Implementation of IEEE 802.21 based Media Independent Handover Services

Ankith Agarwal*, Pramod P. J., and D. K. Jain

Center for Development of Advanced Computing (CDAC), Hyderabad.

e-Mails: ankitha@cdac.in; pramodpj@cdac.in; dkjain@cdac.in

* Tel.: +91-040-23737124; Fax: +91-040-23743382

Abstract: The evolution of mobile technology has increased the demand for multimedia services on portable devices. Multiple access technologies are being integrated to modern mobile devices, which provide an opportunity to the network operators to improve the quality of services they provide. To take the advantage of these opportunities, IEEE 802.21 Media Independent Handover (MIH) standard was introduced to facilitate seamless mobility in a heterogeneous network ecosystem comprising of various IEEE 802 and cellular access technologies. In this paper we focus on design and implementation of MIH services as part of Stream over IP project funded by Department of IT, Ministry of Communication & IT, Govt of India. Currently, the services are implemented over Ethernet, Wi-Fi and Bluetooth technologies, enabling horizontal and vertical handovers across them.

Keywords: IEEE 802.21, Heterogeneous Networks, Seamless Mobility, Media Independent Handover, MIH.

1. Introduction

A major trend in the communication era is the convergence of networking infrastructures and the integration of various types of access technologies to create a common hybrid network. However, the key challenge is to interconnect these networks to provide seamless mobility to portable wireless devices. To address this issue, IEEE 802.21 working group has specified an open standard called MIH Services [1].
This paper presents the design of IEEE 802.21 standard, through high level design descriptions, leading to an implementation of a prototype model capable of demonstrating vertical handovers among 802.3, 802.11 and 802.15.1 access technologies. The remainder of the paper is organised as follows. Section 2 gives a brief overview of the standard; Section 3 provides a high-level implementation overview and elaborates link information gathering methods; Section 4 presents empirical evaluation results and Section 5 concludes the paper by outlining the scope for future work.

2. Overview of IEEE 802.21

The purpose of IEEE 802.21 is to improve the user experience by providing a middleware i.e. MIH layer, which facilitates both mobile-initiated and network-initiated handovers. This functionality ensures seamless connectivity across same and different access technologies. As depicted in Figure 1 the standard consists of the following modules-

- MIHF (MIH functions) – This is a logical entity which provides abstract services to the higher layers through media independent interface and obtains information from lower layers through media specific interfaces. This layer provides the following functionalities-
  - Media Independent Event Service (MIES) – accumulates changes in state and transmission behavior of physical, data link and logical layers, or predict state changes in both local and remote interfaces and reports these events to the higher layers.
  - Media Independent Command Services (MICS) – provides a set of commands for both local and remote MIH users to govern the link state of the network interfaces.
  - Media Independent Information Services (MIIS) – provides information about neighboring networks including their location, properties and related services. The MIIS also allows this collective information to be accessed from any single network. For example, by using an IEEE 802.11 access network the MIH User gets information not only about all other IEEE 802 based networks in a particular region but also about 3GPP and 3GPP2 networks.
• Service Access Points (SAPs) - define both media-independent and media-specific interfaces. These are required for accessing the services provided by each entity. In particular, the SAPs include:
  - MIH_SAP, a media independent SAP that provides a uniform interface for higher layers to control and monitor different links regardless of access technology. This along with associated primitives is particularly defined for MIH_USER to access the services provided by MIHF layer.
  - MIH_LINK_SAP, a media specific SAP that provides an interface for the MIHF to control and monitor media specific links. For the MIHF to provide MIES and MICS for a specific link layer, it must implement the MIH_LINK_SAP for that specific link layer.
  - MIH_NET_SAP, a media-dependent SAP that provides transport services over the data plane on the local node, supporting the exchange of MIH information and messages with the remote MIHF.
• MIH users - functional entities that use MIH services. It comprises of higher layers which take the decision to trigger a handover based on the information and event services provided to it. This decision is executed using the command services.

3. Implementation

The general architecture of our prototype implementation is given in figure 2. The implementation of IEEE 802.21 consists of three major sub-systems which are again sub divided into many modules. The implementation of each sub-system and its modules are described below-

![Figure 2. Architecture of prototype implementation.](image)

3.1. MIHF Subsystem

The MIHF is a set of handover enabling functions within the protocol stack, which provide services to upper layers by facilitating a media independent interface for all the media specific services. The MIHF subsystem provides many services to the MIH_USER independent of the media being used. The services of MIHF are divided into three components – Event services,
Command services and Information services. A portion of this subsystem is majorly adopted from activities carried out by Heterogeneous Network Group of Institute of Telecommunications, Averio[2]. This component a C++ based generic implementation of Media independent handover Functions and their services.

3.2. MIH_LINK_SAP Subsystem

The standard defines MIH_LINK_SAP for each types of interface. Its main aim is to collect link specific information and provide it to the upper layers through media independent interfaces. Thus, for inclusion of each type of link, this subsystem implementation is to be done for that link.

All the parameters related to a link are divided into two major parts - State Parameters and Dynamic Parameters. State parameters are those which affect the status of the link. The link-up, link-down, link-detected etc. primitives, as defined in the standard are affected by these parameters. Dynamic parameters are dynamic information related to the specific link, for example, RSSI, SINR link level etc. are dynamic parameters for Wi-Fi interface. The Link_Parameter_Report and Link_Going_Down primitives are triggered by these parameters. The calculation of dynamic parameters is triggered by the state parameters as shown in figure3. For example, the calculation of link level for Wi-Fi interface is done only if the Link_Up indication is received from that interface.

![Figure 3. Link Event Collector.](image)

The following sections describe the implementation of this module for different type of links-

- **IEEE 802.3**: The major information required from Ethernet is the presence of the link i.e. the information collector should give the current state of the link. For link-down event ioctl command, SIOCETHTOOL [3] is used and for link-up event RTnetlink [4] sockets are used. Currently, there are no dynamic parameters analyzed in this access technology. The commands received by the MICS module are executed using the ioctl command and RTnetlink sockets.

- **IEEE 802.11**: The Linux Wireless Extension (WE) API [5] is a generic API allowing a driver to expose to the user space configuration and statistics specific to common Wireless LANs. The advantage of it is that a single set of tool can support all the variations of Wireless LANs and may be changed on the fly without resetting the device driver. WE APIs are based on a set of ioctl calls and the dynamic /proc file system [6]. The best way to identify a network interface as a Wi-Fi is to check if that interface supports WE APIs.
The state parameters are obtained from the WE APIs and RTnetlink sockets. These parameters include the presence of link and wireless extensions for that particular interface. Using these parameters primitives such as Link_Detected, Link_Up and Link_Down are implemented.

The dynamic parameters for this interface include link quality, link level and link noise. Where link level and link noise are measured at the receiver and are given either in dBm or in relative values. According to IEEE 802.11g specifications [7], the received RF energy is defined as RSSI and is a proportional value to signal noise. The link quality metric, defined in the standard, indicates the quality of received signal. According to WE, it is a combination of different measurements, such as percentage of retries, Signal-to-Noise Ratio (SNR), and missed beacons. Thus, these parameters are reliable and sufficient enough for event triggering purposes. These parameters are obtained continuously using the WE APIs.

The MIH_LINK_SAP module for Wi-Fi also maintains threshold levels which are configured dynamically by the MIH_USER or the peer MIHF using the Link_Configure_Thresholds primitive. Each time this module acquire the parameters, it compares the current values with the configured threshold and fires a Link_Parameters_Report primitive towards the MIHF, if the threshold is crossed.

The commands issued by local user and peer MIHFs are executed using WE APIs and RTnetlink sockets. Currently, we have implemented Link_configure_thresholds, Link_get_parameters and Link_actions (up, down and scan) commands in the Wi-Fi MIH_LINK_SAP.

IEEE 802.15.1: In particular, there is no support for Bluetooth interface according to the IEEE 802.21 standard. But the standard is flexible enough to accommodate new type of interfaces. As Bluetooth is one of the most widely adopted wireless communication protocol, we felt it relevant to include it in our implementation. This implementation was done by the use of BNEP (Bluetooth Network Encapsulation Protocol) [8] in Bluetooth stack.

The state parameters in this interface include the detection of the link and its current state (up or down). These parameters are obtained using the /sysfile system [9]. The dynamic parameters are same as that of Wi-Fi, except that the link noise level is not obtained. These parameters are acquired using the combination of ioctl command, HCIGETCONNINFO and hcitool [10]. The ioctl calls on this interface is made using the L2CAP sockets. Hcitool is a tool provided by BlueZ, which is an open source implementation of Bluetooth stack in Linux. This tool provides APIs like hci_read_rssi and hci_read_link_quality, which are helpful in attaining the RSSI and link quality respectively. The commands given to this interface specific MIH_LINK_SAP is executed using the hcitool.

3.3. MIH_USER Subsystem
This module is the highest part of the layered architecture, which acts as an end user of various services provided by the underlying layers. It majorly constitutes the policy engine and the higher layer mobility protocols. The description of the implementation of higher layer mobility protocols is out of the scope of this paper.

As per the standard, the key primitives issued by MIH_USER are the service primitives. The following are the description of these primitives-

- MIH_Capability_Discover - It is used to discover local or remote MIHF’s capabilities in terms of services (Event, Command and Information). In our implementation this is performed at the startup of the module and is achieved using the MIH protocol. The description of MIH protocol and its implementation is detailed in the later sections.

- MIH_Register and MIH_Deregister – These primitives are used to register and deregister the user to access specific services of MIHF. It is mandatory for the use of MIH command services and the push mode of MIIS. For reducing the complexity, the registration process is done soon after the capability discovery primitive is fired and it is deregistered before closing the application.

- MIH_Event_Subscribe and MIH_Event_Unsubscribe- This mechanism allows an MIH user to subscribe or unsubscribe a particular set of events that originates from a local or remote MIHF entity. In our prototype, the subscription of the events is done after discovering the capabilities of the MIHF. Only those events which are relevant to the user are subscribed. For example, a user which maintains the VOIP call continuity may not be interested in Bluetooth events, because the voice packets may not be directed towards the Bluetooth interface.

3.4. Communication

![Class diagram for message class](image)

For inter MIHF, communication between different subsystem of the implementation and peer message exchange, the standard specific MIH protocol is used. This makes it simple to construct and parse local and remote MIH messages. This protocol defines a header format, content and encoding for a set of protocol messages. The protocol messages are composed of eight byte long header part and a TLV (type-length value) encoded payload part. The header includes a MIH
message ID (MID) which in turn is composed of three other fields and is required to identify the service, type and the action of the message. It also includes a Transaction ID (TID) which is required to match the request and response of the message.

The communication module is implemented using the synchronous UDP message transfer APIs provided by Boost libraries [11]. The address of the destination is taken from the directory of the module, which is updated whenever a new MIHF or MIH user is discovered. The MID is generated from a lookup table and TID is generated using a random number generator API of Boost library. Figure 4 shown above depicts the class diagram for the implementing the construction and parsing of the MIH protocol messages.

4. Results

The modules and subsystems, as discussed in the previous section were implemented in C++ language and were tested under Linux operating system (kernel 2.6.32). For performing empirical evaluation of the framework, VOIP call was taken as the end application. The following table depicts the vertical handover time delay between different access technologies. The delay is calculated by taking the difference between average packet delay and the difference between timestamps of two consecutive packets received at the time of handover. All the time intervals were calculated using the RTP plugin of Wireshark [12].

Table 1. Handover delays for different scenarios

<table>
<thead>
<tr>
<th>Sno.</th>
<th>Scenario</th>
<th>Handover delay(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ethernet (802.3) to Bluetooth (802.15.1)</td>
<td>5510</td>
</tr>
<tr>
<td>2</td>
<td>Ethernet (802.3) to Wi-Fi (802.11)</td>
<td>1055</td>
</tr>
<tr>
<td>3</td>
<td>Wi-Fi (802.11) to Bluetooth (802.15.1)</td>
<td>3060</td>
</tr>
<tr>
<td>4</td>
<td>Bluetooth (802.15.1) to Wi-Fi (802.11)</td>
<td>645</td>
</tr>
</tbody>
</table>

From the table it is inferred that Bluetooth related handovers take maximum amount of time when compared to Wi-Fi & Ethernet technologies. This is possibly because of the inherent delay in Bluetooth protocol stack due to scanning of networks and encapsulation of IP (BNEP).

5. Conclusion

This paper presents an implementation of IEEE 802.21 as specified by the standard. We depicted how the MIH_Link_SAP module and event information collector can be implemented over various access technologies from various vendors. We also evaluated the implementation by calculating the handover delay between major access technologies which are popular across. The delay inferred can be significantly reduced by network topology server implementation i.e., MIIS services of the standard, which is left as future scope of the project.
Acknowledgements

We are grateful to the Department of Information Technology (DIT), government of India for encouraging our efforts in pursuing Research and Development in the areas of Heterogeneous Wireless Networks. This work is funded by DIT, Ministry of Communication and IT, Govt. of India under the project “Development of a State of the art Testbed for advanced Research in Multimedia over IP in an inter-networking environment for guaranteed & secured Quality of Service with content customization capability.”

References

10. Maxim Krasnyansky; Marcel Holtmann. Hcitool man page, URL: http://linux.die.net/

© 2011 by the authors; licensee Asia Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).