

DMM Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2015

A. Yegin
Unaffiliated
K. Kweon
J. Lee
J. Park
Samsung
D. Moses
Intel
March 7, 2015

On Demand Mobility Management
draft-yegin-dmm-ondemand-mobility-03

Abstract

Applications differ with respect to whether they need IP session continuity and/or IP address reachability. The network providing the same type of service to any mobile host and any application running on the host yields inefficiencies. This document describes a solution for taking the application needs into account in selectively providing IP session continuity and IP address reachability on a per-socket basis.

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 September 8, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Notational Conventions	4
3. Solution	4
3.1. Types of IP Addresses	4
3.2. Granularity of Selection	5
3.3. On Demand Nature	5
3.4. Conveying the Selection	6
4. Backwards Compatibility Considerations	7
4.1. Applications	8
4.2. IP Stack in the Mobile Host	8
4.3. Network Infrastructure	8
5. Security Considerations	8
6. IANA Considerations	9
7. Acknowledgements	9
8. References	9
8.1. Normative References	9
8.2. Informative References	9
Authors' Addresses	10

1. Introduction

In the context of Mobile IP [RFC5563][RFC6275][RFC5213][RFC5944], following two attributes are defined for the IP service provided to the mobile hosts:

IP session continuity: The ability to maintain an ongoing IP session by keeping the same local end-point IP address throughout the session despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may change between two independent IP sessions, but that does not jeopardize the IP session continuity. IP session continuity is essential for mobile hosts to maintain ongoing flows without any interruption.

IP address reachability: The ability to maintain the same IP address for an extended period of time. The IP address stays the same across independent IP sessions, and even in the absence of any IP session. The IP address may be published in a long-term registry (e.g., DNS), and it is made available for serving incoming (e.g., TCP)

connections. IP address reachability is essential for mobile hosts to use specific/published IP addresses.

Mobile IP is designed to provide both IP session continuity and IP address reachability to mobile hosts. Architectures utilizing these protocols (e.g., 3GPP, 3GPP2, WIMAX) ensure that any mobile host attached to the compliant networks can enjoy these benefits. Any application running on these mobile hosts is subjected to the same treatment with respect to the IP session continuity and IP address reachability.

It should be noted that in reality not every application may need those benefits. IP address reachability is required for applications running as servers (e.g., a web server running on the mobile host). But, a typical client application (e.g., web browser) does not necessarily require IP address reachability. Similarly, IP session continuity is not required for all types of applications either. Applications performing brief communication (e.g., DNS client) can survive without having IP session continuity support.

Achieving IP session continuity and IP address reachability by using Mobile IP incurs some cost. Mobile IP protocol forces the mobile host's IP traffic to traverse a centrally-located router (Home Agent, HA), which incurs additional transmission latency and use of additional network resources, adds to the network CAPEX and OPEX, and decreases the reliability of the network due to the introduction of a single point of failure [I-D.ietf-dmm-requirements]. Therefore, IP session continuity and IP address reachability should be provided only when needed.

Furthermore, when an application needs session continuity, it may be able to satisfy that need by using a solution above the IP layer, such as MPTCP [RFC6824], SIP mobility [RFC3261], or an application-layer mobility solution. Those higher-layer solutions are not subject to the same issues that arise with the use of Mobile IP since they can utilize the most direct data path between the end-points. But, if Mobile IP is being applied to the mobile host, those higher-layer protocols are rendered useless because their operation is inhibited by the Mobile IP. Since Mobile IP ensures the IP address of the mobile host remains fixed (despite the location and movement of the mobile host), the higher-layer protocols never detect the IP-layer change and never engage in mobility management.

This document proposes a solution for the applications running on the mobile host to indicate whether they need IP session continuity or IP address reachability. The network protocol stack on the mobile host, in conjunction with the network infrastructure, would provide the required type of IP service. It is for the benefit of both the users

and the network operators not to engage an extra level of service unless it is absolutely necessary. So it is expected that applications and networks compliant with this specification would utilize this solution to use network resources more efficiently.

2. Notational Conventions

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].

3. Solution

3.1. Types of IP Addresses

Three types of IP addresses are defined with respect to the mobility management.

- Fixed IP Address

This is what standard Mobile IP provides with a Home Address (HoA). The mobile host is configured a HoA from a centrally-located Home Network. Both IP session continuity and IP address reachability are provided to the mobile host with the help of a router in the Home Network (Home Agent, HA). This router acts as an anchor for the IP address of the mobile host.

- Sustained IP Address

This type of IP address provides IP session continuity but not IP address reachability. It is achieved by ensuring that the IP address used at the beginning of the session remains usable despite the movement of the mobile host. The IP address may change after the termination of the IP session(s), therefore it does not exhibit persistence.

A sustained IP address may be configured and maintained by using access network anchoring, corresponding network anchoring, or some other solution.

- Nomadic IP Address

This type of IP address provides neither IP session continuity nor IP address reachability. The IP address is obtained from the serving IP gateway and it is not maintained across gateway changes. In other words, the IP address may be released and replaced by a new IP address when the IP gateway changes due to the movement of the mobile host.

Applications running as servers at a published IP address require a Fixed IP Address. Long-standing applications (e.g., an SSH session) may also require this type of address. Those applications could use a Sustained IP Address, but that can produce sub-optimal results if the mobile host ends up far from the anchor gateway. Enterprise applications that connect to an enterprise network via virtual LAN require a Fixed IP Address.

Applications with short-lived transient IP sessions can use Sustained IP Addresses. For example: Web browsers.

Applications with very short IP sessions, such as DNS client and instant messengers, can utilize Nomadic IP Addresses. Even though they could very well use a Fixed or Sustained IP Addresses, the transmission latency would be minimized when a Nomadic IP Address is used.

3.2. Granularity of Selection

The IP address type selection is made on a per-socket granularity. Different parts of the same application may have different needs. For example, control-plane of an application may require a Fixed IP Address in order to stay reachable, whereas data-plane of the same application may be satisfied with a Sustained IP Address.

3.3. On Demand Nature

At any point in time, a mobile host may have a combination of IP addresses configured. Zero or more Nomadic, zero or more Sustained, and zero or more Fixed IP addresses may be configured on the IP stack of the host. The combination may be as a result of the host policy, application demand, or a mix of the two.

When the application requires a specific type of IP address and such an IP address is not already configured on the host, then the IP stack shall attempt to configure one. For example, a host may not always have a Fixed IP address available as such an address is rarely used. In case an application requests one, then the IP stack shall make an attempt to configure one using Mobile IP. If Mobile IP protocol is not available on the stack, or if its operation fails, then the IP stack shall fail the associated socket request. In case of successful Mobile IP operation, a Fixed IP Address gets configured on the mobile host. If another socket requests a Fixed IP address at a later time, then the same IP address may be served to that socket as well. When the last socket using the requested IP address is closed, the IP address may be released or kept for future applications that may be launched and require a Fixed IP address.

The following are matters of policy, which may be dictated by the host itself, the network operator, or the system architecture standard:

- The initial set of IP addresses configured on the host at the boot time.
- Permission to grant various types of IP addresses to a requesting application.
- Determination of a default address type when an application does not make any explicit indication, whether it already supports the required API or it is just a legacy application.

3.4. Conveying the Selection

The selection of the address type is conveyed from the applications to the IP stack in a way to influence the source address selection algorithm [RFC6724].

The current source address selection algorithm operates on the available set of IP addresses when selecting an address. According to the proposed solution, if the requested type IP address is not available at the time of the request, then the IP stack shall make an attempt to configure one such IP address. The selected IP address shall be compliant with the requested IP address type, whether it is selected among available addresses or dynamically configured. In the absence of a matching type (because it is not available and not configurable on demand), the source address selection algorithm shall return an empty set.

A Socket API-based interface for enabling applications to influence the source address selection algorithm is described in [RFC5014]. That specification defines IPV6_ADDR_PREFERENCES option at the IPPROTO_IPV6 level. That option can be used with setsockopt() and getsockopt() calls to set and get address selection preferences.

Furthermore, that RFC also specifies two flags that relate to IP mobility management: IPV6_PREFER_SRC_HOME and IPV6_PREFER_SRC_COA. These flags are used for influencing the source address selection to prefer either a Home Address or a Care-of Address.

Unfortunately, these flags do not satisfy the aforementioned needs due to the following reasons, therefore new flags are proposed in this document:

- Current flags indicate a "preference" whereas there is a need for indicating "requirement". Source address selection algorithm does

not have to produce an IP address compliant with the "preference" , but it has to produce an IP address compliant with the "requirement".

- Current flags influence the selection made among available IP addresses. The new flags force the IP stack to configure a compliant IP address if none is available at the time of the request.

- The Home vs. Care-of Address distinction is not sufficient to capture the three different types of IP addresses described in Section 2.1.

The following new flags are defined in this document and they shall be used with Socket API in compliance with the [RFC5014]:

IPV6_REQ_FIXED_IP /* Require a Fixed IP address as source */

IPV6_REQ_SUSTAINED_IP /* Require a Sustained IP addr. as source */

IPV6_REQ_NOMADIC_IP /* Require a Nomadic IP address as source */

More than one of these flags may be set on the same socket. In that case, an IP address compliant with any one of them shall be selected. TBD: Disallow this case?

When any of these new flags is used, then the IPV6_PREFER_SRC_HOME and IPV6_PREFER_SRC_COA flags, if used, shall be ignored.

These new flags are used with `setsockopt()/getsockopt()`, `getaddrinfo()`, and `inet6_is_srcaddr()` functions [RFC5014]. Similar with the `setsockopt()/getsockopt()` calls, `getaddrinfo()` call shall also trigger configuration of the required type IP address, if one is not already available. When the new flags are used with `getaddrinfo()` and the triggered configuration fails, the `getaddrinfo()` call shall ignore that failure (i.e., not return an error code to indicate that failure). Only the `setsockopt()` shall return an error when configuration of the requested type IP address fails.

Application of this solution to IPv4 is TBD.

4. Backwards Compatibility Considerations

Backwards compatibility support is required by the following 3 types of entities:

- The Applications on the mobile host
- The IP stack in the mobile host

- The network infrastructure

4.1. Applications

Legacy applications that do not support the new flags will use the legacy API to the IP stack and will not enjoy On-Demand Mobility feature.

Applications using the new flags must be aware that they may be executed in environments that do not support On-Demand Mobility feature. Such environments may include legacy IP stack in the mobile host, legacy network infrastructure, or both. In either case, the API will return an error code and the invoking applications must respond with using legacy calls without On-Demand Mobility feature.

4.2. IP Stack in the Mobile Host

New IP stacks must continue to support all legacy operations. If an application does not use On-Demand Mobility feature, the IP stack must respond in a legacy manner.

If the network infrastructure supports On-Demand Mobility feature, the IP stack may still request specific types of source IP address transparently to legacy applications. This may be useful for environments in which both legacy and new applications are executed.

The definition of what type of addresses to request and how they are assigned to legacy applications are outside of the scope of this specification.

4.3. Network Infrastructure

The network infrastructure may or may not support the On-Demand Mobility feature. How the IP stack on the host and the network infrastructure behave in case of a compatibility issue is outside the scope of this API specification.

5. Security Considerations

The setting of certain IP address type on a given socket may be restricted to privileged applications. For example, a Fixed IP Address may be provided as a premium service and only certain applications may be allowed to use them. Setting and enforcement of such privileges are outside the scope of this document.

6. IANA Considerations

TBD

7. Acknowledgements

We would like to thank Alexandru Petrescu, John Kaippallimalil, Jouni Korhonen, Seil Jeon, and Sri Gundavelli for their valuable comments and suggestions on this work.

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.
- [RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, September 2007.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

8.2. Informative References

- [I-D.ietf-dmm-requirements] Chan, A., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", draft-ietf-dmm-requirements-17 (work in progress), June 2014.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.
- [RFC5563] Leung, K., Dommety, G., Yegani, P., and K. Chowdhury, "WiMAX Forum / 3GPP2 Proxy Mobile IPv4", RFC 5563, February 2010.
- [RFC5944] Perkins, C., "IP Mobility Support for IPv4, Revised", RFC 5944, November 2010.

[RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

Authors' Addresses

Alper Yegin
Unaffiliated
Istanbul
Turkey

Email: alper.yegin@yegin.org

Kisuk Kweon
Samsung
Suwon
South Korea

Email: kisuk.kweon@samsung.com

Jinsung Lee
Samsung
Suwon
South Korea

Email: js81.lee@samsung.com

Jungshin Park
Samsung
Suwon
South Korea

Email: shin02.park@samsung.com

Danny Moses
Intel Corporation
Petah Tikva
Israel

Email: danny.moses@intel.com