Communication in Machine-to-Machine Environments

Iva Bojic  
University of Zagreb  
Faculty of Electrical Engineering and Computing  
HR-10000 Zagreb, Croatia  
iva.bojic@fer.hr

Damjan Katusic  
University of Zagreb  
Faculty of Electrical Engineering and Computing  
HR-10000 Zagreb, Croatia  
damjan.katusic@fer.hr

Mario Kusek  
University of Zagreb  
Faculty of Electrical Engineering and Computing  
HR-10000 Zagreb, Croatia  
mario.kusek@fer.hr

Gordan Jezic  
University of Zagreb  
Faculty of Electrical Engineering and Computing  
HR-10000 Zagreb, Croatia  
gordan.jezic@fer.hr

Sasa Desic  
Ericsson Nikola Tesla  
R&D Center  
HR-10000 Zagreb, Croatia  
sasa.desic@ericsson.com

Darko Huljenic  
Ericsson Nikola Tesla  
R&D Center  
HR-10000 Zagreb, Croatia  
darko.huljenic@ericsson.com

ABSTRACT

It has been estimated that by the end of 2020 there will be 50 billion connected devices in Machine-to-Machine (M2M) networks. Such projections should encourage us to deal with the corresponding problems in heterogeneous M2M systems. First of all, devices can communicate through different access technologies (e.g. wireline, 2G/3G, WiFi, Bluetooth) and their communication can be classified as direct or indirect, internal or external. In this paper we explain differences between those types of communication and propose a new identification scheme that allows M2M devices to establish communication in every possible way. Secondly, there is a problem of device hardware and software diversity. To overcome this problem, we propose the usage of the Open Service Gateway Initiative (OSGi) framework.

1. INTRODUCTION

The trend of today's technology development is to provide networking and collaboration between various computing machines. The goal is to expand the possibilities of devices used in daily life (e.g. PCs, smartphones, positioning devices, health monitors) by connecting them together to achieve new functionalities or create new services which can improve the quality of life. Machine-to-Machine (M2M) communication [9] is a concept that defines the rules and relationships between devices while cooperating. It implies a highly automated usage of a set of devices simultaneously, without any need for human interaction.

Since the number of connected devices is rapidly growing, the M2M concept is gaining significance. The idea is to connect a plethora of different devices that communicate through different technologies (e.g. wireline, 2G/3G, WiFi, Bluetooth) and thus create a heterogeneous environment.

System architecture for the M2M system can be found in [7]. Another challenge in M2M networks is to overcome devices' hardware and software diversity.

In our previous work [6], we addressed the problem of hardware and software diversity in M2M environment introducing a context-aware framework for sensor data retrieval from mobile devices. Mobile operating systems are of help in M2M application development since they enable running the same software on devices that are different hardware-wise. On the other hand, the downside is that there are a lot of operating systems on the market making software development a demanding task. We overcame the problem of various existing mobile operating systems (e.g. Android, iOS, Windows Phone 7) by using the Open Service Gateway Initiative (OSGi) framework. With solutions like the Prosys mBS it is possible to run the same software on operating systems like Android and Symbian, even on embedded low power hardware hosting Java Virtual Machine (JVM) [11].

The contribution of this paper is twofold: in Section 2 we describe different types of communication (i.e. internal, external, direct and indirect) in M2M environment, while Section 4 presents a new identification scheme that supports previously explained communication. Section 3 presents related work in a field of addressing and identification and Section 5 elaborates design and implementation of a proof-of-concept described as a case study. Section 6 concludes the paper and proposes directions for future work.

2. TYPES OF INTERACTIONS

The industry has become more active in the standardization process in M2M domain because of growing number of connected devices. Although M2M is mostly related to the application level, which is an area typically outside the scope of the standard bodies, some efforts have been made.

Wireless access standard groups such as 3GPP and ETSI are looking into the impacts to the existing network due to the potentially heavy use of M2M devices. In our previous work [5] we presented some of the standards from the 3GPP and ETSI. Additionally, Chang et al. [3] presented a view at the progress made in developing M2M communication standards. In this paper, we will mention only those standards that are dealing with requirements for M2M communication.
Because of the differences in communication technologies that are supported in M2M networks, none of the existing standards defines the entire process of communication. However, ETSI's Technical Specification 102 689 M2M service requirements [4] identified general requirements for the correct establishment of M2M communications. Additionally, 3GPP Technical Report 22.888 System Improvements for Machine-Type Communications [2] explains some of the end-to-end aspects of communication between M2M devices and servers.

Therefore, in this paper we identify four different types of communication in M2M environment. Without loss of generality, these types of communication will be explained in communication process between two devices (see Table 1). When both M2M devices are within the same private address space (no matter if it is an IPv4 or IPv6 address space or some other) their communication is called internal. When two devices can communicate without any intermediary (e.g. Bluetooth ad-hoc network), then their communication is direct, otherwise indirect (e.g. WiFi network). Communication in M2M system is external and direct when both M2M devices use public address space. And finally, when one device has public address and the other one private, then communication between them is indirect and external.

### Table 1: M2M Communication Types

<table>
<thead>
<tr>
<th>Type</th>
<th>Direct</th>
<th>Indirect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Both M2M devices are within one private address space (without intermediary).</td>
<td>Both M2M devices are within one private address space (with intermediary).</td>
</tr>
<tr>
<td></td>
<td>A device may be able to communicate in a P2P manner with any other device.</td>
<td>The M2M system shall be capable of interfacing heterogeneous M2M area networks.</td>
</tr>
<tr>
<td></td>
<td>Both M2M devices have address from the public address space.</td>
<td>One device has public while the other has private address.</td>
</tr>
<tr>
<td></td>
<td>The system should be able to allow communication between device and gateway, by using multiple communication means.</td>
<td>The system should be able to communicate with devices behind a M2M gateway.</td>
</tr>
</tbody>
</table>

Table 1 also summarizes requirements for M2M communication from ETSI's technical specification [4]. This standard defines that the M2M system should be capable of interfacing heterogeneous M2M area networks and proposes that this is achieved at the M2M gateway. Characteristics of the M2M gateway may be either preconfigured in the M2M system or provided by the M2M gateway to the M2M system.

Although this standard does not define M2M NAT (Network Address Translator), we distinguish two types of intermediaries: M2M NATs and gateways. Difference between them is that the M2M NAT has limited routing capabilities and does not possess any kind of logic, while the M2M gateway is network component that has ability to store information about M2M devices that are located behind it.

### 3. RELATED WORK

Type of communication has a direct effect on identification and addressing processes that are necessary prerequisite for any communication. ETSI's technical specification [4] defines general requirements for naming, numbering and addressing in M2M environments. In 3GPP's technical report [2] identification and addressing processes are explained for 3GPP networks (e.g. UTRAN) and IP networks, while another 3GPP's technical report [1] explains special requirements considering addressing based on IMSIs and MSISDNs.

General requirements for identification and addressing in M2M systems are given in this technical specification [4] the M2M system should be able to reach the M2M devices or M2M gateways using M2M device names or M2M gateway names respectively. The M2M system should be flexible in supporting more than one naming scheme. The M2M system should also allow flexible addressing schemes, including: IP and E.164 (e.g. MSISDN) addresses of M2M devices, and addresses of groups of devices (including multicast addresses).

Group addressing scheme for 3GPP networks was proposed in application for US patent [8] under title M2M group based addressing using cell broadcast service. They benefited from Cell Broadcast Service defined in [12]. First step in their method is identification of a group of devices that the message should be sent to. Then, after the group is indicated in a cell broadcast service message and the message is sent to the identified group.

The standard [2] defines two types of identifiers for 3GPP networks: external and internal. External identifier is used by M2M servers outside the 3GPP networks, to identify a M2M device within the 3GPP system. Internal identifier is the subscription related identifier used within the 3GPP system to uniquely identify a M2M device (e.g. IMSI), and is not necessarily known by the M2M server. Within IP networks three possibilities of M2M addressing are analyzed.

IPv6-based addressing, in which both the M2M server and device are located in the IPv6 address space, is considered the primary addressing solution. The number of available IPv6 prefixes in IPv6 address space is abundant and thus there is almost no limitation for its usage. Second scenario involves the M2M server located in a public IPv4 address space and a device with assigned private IPv4 address, while in the third scenario the M2M server is located in a private IPv4 address space, same as the M2M device.

Technical report [1] also elaborates different addressing schemes that could be used in M2M systems. IMSI based addressing provides only limited connectivity in the M2M system but could serve as a first step of a remote activation of an M2M device. In theory, this addressing would provide the possibility for addressing up to 10 billion different devices within one mobile network. However, in practice this may collide with existing IMSIs of the operators. Therefore, operators would very likely be forced to apply a separate machine network management for M2M communication.

IMSI based addressing is also proposed in application for US patent entitled Method and apparatus for solving limited addressing space in M2M environments [10]. This patent deals with methods and apparatus for M2M communication in limited address space. They propose a group based communication between controller and a group of wireless transmit/receive units where these wireless units within the same group have the same IMSI.
Furthermore, MSISDN based addressing would also provide the possibility for addressing up to 10 billion different terminals. But because of existing numbering plans, the real number is much more limited and would also require a separate machine network management for M2M communication, same as in IMSI based addressing. And finally, the IP addressing (IPv6) would provide a large number of addresses but would not support devices that do not have IP connectivity (e.g. sensors).

4. NEW IDENTIFICATION SCHEME

It can be concluded from a previous section that addressing and identification in M2M are still open issues. There are some standards and applications for patents that address this issue, but they do not simultaneously support heterogeneity and scalability in M2M systems as we believe our approach does. In both situations (i.e. 3GPP and IP networks) identification process is tightly connected with addressing scheme. However, as mentioned before, M2M devices can use different network technologies for communication (i.e. not only IP). For those purposes, we propose identification scheme independent of used communication technologies where device’s identifier is constant and unchanged for the whole time, while current address can change in respect to communication technology currently used.

The unique M2M identifiers. Our identifier has 27 bytes and consists of four parts, where each part represents a single identifier for a specific hardware component (see Figure 1). Length of 27 bytes should ensure that this identifier is unique and unchangeable in the whole M2M system the whole time (scalability), while usage of different hardware component marks should support heterogeneity.

The first part of our identifier is the IMEI that presents a number of a mobile device and has a length of 15 digits. The second part, which length is also 15 digits, is an IMSI number. IMSI number connects a subscriber with a particular telecommunication operator. The third part is the WLAN or Ethernet MAC address, while the fourth part is Bluetooth MAC address. Both 6 are bytes long. It should be noted that the first two parts of identifier are numeric values, while the MAC addresses are stored in hexadecimal notation.

The variable M2M addresses. After investigating different possibilities for identification of M2M devices, and have proposed one format for globally unique ID for each M2M device in M2M network, we will present types of communication during the software update process. As mentioned in the previous section, M2M devices can be located within public IP space, behind M2M NATs or gateways. The first communication scenario is classified as direct and external, while the second and third one are indirect external. In this paper we will explain the first and second scenario (see Figure 2), while the third one will be explained in future work.

In these three communication scenarios every M2M device has its own identifier that does not change over time and address that is variable. If we take for example that M2M device is part of an IP network, then in the first case M2M device would have public IP address (IPv4 or IPv6), and in the remaining two cases private IP addresses.

5. CASE STUDY

This case study presents the process of a software update. It consists of identification, addressing, software provisioning and configuration management. In software provisioning process, there is a need for the M2M server to be the initiator of communication (i.e. push mechanism) towards the M2M device that needs a new version of software (e.g. because of the need for automated software provisioning process). Moreover, mechanisms for software provisioning and configuration management should support remote software updates (i.e. without visiting the equipment).

The purpose of this case study is to show how the proposed identification scheme fits in different types of communication during the software update process. As mentioned in the previous section, M2M devices can be located within public IP space, behind M2M NATs or gateways. The first communication scenario is classified as direct and external, while the second and third one are indirect external. In this paper we will explain the first and second scenario (see Figure 2), while the third one will be explained in future work.

In both scenarios we used the same Android based mobile phone with Prosyst mBS OSGi solution for mobile devices that provides mechanisms for software provisioning and configuration management. Moreover, in both cases the identifier is the same and is equal to 352668046631360 2140701234 56789 F8DB7F51BFAE F8DB7F62D2C2, where:

- IMEI = 352668046631360;
- IMSI = 214070123456789;
- WLAN MAC = F8DB7F51BFAE and
- BT MAC = F8DB7F62D2C2.

The only difference between two described scenarios is the way Android devices are addressed, while identification and software update process are the same. The addressing process for Android in the first scenario is accomplished using a mobile network (i.e. 3G) that provides public IP address. In that case, since server also has a public IP address, server can easily initialize the communication. Therefore, the only
thing M2M server has to store in its database is a device identifier together with its public IP address (i.e. registration process). In the second scenario, when a device is behind a M2M NAT (i.e. has a private IP address), the server cannot be the initiator of communication. Thus, in order to have automated software provisioning, server has to stimulate M2M device to establish connection with the server. Although there are many possible solutions for that (e.g. Message Queuing Telemetry Transport, SMS), in this work we present a prototype which uses SMS messaging in order to enable push messages (see Figure 3).

Software update process is divided in two parts: registration and software retrieval. First of all, each M2M device has to register itself by sending its own identifier (see Registration process in Figure 1). This happens only one time when M2M server stores device’s identity into its database. Together with this identity, the server stores private IP address, mobile phone number and current version of software from that M2M device. Second step of software update process is initiated from the server side when a new version of software is available. In that case, the server sends SMS to the device that needs to be updated (see Update process in Figure 1). This can be done since M2M server knows the telephone number of the device. SMS is used as a wake up call for the device that then establishes a TCP connection with the M2M server. Finally, the M2M server can send a new version of software and after it is downloaded and installed, the software update procedure ends.

6. CONCLUSION

Very soon the number of interconnected machines will exceed the overall population count and thus it is of vital importance to understand basic M2M principles. M2M systems are heterogeneous systems in sense of device (hardware and software) and communication diversity. Both these aspects of M2M systems should be taken in consideration when developing applications for M2M environments.

We addressed the problem of different types of connections in M2M systems regarding M2M device availability. For instance, M2M devices can have public IP addresses, or be located behind M2M NATs or gateways. Taking these parameters into account, we classified communication to be direct or indirect (i.e. with or without intermediary), internal or external (inside private or public IP address space).

Our contribution is the proposal of a new identification scheme that we have shown to be scalable and supports M2M heterogeneity. To show how our approach can be used no matter which type of communication is used, we proposed proof-of-concept for the software update process. This process is used for the M2M devices that have globally known IP addresses, and those that are behind M2M NATs.

We used OSGi framework in our implementation since it is shown to be better solution than native Android application in terms of scalability and more efficient usage of system resources. Moreover, we believe that usage of OSGi could speed up application development because it can be run on different mobile operating systems (e.g. Android and Symbian) and even on embedded low power hardware that can host JVM.

Acknowledgment

This work was supported by two research projects: "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639), funded by the Ministry of Science, Education and Sports of the Republic of Croatia and "Machine-to-Machine Communication challenges", funded by Ericsson Nikola Tesla, Croatia.

7. REFERENCES