
In all of these cases, the DNS server addresses SHOULD be associated
with the corresponding PvD. Specifically, queries sent to a
configured recursive DNS server SHOULD be sent from a local IP
address that was provisioned by the PvD via RA or DHCP. Answers
received from the DNS server SHOULD only be used on the same PvD.

PvD-aware applications will be able to select which PvD(s) to use for
DNS resolution and connections, which allows them to effectively use
multiple Explicit PvDs. In order to support non-PvD-aware
applications, however, PvD-aware hosts SHOULD ensure that non-PvD-
aware name resolution APIs like "getaddrinfo" only use resolvers from
a single PvD for each query. More discussion is provided in
Section 5.2.1 of [RFC7556].

Maintaining the correct usage of DNS within PvDs avoids various
practical errors, such as:

o A PvD associated with a VPN or otherwise private network may
   provide DNS answers that contain addresses inaccessible over
   another PvD.

o A PvD that uses a NAT64 [RFC6146] and DNS64 [RFC6147] will
   synthesize IPv6 addresses in DNS answers that are not globally
   routable, and would be invalid on other PvDs. Conversely, an IPv4
   address resolved via DNS on another PvD cannot be directly used on
   a NAT64 network.

4. Provisioning Domain Additional Information

Additional information about the network characteristics can be
retrieved based on the PvD ID. This set of information is called PvD
Additional Information, and is encoded as a JSON object [RFC8259].
This JSON object is restricted to the restricted profile of I-JSON,
as defined in [RFC7493].

The purpose of this JSON object is to provide additional information
to applications on a client host about the connectivity that is
provided using a given interface and source address. It typically

includes data that would be considered too large, or not critical
enough, to be provided within 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 3.4.

The additional information related to a PvD is specifically intended
to be optional, and is targeted at optimizing or informing the
behavior of user-facing hosts. This information can be extended to
provide hints for host system behavior (such as captive portal or
walled-garden PvD detection) or application behavior (describing
application-specific services offered on a given PvD). This content
may not be appropriate for light-weight Internet of Things (IoT)
devices. IoT devices might need only a subset of the information,
and would in some cases prefer a smaller representation like CBOR
([RFC7049]). Delivering a reduced version of the PvD Additional
Information designed for such devices is not defined in this
document.

4.1. Retrieving the PvD Additional Information

When the H-flag of the PvD 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>/.well-known/pvd [RFC8615]. Inversely, hosts MUST NOT do so
whenever the H-flag is not set.

HTTP requests and responses for PvD additional information use the
"application/pvd+json" media type (see Section 8). Clients SHOULD
include this media type as an Accept header in their GET requests,
and servers MUST mark this media type as their Content-Type header in
responses.

Note that the DNS name resolution of the PvD ID, the PKI (Public Key
Infrastructure) checks as well as the actual query MUST be performed
using the considered PvD. In other words, the name resolution, PKI
checks, 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 3.4. In some cases, it may therefore be necessary to wait
for an address to be available for use (e.g., once the Duplicate
Address Detection or DHCPv6 processes are complete) before initiating
the HTTP over TLS query. If the host has a temporary address per
[RFC4941] in this PvD, then hosts SHOULD use a temporary address to
fetch the PvD Additional Information and SHOULD deprecate the used
temporary address and generate a new temporary address afterward.

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 and 399, inclusive, it MUST follow the redirection(s). If the HTTP status of the answer is between 200 and 299, inclusive, the host MAY get a file containing a single JSON object.

After retrieval of the PvD Additional Information, hosts MUST remember the last Sequence Number value received in the RA including the same PvD ID. Whenever a new RA for the same PvD is received with a different Sequence Number value, or whenever the expiry date for the additional information is reached, hosts MUST deprecate the additional information and stop using it until a new JSON object is retrieved.

Hosts retrieving a new PvD Additional Information object MUST check for the presence and validity of the mandatory fields specified in Section 4.3. A retrieved object including an expiration time that is already past or missing a mandatory element MUST be ignored.

In order to avoid synchronized queries toward the server hosting the PvD Additional Information when an object expires, object updates are delayed by a randomized backoff time.

o When a host performs a JSON object update after it detected a change in the PvD Option Sequence Number, it MUST add a delay before sending the query. The target time for the delay is calculated as a random time between zero and 2**(Delay * 2) milliseconds, where 'Delay' corresponds to the 4-bit unsigned integer in the last received PvD Option.

o When a host last retrieved a JSON object at time A that includes a expiry time B using the "expires" key, and the host is configured to keep the PvD information up to date, it MUST add some randomness into its calculation of the time to fetch the update. The target time for fetching the updated object is calculated as a uniformly random time in the interval [(B-A)/2,B].

In the example Figure 2, the delay field value is 5, this means that host calculates its delay by choosing a random number between 0 and 2**(5 * 2) milliseconds, i.e., between 0 and 1024 milliseconds.

Since the 'Delay' value is directly within the PvD Option rather than the object itself, an operator may perform a push-based update by incrementing the Sequence value while changing the Delay value depending on the criticality of the update and its PvD Additional Information servers capacity.

The PvD Additional Information object includes a set of IPv6 prefixes
(under the key "prefixes") which MUST be checked against all the
Prefix Information Options advertised in the RA. If any of the
prefixes included in the PIO is not covered by at least one of the
listed prefixes, the associated PvD information MUST be considered to
be a misconfiguration, and MUST NOT be used by the host. See
Section 4.4 for more discussion on handling such misconfigurations.

4.2. Operational Consideration to Providing the PvD Additional Information

Whenever the H-flag is set in the PvD Option, a valid PvD Additional
Information object MUST be made available to all hosts receiving the
RA by the network operator. In particular, when a captive portal is
present, hosts MUST still be allowed to perform DNS, PKI and HTTP
over TLS operations related to the retrieval of the object, even
before logging into the captive portal.

Routers SHOULD increment the PVD Option Sequence Number by one
whenever a new PvD Additional Information object is available and
should be retrieved by hosts. If the value exceeds what can be
stored in the Sequence Number field, it SHOULD wrap back to zero.

The server providing 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.

4.3. PvD Additional Information Format

The PvD Additional Information is a JSON object.

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

```
+-----------+----------------+----------+-----------------------+
| JSON key  | Description    | Type     | Example               |
+-----------+----------------+----------+-----------------------+
| identifier| PvD ID FQDN    | String   | "pvd.example.com."    |
|           |                |          |                       |
| expires   | Date after     | [RFC3339]| "2017-07-23T06:00:00Z"|
|           | which this     | Date     |                       |
|           | object is no   |          |                       |
|           | longer valid   |          |                       |
|           |                |          |                       |
| prefixes  | Array of IPv6  | Array of | ["2001:db8:1::/48",   |
|           | prefixes valid | strings  | "2001:db8:4::/48"]    |
|           | for this PvD   |          |                       |
+-----------+----------------+----------+-----------------------+
```

A retrieved object which does not include all three of these keys at
the root of the JSON object MUST be ignored. All three keys need to
be validated, otherwise the object MUST be ignored. The value stored
for "identifier" MUST be matched against the PvD ID FQDN presented in
the PvD RA option using the comparison mechanism described in
Section 3.4. The value stored for "expires" MUST be a valid date in
the future. If the PIO of the received RA is not covered by at least
one of the "prefixes" key, the retrieved object SHOULD be ignored.

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

```
+-----------+---------------------+----------+-------------------+
| JSON key  | Description         | Type     | Example           |
+-----------+---------------------+----------+-------------------+
| dnsZones  | DNS zones searchable| Array of | ["example.com",   |
|           | and accessible      | strings  |                   |
|           |                     |          |                   |
|           |                     |          | "sub.example.com"]|
|           |                     |          |                   |
| noInternet| No Internet, set    | Boolean  | true              |
|           | when the PvD is     |          |                   |
|           | restricted.         |          |                   |
+-----------+---------------------+----------+-------------------+
```

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.

Private-use or experimental keys MAY be used in the JSON dictionary.
In order to avoid such keys colliding with IANA registry keys,
implementers or vendors defining private-use or experimental keys
MUST create sub-dictionaries, where the sub-dictionary is added into

the top-level JSON dictionary with a key of the format "vendor-*"
where the "*" is replaced by the implementer's or vendor's
identifier. For example, keys specific to the FooBar organization
could use "vendor-foobar". Upon receiving such a sub-dictionary,
host MUST ignore this sub-dictionary if it is unknown. When the
vendor or implementer is part of an IANA URN namespace [URN], the URN
namespace SHOULD be used rather than the "vendor-*" format.

4.3.1. Example

The following two examples show how the JSON keys defined in this
document can be used:

```
{
  "identifier": "cafe.example.com",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
}

{
  "identifier": "company.foo.example.com",
  "expires": "2017-07-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "vendor-foo": { "private-key": "private-value" },
}
```

4.4. Detecting misconfiguration and misuse

When a host retrieves the PvD Additional Information, it MUST verify
that the TLS server certificate is valid for the performed request
(e.g., that the Subject Name is equal to the PvD ID expressed as an
FQDN). This authentication creates a secure binding between the
information provided by the trusted Router Advertisement, and the
HTTPS server. However, this does not mean the Advertising Router and
the PvD server belong to the same entity.

Hosts MUST verify that all prefixes in the RA PIO are covered by a
prefix from the PvD Additional Information. An adversarial router
attempting to spoof the definition of an Explicit PvD, without the
ability to modify the PvD Additional Information, would need to
perform NAT66 in order to circumvent this check. Thus, this check
cannot prevent all spoofing, but it can detect misconfiguration or
mismatched routers that are not adding a NAT.

If NAT66 is being added in order to spoof PvD ownership, the HTTPS
server for additional information can detect this misconfiguration.
The HTTPS server SHOULD validate the source addresses of incoming
connections (see Section 4.1). This check gives reasonable assurance

that neither NPTv6 [RFC6296] nor NAT66 were used and restricts the
information to the valid network users. If the PvD does not
provision IPv4 (it does not include the 'L' bit in the RA), the
server cannot validate the source addresses of connections using
IPv4. Thus, the PvD ID FQDN for such PvDs SHOULD NOT have a DNS A
record.

5. Operational Considerations

This section describes some example use cases of PvD. For the sake
of simplicity, the RA messages will not be described in the usual
ASCII art but rather in an indented list.

5.1. Exposing Extra RA Options to PvD-Aware Hosts

In this example, there is one RA message sent by the router. This
message contains some options applicable to all hosts on the network,
and also a PvD Option that also contains other options only visible
to PvD-aware hosts.

o RA Header: router lifetime = 6000

o Prefix Information Option: length = 4, prefix = 2001:db8:cafe::/64

o PvD Option header: length = 3 + 5 + 4 , PvD ID FQDN =
   example.org., R-flag = 0 (actual length of the header with padding
   24 bytes = 3 * 8 bytes)

   * Recursive DNS Server: length = 5, addresses =
      [2001:db8:cafe::53, 2001:db8:f00d::53]

   * Prefix Information Option: length = 4, prefix =
      2001:db8:f00d::/64

Note that a PvD-aware host will receive two different prefixes,
2001:db8:cafe::/64 and 2001:db8:f00d::/64, both associated with the
same PvD (identified by "example.org."). A non-PvD-aware host will
only receive one prefix, 2001:db8:cafe::/64.

5.2. Different RAs for PvD-Aware and Non-PvD-Aware Hosts

It is expected that for some years, networks will have a mixed
environment of PvD-aware hosts and non-PvD-aware hosts. If there is
a need to give specific information to PvD-aware hosts only, then it
is recommended to send two RA messages (one for each class of hosts).
For example, here is the RA sent for non-PvD-aware hosts:

   o RA Header: router lifetime = 6000 (non-PvD-aware hosts will use
      this router as a default router)

   o Prefix Information Option: length = 4, prefix = 2001:db8:cafe::/64

   o Recursive DNS Server Option: length = 3, addresses=
      [2001:db8:cafe::53]

   o PvD Option header: length = 3 + 2, PvD ID FQDN = foo.example.org.,
      R-flag = 1 (actual length of the header 24 bytes = 3 * 8 bytes)

      * RA Header: router lifetime = 0 (PvD-aware hosts will not use
         this router as a default router), implicit length = 2

   And here is the RA sent for PvD-aware hosts:

   o RA Header: router lifetime = 0 (non-PvD-aware hosts will not use
      this router as a default router)

   o PvD Option header: length = 3 + 2 + 4 + 3, PvD ID FQDN =
      bar.example.org., R-flag = 1 (actual length of the header 24 bytes
      = 3 * 8 bytes)

      * RA Header: router lifetime = 1600 (PvD-aware hosts will use
         this router as a default router), implicit length = 2

      * Prefix Information Option: length = 4, prefix =
         2001:db8:f00d::/64

      * Recursive DNS Server Option: length = 3, addresses =
         [2001:db8:f00d::53]

   In the above example, non-PvD-aware hosts will only use the first RA
   sent from their default router and using the 2001:db8:cafe::/64
   prefix. PvD-aware hosts will autonomously configure addresses from
   both PIOs, but will only use the source address in 2001:db8:f00d::/64
   to communicate past the first hop router since only the router
   sending the second RA will be used as default router; similarly, they
   will use the DNS server 2001:db8:f00d::53 when communicating with
   this address.

5.3. Enabling Multi-homing for PvD-Aware Hosts

   In this example, the goal is to have one prefix from one RA be usable
   by both non-PvD-aware and PvD-aware hosts; and to have another prefix
   usable only by PvD-aware hosts. This allows PvD-aware hosts to be
   able to effectively multi-home on the network.

The first RA is usable by all hosts. The only difference for PvD-
aware hosts is that they can explicitly identify the PvD ID
associated with the RA. PvD-aware hosts will also use this prefix to
communicate with non-PvD-aware hosts on the same network.

o RA Header: router lifetime = 6000 (non-PvD-aware hosts will use
   this router as a default router)

o Prefix Information Option: length = 4, prefix = 2001:db8:cafe::/64

o Recursive DNS Server Option: length = 3, addresses=
   [2001:db8:cafe::53]

o PvD Option header: length = 3, PvD ID FQDN = foo.example.org.,
   R-flag = 0 (actual length of the header 24 bytes = 3 * 8 bytes)

The second RA contains a prefix usable only by PvD-aware hosts. Non-
PvD-aware hosts will ignore this RA.

o RA Header: router lifetime = 0 (non-PvD-aware hosts will not use
   this router as a default router)

o PvD Option header: length = 3 + 2 + 4 + 3, PvD ID FQDN =
   bar.example.org., R-flag = 1 (actual length of the header 24 bytes
   = 3 * 8 bytes)

   * RA Header: router lifetime = 1600 (PvD-aware hosts will use
      this router as a default router), implicit length = 2

   * Prefix Information Option: length = 4, prefix =
      2001:db8:f00d::/64

   * Recursive DNS Server Option: length = 3, addresses =
      [2001:db8:f00d::53]

6. Security Considerations

Although some solutions such as IPsec or SeND [RFC3971] can be used
in order to secure the IPv6 Neighbor Discovery Protocol, in practice
actual deployments largely rely on link layer or physical layer
security mechanisms (e.g. 802.1x [IEEE8021X]) in conjunction with RA
Guard [RFC6105].

This specification does not improve the Neighbor Discovery Protocol
security model, but extends the purely link-local trust relationship
between the host and the default routers with HTTP over TLS
communications which servers are authenticated as rightful owners of
the FQDN received within the trusted PvD ID RA option.

It must be noted that [Section 4.4](#) of this document only provides
reasonable assurance against misconfiguration but does not prevent an
hostile network access provider to advertise wrong information that
could lead applications or hosts to select a hostile PvD.

Users cannot be assumed to be able to meaningfully differentiate
between "safe" and "unsafe" networks. This is a known attack surface
that is present whether or not PvDs are in use, and hence cannot be
addressed by this document. However, a host that correctly
implements the MPvD architecture ([[RFC7556](#)]) using the mechanism
described in this document will be less susceptible to such attacks
than a host that does not by being able to check for the various
misconfigurations described in this document.

[7](#). Privacy Considerations

Retrieval of the PvD Additional Information over HTTPS requires early
communications between the connecting host and a server which may be
located further than the first hop router. Although this server is
likely to be located within the same administrative domain as the
default router, this property can't be ensured. Therefore, hosts
willing to retrieve the PvD Additional Information before using it
without leaking identity information, SHOULD make use of an IPv6
Privacy Address and SHOULD NOT include any privacy sensitive data,
such as User Agent header or HTTP cookie, while performing the HTTP
over TLS query.

From a privacy perspective, retrieving the PvD Additional Information
is not different from establishing a first connection to a remote
server, or even performing a single DNS lookup. For example, most
operating systems already perform early queries to well known web
sites, such as http://captive.example.com/hotspot-detect.html, in
order to detect the presence of a captive portal.

There may be some cases where hosts, for privacy reasons, should
refrain from accessing servers that are located outside a certain
network boundary. In practice, this could be implemented as a
whitelist of 'trusted' FQDNs and/or IP prefixes that the host is
allowed to communicate with. In such scenarios, the host SHOULD
check that the provided PvD ID, as well as the IP address that it
resolves into, are part of the allowed whitelist.

[8](#). IANA Considerations

Upon publication of this document, IANA is asked to remove the
'reclaimable' tag off the value 21 for the PvD Option (from the IPv6
Neighbor Discovery Option Formats registry).

IANA is asked to assign the value "pvd" from the Well-Known URIs registry.

8.1. Additional Information PvD Keys Registry

IANA is asked to create and maintain a new registry called "Additional Information PvD Keys", which will reserve JSON keys for use in PvD additional information. The initial contents of this registry are given in Section 4.3.

New assignments for Additional Information PvD Keys Registry will be administered by IANA through Expert Review [RFC8126].

8.2. PvD Option Flags Registry

IANA is also asked to create and maintain a new registry entitled "PvD Option Flags" reserving bit positions from 0 to 15 to be used in the PvD Option bitmask. Bit position 0, 1 and 2 are reserved by this document (as specified in Figure 1). Future assignments require Standards Action [RFC8126], via a Standards Track RFC document.

8.3. PvD JSON Media Type Registration

This document registers the media type for PvD JSON text, "application/pvd+json".

Type Name: application

Subtype Name: pvd+json

Required parameters: None

Optional parameters: None

Encoding considerations: Encoding considerations are identical to those specified for the "application/json" media type.

Security considerations: See Section 6.

Interoperability considerations: This document specifies format of conforming messages and the interpretation thereof.

Published specification: This document

Applications that use this media type: This media type is intended to be used by network advertising additional Provisioning Domain information, and clients looking up such information.

Additional information: None

Person and email address to contact for further information: See
Authors' Addresses section

Intended usage: COMMON

Restrictions on usage: None

Author: IETF

Change controller: IETF

9. Acknowledgments

Many thanks to M. Stenberg and S. Barth for their earlier work:
[I-D.stenberg-mif-mpvd-dns], as well as to Basile Bruneau who was
author of an early version of this document.

Thanks also to Marcus Keane, Mikael Abrahamsson, Ray Bellis, Zhen
Cao, Tim Chown, Lorenzo Colitti, Michael Di Bartolomeo, Ian Farrer,
Phillip Hallam-Baker, Bob Hinden, Tatuya Jinmei, Erik Kline, Ted
Lemon, Paul Hoffman, Dave Thaler, Suresh Krishnan, Gorry Fairhurst,
Jen Lenkova, Veronika McKillop, Mark Townsley and James Woodyatt for
useful and interesting discussions and reviews.

Finally, special thanks to Thierry Danis and Wenqin Shao for their
valuable inputs and implementation efforts, Tom Jones for his
integration effort into the NEAT project and Rigil Salim for his
implementation work.

10. References

10.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and
           specification", STD 13, RFC 1035, DOI 10.17487/RFC1035,
           November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic
           Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646,
           DOI 10.17487/RFC3646, December 2003,
           <https://www.rfc-editor.org/info/rfc3646>.

[RFC4343] Eastlake 3rd, D., "Domain Name System (DNS) Case
           Insensitivity Clarification", RFC 4343,
           DOI 10.17487/RFC4343, January 2006,
           <https://www.rfc-editor.org/info/rfc4343>.

   [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
              "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
              DOI 10.17487/RFC4861, September 2007,
              <https://www.rfc-editor.org/info/rfc4861>.

   [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493,
              DOI 10.17487/RFC7493, March 2015,
              <https://www.rfc-editor.org/info/rfc7493>.

   [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
              Writing an IANA Considerations Section in RFCs", BCP 26,
              RFC 8126, DOI 10.17487/RFC8126, June 2017,
              <https://www.rfc-editor.org/info/rfc8126>.

   [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data
              Interchange Format", STD 90, RFC 8259,
              DOI 10.17487/RFC8259, December 2017,
              <https://www.rfc-editor.org/info/rfc8259>.

   [RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers
              (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019,
              <https://www.rfc-editor.org/info/rfc8615>.

10.2. Informative References

   [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.

   [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.

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

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",
              RFC 2131, DOI 10.17487/RFC2131, March 1997,
              <https://www.rfc-editor.org/info/rfc2131>.

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

   [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
              Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
              <https://www.rfc-editor.org/info/rfc3339>.

   [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander,
              "SEcure Neighbor Discovery (SEND)", RFC 3971,
              DOI 10.17487/RFC3971, March 2005,
              <https://www.rfc-editor.org/info/rfc3971>.

   [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and
              More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191,
              November 2005, <https://www.rfc-editor.org/info/rfc4191>.

   [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery
              Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April
              2006, <https://www.rfc-editor.org/info/rfc4389>.

   [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy
              Extensions for Stateless Address Autoconfiguration in
              IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007,
              <https://www.rfc-editor.org/info/rfc4941>.

   [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J.
              Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105,
              DOI 10.17487/RFC6105, February 2011,
              <https://www.rfc-editor.org/info/rfc6105>.

   [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
              NAT64: Network Address and Protocol Translation from IPv6
              Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146,
              April 2011, <https://www.rfc-editor.org/info/rfc6146>.

   [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van
              Beijnum, "DNS64: DNS Extensions for Network Address
              Translation from IPv6 Clients to IPv4 Servers", RFC 6147,
              DOI 10.17487/RFC6147, April 2011,
              <https://www.rfc-editor.org/info/rfc6147>.

   [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix
              Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011,
              <https://www.rfc-editor.org/info/rfc6296>.

   [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown,
              "Default Address Selection for Internet Protocol Version 6
              (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012,
              <https://www.rfc-editor.org/info/rfc6724>.

   [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object
              Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049,
              October 2013, <https://www.rfc-editor.org/info/rfc7049>.

   [RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6
              /64 Prefix from a Third Generation Partnership Project
              (3GPP) Mobile Interface to a LAN Link", RFC 7278,
              DOI 10.17487/RFC7278, June 2014,
              <https://www.rfc-editor.org/info/rfc7278>.

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

   [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,
              <https://www.rfc-editor.org/info/rfc8028>.

   [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli,
              "IPv6 Router Advertisement Options for DNS Configuration",
              RFC 8106, DOI 10.17487/RFC8106, March 2017,
              <https://www.rfc-editor.org/info/rfc8106>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A.,
              Richardson, M., Jiang, S., Lemon, T., and T. Winters,
              "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)",
              RFC 8415, DOI 10.17487/RFC8415, November 2018,
              <https://www.rfc-editor.org/info/rfc8415>.

   [URN]   "URN Namespaces", n.d..

Authors' Addresses

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

   Email: ppfister@cisco.com

   Eric Vyncke
   Cisco
   De Kleetlaan, 6
   Diegem 1831
   Belgium

   Email: evyncke@cisco.com

   Tommy Pauly
   Apple Inc.
   One Apple Park Way
   Cupertino, California 95014
   United States of America

   Email: tpauly@apple.com

   David Schinazi
   Google LLC
   1600 Amphitheatre Parkway
   Mountain View, California 94043
   United States of America

   Email: dschinazi.ietf@gmail.com

   Wenqin Shao
   Cisco
   11 Rue Camille Desmoulins
   Issy-les-Moulineaux 92130
   France

   Email: wenshao@cisco.com