

DMM Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 31, 2020

A. Yegin  
Actility  
D. Moses  
Intel  
S. Jeon  
Sungkyunkwan University  
July 30, 2019

On Demand Mobility Management  
draft-ietf-dmm-ondemand-mobility-18

## Abstract

Applications differ with respect to whether they need 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, as described in [RFC7333]. This document defines a new concept of enabling applications to influence the network's mobility services (session continuity and/or IP address reachability) on a per-Socket basis, and suggests extensions to the networking stack's API to accommodate this concept.

## 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 <https://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 31, 2020.

## Copyright Notice

Copyright (c) 2019 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 (<https://trustee.ietf.org/license-info>) in effect on the date of

Internet-Draft

On Demand Mobility

July 2019

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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Notational Conventions . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Solution . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	High-level Description . . . . .	<a href="#">4</a>
<a href="#">3.2.</a>	Types of IP Addresses . . . . .	<a href="#">5</a>
<a href="#">3.3.</a>	Granularity of Selection . . . . .	<a href="#">6</a>
<a href="#">3.4.</a>	On Demand Nature . . . . .	<a href="#">6</a>
<a href="#">4.</a>	Backwards Compatibility Considerations . . . . .	<a href="#">7</a>
<a href="#">4.1.</a>	Applications . . . . .	<a href="#">8</a>
<a href="#">4.2.</a>	IP Stack in the Mobile Host . . . . .	<a href="#">8</a>
<a href="#">4.3.</a>	Network Infrastructure . . . . .	<a href="#">8</a>
<a href="#">4.4.</a>	Merging this work with <a href="#">RFC5014</a> . . . . .	<a href="#">8</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">9</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">10</a>
<a href="#">7.</a>	Contributors . . . . .	<a href="#">10</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">10</a>
<a href="#">9.</a>	References . . . . .	<a href="#">10</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">10</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">11</a>
<a href="#">Appendix A.</a>	Conveying the Desired Address Type . . . . .	<a href="#">11</a>
	Authors' Addresses . . . . .	<a href="#">12</a>

## [1.](#) Introduction

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

### - Session Continuity

The ability to maintain an ongoing transport interaction by keeping the same local end-point IP address throughout the life-time of the IP socket despite the mobile host changing its point of attachment within the IP network topology. The IP address of the host may

change after closing the IP socket and before opening a new one, but that does not jeopardize the ability of applications using these IP sockets to work flawlessly. 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 sessions, and even in the absence of any session. The IP address may be published in a long-term registry (e.g., DNS), and 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 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 session continuity and IP address reachability.

Achieving session continuity and IP address reachability with 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 [[RFC7333](#)]. Therefore, session continuity and IP address reachability SHOULD be provided only when necessary.

In reality not every application may need these 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, session continuity is not required for all types of applications either. Applications performing brief communication (e.g., text messaging) can survive without having session continuity support.

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. These 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, the higher-layer protocols are rendered useless because their operation is inhibited by Mobile IP. Since Mobile IP ensures that 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 applications running on mobile hosts to indicate when establishing the network connection ('on demand') whether they need session continuity or IP address reachability. The network protocol stack on the mobile host, in conjunction with the network infrastructure, provides the required type of 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. It is expected that applications and networks compliant with this specification will 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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#), [[RFC2119](#)] [[RFC8174](#)] when, they appear in all capitals, as shown here.

## [3.](#) Solution

### [3.1.](#) High-level Description

Enabling applications to indicate their mobility service requirements e.g. session continuity and/or IP address reachability, comprises the following steps:

- The application indicates to the network stack (local to the mobile host) the desired mobility service.

- The network stack assigns a source IP address based on an IP prefix with the desired services that was previously provided by the network. If such an IP prefix is not available, the network stack performs the additional steps below.
- The network stack sends a request to the network for a new source IP prefix that is associated with the desired mobility service.
- The network responds with the suitable allocated source IP prefix (or responds with a failure indication).
- If the suitable source IP prefix was allocated, the network stack constructs a source IP address and provides it to the application.

This document specifies the new address types associated with mobility services and details the interaction between the applications and the network stack steps. It uses the Socket

interface as an example for an API between applications and the network stack. Other steps are outside the scope of this document.

### [3.2.](#) Types of IP Addresses

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

#### - Fixed IP Address

A Fixed IP address is an address with a guarantee to be valid for a very long time, regardless of whether it is being used in any packet to/from the mobile host, or whether or not the mobile host is connected to the network, or whether it moves from one point-of-attachment to another (with a different IP prefix) while it is connected.

Fixed IP addresses are required by applications that need both session continuity and IP address reachability.

#### - Session-lasting IP Address

A session-lasting IP address is an address with a guarantee to be valid throughout the life-time of the socket(s) for which it was requested. It is guaranteed to be valid even after the mobile host had moved from one point-of-attachment to another (with a different IP prefix).

Session-lasting IP addresses are required by applications that need session continuity but do not need IP address reachability.

#### - Non-persistent IP Address

This type of IP address has no guarantee to exist after a mobile host moves from one point-of-attachment to another, and therefore, no session continuity nor IP address reachability are provided. The IP address is created from an IP prefix that is obtained from the serving IP gateway and is not maintained across gateway changes. In other words, the IP prefix may be released and replaced by a new one when the IP gateway changes due to the movement of the mobile host forcing the creation of a new source IP address with the updated allocated IP prefix.

#### - Graceful Replacement IP Address

In some cases, the network cannot guarantee the validity of the provided IP prefix throughout the duration of the opened socket, but can provide a limited graceful period of time in which both the

original IP prefix and a new one are valid. This enables the application some flexibility in the transition from the existing source IP address to the new one.

This gracefulness is still better than the non-persistence type of address for applications that can handle a change in their source IP address but require that extra flexibility.

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. Enterprise applications that connect to an enterprise network via virtual LAN require a Fixed IP Address.

Applications with short-lived transient sessions can use Session-

lasting IP Addresses. For example: Web browsers.

Applications with very short sessions, such as DNS clients and instant messengers, can utilize Non-persistent IP Addresses. Even though they could very well use Fixed or Session-lasting IP Addresses, the transmission latency would be minimized when a Non-persistent IP Addresses are used.

Applications that can tolerate a short interruption in connectivity can use the Graceful-replacement IP addresses. For example, a streaming client that has buffering capabilities.

### [3.3.](#) Granularity of Selection

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

### [3.4.](#) On Demand Nature

At any point in time, a mobile host may have a combination of IP addresses configured. Zero or more Fixed, zero or more Session-lasting, zero or more Non-persistent and zero or more Graceful-Replacement IP addresses may be configured by 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 an application requires a specific type of IP address and such an address is not already configured on the host, the IP stack SHALL attempt to configure one. For example, a host may not always have a Session-lasting IP address available. When an application requests

one, the IP stack SHALL make an attempt to configure one by issuing a request to the network. If the operation fails, the IP stack SHALL fail the associated socket request and return an error. If successful, a Session-lasting IP Address gets configured on the mobile host. If another socket requests a Session-lasting IP address at a later time, the same IP address may be served to that socket as well. When the last socket using the same configured IP address is closed, the IP address may be released or kept for future

applications that may be launched and require a Session-lasting IP address.

In some cases it might be preferable for the mobile host to request a new Session-lasting IP address for a new opening of an IP socket (even though one was already assigned to the mobile host by the network and might be in use in a different, already active IP sockets). It is outside the scope of this specification to define criteria for choosing to use available addresses or choosing to request new ones. It supports both alternatives (and any combination).

It is outside the scope of this specification to define how the host requests a specific type of prefix and how the network indicates the type of prefix in its advertisement or in its reply to a request.

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 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.

#### 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 On-Demand functionality will use the legacy API and will not be able to take advantage of the On-Demand Mobility feature.

Applications using the new On-Demand functionality should be aware that they may be executed in legacy environments that do not support it. Such environments may include a legacy IP stack on the mobile host, legacy network infrastructure, or both. In either case, the API will return an error code and the invoking applications may just give up and use legacy calls.

#### [4.2.](#) IP Stack in the Mobile Host

New IP stacks (that implement On Demand functionality) MUST continue to support all legacy operations. If an application does not use On-Demand functionality, the IP stack MUST respond in a legacy manner.

If the network infrastructure supports On-Demand functionality, the IP stack SHOULD follow the application request: If the application requests a specific address type, the stack SHOULD forward this request to the network. If the application does not request an address type, the IP stack MUST NOT request an address type and leave it to the network's default behavior to choose the type of the allocated IP prefix. If an IP prefix was already allocated to the host, the IP stack uses it and may not request a new one from the network.

#### [4.3.](#) Network Infrastructure

The network infrastructure may or may not support the On-Demand functionality. 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.

#### [4.4.](#) Merging this work with [RFC5014](#)

[RFC5014] defines new flags that may be used with `setsockopt()` to influence source IP address selection for a socket. The list of flags include: source home address, care-of address, temporary address, public address CGA (Cryptographically Created Address) and non-CGA. When applications require session continuity service, they SHOULD NOT set the flags specified in [[RFC5014](#)].

However, if an application erroneously performs a combination of (1) Use `setsockopt()` to set a specific option (using one of the flags specified in [[RFC5014](#)]) and (2) Selects a source IP address type, the

---

IP stack will fulfill the request specified by (2) and ignore the flags set by (1).

## 5. Security Considerations

The different service types (session continuity types and address reachability) associated with the allocated IP address types, may be associated with different costs. The cost to the operator for enabling a type of service, and the cost to applications using a selected service. A malicious application may use these to generate extra billing of a mobile subscriber, and/or impose costly services on the mobile operator. When costly services are limited, malicious applications may exhaust them, preventing other applications on the same mobile host from being able to use them.

Mobile hosts that enables such service options, should provide capabilities for ensuring that only authorized applications can use the costly (or limited) service types.

The ability to select service types requires the exchange of the association of source IP prefixes and their corresponding service types, between the mobile host and mobile network. Exposing these associations may provide information to passive attackers even if the traffic that is used with these addressed is encrypted.

To avoid profiling an application according to the type of IP addresses, it is expected that prefixes provided by the mobile operator are associated to various type of addresses over time. As a result, the type of address could not be associated to the prefix, making application profiling based on the type of address harder.

The application or the OS should ensure that IP addresses regularly change to limit IP tracking by a passive observer. The application should regularly set the On Demand flag. The application should be able to ensure that session lasting IP addresses are regularly changed by setting a lifetime for example handled by the application. In addition, the application should consider the use of graceful replacement IP addresses.

Similarly, the OS may also associated IP addresses with a lifetime. Upon receiving a request for a given type of IP address, after some time, the OS should request a new address to the network even if it already has one IP address available with the requested type. This includes any type of IP address. IP addresses of type graceful replacement or non persistent should be regularly renewed by the OS.

The lifetime of an IP address may be expressed in number of seconds or in number of bytes sent through this IP address.

## 6. IANA Considerations

This document has no IANA considerations.

## 7. Contributors

This document was merged with [[I-D.sijeon-dmm-use-cases-api-source](#)]. We would like to acknowledge the contribution of the following people to that document as well:

Sergio Figueiredo  
Altran Research, France  
Email: sergio.figueiredo@altran.com

Younghan Kim  
Soongsil University, Korea  
Email: younghak@ssu.ac.kr

John Kaippallimalil  
Huawei, USA  
Email: john.kaippallimalil@huawei.com

## 8. Acknowledgements

We would like to thank Wu-chi Feng, Alexandru Petrescu, Jouni Korhonen, Sri Gundavelli, Dave Dolson Lorenzo Colitti and Daniel Migault for their valuable comments and suggestions on this work.

## 9. References

### 9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", [RFC 5014](#),

DOI 10.17487/RFC5014, September 2007,  
<<https://www.rfc-editor.org/info/rfc5014>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Yegin, et al.

Expires January 31, 2020

[Page 10]

---

Internet-Draft

On Demand Mobility

July 2019

## [9.2](#). Informative References

- [I-D.sijeon-dmm-use-cases-api-source]  
Jeon, S., Figueiredo, S., Kim, Y., and J. Kaippallimalil,  
"Use Cases and API Extension for Source IP Address  
Selection", [draft-sijeon-dmm-use-cases-api-source-07](#) (work  
in progress), September 2017.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,  
A., Peterson, J., Sparks, R., Handley, M., and E.  
Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#),  
DOI 10.17487/RFC3261, June 2002,  
<<https://www.rfc-editor.org/info/rfc3261>>.
- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V.,  
Chowdhury, K., and B. Patil, "Proxy Mobile IPv6",  
[RFC 5213](#), DOI 10.17487/RFC5213, August 2008,  
<<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC5563] Leung, K., Dommety, G., Yegani, P., and K. Chowdhury,  
"WiMAX Forum / 3GPP2 Proxy Mobile IPv4", [RFC 5563](#),  
DOI 10.17487/RFC5563, February 2010,  
<<https://www.rfc-editor.org/info/rfc5563>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised",  
[RFC 5944](#), DOI 10.17487/RFC5944, November 2010,  
<<https://www.rfc-editor.org/info/rfc5944>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility  
Support in IPv6", [RFC 6275](#), DOI 10.17487/RFC6275, July  
2011, <<https://www.rfc-editor.org/info/rfc6275>>.

- [RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", [RFC 6824](https://www.rfc-editor.org/info/rfc6824), DOI 10.17487/RFC6824, January 2013, <<https://www.rfc-editor.org/info/rfc6824>>.
- [RFC7333] Chan, H., Ed., Liu, D., Seite, P., Yokota, H., and J. Korhonen, "Requirements for Distributed Mobility Management", [RFC 7333](https://www.rfc-editor.org/info/rfc7333), DOI 10.17487/RFC7333, August 2014, <<https://www.rfc-editor.org/info/rfc7333>>.

## [Appendix A](#). Conveying the Desired Address Type

Following are some suggestions of possible extensions to the Socket API for enabling applications to convey their session continuity and address reachability requirements.

Yegin, et al.

Expires January 31, 2020

[Page 11]

---

Internet-Draft

On Demand Mobility

July 2019

[RFC5014] introduced the ability of applications to influence the source address selection with the IPV6\_ADDR\_PREFERENCE option at the IPPROTO\_IPV6 level. This option is used with setsockopt() and getsockopt() calls to set/get address selection preferences.

One alternative is to extend the definition of the IPV6\_ADDR\_PREFERENCE option with flags that express the invoker's desire. An "OnDemand" field could contain one of the values: FIXED\_IP\_ADDRESS, SESSION\_LASTING\_IP\_ADDRESS, NON\_PERSISTENT\_IP\_ADDRESS or GRACEFUL\_REPLACEMENT\_IP\_ADDRESS.

Another alternative is to define a new Socket function used by the invoker to convey its desire. This enables the implementation of two behaviors of Socket functions: The existing "setsockopt()" is a function that returns after executing, and the new "setsc()" (Set Service Continuity) function that may initiate a request for the desired service, and wait until the network responds with the allocated resources, before returning to the invoker.

After obtaining an IP address with the desired behavior the application can call the bind() Socket function to associate that received IP address with the socket.

Authors' Addresses

Alper Yegin  
Actility  
Istanbul  
Turkey

Email: [alper.yegin@actility.com](mailto:alper.yegin@actility.com)

Danny Moses  
Intel Corporation  
Petah Tikva  
Israel

Email: [danny.moses@intel.com](mailto:danny.moses@intel.com)

Seil Jeon  
Sungkyunkwan University  
Suwon  
South Korea

Email: [seiljeon@skku.edu](mailto:seiljeon@skku.edu)