MIF WG
Internet-Draft
Intended status: Informational
Expires: March 24, 2013

P. Seite
France Telecom - Orange
JC. Zuniga
InterDigital Communications, LLC
September 20, 2012

MIF API Conn Mngr Considerations
draft-seite-mif-cm-00.txt

Abstract

There is currently a need to present a coherent connection management behaviour for different terminal platforms (e.g. mobile phones, PCs, tablets, etc.). This document discusses how a connection manager can use the MIF API to provide this coherent behaviour and enhance the end user's experience when a terminal is able to connect to multiple interfaces. The goal of this document is not to define a connection manager specification, but to focus on the interaction with the MIF API and suggest relevant generic messages for the interface.

This document is for discussion and its intention is to help clarifying the utilization of the MIF API in a connection management context and propose some relevant considerations.

Table of Contents

1. Introduction

[I-D.ietf-mif-api-extension] describes an abstract API that provides commands and services for applications and higher layer APIs running on a terminal with more than one interface. There is currently a need to present a coherent connection management behaviour for different terminal platforms (e.g. mobile phones, PCs, tablets, etc.), as users often experience a very different behaviour when connecting with various platforms to the same networks and for the same purposes (e.g. web browsing, email access, dedicated applications, etc.). This document builds on top of the MIF API and aims to discuss how connection managers can use the MIF API to provide a coherent and constant behaviour to the users. The goal of this document is not to define a connection manager specification, but to focus on the interaction with the MIF API and suggest relevant generic messages for the interface

This document is for discussion and its intention is to help clarifying the utilization of the MIF API in a connection management context.

2. MIF API model

The terminal's API model is depicted below:

- MIF API: Provides information as per [I-D.ietf-mif-api-extension].

- Connection Manager: Relies on the MIF API to decide on the mapping between flows and interfaces and configures their operation accordingly, e.g. configuration of the routing table. The decision process relies on information provided by the MIF API as well as by other functions, such as 3GPP/ANDSF [TS23.402] or the OS (via the OS API), providing information which is not in the scope of the MIF API, e.g. battery status.

- OS API: Provides the interface to manipulate object configuration, e.g. routing table, virtual interface, etc.

- Virtual interface: Hides the multiple interfaces environment to the application. It applies connection management decisions, mapping flows to the appropriate interface. If supported, the connection manager may also request the virtual interface to provide IP flow mobility support [RFC6089], [I-D.ietf-netext-logical-interface-support].

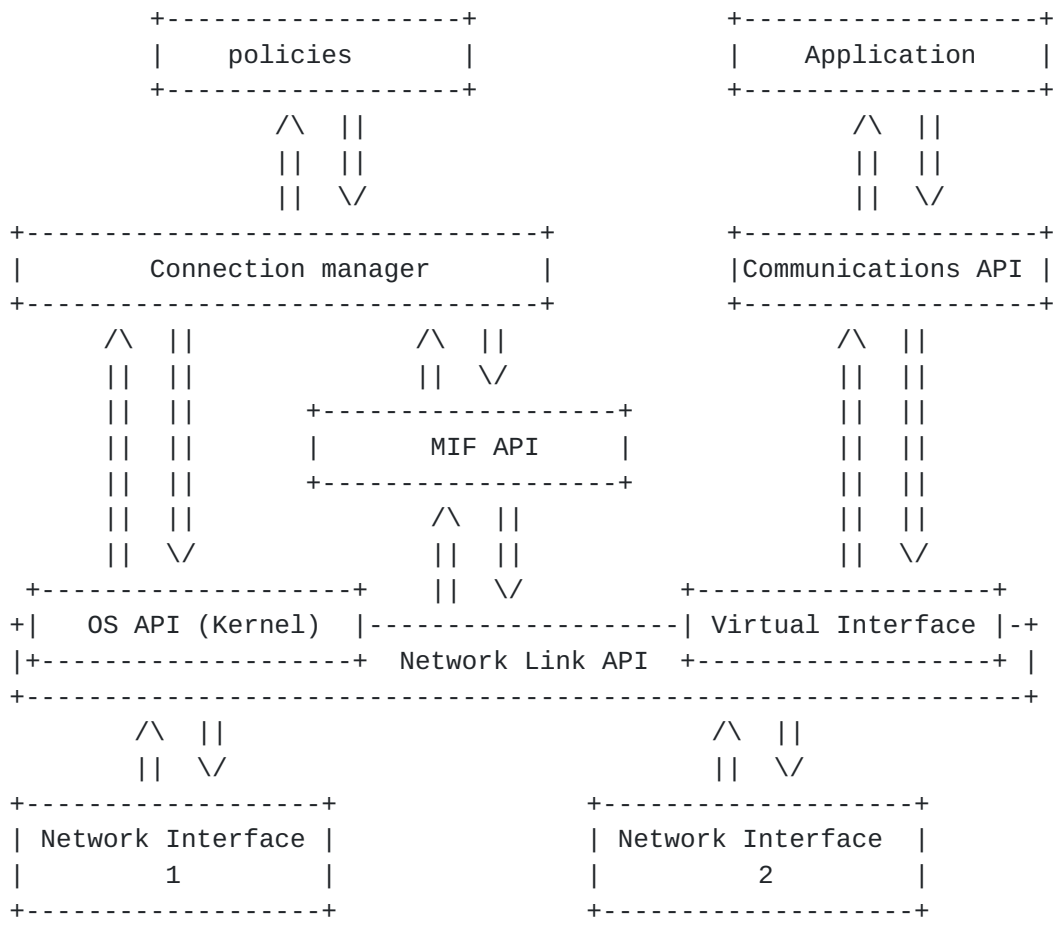

Figure 1: MIF API framework

3. Use-case

The presented use-case aims to illustrate the behaviour of the MIF API in a concrete situation. The use-case is as follows:

1. Multiple IP communications are running simultaneously; each can be mapped to different interface/provisioning domain at the same time.

2. The connection manager selects the appropriate interface for the application. The connection manager makes a decision according to various criteria; including information provided by the MIF and lower layers APIs as well as selection policies, such as user preferences and/or network operator policies provided by the 3GPP Access Network and Discovery Function (ANDSF) [TS23.402]).

3. The connection manager, together with the terminal's APIs, shall make the applications agnostic about the multiple interfaces management.

The interaction between the different APIs is depicted in Figure 2. It is assumed that at least one IP communication is running. Then, an interface event occurs and the connection manager decides to move the communication to a different interface.

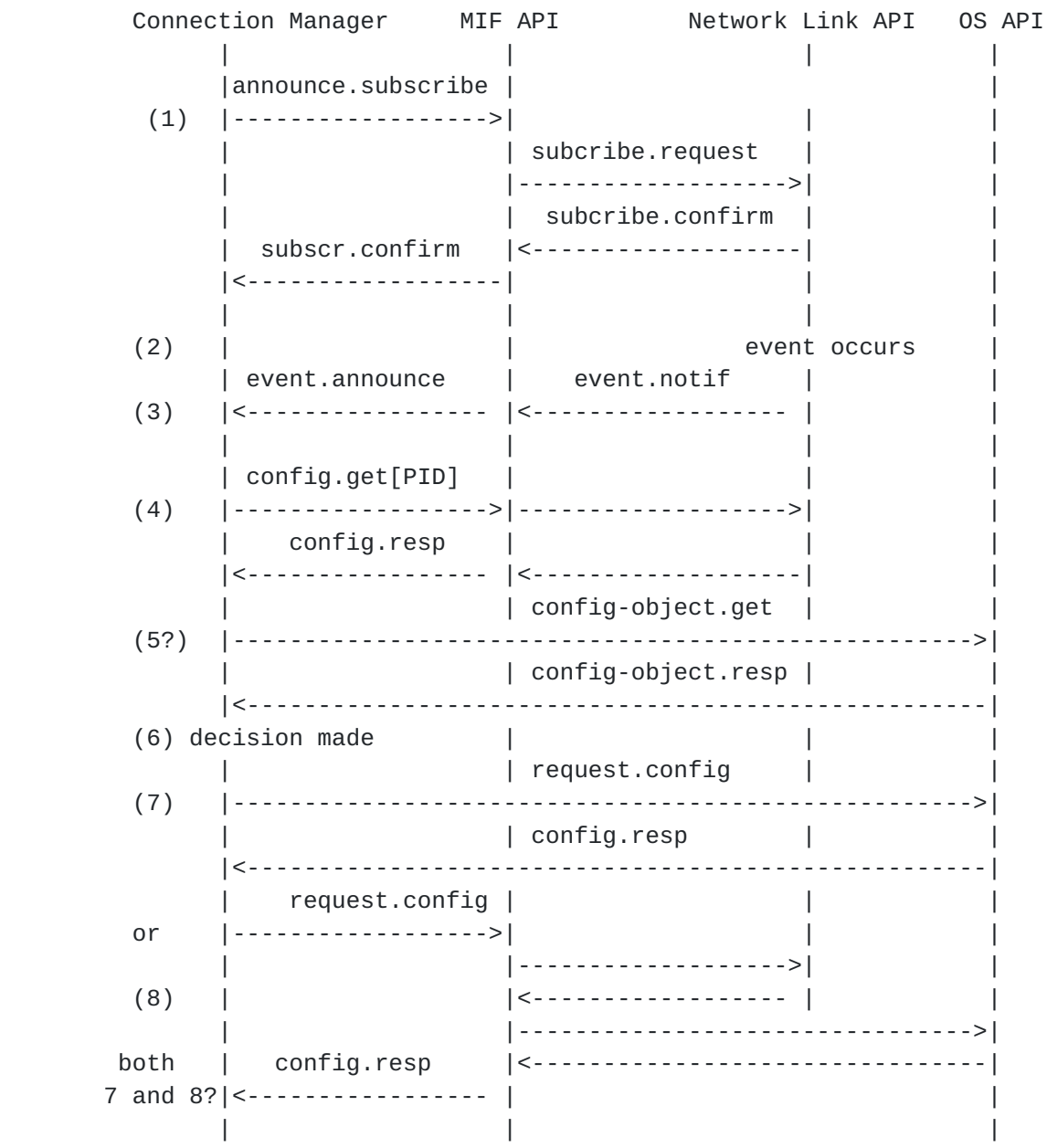

Figure 2: APIs interaction

Operations are as follows:

1. The connection manager subscribes to the MIF API notifications [I-D.ietf-mif-api-extension].

2. An event, to which the connection manager has subscribed, occurs; e.g. a new interface becomes available or a low radio signal

level is crossed.

3. The connection manager is notified about the event.

4. In order to take its decision, the connection manager gets some configuration information from the MIF API.

5. The connection manager fetches additional information from the OS API

6. The connection manager decides to move the ongoing IP communication to another interface.

7. The connection manager requests the OS API to reconfigure one or multiple interfaces according to the decision; for example, the connection manager could request reconfiguration of the routing table or trigger a MIP operation.

To Be Discussed:

- Could all the IP configuration objects be provided by the MIF API? The decision of the CM may impact some IP configuration objects (e.g. terminal's routing table, source address, etc.), but the current MIF API does not provide such a service. The CM can proceed via the OS API, but shouldn't the MIF API be the unique API for any manipulation of IP objects?

4. Functions of the connection manager

This section focuses on the interactions between the connection manager and the MIF API and OS API. The interactions between the connection manager and other complementary APIs, like user preferences and/or ANDSF network operator policies are out of the scope of this document.

A connection manager may also rely on different abstraction layers together with the MIF API. The IEEE 802.21 MIH SAP [IEEE802.21] is an example of such an abstraction layer, which can be seen as a partial instantiation of the MIF API. A brief description is provided in the Annex in Section 7.

Generic connection manager functions in the MIF API scope are as follows:

Subscribe(eventID)

Description: register for a MIF API event notification, e.g. WLAN scan results ready, WLAN connected, WLAN disconnected, interface is going to be disconnected detected (e.g. because of low radio signal level detected), Cellular connected, Cellular disconnected, etc.

Input: identifier of the event to be notified. Some events are defined in [I-D.ietf-mif-api-extension]

API: MIF API

UnSubscribe(eventID)

Description: unregister to a MIF API notification.

Input: identifier of event. Some events are defined in [I-D.ietf-mif-api-extension]

API: MIF API

ListInterfaces()

Description: return the list of available interfaces with their characteristics. Interfaces may have different access technologies.

Input: n/a

API: OS API / MIF API

ListProvisioningDomains()

Description: return the list of available provisioning domains with their characteristics.

Input: n/a

API: MIF API

GetStatus(IID)

Description: provide the status of the interface, e.g. enabled/ disabled, active, idle, connection failed, connecting, disconnecting, scanning, unknown state, etc.

Input:Interface Identifier

API: MIF API

IPconnectivityCheck(PID, IP[])

Description: check IP connectivity to the intranet/Internet: the interface may have a valid IP address but no IP connectivity to data networks (e.g. web based authentication through a captive portal).

Input: Provisioning domain Identifier, IP addresses to be tested

API: MIF API

GetConfiguration(IID)

Description: retrieve layer 2 configuration information for a given interface.

Input: Interface Identifier

API: OS API

SetConfiguration(IID)

Description: configures an interface, e.g. enable/disable, scan, etc.

Input: Interface Identifier

API: OS API

GetConfiguration(PID)

Description: retrieve configuration information for a given provisioning domain(IP address(es), DNS, default gateway, authentication method, associated interface(s))

Input: Provisioning domain Identifier

API: MIF API

SetConfiguration(PID)

Description: configure provisioning domain information (IP addresse(s), default gateway, authentication method, associated interface, routing table, etc.)

Input: Provisioning domain Identifier

API: MIF API

GetTheoriticalQoS(IID)

Description: provide information on the theoretical interface capabilities (e.g. upload/download speed)

Input: Interface Identifier

API: MIF API

GetAvailableQoS(IID)

Description: provide information on the quality of communication (Jitter, delay, average upload data rate, average Download data rate, signal strength, etc.)

Input: Interface Identifier

API: MIF API

GetIPtype(IP address)

Description: return the type of address and properties (e.g. local, remote, mobile IP anchored, etc.) [I-D.korhonen-dmm-prefix-properties].

Input: IP address

API: OS API or MIF API?

AssociateRouting(RT-TABLE-ID, FlowID)

Description: associate a routing table, RT-TABLE-ID, to the IP flow identified by FlowID, e.g. as defined in [RFC6088].

Input: routing table identifier, flow identifier

API: OS API or MIF API?

SetSourceAddress(IP, FlowID)

Description: influence source address selection for a given IP flow.

Input: IP source address, flow identifier

API: OS API or MIF API?

5. Security Considerations

TBD.

6. IANA Considerations

This document has no actions for IANA.

7. Annex: Interaction of the MIF API with the IEEE 802.21 Framework

Some of the connection management services described in section Section 4 may rely on standardized abstraction layers such as the IEEE 802.21 framework [IEEE802.21].

The IEEE 802.21-2008 Media Independent Handover (MIH) Services specification defines three type of services: Information Services (IS), Event Services (ES) and Command Services (CS). Each one of these services has a different purpose. The IS provides information about existing networks and services in a potential target network. The ES provides triggers, measurements and events from lower layers (e.g. network access layers) that can be used for instance to proactively trigger actions like a handing over or establishing connection to a new network. The CS allows configuring lower layer interfaces and events. Both ES and CS provide an abstraction layer to upper layers that allows configuring and interacting with different types of network link interfaces in a coherent manner.

The basics of the 802.21 ES/CS flow are depicted in Figure 3.

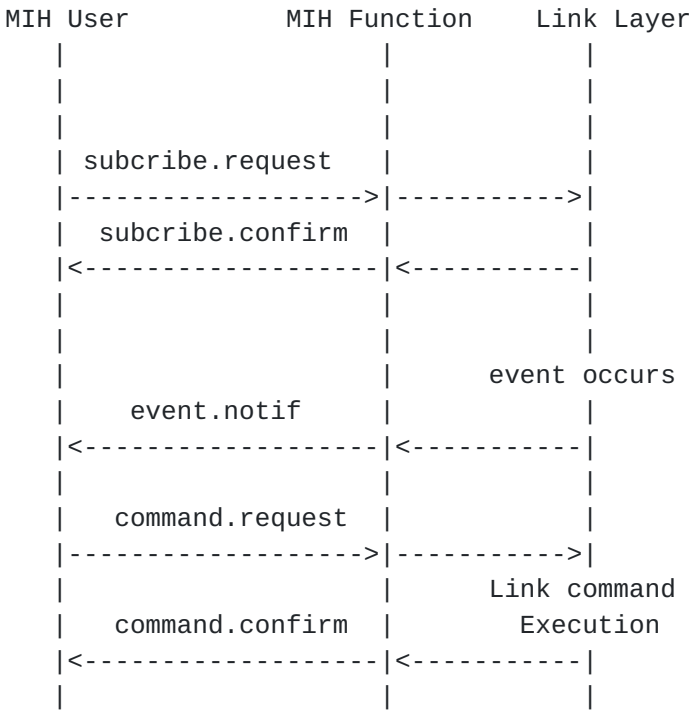

Figure 3: IEEE 802.21 event and command services flow

An MIH user may use the following IEEE 802.21 events and commands to

complete connection management operations:

Link_Event_Subscribe

Category: command

Description: Subscribe to one or more events from a link.

Link_Event_Unsubscribe

Category: command

Description: Unsubscribe from a set of link-layer events.

Link_Parameters_Report

Category: Event

Description: Link parameters have crossed a specified threshold and need to be reported.

Link_Get_Parameters

Category: command

Description: Get parameters measured by the active link, such as signal-to-noise ratio (SNR), BER, received signal strength indication (RSSI).

Link_Detected

Category: Event

Description: Link of a new access network has been detected.

Link_Up

Category: Event

Description: L2 connection is established and link is available for use.

Link_Down

Category: Event

Description: L2 connection is broken and link is not available for use.

Link_Going_Down

Category: Event

Description: Radio link conditions are degrading and connection loss is very likely.

Link_Handover_Imminent

Category: Event

Description: L2 handover is imminent based on either the changes in the link conditions or additional information available at layer 2.

Link_Handover_Complete

Category: Event

Description: L2 handover has been completed.

Link_PDU_Transmit_Status

Category: Event

Description: indicate transmission status of a PDU.

Link_Capability_Discover

Category: command

Description: Query and discover the list of supported link-layer events and link-layer commands.

Link_Configuration_Thresholds

Category: command

Description: Configure thresholds for future Link Parameters Report events.

Link_Action

Category: command

Description: request an action on a link-layer connection, e.g. perform a scan, shut down an interface, etc.

Handover_Query

Category: command

Description: query and obtain handover related information about possible candidate networks.

Handover_Commit

Category: command

Description: notify the MIH function of the decided target network.

Handover_Complete

Category: command

Description: indicate the status of the handover completion to the MIH function.

8. Acknowledgements

The authors would like to express their gratitude to Ralph Droms, Ted Lemon and Dave Thaler for the fruitful discussions regarding MIF API and connection managers.

9. References

9.1. Normative References

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

[RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, January 2011.

[RFC6089] Tsirtsis, G., Soliman, H., Montavont, N., Giaretta, G., and K. Kuladinithi, "Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support", RFC 6089, January 2011.

9.2. Informative References

[I-D.deng-mif-api-session-continuity-guide] Deng, H., Krishnan, S., Lemon, T., and M. Wasserman, "Guide for application developers on session continuity by using MIF API", draft-deng-mif-api-session-continuity-guide-02 (work in progress), July 2012.

[I-D.ietf-mif-api-extension] Liu, D., Lemon, T., Ismailov, Y., and Z. Cao, "MIF API consideration", draft-ietf-mif-api-extension-01 (work in progress), July 2012.

[I-D.ietf-netext-logical-interface-support] Melia, T. and S. Gundavelli, "Logical Interface Support for multi-mode IP Hosts", draft-ietf-netext-logical-interface-support-05 (work in progress), April 2012.

[I-D.korhonen-dmm-prefix-properties] Korhonen, J., Patil, B., Gundavelli, S., Seite, P., and D. Liu, "IPv6 Prefix Mobility Management Properties", draft-korhonen-dmm-prefix-properties-02 (work in progress), July 2012.

[IEEE802.21] IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Part 21: Media Independent Handover Services", IEEE LAN/MAN Std 802.21-2008, January 2009.", 2009, < http ://www.ieee802.org/21/private/Published%20Spec/ 802.21-2008.pdf>.

[TS23.402]
3GPP, "3GPP TS 23.402; Architecture enhancements for non-3GPP accesses", 2010.

Authors' Addresses

Pierrick Seite
France Telecom - Orange
4, rue du Clos Courtel, BP 91226
Cesson-Sevigne 35512
France

Email: pierrick.seite@orange.com

Juan Carlos Zuniga
InterDigital Communications, LLC
1000 Sherbrooke Street West, 10th floor
Montreal, Quebec H3A 3G4
Canada

Email: JuanCarlos.Zuniga@InterDigital.com