

MPVD (Multiple ProVisioning Domain) architecture status (draft-anipko-mif-mpvd-arch)

Dmitry Anipko (editor)

Cao Zhen (presenter)

IETF 87 MIF WG

Draft progress so far

- Great MIF-ARCH-DT work on several teleconfs since IETF 86!
- draft-anipko-mif-mpvd-arch-00 published before the cut-off date
- -02 just published at the IETF opening, please read and comment

Definition: ProVisioning Domain (PVD)

- A consistent set of network configuration information.
- PVD consists of PVD elements: IP/DNS/DHCP/Route/...
- Classically, all info available on 1 interface is provided by a single source (e.g., net admin) and can be treated as 1 PVD
- Multihoming can result in >1 PVDs on a single link
- PVD can span multiple interfaces and links
- PVDs have IDs, the exact form is still TBD
- Dual-stack PVDs are supported
 - Both address families are included in the same PVD by default

“Explicit” vs “Implicit” PVDs

- **Explicit:** created when a node receives explicit information about PVDs and associated info
- **Implicit:** automatically created for configuration received on different interfaces, when connected to networks that don't advertise explicit PVD info

Relationship to existing processes discussed

- Next-hop and source address selection (RFC 6724)
- If start with name resolution
 - By default send data using same PVD as where answer to name lookup came from
 - What component ensures using same PVD as name resolution used? Could be app but better if under a platform API
- App might do something other than name resolution
 - App might want to specify a specific PVD

Rough framework for impact on APIs

- Basic, intermediate, advanced for different type of apps / scenarios
 - Basic: applications are not PVD-aware in any manner, and only submit connection requests.
 - Intermediate: applications indirectly participate in selection of PVD by specifying hard requirements and soft preferences.
 - Advanced: PVDs are directly exposed to applications, for enumeration and selection

Model for trust to PVDs

- Trust = Authentication of PVD ID + Relationship with source (Authorization)
 - Authentication explicit or by attachment
- Untrusted and Trusted PVDs
 - Trusted: node has some strong trust in source of information (e.g., admin config, SeND, secure DHCP)
 - Untrusted: info is a hint from an untrusted source (e.g., RA on open network)

Potential next items for elaboration

- Cover a list of PVD elements
- Discussion of design approach for PVD ID and information elements associations:
 - Solely DHCP based
 - DHCP + per-layer PVD information (e.g., ND options)
 - Something else?
- Connectivity tests – how they are done and how the results can/shall be used
- Volunteers and text suggestions welcome!

Some issues raised in the discussions

- 1 implicit PVD per interface – can we do better than that? If yes, how?
- Should app configuration be part of PVDs? What are the scenarios?
- How should one deal with multiple PVDs? (merge, select, ...?)
- How should one deal with PVDs with incomplete information?