Support for multiple provisioning domains in DHCPv6

draft-kkb-mpvd-dhcp-support-00

Suresh Krishnan, Jouni Korhonen, Shwetha Bhandari

Goals

- Describe how to associate configuration information with provisioning domains
- Describe a mechanism for identifying provisioning domains
- Describe the authentication and authorization issues with the use of mPVDs

Basic concepts

- The basic construct for compartmentalizing the configuration information per PVD is realized using a container option
  - Encapsulates all configuration information pertaining to a given PVD
  - Multiple PVD containers can occur inside the same DHCPv6 message
    - The PVD identities need to be different though
- Configuration information can still be conveyed without using PVDs

Container option format

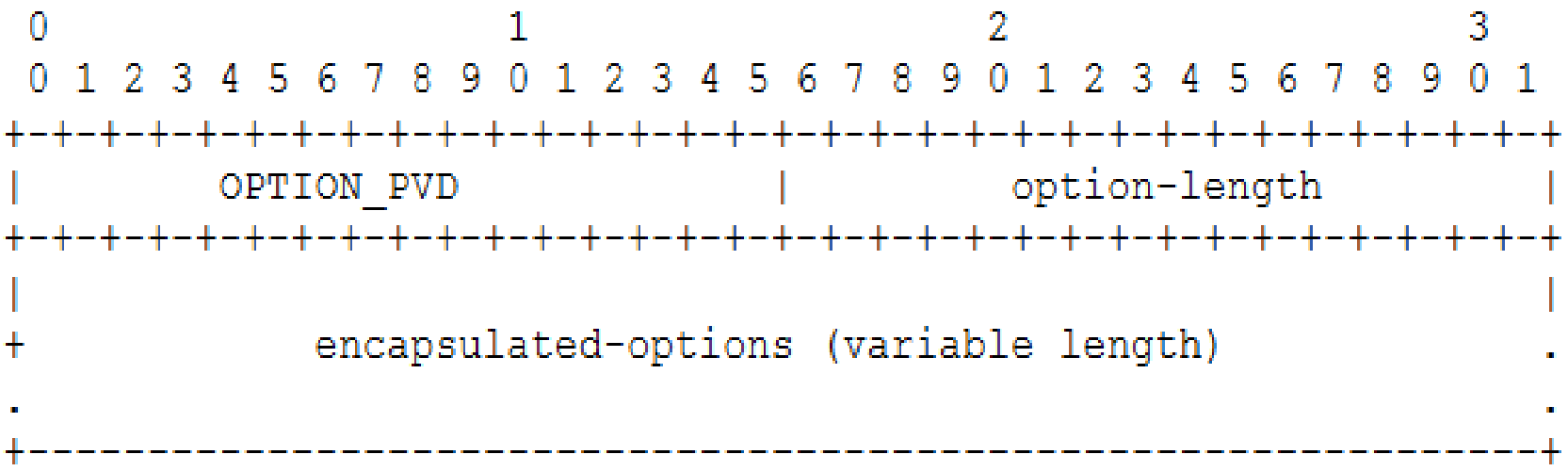


Figure 1: PVD Container Option

Identifying PVDs

- We wanted to have some flexibility on how we identify the PVDs
  - A one-size-fits-all approach didn't seem too likely to be universally acceptable
  - Decided to use a mechanism where we can start of with a few well known types and register new ID types if needed later
- The PVD identity information is carried in a PVD ID option
  - Exactly one PVD ID per PVD container

PVD ID option format

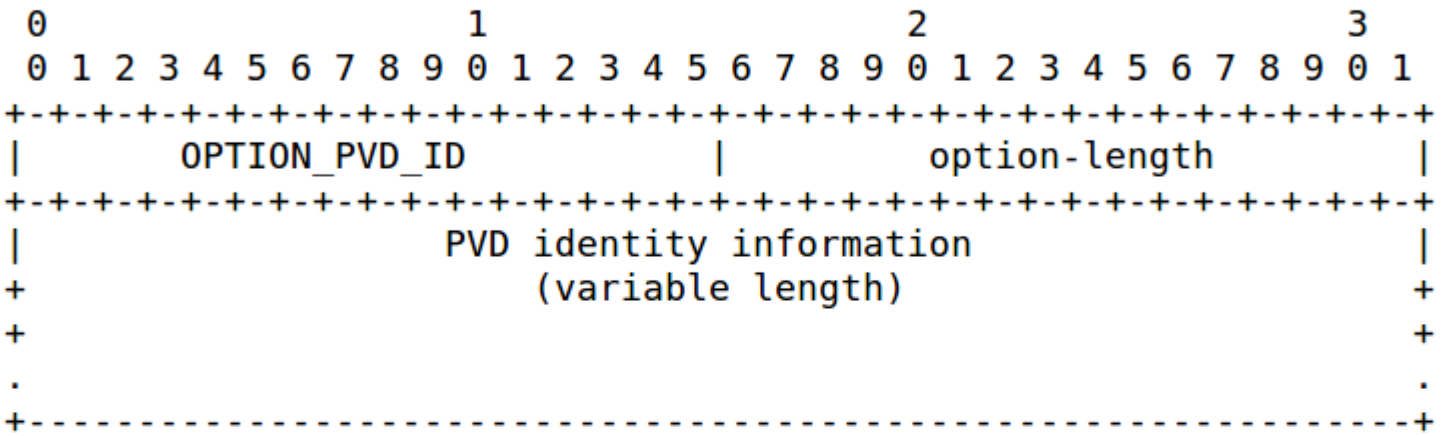


Figure 2: PVD ID Option

- option-code: OPTION_PVD_ID (TBA2)

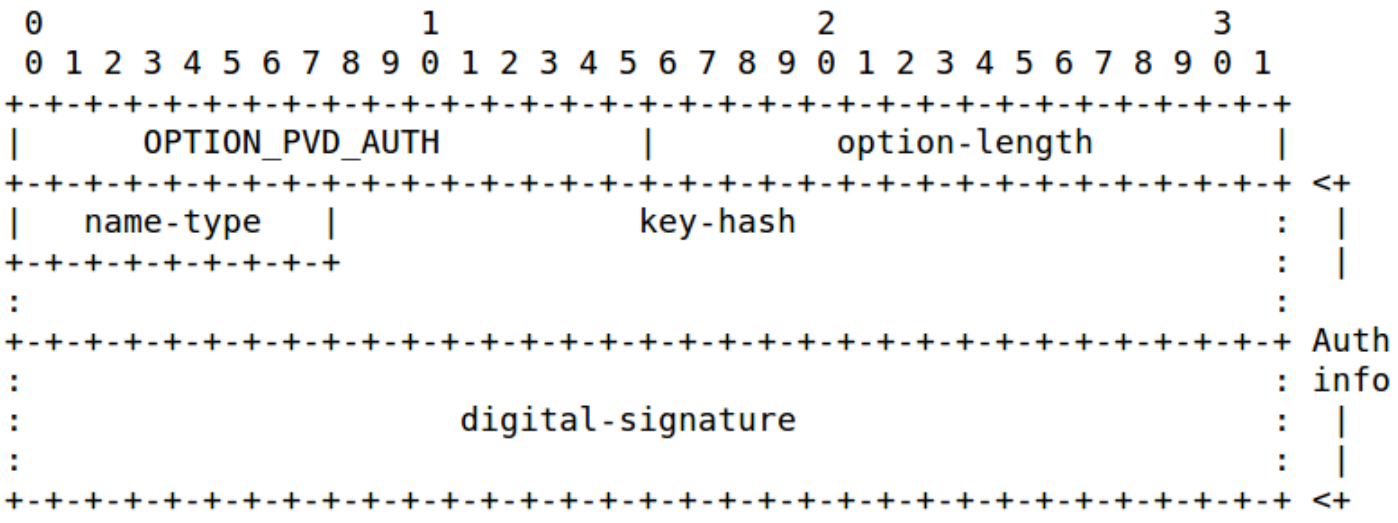
- option-length: Length of PVD identity information

- PVD identity information: The provisioning domain identity. The contents of this field is defined in a separate document [PVDIDS].

Authentication/Authorization

- The PVD Auth option is a mechanism for tying the configuration information inside a container to the *original source* of the information
  - Not for authenticating the configuration source (i.e. the DHCPv6 server)
- Strive to use a common mechanism for DHCPv6 and RA
  - Propose to reuse mechanisms specified for SeND (RFC6494/RFC6495)

PVD Auth option format



- option-code: OPTION_PVD_AUTH (TBA3)

- option-length: Length of the Auth info

- name-type: Names the algorithm used to identify a specific X.509 certificate using the method defined for the Subject Key Identifier (SKI) extension for the X.509 certificates. The usage and the Name Type registry aligns with the mechanism defined for SeND [RFC6494][RFC6495]. Name Type values starting from 3 are supported and an implementation MUST at least support SHA-1 (value 3).

- key-hash: A hash of the public key using the algorithm identified by the Name Type. The procedure how the Key Hash is calculated is defined in [RFC3971] and [RFC6495]

- digital-signature: A signature calculated over the encapsulating OPTION_PVD including all option data from the beginning of the option while setting the digital-signature field to zero. The procedure of calculating the signature is identical to the one defined for SeND [RFC3971].

Features

- Backward compatible
  - Clients indicate support using an ORO
    - Legacy clients will not request this option
    - Legacy servers will ignore option
- Allows clients to request information for selected pvds by including one or more OPTION_PVD_IDs
  - Default is to provide info for all available PVDs

Status

- The draft is pending adoption in the mif working group
- We would highly appreciate comments and suggestions from the dhc community