Automatic Over-the-Air Provisioning for Wi-Fi Equipped M2M Devices

Kenji Hori, Tomohiko Ogishi
Smart Network Administration Lab.
KDDI R&D Labs. Inc.
Saitama, Japan
[hori, ogishi]@kddilabs.jp

Ming-Yee Lai, Dana Chee, Kaustubh Sinkar
Applied Research
Applied Communication Sciences
Basking Ridge, New Jersey, U.S.A
[mlai, dchee, ksinkar]@appcomsci.com

***Abstract*— In this paper, we first describe the challenges of provisioning Wi-Fi equipped M2M devices, which may be behind Network Address Translation (NAT). We then articulate an automatic over-the-air provisioning mechanism that includes four steps: device discovery, user account setup for device, ESSID/IP configuration, and device registration (including NAT tunnel configuration). The mechanism enables M2M service providers to manage the registered M2M devices and allows M2M users to access these M2M devices behind NAT using a unique fully qualified domain name (FQDN). Finally, we elaborate the performance analysis and technical discussions related to the automatic provisioning mechanism.**

Keywords- M2M; device management; configuration management;

I. INTRODUCTION

Machine-to-machine (M2M) devices are the devices that have communication interface(s) to interact with other devices or servers, but have no or limited human interface. The examples M2M devices are smart meters, cleaning robots, smart appliances, home security systems, e-health monitors, and telemactics on-board-units. By this definition, smart phones, tablets, and laptops are not M2M devices.

M2M communication is mainly data communication, in contrast to voice communication dominant in human communication. The communication interface(s) in a M2M device can be a Wide Area Network (WAN) interface, a Local Area Network (LAN) interface, or combinations of them. A WAN can be a wireless WAN (WWAN, e.g., 3/4G) or a fixed (wireline) WAN (FWAN, e.g., ADSL, FTTH). Similarly, a LAN can be a wireless LAN (WLAN, e.g., Wi-Fi, Bluetooth) or a fixed LAN (FLAN, e.g., Ethernet).

Many M2M devices (e.g., cleaning robots, smart appliances) distributed via open retail sales channel (e.g., amazon.com, Best Buy) only have a WLAN interface, which may be used to connect with a gateway to WAN if a wide area remote management is called for. Such M2M devices are difficult to manage as there is no established automatic over-the-air provisioning mechanism like that based on OMA-DM[1], which is a device management protocol widely used for cellular handsets with a direct interface with WWAN. Although OMA just completed the 1st version of the Gateway Management Object (GwMO) technical specification [2], the GwMO adaption mode for Wi-Fi equipped M2M devices has not been addressed yet. This paper sheds lights on what needs to be considered in the evolution of GwMO via an important gateway area wireless protocol – Wi-Fi.

A Wi-Fi equipped device may have its Wi-Fi initially configured to be in infrastructure mode or in ad hoc mode. Most of the laptops and smart phones have their Wi-Fi initially configured in infrastructure mode which requires users to perform manual operations on the devices to set the Wi-Fi parameters (e.g., Setting ESSID and WEP key by pressing down the buttons for the Wi-Fi Protected Setup (WPS) [3]). Since M2M devices may be placed in a location hard to reach or in large quantity, they are best initially configured in ad hoc mode for automatic provisioning. With ad hoc mode, the gateway could access the M2M devices without requiring manual operations on the devices..Once an M2M device is connected with the gateway, it needs to be configured to communicate with servers, controlling hosts, and other M2M devices over the Internet .

In this paper, we describe an automatic provisioning mechanism for all types of Wi-Fi equipped M2M devices with Wi-Fi initially configured in ad hoc mode. The mechanism includes four main steps: device discovery, user account setup, ESSID/IP configuration, and device registration. The mechanism also enables M2M service providers to manage the registered M2M devices and allows M2M users to access these M2M devices (which may or may not be behind NAT) anywhere using a unique fully qualified domain name (FQDN), instead of IP address and port number.

We have implemented the automatic provisioning mechanism in our "MOPAD" (M2M OTA Provisioning and Activation for Diverse devices) Phase 1 project successfully. The implementation, among other things, uses the Wi-Fi equipped surveillance robots as the M2M devices and IETF Constrained Application Protocol (CoAP) [5] as the device management protocol, which is UDP-based and lighter weight than the OMA HTTP/TCP based SyncML [1]. The surveillance robots ("Rovio") with web camera, indoor navigation, and auto charging capabilities can operate in a residential or working environment autonomously and semi-autonomously (i.e. remote control option to override pre-recorded paths). Performance measurements and analysis are performed for the whole mechanism and four individual steps. In this paper, we focus on the analysis on elapse time to execute the automatic provisioning mechanism.

978-1-4799-0537-9/13/$31.00 ©2013 IEEE

II. CURRENT METHOD AND PROMBLEMS OF PROVISIONING M2M DEVICES

A. Overview of Current Provisioning of M2M Devices with WiFi Interfaces

Figure 1 shown below is used to illustrate the current provisioning method. The M2M device A with the Wi-Fi interface is provisioned manually by a local configuration host B (e.g., a laptop PC) to be accessible by a remote controlling host C, which could be a smart phone with public IP address, or another M2M Device D with a private address.

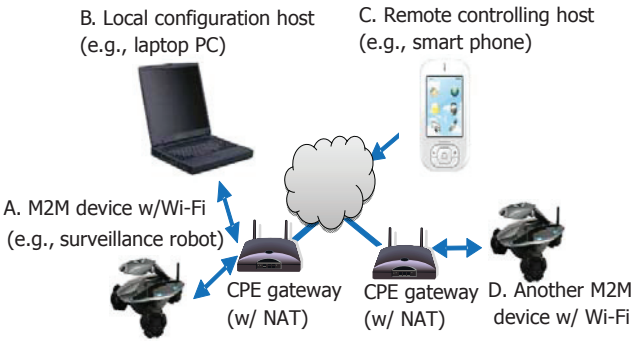

Figure 1. Architecture of Manual M2M Device Provisioning.

Note that a M2M device with Wi-Fi interface can connect to a WAN via a wireless router (i.e., a Wi-Fi access point) and a WAN gateway (e.g., a cable/DSL/fiber/WiMAX modem, router, or a 3G/4G radio device) or a CPE gateway (e.g., a 3G/4G Mi-Fi device, or a combined box with the Fiber modem and wireless router functionalities) which combines the functions of wireless router and WAN gateway. The CPE gateway also provides the network address translation (NAT) function for M2M devices with a private IPv4 address. In Figure 1, a CPE gateway is used to connect A and D.

A typical manual provisioning procedure of a M2M device (e.g., [6], [7]) with Wi-Fi interface is as follows:

(1) User U connects the M2M device with the CPE gateway via its Wi-Fi interface initially configured in ad hoc mode using the local configuration host (B in Figure 1) connected to the same gateway,
(2) User U configures the M2M device with a dynamic DNS server in the IP network to enable remote Internet access to the device behind the CPE gateway,
(3) User U configures the device in Wi-Fi infrastructure mode with IP address (private or public) and external IP port number,
(4) User U configures the CPE gateway with port forwarding for remote access.

After the above manual provisioning procedure is complete, User U can use a remote controlling host with the above IP address and port number to access the provisioned M2M device via Internet.

B. Problems with Current Provisioning of M2M Devices with Wi-Fi Interfaces

The problems with current provisioning of M2M devices with Wi-Fi interface are as follows:

(a) The current provisioning method requires manual procedures to connect to a WAN via a CPE gateway; (b) The manual procedures require the user of the M2M devices to have expertise in configuring Wi-Fi and IP networks; (c) The manual procedures require trial-and-error even for users with knowledge in Wi-Fi and IP networking; (d) Incorrect manual provisioning setup and steps may cause lengthy service interruption of active devices while provisioning a new device in the same gateway area; (e) The M2M devices of different types or made by different vendors have their own distinct or varying provisioning and configuration procedure; (f) For a large number of M2M devices in a gateway area (e.g., in office, factory, service area, multi-unit dwelling, house), manual provisioning takes time and efforts even for professional installers; (g) The manually provisioned devices cannot be managed by a M2M or network service provider centrally; (h) When a M2M device is reset to a factory default setting (e.g., after power failures), the user needs to repeat the manual procedure for re-provisioning; (i) The manual provisioning or re-provisioning procedure requires a user being physically in the gateway area.

III. MOPAD-I ARCHITECTURE

Figure 2 shows the MOPAD Phase 1 (MOPAD-I) architecture with a fixed (i.e., equipped with an FLAN interface) Device Management Gateway (abbreviated as DM Gateway for the rest of the paper). Besides the FLAN interface that connects to the service provider core network through a CPE gateway and routers, the DM Gateway has Wi-Fi interface(s) with the M2M devices in the gateway area. The service provider core network area contains the Device Management (DM) Server and the DNS Server. The laptops and smart phones in the remote access area serve as controlling hosts to operate the M2M devices from remote.

All laptops used in the architecture, including those running the software of DM Gateway, DM Sever, DNS Server, controlling hosts, router, are powered by Fedora 14 Linux. The smart phones run on Android 2.3 OS.

As M2M device, we use Rovio surveillance robot [6], which has an onboard web server enables access to Rovio from anywhere.

The MOPAD-I DNS Server, used for M2M device automatic provisioning, is different from the commercial dynamic DNS server in that the 65534 record type (also called record type for NAT3D, as mentioned in the next paragraph) is supported to allow connectivity behind a NAT without user's manual configuration for port forwarding as described in Section II.B. A commercial dynamic DNS supports record type A for FQDN to IP address mapping, but not the 65534 record type.

The automatic provisioning mechanism implemented in MOPAD-I supports wireless and wired controlling hosts either behind NAT or not. NAT Through Tunneling (NATTT) [8] is used to enable those controlling hosts to access M2M devices behind NAT. NATTT Demon (NAT3D) server software runs on the DM Gateway while NAT3D client software runs on the controlling hosts and the DM Server. Since NATTT addressing mapping is used only after initial provisioning, its detailed operation on inner and outer (source IP, destination IP) pairs in NAT3D server and client will not be discussed for the rest of this paper which focuses on initial provisioning.

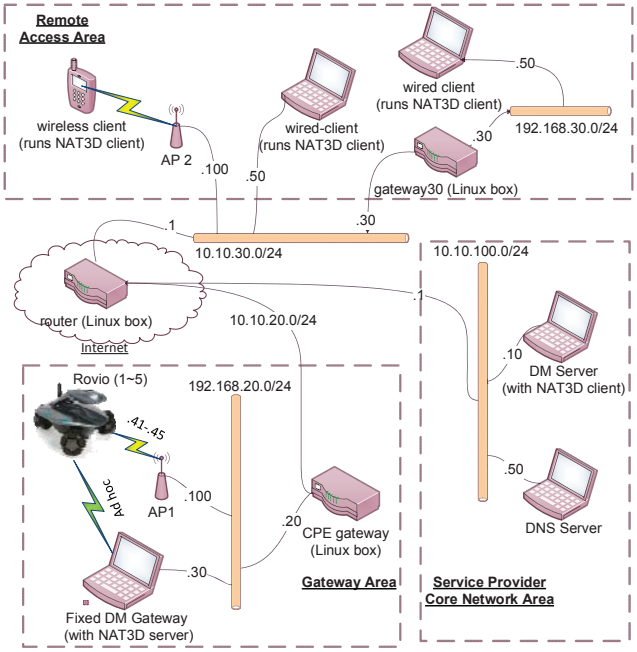

Figure 2: MOPAD-I Architecture - Fixed DM Gateway.

IV. AUTOMATIC PROVISIONING MECHANISM AND IMPLEMENTATION

We consider the scenario where the M2M Devices (Rovio) and the Controlling Android handset or wired laptops are both located inside a private network (behind a NAT), Figure 3 shows the automatic initial provisioning flow for M2M devices (Rovios in MOPAD-I project). The flow is decomposed into the following four main steps:

1. Device discovery: The DM Gateway continuously scans for available t ESSID of M2M devices with Wi-Fi in ad hoc mode. When the DM Gateway discovers one, it gets the MAC address and additional information such as the version of software/firmware. In the MOPAD-I implementation, the Rovio the API [9] is used to get the MAC address and firmware version of the device.

2. Device discovery: The DM Gateway continuously scans for available t ESSID of M2M devices with Wi-Fi in ad hoc mode. When the DM Gateway discovers one, it gets the MAC address and additional information such as the version of software/firmware. In the MOPAD-I implementation, the Rovio the API [9] is used to get the MAC address and firmware version of the device.

3. User account setup: The DM Gateway enables the device, performs mutual authentication, and sets up account for Wi-Fi equipped M2M devices containing built-in web server. Credentials can be pre-provisioned to avoid manual interactions.

4. IP address and ESSID configuration: The DM Gateway switches the connection of the Rovio from ad hoc mode into the infrastructure mode and sets (if manually assign) or gets (via a DHCP server) the IP address using the Rovio API. During IP address and ESSID configuration, a challenging sub-step is also performed.

5. Device registration: The DM Gateway registers the Rovio (M2M Device) to the DM Server by sending the name, MAC address, and IP address information. The DM Server processes this information, interacts with the DNS Server, and returns the Fully Qualified Domain Name (FQDN) for the Rovio to the DM Gateway.

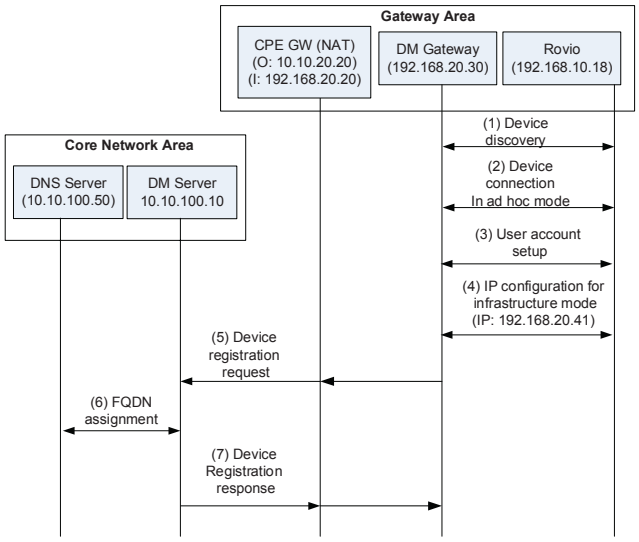

Figure 3: Automatic Provisioning Flow.

In Figure 3, when the Rovio is connected in ad hoc mode, its IP address is set to 192.168.10.18, which is a default factory set IP address for any Rovio. After connecting in infrastructure mode, the IP address becomes 192.168.20.41.

IV. PERFORMANCE EVALUTAION

In Figure 3, the messages <1>, <2>, <3>, <4> for device discovery, device connection in ad hoc mode, user account setup, and IP configuration use HTTP and TCP as the transport protocol transmitted over local area network in the gateway area.

The registration request/response messages (<5>, <7> in Figure 3) are sent using Constrained Application Protocol

(CoAP) [5] over UDP as the transport protocol transmitted over a wide area network.

The elapse time measurements for these steps among our tests are consistent. Six sample results are shown in Table I. The User account setup, IP and ESSID configuration, and device registration are the three most time consuming steps, which takes about 3.2 seconds, 3.1 seconds, and 0.6 second respectively.

TABLE I. ELAPSE TIME MEASUREMENTS (MSEC)

Test	Discovery		User Account Setup			IP and ESS-ID Configuration		Registration	Total
	MAC	*Ver*	*Enable*	*Authentication*	*Account*	*Authentication*	*IP & ESS-ID*		
1	4.5	5.4	1552	5.5	1581	10.4	3150	539	6849
2	5.0	5.1	1570	6.5	1575	7.2	3163	233	6566
3	5.1	6.9	1577	6.5	1572	10.2	3140	565	6884
4	5.5	10.7	1584	5.0	1571	6.4	3150	562	6896
5	10.8	5.8	1562	5.6	1584	6.8	3152	228	6556
6	6.1	5.3	1598	21.7	1583	6.7	3189	567	6978

For ESSID and IP address configuration, the link layer (L2) switching delay from ad hoc mode to infrastructure mode is caused by L2 authentication and association, flushing Address Resolution Protocol (ARP), updating the routing table, and getting a new IP address.

For the device registration step, the interaction between the DM Server and the DNS Server is most time consuming. Specifically, when the DM Server receives a registration request, it needs to update the DNS server (see "5. Device Registration" in Chapter IV) before creating a new CoAP message containing the FQDN of the M2M device and then sends the information to the DM Gateway.

From the Total column in Table I, the total time for automatic provisioning of a Rovio is around 7 seconds, which is significantly lower than the time for manual provisioning that takes in minutes even for Wi-Fi and IP experts. The time saving from automatic provisioning is critical to M2M service providers and M2M users, especially when the number of Wi-Fi equipped M2M devices in a gateway area is large.

V. DISCUSSION

In the case where there is only one Wi-Fi connection to the M2M device from the DM Gateway and the CPE gateway within a gateway area, the Wi-Fi radio needs to operate alternatively between the ad hoc mode for initial provisioning of new M2M devices and the infrastructure mode for normal operation of provisioned devices. The switching from the infrastructure mode to the ad hoc mode incurs Internet connection loss, and thus service lockout for the provisioned M2M devices.

To avoid the service lockout issue, the DM Gateway needs to have a Wi-Fi connection with the M2M device for provisioning and a non-Wi-Fi connection (e.g., Ethernet, Bluetooth, or internal bus) with a Wi-Fi access point and the CPE gateway to switch between Wi-Fi ad hoc model and infrastructure mode, as shown in Figure 2, where the laptop running with the DM Gateway software also has a Wi-Fi interface that connects the provisioned M2M devices in infrastructure mode for Internet connection. In contrast, if the DM Gateway is implemented in a laptop without Wi-Fi interface or in a smart phone with single Wi-Fi radio used for stand-alone Wi-Fi (to connect with the M2M device in ad hoc mode) and Mi-Fi (to connect to Internet in infrastructure mode) alternatively, then the service lockout issue needs to be addressed using other solutions.

VI. CONCLUSION

In this paper, we describe an efficient (short elapse time) automatic provisioning mechanism for Wi-Fi equipped M2M devices initially configured in ad hoc mode. The mechanism includes four steps: device discovery, user account setup, ESSID/IP configuration, and device registration. The mechanism enables M2M service providers to manage the registered M2M devices remotely, M2M devices to communicate with servers and other devices (behind NAT or not), and allows M2M users to access M2M devices anywhere using a unique fully qualified domain name (FQDN.

ACKNOWLEDGMENT

We express our thanks to Kiyohito Yoshihara and Akira Idoue of KDDI R&D Labs. as well as Christian Makaya and Joe Lin affiliated with Telcordia during the MOPAD Phase 1 project for their invaluable contributions to the MOPAD-I design and prototyping which the paper is based on. This work is partially supported by Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan.

REFERENCES

[1] "OMA Device Management Protocol , " Version 1.2.1, June, 2008

[2] "OMA Gateway Management Object Technical Speification, Version 1.0, " Dec., 2012.

[3] "Wi-Fi Alliance: Wi-Fi Simple Configuration Technical Specification, Version 2.0.2, " Jan, 2012.

[4] P. Srisuresh, B. Ford and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", IETF RFC5128, March, 2008.

[5] Z. Shelby, K. Hartke, C. Bormann and B. Frank, "Constrained Application Protocol (CoAP)," IETF Internet Draft, http://tools.ietf.org/html/draft-ietf-core-coap-06, May, 2011.

[6] "WowWee Rovio IPcam Surveillance Robot," http://www.wowwee.com/en/products/tech/telepresence/rovio/rovio.

[7] "Parrot AR.Drone User Guide," http://ardrone2.parrot.com/media/uploads/support_ardrone_1/ar.drone_user-guide_uk.pdf.

[8] B. Zhang, "NATTT", http://www.cs.arizona.edu/~bzhang/nat/.

[9] "API Specification for Rovio Version 1.2," http://www.wowweesupport.com/pdf/Rovio_API_Specifications_v1.2.pdf, Oct., 2008.