Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Le Pape
S. Bhandari
Cisco Systems
I. Farrer
Deutsche Telekom AG

July 15, 2013

IPv6 Prefix Meta-data and Usage
draft-lepape-6man-prefix-metadata-00

Abstract

This document presents a method for applications to influence the IPv6 source selection algorithm used by the IP stack in a host. To do so, IPv6 prefixes are associated with meta-data when configured by the network. This meta-data allows the network to describe the purpose and properties of the prefix enabling applications to make intelligent decision when selecting a prefix.

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 RFC2119 [RFC2119].

Status of This Memo

Table of Contents

1. Introduction

IPv6 provides not only a larger address space than IPv4, but also allows host interfaces to have more than one IPv6 address of the same or different scope(s). When multiple prefixes are assigned to one or more network interfaces each of the prefixes can have a specific property and purpose associated with it. For example: In a mobile network, a mobile device can be assigned a prefix from its home network and another from the visiting network that it is currently attached to. Another example is a public WLAN hotspot configured with two prefixes offering Internet access. One is free, but low-quality, whilst the other is charged and offers service level guarantees.

A prefix may have well defined properties that are universal and have additional meta-data associated with it in order to communicate the prefixes local significance. When multiple prefixes are provisioned to the host, this additional information allows the host and applications to make more intelligent decisions about the best IPv6 address to select when sourcing connections.

This document introduces the motivations and considerations for having additional meta-data associated with a prefix and also proposes a format for the meta-data itself.

The underlying assumption is that a endpoint or an application has multiple prefixes to choose from. Typically this means either the endpoint has multiple interfaces or an interface has been configured with multiple prefixes. This specification does not make a distinction between these alternatives.

1.1. Motivation

In this section, the motivation for attaching meta-data to IPv6 prefixes is described in the context of both mobile and home networks. The meta-data helps to distinguish an IPv6 prefix and aids with the selection of the prefix by different applications.

1.1.1. Home networks

In a fixed network environment, the homenet CPE may also function as both a DHCPv6 client (requesting IA_PD(s)) and a DHCPv6 server allocating prefixes from delegated prefix(es) to downstream home network hosts. Some service providers may wish to delegate multiple globally unique prefixes to the CPE for use by different services classes and traffic types.

Motivations for this include:

- Using source prefix to identify the service class / traffic type that is being transported. The source prefix may then reliably be used as a parameter for differentiated services or other purposes. E.g. [I-D.jiang-v6ops-semantic-prefix]

- Using the specific source prefix as a host identifier for other services.

- In multi-homed environments, a single homenet LAN may have multiple globally unique prefixes provided by the different service providers. In this scenario, correct source address selection is fundamental to successfully establishing connections. E.g. [I-D.troan-homenet-sadr]

Any host which is configured with multiple prefixes must perform a source address selection process when initiating a connection. Any client that has multiple globally unique prefixes only has source and destination longest-prefix matching policy [RFC6724] in order to make this selection. For cases such as those listed above, longest-prefix matching can not assist the client in selecting the correct source address to use. Addition information is needed to assist the client in making the correct source address selection.

1.1.2. **Mobile networks**

In mobile network architecture, a mobile node can be associated with multiple IPv6 prefixes belonging to different domains. E.g. home address prefix, care of address prefix (as specified in [RFC3775]). The delegated prefixes may be topologically local and some may be remote prefixes anchored on a global anchor, but available to the local anchor by means of tunnel setup in the network between the local and global anchor. Some prefixes may be local with low latency characteristics suitable for voice call break-out, some may have global mobility support.

So, the prefixes have different properties and it is necessary for the application using the prefix to learn about this property in order to use it intelligently. An example is determining if the prefix is a home address or care of address or other network characteristics that can be offered.

2. Overview

The mechanism that is described in this document describes two different types of meta data which can be used in different ways:

Prefix Properties	Provides a method for an application to "hint" required source address properties to the kernel. These properties are universal and expressed as a set of flags.
Prefix Color	Provides an arbitrary color value to prefixes (of local significance) enabling an application to request a source prefix with a specific color.

These two meta data types are described in more detail below.

Prefix Properties functions as follows:

- o The client receives multiple prefixes, with relevant Prefix Property meta-data attached to each prefix

- o Prefix property aware applications running on the client have a policy defining that they prefer prefixes that have specific properties.

- o On initiating a connection, the Prefix Property aware application passes the required prefix properties to the kernel along with the connect request

- o The kernel checks the requested properties against the available prefixes. If a match is found, the matching prefix is passed back to the application

- o The application uses the returned prefix when making the call to the socket API to create the connection

- o If no prefix matching the requested properties is available, then the kernel uses [RFC6724] for source address selection as normal

Prefix property offers well defined universally understood information about the prefix. Example properties include whether a prefix can provide Internet reachability, if the prefix offers application specific Internet service level, if the prefix usage is free/charged, if the prefix offers security guarantees etc. This is maintained as a global registry.

Prefix Coloring functions as follows:

- The client receives multiple prefixes, with relevant meta-data attached

- Color aware applications running on the client are provisioned with policy telling them which prefix color to request

- On initiating a connection, the meta data aware application passes the required prefix color to the kernel along with the connect request

- The kernel takes this color and selects the prefix matching the requested color and passes this back to the application

- The application uses the returned prefix when making the call to the socket API to create the connection

Prefix colour conveys information of the prefix that is of relevance to the network where the prefix is provisioned and application using it. Example usage of prefix color include color that is provisioned to offer better video application experience. The prefix color is defined as a 16 bit numerical value.

Figure 1 illustrates a typical network with different components that can add, understand and use the meta-data attached to a prefix.

- Mobile or ISP Network - Provisioned with prefixes that offer specific network characteristic. e.g. prefixes that do not have internet reach but can offer quality of service required for better video application experience. Includes address delegation server that associate prefixes with this information, selects and offers this information during prefix delegation

- Home/Mobile gateway - Learns or determines characteristic of the prefix and propagates it along with prefix delegation. e.g. Determines if the prefix is locally anchored or learns the prefix meta-data from the ISP prefix delegation server and includes this information in prefix delegation to endpoints

- Endpoint network stack - Learns the additional information associated with the prefix and offers interface to applications for listing and selecting the available prefixes

- Prefix selection policy - Either embedded in the application/ endpoint or learnt from a server that helps choose the prefix with specific characteristic for the application based on predetermined service agreement between the application/endpoint/application service provider and network service provider

   o Applications - That can utilize the prefix with specific
      characteristic for enhanced application user experience e.g. On
      demand video application, by choosing the prefix with appropriate
      prefix selection policy while connecting and delivering the
      application over the network

This prefix meta-data could be further extended to have more attributes such as the administrative domain of the prefix.

```
         +-----------------------+        +-------------------------+
         |                       |        |                         |
         |     Application       |        |                         |
         |       prefix          |        |    ISP 1, ..., n        |
         |       policy          |        |                         |
         |                       |        |                         |
         +-----------------------+        +-------------------------+
                    :                             |       |
                    :                             |       |
                    :                             |---n---|
              +--------------+                    |       |
              |  Endpoint    |                    |       |
              | application  |                    |       |
              +- - - - - - - +               +------------------+
              |  Endpoint    |               |                  |
              |  networking  |---------------|  Home Gateway/   |
              |   stack      |               | Mobile Gateway   |
              +--------------+               +------------------+
```

Figure 1

3. Considerations

3.1. Prefix meta-data propogation

The prefix meta-data can be delivered using DHCPv6 prefix delegation and address allocation as elaborated in [I-D.bhandari-dhc-class-based-prefix] or via IPv6 Neighbour discovery (ND) as defined in [I-D.korhonen-6man-prefix-properties].

3.2. Configuring Applications

Applications supporting multiple prefixes obtain the prefixes from the host kernel along with their meta-data.

The policy can then be contained either locally (e.g. If the application is intended only for use within a specific network, linked to a particular ISP comes prepackages with prefix color to

use), or be contained on a remote policy server. The mechanism used to exchange the meta-data information and selection between application/host with a remote server is beyond the scope of this document.

3.3. Application to network stack communication

Once an application has determined the appropriate property and color for its use it has to communicate with the network stack to select the prefix. The host internal data structures need to be extended with the 'prefix property' and the 'prefix color' information associated to the learnt prefix and configured addresses. How this is accomplished is host implementation specific. It is also a host implementation issue how an application can learn or query both properties and color of an address or a prefix. One possibility is to provide such information through the socket API extensions. Other possibilities include the use of e.g., ioctl() or NetLink [RFC3549] extensions or by using the IPv6 address scope [RFC4007].

Discussion point: Should prefix property and color be mutually exclusive? This would avoid complexities which takes precedence when one prefix matches color and another matches property. Possibly a prefix may be advertised with both, but the application can only request property or color.

3.4. Default Address Selection

[RFC6724] provides a mechanism for selecting which source address to use, in the absence of an application or upper layer protocol's explicit choice of a legal destination or source address.

The use of prefix meta-data allows an application to express property preferences through socket API extensions, meaning that when used for creating a socket, [RFC6724] source address selection is not required.

If a higher layer protocol or application does not include a prefix property preference when making a create socket request, then source address selection according to [RFC6724] is followed as normal.

3.5. Scope of Prefix Color

Since a home can be connected to multiple ISPs, it is possible that it receives multiple prefixes with the same color from different ISPs. Since the application coloring policy is not received with the color, multiple ISPs may use different coloring policy for a single color. For example: One ISP could use color 50 for video whilst a second ISP is using color 50 for audio.

This section presents some alternatives to handle this problem.

3.5.1. Local scoping

A locally scoped color is a value which is selected by the network and application providers with no central registry. In a multi homed network, this may result in two providers selecting the same color for different behaviors. A color translation could be performed to ensure unique color at the device that connects to multiple providers.

3.5.2. Local scoping with fuzzy matching

To avoid having to maintain multiple colors for each prefix for translating the color, a specific algorithm can be used to determine the new color from the old one on conflict.

For example, when a collision is detected, the new color value may be incremented. Further, colors could be defined to be equally spaced (e.g., 10s or 100s).

Many other encodings are possible as well, as long as obtaining the original color communicated by the ISP may be recovered in the event the application policy server requires this.

3.5.3. Global scoping

A globally scoped color avoids the need for responding to collisions. This can be achieved by disambiguating the color by attaching the domain that provisions the color to the prefix meta-data or by assigning colors from a global registry that comes with the overhead of maintaining such a registry.

3.6. Compatibility with Existing Implementations

The prefix meta-data mechanism that is described in this document provides a way of improving source address selection over the longest-prefix matching method used by [RFC6724].

However, all IPv6 capable hosts deployed at the time of writing do not have the capability of understanding and processing prefix meta-data. This means that any new mechanism must be backwards compatible with existing implementations. Also, clients which understand prefix meta-data need to support applications which do not have meta-data awareness.

There are a number of possible approaches that could be taken here. The following list is included as ideas for further development:

- In DHCPv6 only clients which request prefixes with meta-data (e.g. signalled through OPTION_ORO in the IA_NA or IA_PD request) will receive them.

- In case of prefix delegated using IPv6 Neighbour discovery (ND) both forms of prefix i.e with and without meta-data can be offered.

- If an application makes a socket API request and does not include meta-data as part of the request, follow [RFC6724] source address selection, but remove any prefixes that have meta-data from the list of candidate addresses. It follows that there should be a GU prefix advertised that does not have any meta-data associated that would act as the default choice for non prefix meta-data aware clients and applications.

4. IANA Considerations

Should the global scoping for prefix color be chosen, a new registry should be created by IANA to store colors.

5. Security Considerations

TBD

6. Acknowledgements

The authors would like to acknowledge review and guidance received from

7. Change History (to be removed prior to publication as an RFC)

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[I-D.bhandari-dhc-class-based-prefix]
Systems, C., Halwasia, G., Gundavelli, S., Deng, H., Thiebaut, L., and J. Korhonen, "DHCPv6 class based prefix", draft-bhandari-dhc-class-based-prefix-04 (work in progress), February 2013.

[I-D.ietf-dhc-dhcpv4-over-ipv6]

Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-06 (work in progress), March 2013.

[I-D.jiang-v6ops-semantic-prefix]
Jiang, S., Sun, Q., Farrer, I., and Y. Bo, "A Framework for Semantic IPv6 Prefix", draft-jiang-v6ops-semantic-prefix-03 (work in progress), May 2013.

[I-D.korhonen-6man-prefix-properties]
Korhonen, J., Patil, B., Gundavelli, S., Seite, P., and D. Liu, "IPv6 Prefix Properties", draft-korhonen-6man-prefix-properties-02 (work in progress), July 2013.

[I-D.troan-homenet-sadr]
Troan, O. and L. Colitti, "IPv6 Multihoming with Source Address Dependent Routing (SADR)", draft-troan-homenet-sadr-00 (work in progress), February 2013.

[RFC3549] Salim, J., Khosravi, H., Kleen, A., and A. Kuznetsov, "Linux Netlink as an IP Services Protocol", RFC 3549, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, March 2005.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

Appendix A. Prototype notes

A.1. Homenet prototype implementation notes

This section provides the implementation details of a prototype video application on Android for a Galaxy Nexus device developed for the home network.

A.1.1. Video provider service

A possible use of this prefix coloring is a video service, which requires the network to guarantee a minimal throughput for streaming video. A prefix could be colored by the ISP to indicate that traffic sourced from that prefix will have a certain service level. Using prefix coloring avoids having to set up a separate network for this usage, or implement QoS traffic identification, classification and marking.

An agreement could then be established between the video service provider and the ISP, telling the video provider to use the specific color when streaming video. In the following example, the color 50 was used.

A.1.2. Prefix Color delegation

The CPE routers request prefixes using prefix delegation [RFC3633] with the OPTION_PREFIX_CLASS option [I-D.bhandari-dhc-class-based-prefix]. This informs the upstream provider that the CPE supports colored prefixes. If an ISP does not support this option, it will be ignored, and the CPE will only get colorless prefixes. Otherwise, the ISP returns multiple prefixes each with their associated color. A color of '0' is identical to an uncolored prefix, for application compatibility, as explained in Appendix A.1.5. If the CPE does not support colored prefixes, the ISP could decide to delegate a normally colored prefix as an colorless one, though this means hosts will use this prefix according to the default source address selection algorithm, and will not associate any meaning to it.

Once the CPE receives those prefixes, it distributes them, along with their color, using OSPF and the homenet protocols. [I-D.troan-homenet-sadr] defines "Source Address Dependent Routing" (SADR) which ensures that packets are routed based on their destination as well as source address. SADR is necessary to ensure that a multihomed network using provider aggregatable addresses will send the packet out the right path to avoid violating the provider's ingress filtering.To ensure that those prefixes keep their meaning, Source Address Dependent Routing [I-D.troan-homenet-sadr] is implemented and used.

Colored addresses are advertised to hosts through DHCPv6, to associate the color to the address. Colorless addresses may be distributed through DHCPv6 or through Router Advertisements. Hosts supporting colored prefixes include the OPTION_PREFIX_CLASS, and receive colored addresses. For legacy hosts, who do not include this option, there are two possibilities :

- o Those hosts can receive all available prefixes, including colored ones, as uncolored. This allows a legacy host in a fully colored homenet to still have access to IPv6. However, those hosts may use prefixes for the wrong purposes.

- o Those hosts can receive only colorless prefixes. This ensures that a prefix will not be used for the wrong purpose. However, hosts in a fully colored environment will not get access to IPv6. This can however be what the ISP originally intended, for example if the ISP does not provide access to the IPv6 Internet, but uses IPv6 for wall gardened services, which their specific devices know how to use.

A.1.3. Configuring Applications

Applications supporting multiple prefixes obtain the prefixes from the host kernel, along with their color.

The policy can be contained either in a local database (e.g. If the application is intended only for use within a specific network, linked to a particular ISP), or be contained on a distant server.

For applications that do not contain a local database, an HTTP POST request is sent to a predefined server using a colorless prefix. This server, through means that are out of the scope of this document, selects the most appropriate color for the URIs used by the application. It then returns an XML document containing the mapping between the URIs and the colors. URIs in this document MAY use wild cards.

When the application is started, it sends the available prefixes and their color to the video provider server which answers with a wild card URI videos.example.com associated to color 50. In this example application it receives:

```
<?xml version="1.0" encoding="UTF-8"?>
<mappings>
    <mapping>
        <URI>*://audio.example.com/*</URI>
        <color>40</color>
    </mapping>
    <mapping>
        <URI>rtsp://video.example.com/*</URI>
        <color>50</color>
    </mapping>
</mappings>
```

The server is expected to know the application, and thus to list all URIs that could be of use to the application. The application will not ask the server if it has to contact an address not in the list and will use the colorless prefix. This avoids an additional delay when trying to contact an unlisted URI.

Example: While the application is browsing the video list, it is using www.example.com, and thus the colorless prefix. However as soon as a video is chosen, it starts streaming from videos.example.com, and asks to connect to host videos.example.com with color 50, indicating that it wishes to use the colored prefix.

A.1.4. Android DHCPv6

Considering that this prototype is being implemented on Android, the first step is to get a running DHCPv6 client on Android, with support for the OPTION_PREFIX_CLASS option.

The odhcp6c client, which already supports OPTION_PREFIX_CLASS, has been ported to Android, and is set to run in parallel to the dhcpcd client used for DHCP. The success of any of the two clients results in the success of the WiFi connection, so as to support IPv6 only networks.

This client configures the IPv6 addresses using calls to IP address, which is modified to support the addition of a class option to set the prefix color.

A.1.5. Application to network stack communication

Once an application has received the appropriate color for its use, in this prototype it specifies the prefix it wishes to use by using the IPv6 address scope [RFC4007]. When resolving this address, the standard library then adds this information in the address information it returns, using the scope field, allowing the kernel to appropriately select the source IP when connecting. For this reason, a color of 0 is identical to an colorless prefix.

In the example, when downloading from video.example.com, the application would request a connection to video.example.com%50.

This allows the user to override the application's default simply by specifying a color in the scope of the URI it is trying to access, and requires little to no change in applications to support it. Applications that allow scope ids do not need to be modified in order to allow the user to use multiple prefixes (though it is then up to the user to select its color). A web browser that allows scope id would allow the user to add a color to the URI, without requiring any modifications.

A.1.6. Android kernel

To reduce the amount of modifications needed by the applications to support this prefix coloring, we need to avoid having to bind to the address in the colored prefix before initiating the connection. The kernel is expected to choose the correct source address when a colored destination is used.

This implies storing the color in the kernel, along with the address, which is done using a new attribute IFA_color to the netlink message RTM_NEWADDR, used by ip address. Setting a colored prefix using ioctls is not supported.

Since colors are put in the scope id part of the destination address, we continue to use the scope element of the sockaddr_in6 structure to store the color when sending connect messages to the kernel. The scope is only used when considering local addresses, so we interpret the presence of a scope on a non link-local address to be a color. Colors can not be assigned to link-local addresses, but since they are on the same link, source address shouldn't impact how the network treats packets. When selecting the source address, we then discard all addresses which do not have the correct color.

A.1.7. Limitations of the current prototype

It does not implement any duplicate color detection. Colors are considered to be unique within the home, and to correspond to the original color provided by the ISP. This is compatible with Global scoping. No changes would be required to the host in order to support Local scoping with fuzzy matching, but OSPF would need to detect collisions, and the server would need to recalculate the original color before making a decision. In this implementation, hosts that do not support colors do not receive colored prefixes.

Authors' Addresses

Maico Le Pape
Cisco Systems
Paris
FR

Email: maico@maicolepape.org

Shwetha Bhandari
Cisco Systems
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4, Landgrabenweg 151
Bonn 53227
Germany

Email: ian.farrer@telekom.de