

MIF WG
Internet-Draft
Intended status: Informational
Expires: January 2, 2015

H. Deng
China Mobile
S. Krishnan
Ericsson
T. Lemon
Nominum
M. Wasserman
Painless Security, LLC
July 1, 2014

Guide for application developers on session continuity by using MIF API
[draft-deng-mif-api-session-continuity-guide-04](#)

Abstract

Today most smart terminals are equipped with multiple interfaces such as 3G/LTE and WiFi, and users experience some loss of connectivity while switching interfaces. The MIF API draft [\[I-D.ietf-mif-api-extension\]](#) has specified an API to announce interface status information to the applications. Once the application receives such information, it can use this information reconnect to its peer(s), and this could significantly improve the user experience.

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2015.

Internet-Draft

MIF API Session Continuity

July 2014

Copyright Notice

Copyright (c) 2014 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.	Related MIF API information	3
3.	Using different source address to reconnect the server . . .	3
4.	Generic guidelines for writing applications to handle new interfaces becoming available	4
5.	Generic guidelines for writing applications to handle interfaces becoming unavailable	5
6.	IANA Considerations	5
7.	Security Considerations	5
8.	Acknowledgements	5
9.	Normative References	6
	Authors' Addresses	6

[1.](#) Introduction

A significant and increasing number of smart mobile terminals have multiple interfaces for connectivity (e.g. Wifi and 3G/LTE). These interfaces may have very characteristics in terms of reliability, available bandwidth, delay/jitter as well as cost per bit. There is some form of connection manager on the end device that picks an interface for communication based on some pre-configured policy and/or based on dynamic conditions. The initially selected interface may become deprioritized (e.g. due to a lower cost interface becoming available) or may become unavailable (e.g. due to loss of coverage when moving out of a WiFi hotspot). New interfaces may become available due to administrative action (e.g. manual activation of a

specific connectivity technology) or due to dynamic conditions (e.g. entering coverage area of a wireless network or plugging in an ethernet cable). In order to handle such changes in connectivity, applications need to be aware of network status changes and react to them. This document provides a guide to writing such applications.

The MIF API [[I-D.ietf-mif-api-extension](#)] document specifies an API that is capable of providing information regarding changes in network and interface connectivity status. By using this information, application developers can develop applications that can survive changes in connectivity and even benefit from them.

The MIF MPVD Architecture [[I-D.ietf-mif-mpvd-arch](#)] document defines the notion of a PVD (set of network configuration information), and PVDs somehow must be exposed, in case applications are not PVD-aware, or indirectly participating the selection of PVD, or knowing of the PVDs based on PVD APIs. The MIF API [[I-D.ietf-mif-api-extension](#)] document specifies "Connect to PVD" message, application developers may develop application that can changes between different PVD connectivity.

[2.](#) Related MIF API information

MIF API draft [[I-D.ietf-mif-api-extension](#)] defines a few messages that are related to notifying whether an interface is available or not. The messages are defined in [Section 3.5.1](#) (Announce Interfaces) and [Section 3.5.4](#) (No Interface). Similar functionality is available for addresses using the messages defined in [Section 3.5.12](#) (Announce Address) and [Section 3.5.14](#) (No Address Announcement). The API also specifies interface change information in [section 3.5.23.5](#) (Interface is going up) and [3.5.23.4](#) (Interface is going away). Both interface and address information could be used by the application to infer the availability of a new endpoint for communication or the loss of an existing endpoint for communication.

[3.](#) Using different source address to reconnect the server

The applications deployed on mobile hosts usually setup the connection with the server, then trying to keep the connection up as long as they can. This works reasonable well when the host has only one communication interface. Once the host has more than one communication interface, such as 3G/LTE and WLAN, such applications

cease to work well. e.g. The per bit cost and the connection speed are different on these two interfaces, and the user would always prefer to change another cheaper and faster connection. e.g. While connecting to a WLAN interface after being connected to LTE, the mobile terminal would get a different set of configuration parameters including the IP address, DNS server and default gateway. Application would normally break after such change in connectivity if the original interface (3G/LTE) is turned off and manual intervention is usually required to reinitiate connectivity.

If the application is designed with changing network connectivity in mind, then the application could be carefully designed reconnect to

its peer based on MIF API notification about new interface(s) and/or new address(es). The application needs to start testing the usability of the new interface(s)/address(es) immediately and determine whether they are usable and, if so, decide what traffic to switch over. Please note that there are other solutions for handling address changes in the lower layers (network and transport) like MIPv6, shim6, and MPTCP that can shield the application from address changes. The guidelines provided in this document are also applicable when these techniques are being used. Also, there might be load balancers present on the server side and it may become very difficult to preserve sessions after an address change has occurred.

In most cases even when a mobile terminal gets WLAN connectivity and gets an IP address assigned, but it could still be disconnected from the Internet due to lack of authentication. As a consequence, the interface needs to be tested for internet connectivity before switching communication from an existing interface to a newly available interface.

[4.](#) Generic guidelines for writing applications to handle new interfaces becoming available

The recommended steps for the application developer to keep the session continuity based on MIF API are listed below:

Step 1: Application subscribes to the MIF API for interface and address change notifications;

Step 2: Application connects to the server based on interface 1

(either 3G/LTE or WLAN);

Step 3: When a new interface comes up or a new address is configured, the MIF API notifies the application.

Step 4: The application tries to re-connect to its peer from the newly available interface. If the connectivity check succeeds, then the application can successfully switch the communication over to the new interface based on policy or user initiated selection. Otherwise communication stays on the existing interface. The decision process on how a preferred interface is selected is out of scope of this document and might be the topic for a separate high level API document.

Step 5: The interface initially used for communication may now be turned off without disrupting communications if no other applications are using it.

[5.](#) Generic guidelines for writing applications to handle interfaces becoming unavailable

The recommended steps for the application developer to keep the session continuity based on MIF API are listed below:

Step 1: Application subscribes to the MIF API for interface and address change notifications;

Step 2: Application connects to the server based on interface 1 (either 3G/LTE or WLAN);

Step 3: When an interface or address, that is currently being used for communication, becomes unavailable the MIF API notifies the application.

Step 4: The application requests the MIF API to acquire a list of interfaces that are currently available. Based on locally configured preferences, the application tries to re-connect to its peer from one of the available interfaces. If the connectivity check succeeds, then the application can successfully switch the communication over to this interface.

Step 5: If the connectivity check fails, the application needs to redo the check for each of the available interfaces in order of preference until it can successfully connect to its peer.

Step 6: If at least one available interface is still able to connect to the peer, the application can switch over to this interface without disrupting communications.

[6.](#) IANA Considerations

This document does not require any IANA actions.

[7.](#) Security Considerations

Some applications may associate the the source address of the communication with the credentials used, it they may require refreshing the credentials after the application switches to using a new source address.

[8.](#) Acknowledgements

The authors would like to thank Pete McCann, Julien Laganier, Dapeng Liu, Dave Thaler, Brian Carpenter and Pierrick Seite for their comments and suggestions for improving this document.

[9.](#) Normative References

[I-D.ietf-mif-api-extension]

Liu, D., Lemon, T., Ismailov, Y., and Z. Cao, "MIF API consideration", [draft-ietf-mif-api-extension-05](#) (work in progress), February 2014.

[I-D.ietf-mif-mpvd-arch]

Anipko, D., "MIF MPVD Architecture", [draft-ietf-mif-mpvd-arch-01](#) (work in progress), May 2014.

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

Hui Deng
China Mobile
No.32 Xuanwumen West Street
Xicheng District,
Beijing 100053
China

Email: denghui02@gmail.com

Suresh Krishnan
Ericsson
8400 Blvd Decarie
Town of Mount Royal, Quebec
Canada

Email: suresh.krishnan@ericsson.com

Ted Lemon
Nominum
Redwood City,
94063
USA

Email: Ted.Lemon@nominum.com

Margaret Wasserman
Painless Security, LLC
356 Abbott Street,
North Andover 01845
USA

Email: mrw@painless-security.com

