Internet Engineering Task Force (IETF)
Request for Comments: 8801
Category: Standards Track
ISSN: 2070-1721

P. Pfister
É. Vyncke
Cisco
T. Pauly
Apple Inc.
D. Schinazi
Google LLC
W. Shao
Cisco

July 2020

Discovering Provisioning Domain Names and Data

Abstract

Provisioning Domains (PvDs) are defined as consistent sets of network configuration information. PvDs 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 PvD Additional Information that can be retrieved using HTTP over TLS.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8801.

described in the Simplified BSD License.

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 Information 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 (FQDNs), called PvD IDs. Those identifiers are advertised in a new Router Advertisement (RA) [RFC4861] option called the PvD 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, derived from it. The PvD Option may also contain a set of other RA options, along with an optional inner Router Advertisement message header. These options and optional inner header are only visible to 'PvD-aware' hosts, allowing such hosts to have a specialized view of the network configuration.

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 a 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].

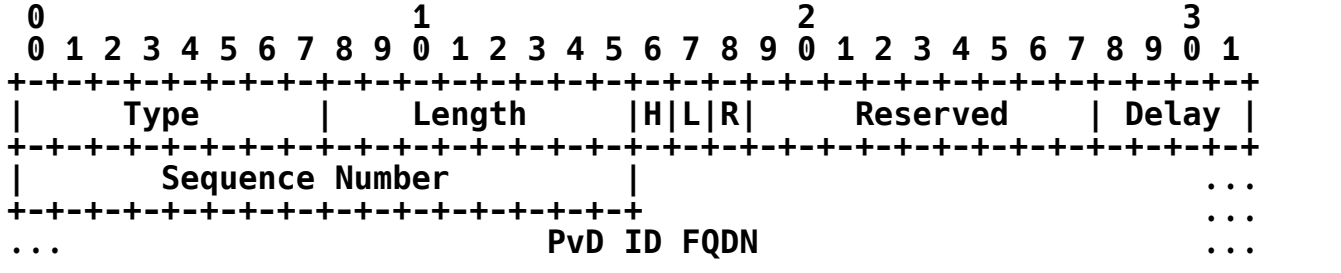
PvD-aware host: A host that supports the association of network configuration information into PvDs and the use of these PvDs as described in this document. Also named "PvD-aware node" in [RFC7556].

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) that identifies the network operator. Network operators MUST use names that they own or manage to avoid naming conflicts. 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 Option for Router Advertisements

This document introduces a Router Advertisement (RA) option called the 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 that would otherwise be valid in the RA.



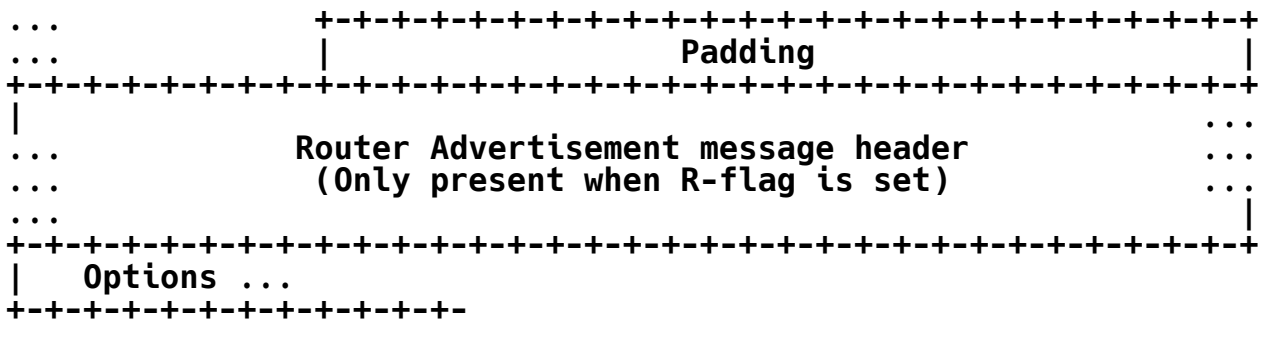


Figure 1: PvD Option Format

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 PvD is associated with IPv4 information assigned using DHCPv4 (see Section 3.4.2).

R-flag: (1 bit) 'Router Advertisement' flag stating whether the PvD Option header is followed (right after padding to the next 64-bit boundary) by a Router Advertisement message header (see Section 4.2 of [RFC4861]). The usage of the inner message header is described in Section 3.4.

Reserved: (9 bits) Reserved for later use. It MUST be set to zero by the sender and ignored by the receiver.

Delay: (4 bits) Unsigned integer used to delay HTTP GET queries from hosts by a randomized backoff (see Section 4.1). If the H-flag is not set, senders SHOULD set the delay to zero, and receivers SHOULD ignore the value.

Sequence Number: (16 bits) Sequence number for the PvD Additional Information, as described in Section 4. If the H-flag is not set, senders SHOULD set the Sequence Number to zero, and receivers SHOULD ignore the value.

PvD ID FQDN: The FQDN used as PvD ID encoded in DNS format, as described in Section 3.1 of [RFC1035]. Domain name compression as 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 field to 134 (the value for "Router Advertisement") and set the Code field to 0. Receivers MUST ignore both of these fields. The Checksum field MUST be set to 0 by the sender; non-zero checksums MUST be ignored by the receiver without causing the processing of the message to fail. 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 so as to be ignored by hosts that are not PvD aware.

Figure 2 shows an example of a PvD Option with "example.org" as the PvD ID FQDN and includes both a Recursive DNS Server (RDNSS) option and a Prefix Information Option. It has a Sequence Number of 123 and indicates the presence of PvD Additional Information that is expected to be fetched with a delay factor of 1.

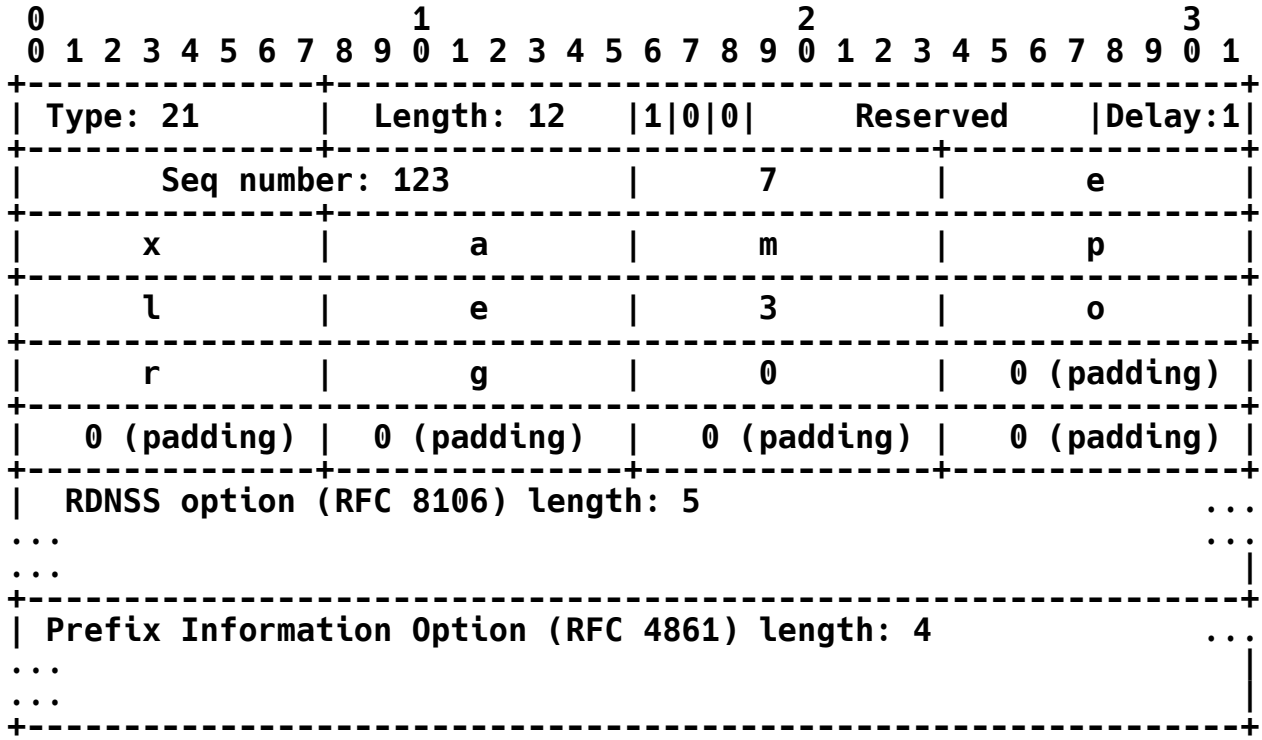


Figure 2: Example PvD Option

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 that would otherwise be valid as part of the same RA. Such options are processed by PvD-aware hosts and ignored by other hosts as per Section 4.2 of [RFC4861].

In order to provide multiple different PvDs, a router MUST send multiple RAs. RAs sent from different link-local source addresses establish distinct Implicit PvDs in the absence of a PvD Option. 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. If a link-local address on the router is changed, then any new RA will be interpreted as a different Implicit PvD by PvD-aware hosts.

As specified in [RFC4861] and [RFC6980], when the set of options causes the size of an advertisement to exceed the link MTU, multiple router advertisements MUST be sent to avoid fragmentation, 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 ensures the backward compatibility required in Section 3.3 of [RFC7556]. This behavior allows for a mixed-mode network where 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], and 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. If an RA message header is present both within the PvD Option and outside it, the header within the PvD Option takes precedence.

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, and DNS Recursive Resolvers) with the PvD associated with the RA that provisioned 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 that 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 performing PvD-specific operations such as resolving names, executing the default address selection algorithm [RFC6724], or executing the default router selection algorithm when forwarding packets [RFC4861] [RFC4191] [RFC8028], hosts and applications MAY consider only the configuration associated with any non-empty subset 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., Internet of Things (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, and the way this selection is enforced are out of the scope of this document. Useful insights about these considerations can be found in [MPVD-API].

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 an RA with the O-flag set [RFC4861].

When a host retrieves stateful assignments using DHCPv6, such assignments MUST be associated with the received PvD that 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 Implicit PvDs 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. Implementations are suggested to flag or log such scenarios as errors to help detect misconfigurations.

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 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 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 in which 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 Option, hosts that are sharing connectivity SHOULD include a PvD Option within the RAs sent downstream with:

* The same PvD ID FQDN

* The same H-flag, Delay, and Sequence Number values

* The L-flag set whenever the host is sharing IPv4 connectivity received from the same upstream interface

* The bits in the Reserved field set to 0

The values of the R-flag, Router Advertisement message header, and Options field depend on whether or not the connectivity should be shared only with PvD-aware hosts (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 Explicit PvD (such as a VPN). In all of these cases, the recursive 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 for 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 a given query. Handling DNS across PvDs is discussed in Section 5.2.1 of [RFC7556], and PvD APIs are discussed in Section 6 of [RFC7556].

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

* A PvD associated with a VPN or otherwise private network may provide DNS answers that contain addresses inaccessible over another PvD. This includes the DNS queries to retrieve PvD Additional Information, which could otherwise send identifying information to the recursive DNS system (see Section 4.1).

* 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 Internet JSON (I-JSON) profile, 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 IoT devices. IoT devices might need only a subset of the information and would in some cases prefer a smaller representation like Concise Binary Object Representation (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". Inversely, hosts MUST NOT do so whenever the H-flag is not set.

Recommendations for how to use TLS securely can be found in [RFC7525].

When a host retrieves the PvD Additional Information, it MUST verify that the TLS server certificate is valid for the performed request, specifically, that a DNS-ID [RFC6125] on the certificate is equal to the PvD ID expressed as an FQDN. This validation indicates that the owner of the FQDN authorizes its use with the prefix advertised by the router. If this validation fails, hosts MUST close the connection and treat the PvD as if it has no Additional Information.

HTTP requests and responses for PvD Additional Information use the "application/pvd+json" media type (see Section 8.5). Clients SHOULD include this media type as an Accept header field in their GET requests, and servers MUST mark this media type as their Content-Type header field in responses.

Note that the DNS name resolution of the PvD ID, any connections made for certificate validation (such as Online Certificate Status Protocol (OCSP) [RFC6960]), and the HTTP request itself 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 exclusively using 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. In order to address privacy concerns around linkability of the PvD HTTP connection with future user-initiated connections, if the host has a temporary address per [RFC4941] in this PvD, then it SHOULD use a temporary address to fetch the PvD Additional Information and MAY 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 close its connection and consider that there is no PvD Additional 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 response is expected to be a single JSON object.

After retrieval of the PvD Additional Information, hosts MUST remember the last Sequence Number value received in an 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.

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.

* 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^((10 + Delay)) milliseconds, where 'Delay' corresponds to the 4-bit unsigned integer in the last received PvD Option.

* When a host last retrieved a JSON object at time A that includes an expiry time B using the "expires" key, and the host is configured to keep the PvD Additional 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 in Figure 2, the Delay field value is 1; this means that the host calculates its delay by choosing a uniformly random time between 0 and 2^((10 + 1)) milliseconds, i.e., between 0 and 2048 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 Number value while changing the Delay value depending on the criticality of the update and the capacity of its PvD Additional Information servers.

In addition to adding a random delay when fetching Additional Information, hosts MUST enforce a minimum time between requesting Additional Information for a given PvD on the same network. This minimum time is RECOMMENDED to be 10 seconds, in order to avoid hosts causing a denial-of-service on the PvD server. Hosts also MUST limit the number of requests that are made to different PvD Additional Information servers on the same network within a short period of time. A RECOMMENDED value is to issue no more than five PvD Additional Information requests in total on a given network within 10 seconds. For more discussion, see Section 6.

The PvD Additional Information object includes a set of IPv6 prefixes (under the key "prefixes") that MUST be checked against all the Prefix Information Options advertised in the RA. If any of the

prefixes included in any associated PIO is not covered by at least one of the listed prefixes, the PvD Additional 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.

If the request for PvD Additional Information fails due to a TLS certificate validation error, an HTTP error, or because the retrieved file does not contain valid PvD JSON, hosts MUST close any connection used to fetch the PvD Additional Information and MUST NOT request the information for that PvD ID again for the duration of the local network attachment. If a host detects 10 or more such failures to fetch PvD Additional Information, the local network is assumed to be misconfigured or under attack and the host MUST NOT make any further requests for any PvD Additional Information, belonging to any PvD ID, for the duration of the local network attachment. For more discussion, see Section 6.

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, certificate validation, 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 MUST wrap back to zero.

The server providing the JSON files SHOULD also check whether the client address is contained by the prefixes listed in 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 which this object is no longer valid	[RFC3339] Date	"2020-05-23T06:00:00Z"
prefixes	Array of IPv6	Array of	["2001:db8:1::/48",

	prefixes valid for this PvD	strings	"2001:db8:4::/48"]

Table 1

A retrieved object that 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 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 that MAY be included in the object.

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

Table 2

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 the keys registered by IANA, implementers or vendors defining private-use or experimental keys MUST create sub-dictionaries. If a set of PvD Additional Information keys are defined by an organization that has a formal URN namespace [IANA-URN], the URN namespace SHOULD be used as the top-level JSON key for the sub-dictionary. For other private uses, the sub-dictionary key SHOULD follow the format of "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". If a host receives a sub-dictionary with an unknown key, the host MUST ignore the contents of the sub-dictionary.

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": "2020-05-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
}

{
  "identifier": "company.foo.example.com.",
  "expires": "2020-05-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "vendor-foo":
    {
        "private-key": "private-value",
    },
}
```

4.4. Detecting Misconfiguration and Misuse

Hosts MUST validate the TLS server certificate when retrieving PvD Additional Information, as detailed in Section 4.1.

Hosts MUST verify that all prefixes in all the RA PIOs 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 IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296] 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 NPTv6 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 NPTv6 was not used and restricts the information to the valid network users.If the PvD does not provision IPv4 (it does not include the L-flag 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 PvDs. For the sake of simplicity, the RA messages will not be described in the usual ASCII art but rather in an indented list. Values in the PvD Option header that are not included in the example are assumed to be zero or false (such as the H-flag, Sequence Number, and Delay fields).

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.

* RA Header: router lifetime = 6000

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

* 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. This approach allows for two distinct sets of configuration information to be sent in a way that will not disrupt non-PvD-aware hosts. It also lowers the risk that a single RA message will approach its MTU limit due to duplicated information.

If two RA messages are sent for this reason, they MUST be sent from two different link-local source addresses (Section 3.2). For example, here is the RA sent for non-PvD-aware hosts:

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

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

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

* 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:

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

* 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 listed RA sent by their default router and use 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 the default router; similarly, they will use the DNS server "2001:db8:f00d::53" when communicating from this address.

5.3. Enabling Multihoming 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 multihome 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.

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

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

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

* 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; hence, only the PvD-aware hosts will be multihomed.

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

* 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]

Note: the above examples assume that the router has received its PvD IDs from upstream routers or via some other configuration mechanism. Another document could define ways for the router to generate its own PvD IDs to allow the above scenario in the absence of PvD ID provisioning.

5.4. Providing Additional Information to PvD-Aware Hosts

In this example, the router indicates that it provides Additional Information using the H-flag. The Sequence Number on the PvD Option is set to 7 in this example.

* RA Header: router lifetime = 6000

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

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

* PvD Option header: length = 3, PvD ID FQDN = cafe.example.com., Sequence Number = 7, R-flag = 0, H-flag = 1 (actual length of the header with padding 24 bytes = 3 * 8 bytes)

A PvD-aware host will fetch <https://cafe.example.com/.well-known/pvd> to get the additional information. The following example shows a GET request that the host sends, in HTTP/2 syntax [RFC7540]:

```
:method = GET
:scheme = https
:authority = cafe.example.com
:path = /.well-known/pvd
accept = application/pvd+json
```

The HTTP server will respond with the JSON Additional Information:

```
:status = 200
content-type = application/pvd+json
content-length = 116

{
  "identifier": "cafe.example.com.",
  "expires": "2020-05-23T06:00:00Z",
  "prefixes": ["2001:db8:cafe::/48"],
}
```

At this point, the host has the PvD Additional Information and knows the expiry time. When either the expiry time passes or a new Sequence Number is provided in an RA, the host will re-fetch the Additional Information.

For example, if the router sends a new RA with the Sequence Number

set to 8, the host will re-fetch the Additional Information:

* PvD Option header: length = 3 + 5 + 4 , PvD ID FQDN = cafe.example.com., Sequence Number = 8, R-flag = 0, H-flag = 1 (actual length of the header with padding 24 bytes = 3 * 8 bytes)

However, if the router sends a new RA, but the Sequence Number has not changed, the host would not re-fetch the Additional Information (until and unless the expiry time of the Additional Information has passed).

6. Security Considerations

Since the PvD Option can contain an RA header and other RA options, any security considerations that apply for specific RA options continue to apply when used within a PvD Option.

Although some solutions such as IPsec or SEcure Neighbor Discovery (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].

If multiple RAs are sent for a single PvD to avoid fragmentation, dropping packets can lead to processing only part of a PvD Option, which could lead to hosts receiving only part of the contained options. As discussed in Section 3.2, routers MUST include the PvD Option in all fragments generated.

This specification does not improve the Neighbor Discovery Protocol security model but simply validates that the owner of the PvD FQDN authorizes its use with the prefix advertised by the router. In combination with implicit trust in the local router (if present), this gives the host some level of assurance that the PvD is authorized for use in this environment. However, when the local router cannot be trusted, no such guarantee is available.

It must be noted that Section 4.4 of this document only provides reasonable assurance against misconfiguration but does not prevent a hostile network access provider from advertising incorrect information that could lead applications or hosts to select a hostile PvD. However, a host that correctly implements the multiple PvD architecture [RFC7556] using the mechanism described in this document will be less susceptible to some attacks than a host that does not by being able to check for the various misconfigurations or inconsistencies described in this document.

Since expiration times provided in PvD Additional Information use absolute time, these values can be skewed due to clock skew or for hosts without an accurate time base. Such time values MUST NOT be used for security-sensitive functionality or decisions.

An attacker generating RAs on a local network can use the H-flag and the PvD ID to cause hosts on the network to make requests for PvD Additional Information from servers. This can become a denial-of-service attack, in which an attacker can amplify its attack by

triggering TLS connections to arbitrary servers in response to sending UDP packets containing RA messages. To mitigate this attack, hosts MUST:

* limit the rate at which they fetch a particular PvD's Additional Information;

* limit the rate at which they fetch any PvD Additional Information on a given local network;

* stop making requests for a PvD ID that does not respond with valid JSON; and

* stop making requests for all PvD IDs once a certain number of failures is reached on a particular network.

Details are provided in Section 4.1. This attack can be targeted at generic web servers, in which case the host behavior of stopping requesting for any server that doesn't behave like a PvD Additional Information server is critical. Limiting requests for a specific PvD ID might not be sufficient if the attacker changes the PvD ID values quickly, so hosts also need to stop requesting if they detect consistent failure when on a network that is under attack. For cases in which an attacker is pointing hosts at a valid PvD Additional Information server (but one that is not actually associated with the local network), the server SHOULD reject any requests that do not originate from the expected IPv6 prefix as described in Section 4.2.

7. Privacy Considerations

Retrieval of the PvD Additional Information over HTTPS requires early communications between the connecting host and a server that 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. To minimize the leakage of identity information while retrieving the PvD Additional Information, hosts SHOULD make use of an IPv6 temporary address and SHOULD NOT include any privacy-sensitive data, such as a User-Agent header field or an HTTP cookie.

Hosts might not always fetch PvD Additional Information, depending on whether or not they expect to use the information. However, if a host allows requesting Additional Information for certain PvD IDs, an attacker could send various PvD IDs in RAs to detect which PvD IDs are allowed by the client. To avoid this, hosts SHOULD either fetch Additional Information for all eligible PvD IDs on a given local network or fetch the information for none of them.

From a user 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 static web sites, such as <http://captive.example.com/hotspot-detect.html>, in order to detect the presence of a captive portal.

The DNS queries associated with the PvD Additional Information MUST

use the DNS servers indicated by the associated PvD, as described in Section 4.1. This ensures the name of the PvD Additional Information server is not unintentionally sent on another network, thus leaking identifying information about the networks with which the client is associated.

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 an allowed list 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 list.

Network operators SHOULD restrict access to PvD Additional Information to only expose it to hosts that are connected to the local network, especially if the Additional Information would provide information about local network configuration to attackers. This can be implemented by allowing access from the addresses and prefixes that the router provides for the PvD, which will match the prefixes contained in the PvD Additional Information. This technique is described in Section 4.2.

8. IANA Considerations

8.1. Change to IPv6 Neighbor Discovery Option Formats Registry

IANA has removed the 'reclaimable' tag for value 21 for the PvD Option in the "IPv6 Neighbor Discovery Option Formats" registry.

8.2. New Entry in the Well-Known URIs Registry

IANA has added a new entry in the "Well-Known URIs" registry [RFC8615] with the following information:

URI suffix: pvd

Change controller: IETF

Specification document: RFC 8801

Status: permanent

Related information: N/A

8.3. New Additional Information PvD Keys Registry

IANA has created and will maintain a new registry called "Additional Information PvD Keys", which reserves JSON keys for use in PvD Additional Information. The initial contents of this registry are given in Section 4.3 (both the table of mandatory keys and the table of optional keys).

The status of a key as mandatory or optional is intentionally not denoted in the table to allow for flexibility in future use cases. Any new assignments of keys will be considered as optional for the

purpose of the mechanism described in this document.

New assignments in the "Additional Information PvD Keys" registry will be administered by IANA through Expert Review [RFC8126]. Experts are requested to ensure that defined keys do not overlap in names or semantics and that they represent non-vendor-specific use cases. Vendor-specific keys SHOULD use sub-dictionaries, as described in Section 4.3.

IANA has placed the "Additional Information PvD Keys" registry within a new registry entitled "Provisioning Domains (PvDs)".

8.4. New PvD Option Flags Registry

IANA has also created and will maintain a new registry entitled "PvD Option Flags". This new registry reserves bit positions from 0 to 11 to be used in the PvD Option bitmask. This document assigns bit positions 0, 1, and 2 as shown in the table below. Future assignments require Standards Action [RFC8126].

Bit	Name	Reference
0	H-flag	RFC 8801
1	L-flag	RFC 8801
2	R-flag	RFC 8801
3-11	Unassigned	

Table 3

Since these flags apply to an IPv6 Router Advertisement Option, IANA has placed this registry under the existing "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry and provided a link on the new "Provisioning Domains (PvDs)" registry.

8.5. 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: N/A

Optional parameters: N/A

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

Security considerations: See Section 6 of RFC 8801.

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

Published specification: RFC 8801

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

Fragment identifier considerations: N/A

Additional information: N/A

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

Intended usage: COMMON

Restrictions on usage: N/A

Author: IETF

Change controller: IETF

9. References

9.1. 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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <https://www.rfc-editor.org/info/rfc6980>.

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

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <https://www.rfc-editor.org/info/rfc7525>.

[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>.

[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>.

[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>.

[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>.

9.2. Informative References

[IANA-URN] IANA, "Uniform Resource Names (URN) Namespaces", <https://www.iana.org/assignments/urn-namespaces/>.

[IEEE8021X] IEEE, "IEEE Standard for Local and Metropolitan Area Networks -- Port-Based Network Access Control", IEEE 802.1X-2020, DOI 10.1109/IEEESTD.2020.9018454, <https://ieeexplore.ieee.org/document/9018454>.

[MPVD-API] Kline, E., "Multiple Provisioning Domains API Requirements", Work in Progress, Internet-Draft, draft-kline-mif-mpvd-api-reqs-00, 1 November 2015, <https://tools.ietf.org/html/draft-kline-mif-mpvd-api-reqs-00>.

[MPVD-DNS] Stenberg, M. and S. Barth, "Multiple Provisioning Domains using Domain Name System", Work in Progress, Internet-Draft, draft-stenberg-mif-mpvd-dns-00, 15 October 2015, <https://tools.ietf.org/html/draft-stenberg-mif-mpvd-dns-00>.

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

[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>.

[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>.

[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>.

[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>.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[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>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[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>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[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>.

[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>.

Acknowledgments

Many thanks to Markus Stenberg and Steven Barth for their earlier work on [MPVD-DNS], as well as to Basile Bruneau, who was author of an early draft 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 for his valuable input and implementation efforts, Tom Jones for his integration effort into the NEAT project, and Rigil Salim for his implementation work.

Authors' Addresses

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

Email: ppfister@cisco.com

Éric Vyncke
Cisco
De Kleetlaan, 6
1831 Diegem
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
92130 Issy-les-Moulineaux
France

Email: wenshao@cisco.com