Stream: Internet Engineering Task Force (IETF)
RFC: 8653
Category: Informational
Published: October 2019
ISSN: 2070-1721
Authors: A. Yegin, *Actility*; D. Moses, *Intel*; S. Jeon, *Sungkyunkwan University*

RFC 8653
On-Demand Mobility Management

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 RFC 7333. 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 document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8653.

Copyright Notice

Table of Contents

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 endpoint IP address throughout the lifetime 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, 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 using these protocols (e.g., 3GPP, 3GPP2, WiMAX) ensure that any mobile host attached to a compliant network 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 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's operating and capital expenditures, 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 Multipath TCP [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 use the most direct data path between the endpoints. 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, and only 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:

1. The application indicates to the network stack (local to the mobile host) the desired mobility service.
2. 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.
3. The network stack sends a request to the network for a new source IP prefix that is associated with the desired mobility service.
4. The network responds with the suitable allocated source IP prefix (or responds with a failure indication).
5. 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 guaranteed 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 guaranteed to be valid for the lifetime of the socket(s) for which it was requested. It is guaranteed to be valid even after the mobile host has 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.

Nonpersistent IP address
: This type of IP address is not guaranteed to exist after a mobile host moves from one point of attachment to another; 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 nonpersistence 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 (e.g., web browsers) can use Session-Lasting IP addresses.

Applications with very short sessions, such as DNS clients and instant messengers, can use Nonpersistent IP addresses. Even though they could very well use Fixed or Session-Lasting IP addresses, the transmission latency would be minimized when a Nonpersistent IP address is 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 Nonpersistent, and zero or more Graceful-Replacement IP addresses may be configured by the IP stack of the host. The combination may be 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 is 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 it may be kept for applications requiring a Session-Lasting IP address that may be launched in the future.

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 socket). 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 explicitly indicate whether it supports the required API or is a legacy application

4. Backwards Compatibility Considerations

Backwards compatibility support is **REQUIRED** by the following three 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 application 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. Instead, the network will choose the type of allocated IP prefix. How the network selects the type of allocated IP prefix is outside the scope of this document. 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 RFC 5014

[RFC5014] defines new flags that may be used with setsockopt() to influence source IP address selection for a socket. The list of flags include the following: 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) using setsockopt() to set a specific option (using one of the flags specified in [RFC5014]) and (2) selecting 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 indirectly generate extra billing of a mobile subscriber, and/or impose costly services on the mobile operator. When expensive 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 enable such service options should provide capabilities for ensuring that only authorized applications can use the expensive (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 addresses is encrypted.

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

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 associate 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 nonpersistent 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 actions.

7. References

7.1. 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>.

7.2. Informative References

[API-EXT] Jeon, S., Figueiredo, S., Kim, Y., and J. Kaippallimalil, "Use Cases and API Extension for Source IP Address Selection", Work in Progress, Internet-Draft, draft-sijeon-dmm-use-cases-api-source-07, 10 September 2017, <https://tools.ietf.org/html/draft-sijeon-dmm-use-cases-api-source-07>.

[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, 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, DOI 10.17487/RFC7333, August 2014, <https://www.rfc-editor.org/info/rfc7333>.

Appendix A. Conveying the Desired Address Type

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

[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_REFERENCE option with flags that express the invoker's desire. An "OnDemand" field could contain one of the following 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) is a 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.

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.

Contributors

This document was merged with "Use Cases and API Extension for Source IP Address Selection" [API-EXT]. 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
Republic of Korea
Email: younghak@ssu.ac.kr
```

John Kaippallimalil
Huawei
United States of America
Email: john.kaippallimalil@huawei.com

Authors' Addresses

Alper Yegin
Actility
Istanbul/
Turkey
Email: alper.yegin@actility.com

Danny Moses
Intel Corporation
Petah Tikva
Israel
Email: danny.moses@intel.com

Seil Jeon
Republic of Korea
Suwon
Sungkyunkwan University
Email: seiljeon.ietf@gmail.com