MIF Working Group
Internet-Draft
Intended status: Informational
Expires: January 22, 2014

D. Anipko
Microsoft Corporation

July 23, 2013

Multiple Provisioning Domain Architecture
draft-anipko-mif-mpvd-arch-01

Abstract

This document is a product of the work of MIF architecture design team. It outlines a solution framework for some of the issues, experienced by nodes that can be attached to multiple networks. The framework defines the notion of a Provisioning Domain (PVD) - a consistent set of network configuration information, and PVD-aware nodes - nodes which learn PVDs from the attached network(s) and/or other sources and manage and use multiple PVDs for connectivity separately and consistently.

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 http://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 January 22, 2014.

Table of Contents

1. Introduction

Nodes attached to multiple networks may encounter problems due to conflict of the networks configuration and/or simultaneous use of the multiple available networks. While existing implementations apply various techniques ([RFC6419]) to tackle such problems, in many cases the issues may still appear. The MIF problem statement document [RFC6418] describes the general landscape as well as discusses many specific issues details.

Across the layers, problems enumerated in [RFC6418] can be grouped

into 3 categories:

1. Lack of consistent and distinctive management of configuration elements, associated with different networks.

2. Inappropriate mixed use of configuration elements, associated with different networks, in the course of a particular network activity / connection.

3. Use of a particular network, not consistent with the intent of the scenario / involved parties, leading to connectivity failure and / or other undesired consequences.

As an illustration: an example of (1) is a single node-scoped list of DNS server IP addresses, learned from different networks, leading to failures or delays in resolution of name from particular namespaces; an example of (2) is use of an attempt to resolve a name of a HTTP proxy server, learned from a network A, with a DNS server, learned from a network B, likely to fail; an example of (3) is a use of employer-sponsored VPN connection for peer-to-peer connections, unrelated to employment activities.

This architecture describes a solution to these categories of problems, respectively, by:

1. Introducing a formal notion of the PVD, including PVD identity, and ways for nodes to learn the intended associations among acquired network configuration information elements.

2. Introducing a reference model for a PVD-aware node, preventing inadvertent mixed use of the configuration information, which may belong to different PVDs.

3. Providing recommendations on PVD selection based on PVD identity and connectivity tests for common scenarios.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Definitions and types of PVDs

Provisioning Domain: a consistent set of network configuration information. Classically, the entire set available on a single interface is provided by a single source, such as network administrator, and can therefore be treated as a single provisioning domain. In modern IPv6 networks, multihoming can result in more than one provisioning domain being present on a single link. In some

scenarios, it is also possible for elements of the same domain to be present on multiple links.

Typical examples of information in a provisioning domain, learned from the network, are: source address prefixes, to be used by connections within the provisioning domain, IP address of DNS server, name of HTTP proxy server if available, DNS suffixes associated with the network etc.

In some cases, other sources, such as e.g., node local policy, user input or other out of band mechanisms may be used to either construct a PVD entirely (analogously to static IP configuration of an interface), or supplement with particular elements all or some PVDs learned from the network.

As an example, node administrator could inject a not ISP-specific DNS server into PVDs for any of the networks the node could become attached to. Such creation / augmentation of PVD(s) could be static or dynamic. The particular implementation mechanisms are outside of the scope of this document.

PVD-aware node: a node that supports association of network configuration information into PVDs, and using the resultant PVDs to serve requests for network connections in a way, consistent with recommendations of this architecture.

2.1. Explicit and implicit PVDs

A node may receive explicit information from the network and/or other sources, about presence of PVDs and association of particular network information with a particular PVD. PVDs, constructed based on such information, are referred to in this document as "explicit".

Protocol changes/extensions will likely be required to support the explicit PVDs. As an example, one could think of one or several DHCP options, defining a PVD identity and elements. A different approach could be to introduce a DHCP option, which only introduces identity of a PVD, while the association of network information elements with that identity, is implemented by the respective protocols - such as e.g., with a Router Discovery [RFC4861] option declaring association of an address range with a particular PVD.

Specific, existing or new, features of networking protocols to enable delivery of PVD identity and association with various network information elements will be defined in companion design documents.

It is likely that for a long time there may be networks which do not advertise any explicit PVD information, since deployment of any new features in networking protocols is a relatively slow process. When connected to such networks, PVD-aware nodes may still provide benefits to their users, compared to non-PVD aware nodes, by creating separate PVDs for configuration received on different interfaces.

Such PVDs are referred to in this document as "implicit". This allows the node to manage and use network information from different

interfaces separately and consistently use the configuration to serve network connection requests.

In the mixed mode, where e.g. multiple networks are available on the link the interface is attached to, and only some of the networks advertize PVD information, the PVD-aware node shall create explicit PVDs based on explicitly learned PVD information, and associate the rest of the configuration with an implicit PVD created for that interface.

It shall be possible for networks to communicate that some of their configuration elements could be used within a context of other networks/PVDs. Based on such declaration and their policies, PVD-aware nodes may choose to inject such elements into some or all other PVDs they connect to.

2.2. Relationship between PVDs and interfaces

Implicit PVDs are limited to network configuration information received on a single interface. Explicit PVDs, in practice will often also be scoped to a configuration related to a particular interface, however per this architecture there is no such requirement or limitation and as defined in this architecture, explicit PVDs may include information related to more than one interfaces, if the node learns presence of the same PVD on those interfaces and the authentication of the PVD ID meets the level required by the node policy.

2.3. PVD identity/naming

For explicit PVDs, PVD ID (globally unique ID, that possibly is human-readable) is received as part of that information. For implicit PVDs, the node assigns a locally generated globally unique ID to each implicit PVD.

PVD-aware node may use these IDs to choose a PVD with matching ID for special-purpose connection requests, in accordance with node policy or choice by advanced applications, and/or to present human-readable IDs to the end-user for selection of Internet-connected PVDs.

2.4. Relationship to dual-stack networks

When applied to dual-stack networks, the PVD definition allows for multiple PVDs to be created, where each PVD contain information for only one address family, or for a single PVD that contains information about multiple address families. This architecture requires that accompanying design documents for accompanying protocol changes must support PVDs containing information from multiple address families. PVD-aware nodes must be capable of dealing with both single-family and multi-family PVDs.

Nevertheless, for explicit PVDs, the choice of either of the approaches is a policy decision of a network administrator and/or node user/administrator. Since some of the IP configuration information that can be learned from the network can be applicable to multiple address families (for instance DHCP address selection option [I-D.ietf-6man-addr-select-opt]), it is likely that dual-stack networks will deploy single PVDs for both address families.

For implicit PVDs, by default PVD-aware nodes shall including multiple IP families into single implicit PVD created for an interface.

A PVD-aware node that provides API to use / enumerate / inspect PVDs and/or their properties shall provide ability to filter PVDs and/or their properties by address family.

2.5. Elements of PVD

3. Example network configurations and number of PVDs

4. Reference model of PVD-aware node

4.1. Constructions and maintenance of separate PVDs

4.2. Consistent use of PVDs for network connections

PVDs enable PVD-aware nodes to use consistently a correct set of configuration elements to serve the specific network requests from beginning to end. This section describes specific examples of such consistent use.

4.2.1. Name resolution

When PVD-aware node needs to resolve a name of the destination used by a connection request, the node could decide to use one, or multiple PVDs for a given name lookup.

The node shall chose one PVD, if e.g., the node policy required to use a particular PVD for a particular purpose (e.g. to download an MMS using a specific APN over a cellular connection). To make the choice, the node could use a match of the PVD DNS suffix or other form of PVD ID, as determined by the node policy.

The node may pick multiple PVDs, if e.g., they are general purpose PVDs providing connectivity to the Internet, and the node desires to maximize chances for connectivity in Happy Eyeballs style. In this case, the node could do the lookups in parallel, or in sequence. Alternatively, the node may use for the lookup only one PVD, based on the PVD connectivity properties, user choice of the preferred

Internet PVD, etc.

In either case, by default the node uses information obtained in a name service lookup to establish connections only within the same PVD from which the lookup results were obtained.

For simplicity, when we say that name service lookup results were obtained from a PVD, what we mean is that the name service query was issued against a name service the configuration of which is present in a particular PVD. In that sense, the results are "from" that particular PVD.

4.2.2. Next-hop and source address selection

For the purpose of this discussion, let's assume the preceding name lookup succeeded in a particular PVD. For each obtained destination address, the node shall perform a next-hop lookup among routers, associated with that PVD. As an example, such association could be determined by the node via matching the source address prefixes/ specific routes advertized by the router against known PVDs, or receiving explicit PVD affiliation advertized through a new Router Discovery [RFC4861] option.

For each destination, once the best next-hop is found, the node selects best source address according to the [RFC6724] rules, but with a constraint that the source address must belong to a range associated with the used PVD. If needed, the node would use the prefix policy from the same PVD for the best source address selection among multiple candidates.

When destination/source pairs are identified, then they are sorted using the [RFC6724] destination sorting rules and the prefix policy table from the used PVD.

4.3. Connectivity tests

4.4. Relationship to interface management and connection managers

5. PVD support in APIs

In all cases changes in available PVDs must be somehow exposed, appropriately for each of the approaches.

5.1. Basic

Applications are not PVD-aware in any manner, and only submit connection requests. The node performs PVD selection implicitly, without any otherwise applications participation, and based purely on node-specific administrative policies and/or choices made by the user in a user interface provided by the operating environment, not by the application.

5.2. Intermediate

Applications indirectly participate in selection of PVD by specifying hard requirements and soft preferences. The node performs PVD selection, based on applications inputs and policies and/or user preferences. Some / all properties of the resultant PVD may be exposed to applications.

5.3. Advanced

PVDs are directly exposed to applications, for enumeration and selection. Node polices and/or user choices, may still override the application preferences and limit which PVD(s) can be enumerated and/or used by the application, irrespectively of any preferences which application may have specified. Depending on the implementation, such restrictions, imposed per node policy and/or user choice, may or may not be visible to the application.

6. PVD-aware nodes trust to PVDs

6.1. Untrusted PVDs

Implicit and explicit PVDs for which no trust relationship exists are considered untrusted. Only PVDs, which meet the requirements in Section 6.2, are trusted; any other PVD is untrusted.

In order to avoid various forms of misinformation that can be asserted when PVDs are untrusted, nodes that implement PVD separation cannot assume that two explicit PVDs with the same identifier are actually the same PVD. A node that did make this assumption would be vulnerable to attacks where for example an open Wifi hotspot might assert that it was part of another PVD, and thereby might draw traffic intended for that PVD onto its own network.

Since implicit PVD identifiers are synthesized by the node, this issue cannot arise with implicit PVDs.

Mechanisms exist (for example, [RFC6731]) whereby a PVD can provide configuration information that asserts special knowledge about the reachability of resources through that PVD. Such assertions cannot be validated unless the node has a trust relationship with the PVD; assertions of this type therefore must be ignored by nodes that receive them from untrusted PVDs. Failure to ignore such assertions could result in traffic being diverted from legitimate destinations to spoofed destinations.

6.2. Trusted PVDs

Trusted PVDs are PVDs for which two conditions apply. First, a trust relationship must exist between the node that is using the PVD configuration and the source that provided that configuration; this

is the authorization portion of the trust relationship. Second, there must be some way to validate the trust relationship. This is the authentication portion of the trust relationship. Two mechanisms for validating the trust relationship are defined.

6.2.1. Authenticated PVDs

One way to validate the trust relationship between a node and the source of a PVD is through the combination of cryptographic authentication and an identifier configured on the node. In some cases, the two could be the same; for example, if authentication is done with a shared secret, the secret would have to be associated with the PVD identifier. Without a (PVD Identifier, shared key) tuple, authentication would be impossible, and hence authentication and authorization are combined.

However, if authentication is done using some public key mechanism such as a TLS cert or DANE, authentication by itself isn't enough, since theoretically any PVD could be authenticated in this way. In addition to authentication, the node would need to be configured to trust the identifier being authenticated. Validating the authenticated PVD name against a list of PVD names configured as trusted on the node would constitute the authorization step in this case.

6.2.2. PVDs trusted by attachment

In some cases a trust relationship may be validated by some means other than described in Section 6.2.1, simply by virtue of the connection through which the PVD was obtained. For instance, a handset connected to a mobile network may know through the mobile network infrastructure that it is connected to a trusted PVD, and whatever mechanism was used to validate that connection constitutes the authentication portion of the PVD trust relationship. Presumably such a handset would be configured from the factory, or else through mobile operator or user preference settings, to trust the PVD, and this would constitute the authorization portion of this type of trust relationship.

7. Acknowledgements

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

All drafts are required to have a security considerations section.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative References

[I-D.ietf-6man-addr-select-opt]
Matsumoto, A., Fujisaki, T. and T. Chown, "Distributing Address Selection Policy using DHCPv6", Internet-Draft draft-ietf-6man-addr-select-opt-10, April 2013.

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

[RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011.

[RFC6419] Wasserman, M. and P. Seite, "Current Practices for Multiple-Interface Hosts", RFC 6419, November 2011.

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

[RFC6731] Savolainen, T., Kato, J. and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, December 2012.

Author's Address

Dmitry Anipko
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

Phone: +1 425 703 7070
Email: dmitry.anipko@microsoft.com