RFP 154 – Network Interface Information Service

Draft

7 Pages

Abstract

This document describes the need for a function that can obtain information from network interfaces in an OSGi environment and the requirements regarding what is to be defined as an OSGi Specification.
0 Document Information

0.1 Table of Contents

0 Document Information.................................................................................................................. 2
  0.1 Table of Contents.................................................................................................................. 2
  0.2 Terminology and Document Conventions............................................................................. 2
  0.3 Revision History.................................................................................................................. 2

1 Introduction................................................................................................................................. 3

2 Application Domain.................................................................................................................... 4

3 Problem Description.................................................................................................................. 4

4 Use Cases.................................................................................................................................. 5

5 Requirements............................................................................................................................. 6

6 Document Support...................................................................................................................... 6
  6.1 References.......................................................................................................................... 6
  6.2 Author’s Address................................................................................................................ 7
  6.3 End of Document................................................................................................................ 7

0.2 Terminology and Document 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 5.1.

Source code is shown in this typeface.

0.3 Revision History

The last named individual in this history is currently responsible for this document.
1 Introduction

Java standard APIs (i.e. java.net.NetworkInterface, java.net.InetAddress) provide functions that allow IP network interface information, such as the IP address and MAC address to be obtained.

However, the bundle that wants to get network interface information has to monitor whether the information has changed or not for a certain period of time. Changes in network interface can be pushed to the bundles concerned, the need for polling by bundles can be eliminated.

In addition, some information cannot be obtained via Java standard APIs.

This RFP describes the need for a mechanism that notifies concerned parties of changes in the network interface and a new API that provides information not obtainable from the standard Java APIs as well as the corresponding requirements.
2 Application Domain

There are many bundles that use the IP network to communicate with other networked devices. In particular, since a Residential Gateway (RGW) may have a number of network interfaces, each bundle running on the RGW needs to obtain an IP address and confirm whether the network interface associated with the allocated IP address suits the bundle’s requirements or not.

For example, a protocol adapter needs the IP address of a network interface on the wide area network side to communicate with an external server. UPnP device service bundle needs the IP address that can be used to communicate with devices in a local area network.

These bundles can acquire information about the network interface via the following Java standard APIs.

- java.net.NetworkInterface
- java.net.InetAddress

3 Problem Description

Many application bundles on the RGW provide services on IP networks. For example, a protocol adapter for DMT Admin Service, a http server established by HTTP Service bundle and UPnP device service bundle use IP networks. In those cases, the bundles need to get information about the network interface on the RGW such as IP address, MAC address, network interface name, and so on.

The information about the network interface can be obtained by using Java standard APIs which are java.net.NetworkInterface and java.net.InetAddress. However, these APIs fail to provide the features needed by the bundles when they use the IP network in the following situations:

[Problem 1] There is no feature that sends a notification when information of the network interface (i.e. IP address) changes during runtime, e.g. the connection status or the assigned IP address.

[Problem 2] There is no feature that can acquire the subnet mask of the network interface.

[Problem 3] Operating System specific bundles must be prepared because some information about network interface depends on the Operating System.
If these functions were available, it would be very useful for bundles that need to use the IP network. However, a standard API does not exist at this time, so it must be prepared for each environment. **3.1 Use Cases**

**Use case 1**

The TR-069 protocol adapter bundle on a RGW needs to communicate with an Auto Configuration Server (ACS). The ACS needs to know the public IP address of the Residential Gateway to send a UDP packet to the protocol adapter bundle for a connection request. In this case, the bundle has to provide the IP address to the ACS when the bundle is started or the IP address has changed.

**Use case 2**

When an HTTP Service bundle is available, at least one HTTP server is expected to run. When the HTTP server needs to be assigned to a specific network interface, the HTTP Service bundle has to know the information of the network interface. In addition, the HTTP Service bundle needs to know when the IP address of the network interface being managed changes.

**Use case 3**

The UPnP Device Service bundle needs to create the DatagramSocket for receiving and sending M-search messages. In the case of devices such as Residential Gateway, which has multi network interfaces, the UPnP bundle has to create a DatagramSocket that is bound to an appropriate local IP address. Therefore, the UPnP bundle needs to know the current IP address of the network interface and the replacement IP address.

**Use case 4**

An application bundle wants to obtain the subnet mask of the IP address to cover the situation in which the bundle needs to execute the Wake-up-On-LAN process.

**Use case 5**

An application wants to obtain information about available network services, such as available DNS Server, Log Server, NTP Server, or network characteristics, such as domain names, ARP cache timeouts, broadcast address, etc. For this, the local DHCP server can be queried to get those information.

**Use case 6**

A device running an OSGi framework in an mixed IPv4/IPv6 environment needs to get specific information about the network interface(s) in order to provide, for example, different services for the IPv4 and IPv6 environments.
4 Requirements

[REQ_1] The solution MUST provide means to send notifications to interested bundles whenever the information of network interface has changed. (i.e. The bundle is notified the information of IP address change from Network Interface Information Service implemented bundle)

[REQ_2] The solution MUST provide an API that can obtain information from a multiple network interfaces (An application bundle needs to know whether the network interface is a LAN interface or a WAN interface.).

[REQ_3] The solution MUST provide a mechanism that can provide the network interface information needed regardless of the Operating System type.

[REQ_4] The solution MUST provide the means of configuring network interface type. It will be defined for each environment (i.e. “LAN”, “WAN” that is bound to each logical interface).

[REQ_5] The solution MUST provide an API that can obtain the subnet mask of each IP address.

[REQ_6] The solution MUST support both IPv4 and IPv6 environments (mixed or separately) and the corresponding characteristics, for example IPv4 and IPv6 addresses, multi-prefixes, multicast etc. .

[REQ_7] The solution SHOULD provide information about the physical type of network interface, e.g. ethernet, DSL, fiber, WIFI.

[REQ_8] The solution SHOULD support the retrieval of MAC addresses for network interfaces.

[REQ_9] The solution SHOULD provide an API that can obtain the list of BOOTP/DHCP command options.

[REQ_10] The solution MAY provide an API that allows alteration of network interface configurations.

5 Document Support

5.1 References


5.2 Author's Address
<table>
<thead>
<tr>
<th>Name</th>
<th>Shigekuni KONDO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>NTT Corporation</td>
</tr>
<tr>
<td>Address</td>
<td>1-1, Hikari-no-oka, Yokosuka-shi, 238-0847, Kanagawa, Japan</td>
</tr>
<tr>
<td>Voice</td>
<td>+81 46 859 3444</td>
</tr>
<tr>
<td>e-mail</td>
<td><a href="mailto:kondo.shigekuni@lab.ntt.co.jp">kondo.shigekuni@lab.ntt.co.jp</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Andreas Kraft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Deutsche Telekom AG</td>
</tr>
<tr>
<td>Address</td>
<td>Telekom Innovation Laboratories, Winterfeldtstrasse 21, 10781 Berlin, Germany</td>
</tr>
<tr>
<td>Voice</td>
<td>+49 30 835358-185</td>
</tr>
<tr>
<td>e-mail</td>
<td><a href="mailto:Andreas.kraft@telekom.de">Andreas.kraft@telekom.de</a></td>
</tr>
</tbody>
</table>

5.3 End of Document