Universal Identification Scheme in Machine-to-Machine Systems

Damjan Katusic*, Pavle Skocir*, Iva Bojic*, Mario Kusek*, Gordan Jezic*, Sasa Desic† and Darko Huljenic†
*University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Telecommunications
Unska 3, HR-10000 Zagreb, Croatia
{damjan.katusic, pavle.skocir, iva.bojic, mario.kusek, gordan.jezic}@fer.hr
†Ericsson Nikola Tesla
R&D Center
Krapinska 45, HR-10000 Zagreb, Croatia
{sasa.desic, darko.huljenic}@ericsson.com

***Abstract*—Number of connected devices in Machine-to-Machine (M2M) systems is rapidly growing: according to famous predictions made by Ericsson, by the end of 2020 there will be 50 billion devices in M2M systems around the world. The idea to connect a plethora of devices, whether they are fixed, movable or fully mobile, that communicate through different technologies (e.g. wireline, 2G/3G/4G, WiFi, ZigBee, Bluetooth) and thus create a heterogeneous environment provides remarkable opportunities. Such device and communication technology diversity offers new business concepts in numerous market verticals, creating a truly connected environment with seamless and automated flow of data and services. In this paper we explain current efforts in the area of M2M identification and addressing, and propose a new identification scheme that allows M2M Devices to establish communication with M2M Server or other M2M Devices in all available scenarios (i.e. with or without intermediary devices).**

I. Introduction

Machine-to-Machine (M2M) communication is established between two or more entities that do not necessarily need any direct human intervention [1] [2]. Actors in such an environment include broad range of communication capable devices: computers, mobile phones, tablets, but also a variety of sensors, smart grid systems, embedded processors, cars, industrial and medical equipment, and countless other everyday devices. The M2M system in a very simplified aspect, as will become clear in the following sections when current considerations regarding architecture standardization will be presented, consists of M2M Devices, M2M Gateways, and M2M Servers.

Apart from heterogeneity in types of M2M Devices, M2M system should also allow communication between different M2M entities, ignoring the differences in the network technologies, including the underlying used addressing mechanism. For example, in an Internet Protocol (IP) based network, the communication establishment between M2M entities should be possible when either static or dynamic IP addressing are used regardless of the use of public or private IP address space. Moreover, it is very important to emphasize that IP connectivity is not the only one, i.e. M2M Devices can be connected using different M2M Area Networks (e.g. Zigbee, Bluetooth, M-BUS, Wireless M-BUS).

This paper is organized as follows. In Section 2, we give an overview of the M2M functional architecture with focus on analyzed communication scenarios, while Section 3 describes procedures involved in establishing connection between M2M Device and M2M Server. Section 4 gives an overview of the various access communication technologies that can be used in an implementation of M2M Area Networks. In Section 5, we introduce our proposed identifier that enables universal and flexible communication establishment in heterogeneous M2M systems. Section 6 brings brief analysis of notification channel management based on long-polling mechanism, and discusses two novel approaches (push over SMS and time-based pull mechanism) that are not considered in the current ETSI standards. Finally, Section 7 concludes the paper.

II. Machine-to-machine system architecture

European Telecommunications Standards Institut (ETSI) is one of the most active standardization bodies in the field of M2M systems. Currently it is joining 6 other standardization organizations from around the world in forming a global M2M initiative: oneM2M. These organizations are: Association of Radio Industries and Businesses (ARIB) in Japan, Alliance for Telecommunications Industry Solutions (ATIS) and Telecommunications Industry Association (TIA) in the USA, China Communications Standards Association (CCSA) in China and Telecommunications Technology Association (TTA) in South Korea. Its goal is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M Servers worldwide [4].

Their work regarding M2M technology has so far focused on different use cases (e.g. smart metering, eHealth, connected consumer, automotive applications, city automation), defining M2M interfaces [5], as well as its service requirements [1] and functional architecture [3]. They closely co-operate with other standardization organizations such as 3rd Generation Partnership Project (3GPP), 3GPP2, Open Mobile Alliance (OMA), and Broadband Forum (BBF) in integration of their respective technologies into M2M systems.

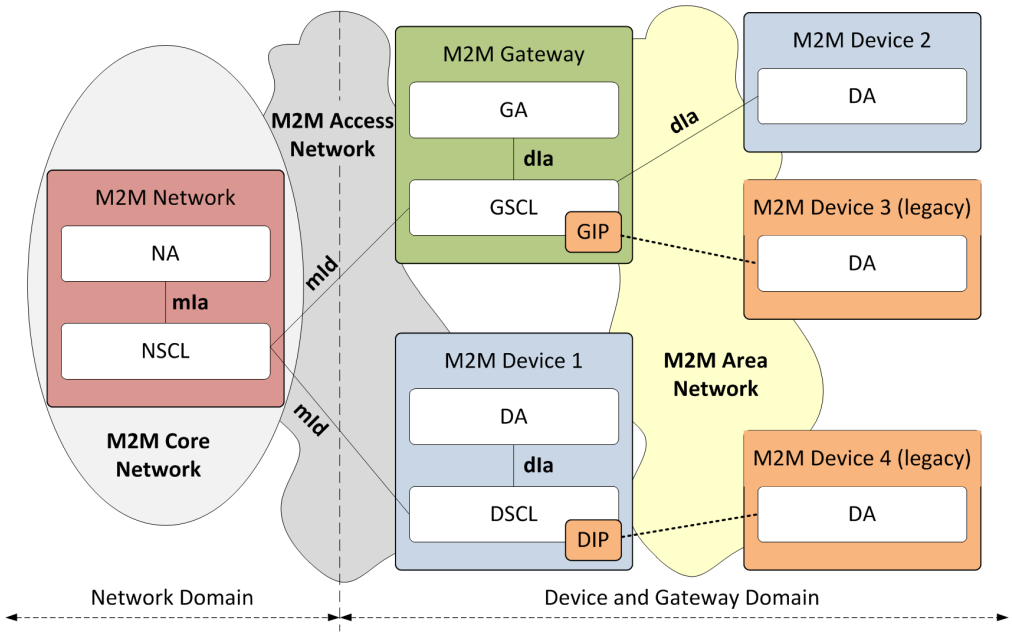

Fig. 1: High level architecture for M2M system [3]

In the ETSI standard *M2M functional architecture* [3] the high level architecture for M2M system consists of a Device and Gateway Domain, and a Network domain (see Figure 1). The Device and Gateway Domain is composed of the following elements:

- **M2M Device** that runs M2M Device Applications (DA) using M2M Device Service Capabilities Layer (DSCL),
- **M2M Gateway** that runs M2M Gateway Applications (GA) using M2M Gateway Service Capabilities Layer (GSCL), and
- **M2M Area Network** that provides connectivity between M2M Devices and M2M Gateways;

while the Network Domain is composed of:

- **M2M Access Network** that allows the M2M Device and Gateway Domain to communicate with the Core Network,
- **M2M Core Network** that (among other things) provides interconnection with other networks,
- **M2M Network Service Capabilities Layer (NSCL)** that provides M2M functions that are shared by different Applications,
- **M2M Network Applications (NA)** that run the service logic and use M2M Service Capabilities accessible via an open interface,
- **M2M Network Management Functions** which consists of all the functions required to manage the Access and Core networks, and
- **M2M Management Functions** which consists of all the functions required to manage M2M Service Capabilities in the Network Domain.

M2M Devices can connect to the Network Domain directly (M2M Device 1 in Figure 1) or through **M2M Gateway** (M2M Device 2 in Figure 1). In the first case M2M Devices connect to the Network Domain via the **M2M Access Network**, while in the second case M2M Devices connect to the M2M Gateway using the **M2M Area Network**. In that case M2M Gateway serves as a proxy for the Network Domain towards the M2M Devices that are connected to it. In both cases, M2M Devices communicate using IP protocol. The only difference is that in the first case, an M2M Device usually has a public IP address while in the second case it has a private one. The 3GPP standard *System Improvements for Machine-Type Communications* [6] analyses three possible M2M addressing schemes of M2M Server located in the M2M Network Domain and M2M Device located in the Device and Gateway Domain:

- M2M Server and the M2M Device are both located in the IPv6 address space,
- M2M Server is located in a public IPv4 address space, while to the M2M Device is assigned a private IPv4 address from an address pool, and
- M2M Server is located in a private IPv4 address space and to the M2M Device is assigned a private IPv4 address within the same address space.

Using a taxonomy proposed in our previous work [7] we can classify aforementioned M2M communication types for M2M Devices. When both M2M Devices are directly connected to the Network Domain via the M2M Access Network, then this type of communication can be classified as *direct* and *external*. Furthermore, when one M2M Device is connected directly to the Network Domain, while the other one is connected to the M2M Gateway using the M2M Area Network, then this type of communication is *indirect* and *external*. Finally, when both devices are connected to the M2M Gateway using the M2M Area Network, then this type of communication is classified as *indirect* and *internal*.

However, an M2M Device may not support IP protocol for communication, in which case it is called a **legacy M2M Device**. A legacy M2M Device can be connected to an M2M Gateway through Gateway Interworking Proxy (GIP), as the M2M Device 3 in Figure 1, or a non-legacy M2M Device through Device Interworking Proxy (DIP), as the M2M Device 4 in Figure 1). In the second case the non-legacy M2M Device provides its service to the legacy M2M Device connected to it that is hidden from the Network Domain. GIP is a part of a GSCL, while DIP is a part of a DSCL.

Different M2M Service Capabilities Layers (i.e. DSCL, GSCL and NSCL) provide functions that are exposed on the mIa, dIa and mId reference points. mIa reference point allows a NA to access the M2M Service Capabilities in the Network Domain. dIa reference point allows a DA residing in a non-legacy M2M Device to access the different M2M Service Capabilities in the same M2M Device or in an M2M Gateway. Moreover, this reference point allows a GA residing in an M2M Gateway to access the different M2M Service Capabilities in the same M2M Gateway. Finally, mId reference point allows an M2M Service Capabilities residing in a non-legacy M2M Device or M2M Gateway to communicate with the M2M Service Capabilities in the Network Domain and vice versa. Functionalities of mIa, dIa and mId reference points are explained in [5].

III. Communication establishment

The process of communication establishment in M2M systems defined in [3] consists of the following six procedures: Application Registration, Network Bootstrap, Network Registration, M2M Service Bootstrap, M2M Service Connection and SCL Registration of D/GSCL with NSCL.

- **Application Registration**: Procedure involves registration of an application (DA, GA or NA) with local SCL. This allows interactions between local applications, i.e. those connected via the local SCL. Enabling M2M communication between applications connected on other SCLs requires involvement of several other procedures.
- **Network Bootstrap**: Network Bootstrap defines initial configuration settings that allow M2M Device/Gateway to connect and register to its access network, whether it is based on fixed or mobile technologies. One example is Bootstrap from Universal Integrated Circuit Card (UICC).
- **Network Registration**: This procedure consists of registration of an M2M Device/Gateway with its access network, taking into account the characteristics of a corresponding access network technologies. For example, registration of an M2M Device in a 3GPP Network such as Universal Mobile Telecommunications System (UMTS) involves IP address assignment, mutual authentication as two sides agree on a set of security keys, authorization for using specific access network services, as well as initiation of potential accounting operations.
- **M2M Service Bootstrap**: M2M Service Bootstrap, with the procedure of M2M Service Connection, defines basic prerequisites for the communication establishment and registration of M2M D/GSCL with the NSCL. It involves, apart from the usual actors, M2M Service Bootstrap Function (MSBF) and M2M Authentication Server (MAS). Former facilitates the bootstrapping of permanent M2M service layer security credentials (M2M Root Key) between the M2M D/G/NSCL entities, as well as MAS, while the latter serves as a safe location for storage (see Figure 2). If the M2M service credentials have been pre-provisioned (e.g. in UICC), no M2M Service Bootstrap procedure is needed. Otherwise, it is conducted with or without the assistance of associated access network layer.
- **M2M Service Connection**: Service connection includes mutual authentication of mId end points (D/GSCL and NSCL), optional agreement on M2M Connection Key derived from M2M Root Key (see Figure 2), as well as optional establishment of a secure encrypted session via mId.
- **SCL Registration**: As its name suggests, it involves D/GSCL registration with NSCL. Successful completion of M2M Service Connection between D/G/N M2M nodes is a prerequisite for performing SCL Registration. This procedure occurs either periodically (frequency of registration updates is decided by the M2M Service Provider) or on demand. Successful registration, among others, results in an exchange of context information between D/GSCL and NSCL.

In other words, when new M2M Device turns on it needs to establish initial contact with corresponding SCL and access network, and then conduct all relevant credential creations and exchanges to establish secure connection with its communication peer (e.g. server in a network layer) within the M2M environment. Only when this process is successfully completed, M2M Device is able to communicate with other nodes in the M2M system, e.g. report measured sensor data to server or update new version of software. In order to successfully complete procedures of the connection establishment process and establish communication, each M2M entity (described in Section II) has to have a proper unique identifier, and eventually address based on the used communication technology so it can be reached by other nodes. ETSI proposes several identifiers which are used in various activities regarding successful connection setup in M2M systems. Some of them, depending on the situation, may be set to the same value [3].

- **Application Identifier (App-ID)**: An application identifier uniquely identifies M2M Application in device (DA), gateway (GA) or network (NA), registered with its corresponding SCL on a global level.
- **SCL Idetifier (SCL-ID)**: A particular SCL is identified by a globally unique identifier, its SCL-ID. It is possible to set SCL-ID and the M2M-Node-ID to the same value.
- **M2M Node Identifier (M2M-Node-ID)**: M2M node represents globally unique logical representation of the M2M components in the M2M Device, M2M Gateway or M2M Network. Such components include one SCL,

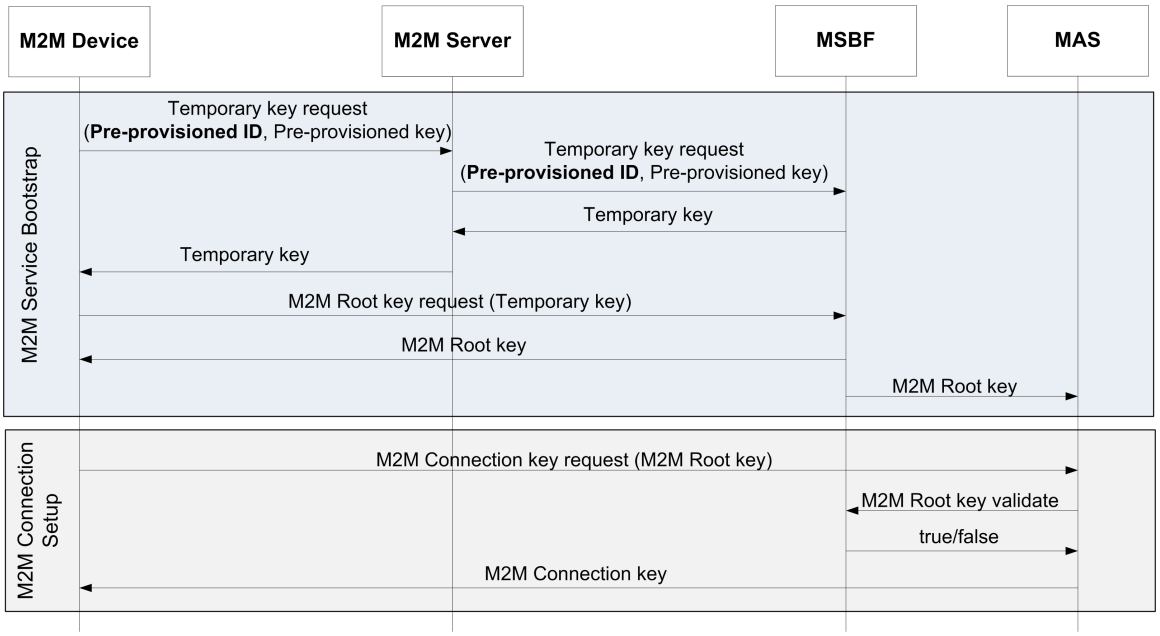

Fig. 2: M2M connection establishment (simplified)

M2M Service Bootstrap function if any, and a M2M Service Connection function. M2M-Node-ID uniquely on a global level identifies particular M2M node.

- **M2M Service Connection Identifier (M2M-Connection-ID)**: M2M-Connection-ID identifies an M2M Service Connection, which is instantiated upon M2M D/GSCL getting authenticated and authorized by an NSCL for connectivity.
- **M2M Service Provider Identifier (M2M-SP-ID)**: This is a static value which uniquely identifies M2M service provider, and is used in the bootstrap procedure for key generation.
- **MSBF Identifier (MSBF-ID)**: Similar to M2M-SP-ID, this is also a static value which uniquely identifies MSBF, assigned by the M2M service provider. Analogously, it is also used in the bootstrap procedure for key generation.
- **M2M Subscription Identifier**: It identifies subscription associated with M2M service provider and M2M SCL either on a device or a network application. It belongs to the service provider and enables communication with it.

Identifiers that are associated with M2M Device/Gateway and are considered inside the scope of current standards are M2M-Node-ID, M2M-Connection-ID, D/GSCL-ID and M2M Subscription ID. Apart from the mentioned identifiers, the creation of M2M Root Key during the M2M Service Bootstrap procedure requires another type of device identification, a pre-provisioned identifier that is typically assigned during manufacturing process of a device. Figure 2 shows simplified version of main activities which occur between M2M entities during either service bootstrap or connection setup procedures. Our proposal for the role of a such pre-provisioned identifier, its format, and (dis)advantages are discussed in the following sections of the paper.

IV. Technologies for M2M Area Network

When talking about M2M connectivity, we usually refer to communication between devices such as smart sensors, actuators and mobile devices with limited human intervention [8]. As previously mentioned in Section II, those devices can connect to the Network Domain directly, using M2M Access Network or via M2M Gateway which acts as proxy for the Network Domain towards M2M Devices that are connected to it [3]. The network which provides connectivity between M2M Devices and M2M Gateway is called M2M Area Network.

In this section we will focus on technologies which enable wireless connectivity in M2M Area Networks. Examples of such technologies are Digital Enhanced Cordless Telecommunications Ultra Low Energy (DECT ULE), Z-Wave, ZigBee, BlueTooth, Wi-Fi, Wireless Meter bus (M-BUS), Near Field Communication (NFC), etc. Most used technologies are shown in Table I, together with their data rates and frequency spectrum used for communication.

TABLE I: M2M Area Network Technologies

Standard	Rate	Frequency
DECT ULE	<500 kb/s	around 1.9 GHz
Z-Wave	<40 kb/s	around 900 MHz
ZigBee	<250 kb/s	around 900 MHz and 2.4 GHz
Bluetooth	<1mb/s	around 2.4 GHz
Wi-Fi	<450 kb/s	around 2.4 GHz and 5 GHz
Wireless M-BUS	<100 kb/s	868-870 MHz
NFC	<424 kb/s	around 13.56 MHz, 106 MHz, 212 MHz

DECT-ULE is an Ultra Low Energy variant of DECT, one of the most reliable and flexible digital radio access standards for cordless communication [9]. DECT standard is widely used in cordless phone system [10]. Devices which communicate using DECT ULE can be used in home control, security, health and smart energy applications, and in tracking devices. Various devices which support DECT ULE are thermostats, sensors for weather reporting, motion detectors, heart rate monitors, consumption displays, inventory trackers, etc. ETSI specifies DECT protocol stack based on the lower layers of the Open Systems Interconnection (OSI) model and includes physical layer, Media Access Control (MAC), data link control layer, and network layer [11]. Every wireless DECT device (Portable Part, PP) has Portable Access Rights Key (PARK) and International Portable User Identity (IPUI). IPUI can be globally or locally unique. PP connects to a fixed part (FP) which has a role of the access point. Every FP contains a globally unique Access Rights Identity (ARI) and broadcasts it. PP is allowed to access any radio FP which broadcasts an ARI that can be identified by any of the PARKs of that PP.

Z-Wave is an interoperable wireless mesh networking technology which is primarily developed to enable communication of home electronics via remote control [12]. It can be used for different purposes: monitoring, thermostats, lights control, smart metering, etc. Z-Wave home automation technology comprises of three layers: radio, network, and application layer [13]. The Z-Wave protocol defines two identifiers for the network organization: Home-ID and Node-ID. Home-ID is the common identification of all nodes belonging to one logical Z-Wave network. It has the length of 4 bytes. Node-ID is the address of a single node in the network with the length of 1 byte. Only nodes within the same home network can communicate with each another. Z-Wave has two types of devices, controllers and slaves. Controllers are factory programmed with a Home-ID which cannot be changed by user. Slaves do not have a pre-programmed Home-ID and they take Home-ID assigned to them by the network.

ZigBee is a standard based wireless technology which enables low-cost and low-power connectivity for sensor and control networks in many markets [14]. ZigBee standards are developed for various domains: building automation, remote control, smart energy, health care, home automation, light link, retail, and telecom services, etc. It is developed for low power, long lifetime wireless devices. By using ZigBee, users can monitor lights, control energy and water consumption, monitor chronic diseases, etc. The ZigBee stack architecture is based on the standard OSI model, but defines only those layers relevant to achieving functionality in the intended marketplace [15]. Two lower layers, physical layer and MAC sublayer are based on Institute of Electrical and Electronics Engineers (IEEE) 802.15.4 standard. The ZigBee network layer supports star, tree, and mesh topologies. Three types of devices are present in ZigBee networks: coordinator, end device, and router. Coordinator is responsible for initiating and maintaining devices on the network and for forming the network. Routing nodes are responsible for forwarding messages to other nodes. Each device is uniquely identified by EUI-64, 64-bit MAC address which must be purchased from IEEE [16]. When establishing a network, coordinating chooses a 16-bit network address which is used to communicate with other devices in the network.

Bluetooth is a wireless technology standard originally developed to enable data exchange over short distances [17]. Today it has various applications in the fields of health and fitness, home entertainment, handsfree calling or home automation. By using Bluetooth connection, wireless earphones can be connected to a mobile phone, temperature or household appliances can be controlled, printers can be connected to computers, etc. Bluetooth protocol stack is defined in IEEE 802.15.1 standard and includes transport layer, middleware layer, and applications layer. Each bluetooth-enabled device has a unique 48-bit address. Devices connect via pairing process. When device is ready to connect, it sends its name, class, list of services, and technical information [18]. The other device sees the information about the device ready to connect. Afterwards they create a connection.

Wi-Fi is a technology that enables wireless Internet connection as well as connection between devices, e.g. mobile phones, media players, etc. [19]. It is widely used all over the world. It is suitable for higher data rate applications such as videos and audio streaming [20]. Wi-Fi Protocol is defined in IEEE 802.11 standard and is compatible with IP protocol stack. Each device is uniquely identified by 48-bit MAC address. Devices using Wi-Fi connect to an access point called hotspot which enables Internet access. Hotspots transmit on radio frequency channels that they are available for connections. Devices pick up transmissions and start a connection procedure.

Wireless M-BUS is a European standard for remote reading of gas, electricity or other consumption meters [21]. It enables collection of data at a centralized aggregator. Applications which can use wireless M-BUS technology are beyond smart metering, such as heating control, lightning, alarm systems, etc. M-Bus architecture is defined in the European norm EN 13757 [22]. Its protocol stack uses a three-layer model: physical layer, data link layer, and application layer. Device addressing is defined in the data link layer. Each device is uniquely addressed by a 6 byte value [23].

NFC is a short-range wireless communication technology that enables the exchange of data between devices over about a 10 cm distance [24]. It can be used for healthcare applications, public transport, consumer electronics, payment purposes, etc. Devices communicate with no more than a simple touch. It is standardized within International Organization for Standardization (ISO), European Computer Manufacturers Association (ECMA), and ETSI standards. Each device has a unique Radio-Frequency IDentification (RFID). Some RFID tags do not even require battery and are powered via magnetic fields. Usually there are two types of devices: a initiator which initiates communication and the target [25]. Initiator (NFC reader) can e.g. read a tag on a credit card and fulfill a payment.

V. Proposed Identifier

We propose an identifier which uniquely identifies an M2M Device regardless of the technologies it uses (e.g. if one sensor uses ZigBee and Bluetooth, it is difficult to realize that it is one single sensor and that its temperature values will always be the same, regardless of the technology used for its readings). Our proposed identification scheme is shown in Figure 3. The first byte (i.e. identifier header) denotes the technologies supported by that device. When needed, technology list can be expanded by allocating another bytes for technology specification.

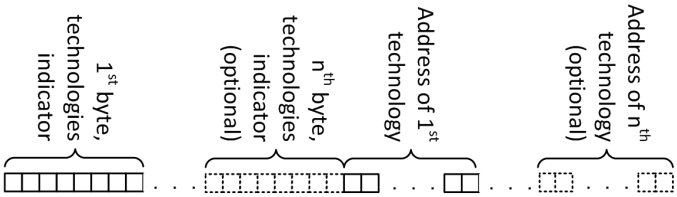

Fig. 3: Proposed Identification Scheme

The last bit of each byte shows that the following byte addresses a list of more technologies, which are not specified in the previous bytes. We will specify only first four bits, others can be specified according to the technologies which are used by larger number of devices. Our specification of the first four bits is the following (in alphabetical order):

- **bit 0**: Bluetooth,
- **bit 1**: Wi-Fi,
- **bit 2**: Wireless M-Bus,
- **bit 3**: ZigBee.

If value of the bit 0 is zero, it denotes that this device does not support Bluetooth. If its value is one, than it supports Bluetooth. The same rule applies for other bits and other technologies.

The remaining bytes show identificators for mentioned technology. For technologies we mentioned in the first four bits, those identifiers are the following:

- **Bluetooth**: 48-bit IEEE 802 address,
- **Wi-Fi**: 48-bit MAC address,
- **Wireless M-Bus**: 6-bit address,
- **ZigBee**: 64-bit EUI address.

In our proposed ID there must be at least one technology identifier, while the upper limit does not exist. Those technologies are enumerated in identifier header and their number depends on the number of technologies that a certain device supports. Therefore, the total length of our identifier is variable.

In the rest of this section we will explain how our identifier can be used for identification of Libelium Waspmote sensors [26]. Waspmote sensors have microcontroller ATmega1281 and work on a frequency of 14.7456 MHz. More importantly they support 8 different radio technologies:

- **long range**: GSM / GPRS,
- **medium range**: Zigbee / 802.15.4 / Wi-Fi, and
- **short range**: RFID / NFC / Bluetooth.

Waspmote can integrate a Global System for Mobile communications (GSM) / General Packet Radio Service (GPRS) module to enable communication using the mobile telephone network. When using this module the following tasks can be done: making/receiving calls, sending/receiving SMS (Short Message Service), single connection and multiple connections TCP/IP and UDP/IP clients, TCP/IP server, Hypertext Transfer Protocol (HTTP) service and File Transfer Protocol (FTP) service for downloading and uploading files.

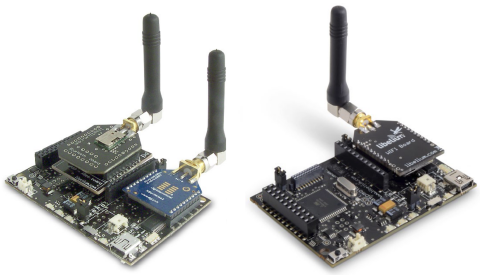

Fig. 4: Waspmote with ZigBee - Bluetooth and with Wi-Fi

Waspmote can also integrate the Digi XBee modules for communication in the Industrial Scientific Medical Band (ISMB) bands. The frequency used is the free band of 2.4GHz, using 16 channels with a bandwidth of 5MHz per channel. The XBee 802.15.4 modules comply with the standard IEEE 802.15.4 which defines the physical level and the link level (MAC layer). Furthermore, the Wi-Fi module for the Waspmote platform enables direct communication of the sensor nodes with any Wi-Fi router in the market. Protocols that are supported are 802.11b/g and the frequency used is the free band of 2.4GHz (same as for XBee).

Waspmote can integrate one RFID module for operating frequency of 125 KHz and one RFID module for frequency of 13.56 MHz which is also compliant with the NFC technology. Finally, in short range communication, Waspmote also supports Bluetooth module for communication in the 2.4GHz ISMB band. Bluetooth uses 79 channels with a bandwidth of 1MHz per channel, as well as adaptive frequency hopping for enhancement of the transmissions.

Waspmote sensor can simultaneously communicate using two communication technologies because of expansion radio board. Namely, the expansion radio board allows to connect two radios at the same time using two different sockets. We will give examples of identifiers for Waspmote sensors with two of the possible combinations: ZigBee - Bluetooth and with only Wi-Fi (see Figure 4).

In order to make our identifier, we have to determine the order of used communication technologies in the first byte. For example, we can use same order proposed earlier in this section. This would mean that the first bit denotes the existence of Bluetooth technology, the second one denotes the existence of Wi-Fi, while the fourth one shows if there is a ZigBee module connected. Other bits within the first byte are not used and thus are set to 0. In case when Waspmote has both ZigBee and Bluetooth modules first byte is equal to ”1001 0000”, i.e. in hexadecimal notation ”0x90”, while when having only Wi-Fi module it is equal to ”0100 0000”, i.e. in hexadecimal notation ”0x40”.

Since a Bluetooth address is 48 bits long and a ZigBee address is 64 bits long, the unique and universal identifier in the first case is 120 bits long. For example if ZigBee address is equal ”0x0013A200406937A0” and Bluetooth address is ”0x0003190D0D7C”, then the whole identifier is equal to ”0x900003190D0D7C0013A200406937A0”. Consequentially, identifier in the second case is 56 bits long since Wi-Fi address is 48 bits long. For example if Wi-Fi address is ”4221CB1F375C”, then the whole identifier is ”0x404221CB1F375C”.

VI. Notification Channel Management

Communication in M2M Applications is based on Representational State Transfer (REST) architecture [27]. This client-server paradigm is composed of four basic interactions (Create, Read, Update, and Delete, also known as CRUD), which are easily mapped into HTTP methods. Main concept of REST is the manipulation of resources through a uniform interface. A resource can be any entity addressed using a HTTP Uniform Resource Identifier (URI). Since M2M systems typically comprise of various types of sensor devices which change states and serve as data sources, the concept that every physical or logical entity is represented as a resource that has particular state that can be manipulated is obviously suitable for M2M modeling. Also, M2M Devices typically have limited processing power available. They benefit from the fact that since every REST communication is stateless, scalability of a system increases [28].

When discussing notification management mechanisms standardized in M2M systems, so far only one (long-poling) is available. However, alongside long-polling solution there are a few other mechanisms that are going to be discussed because they offer better alternatives in certain situations.

A. Long-polling

Client in long-polling mechanism sends its request (long-polling request) to which server responds when new data becomes available, i.e. after a particular event (see Figure 5). Immediately after M2M client receives data it starts another long-polling request and waits for server's response. Obviously, main issue with such an approach is the fact that client's request cannot stay open indefinitely. To tackle the mentioned deficiency, server needs to send empty response before time-to-live of the opened request expires. This informs client that it needs to send a new long-polling request if it wants to continue its "connection" with server side.

In order to implement this approach in practice, first the notification channel resource must be established. Such a resource offers a method for a client to receive asynchronous notifications it has subscribed for. The long-polling variant of a notification channel is discussed and described in more detail in [3]. There are also several alternative approaches which tend to achieve similar results and are so far mostly left out of the scope of ETSI standards: push over SMS mechanism and time-based pull mechanism. Basics of both approaches are briefly analyzed in the following subsections.

B. Push over SMS

As it was discussed in the previous subsection, main disadvantage of long-polling is the fact that typically low-power M2M Devices need to maintain open time-consuming connections with the server side although new data may not be available. Push mechanism over SMS solves this issue because now the server side worries when new data becomes available and informs client about it. This approach is also particularly useful in situations where client M2M Device is not directly reachable from the server side (in term of opening direct communication channel), so it uses SMS to encourage client side to open connection from its side (see Figure 6).

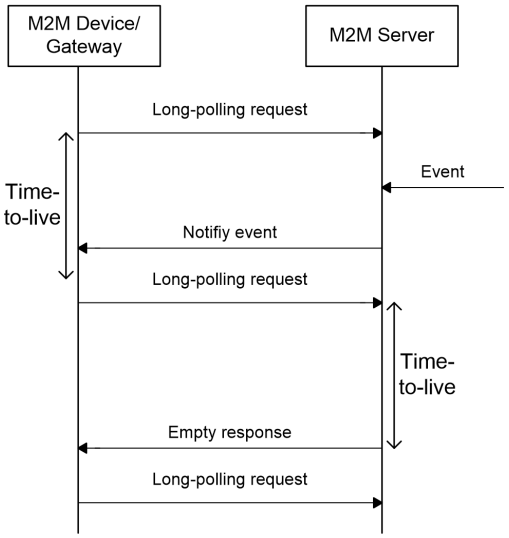

Fig. 5: Long-polling mechanism [3]

Main disadvantage is the fact that client devices in order to be able to receive SMS messages need to have a valid Subscriber Identity Module (SIM) card with International Mobile Subscriber Identity (IMSI) and Mobile Station International Subscriber Directory Number (MSISDN) identities. Also, too often opening and closing of Transmission Control Protocol (TCP) connections to exchange small amounts of data can result in significant communication overhead.

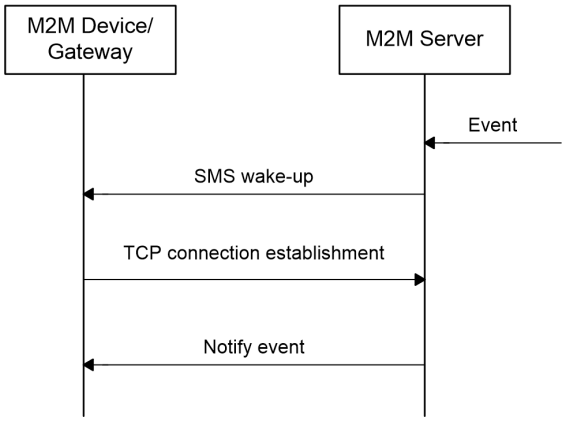

Fig. 6: Push over SMS mechanism

C. Time-based pull

This approach is the most straightforward of the three analyzed mechanisms. Client M2M Device sends a pull request for new data. When expected data becomes available (e.g. sensor measures new data which initiates proper event), server sends it inside notify response. Otherwise, it sends an empty response. Frequency of clients periodic requests depends on the definition of pull time window. One has to adjust this parameter to expected value of new data generation: too small value means requests are coming too fast so server in most situations does not have new data available (resources are needlessly spent), while too big value possibly means that client does not acquire all of interested data.

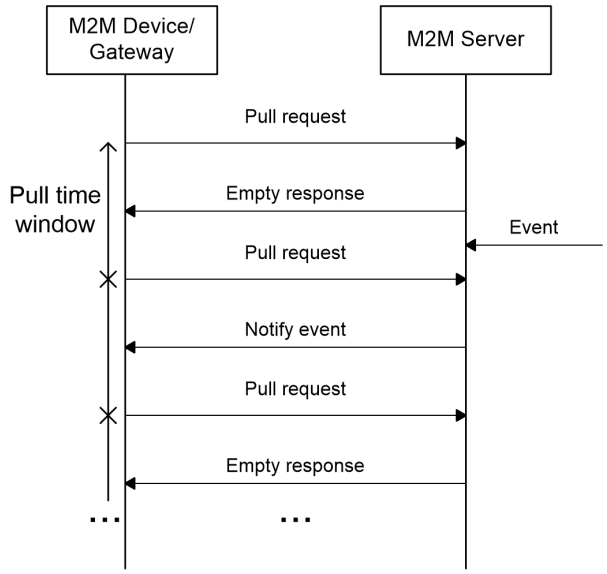

Fig. 7: Time-based pull mechanism

VII. Conclusions and Future Work

M2M Devices have different access or other technologies (e.g. SIM cards) at their disposal, processing capabilities or battery life, and can participate in various communication scenarios. There are many existing technologies which enable connectivity between aforementioned M2M Devices. In this paper we proposed an identifier which can be used in order to uniquely identify an M2M Device regardless of the wireless technologies it uses in M2M Area Networks (e.g. DECT ULE, Z-Wave, ZigBee, BlueTooth, Wi-Fi, Wireless M-BUS). The proposed identifier is of variable length (enabled by the usage of technologies indicator in header) and contains addressing information for all available access technologies for the corresponding M2M Device. Identifier's variable length enables flexible and universal identification scheme, and yet this additional header overhead has a negligible negative effect on the total identifier length.

Moreover, in this paper we analyzed three different mechanisms for notification management in M2M systems: long-polling, push over SMS, and time-based pull. Situations in which new data becomes available every 10 seconds or every 5 minutes cannot be observed identically, although the underlying idea is the same. All three analyzed mechanisms have certain (dis)advantages that produce different positive or negative impacts on established communication depending on the situation. There are some techniques which can enhance mechanisms for notification management in M2M systems (e.g. buffers, proxies). However, their further analysis is out of the scope of this paper and will be analyzed in future work.

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.

References

[1] ETSI TS 102 689 M2M service requirements, *http://www.etsi.org/deliver/etsi-ts/102600-102699/102689/01.01.01-60/ts-102689v010101p.pdf*.

[2] 3GPP TR 22.868 Study on facilitating machine to machine communication in 3GPP systems, *http://www.3gpp.org/ftp/Specs/html-info/22868.htm*.

[3] ETSI TS 102 690 M2M functional architecture, *http://www.etsi.org/deliver/etsi-ts/102600-102699/102690/01.01.01-60/ts-102690v010101p.pdf*.

[4] (2013) Welcome to onem2m. [Online]. Available: http://www.onem2m.org/

[5] ETSI TS 102 921 mIa, dIa and mId interfaces, *http://www.etsi.org/deliver/etsi-ts/102900-102999/102921/01.01.01-60/ts-102921v010101p.pdf*.

[6] 3GPP TR 23.888 System Improvements for Machine-Type Communications, *http://www.3gpp.org/ftp/specs/html-info/23888.htm*.

[7] I. Bojic, G. Jezic, D. Katusic, S. Desic, M. Kusek, and D. Huljenic, "Communication in machine-to-machine environments," in *Proceedings of the Fifth Balkan Conference in Informatics*, 2012, pp. 283–286.

[8] M. Chen, J. Wan, and F. Li, "Machine-to-machine communications: Architectures, standards and applications," *TIIS*, vol. 6, no. 2, pp. 480–497, 2012.

[9] "Dect ultra low energy," White Paper, DECT Forum, 2011.

[10] (2012) Dect^{TM} - introduction. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/dect/

[11] "Radio equipment and systems (res); digital enhanced cordless telecommunications (dect); common interface (ci); part 6: Identities and addressing," Sophia Antipolis, France, Tech. Rep. ETS 300 175-6, 1996.

[12] (2004-2012) About z-wave. [Online]. Available: http://www.z-wave.com/modules/AboutZ-Wave/

[13] (2012) Understanding z-wave networks, nodes and devices. [Online]. Available: http://www.vesternet.com/resources/technology-indepth/understanding-z-wave-networks

[14] (2013) Zigbee technology. [Online]. Available: http://www.zigbee.org/About/AboutTechnology/ZigBeeTechnology.aspx/

[15] L. Pengfei, L. Jiakun, N. Luhua, and W. Bo, "Research and application of zigbee protocol stack," in *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, vol. 2, March, pp. 1031–1034.

[16] N. Rajbharti, "Microchip stack for the zigbee protocol," Microchip Technology Inc., Tech. Rep. AN965, 2006.

[17] (2013) Bluetooth - fast facts. [Online]. Available: http://www.bluetooth.com/Pages/Fast-Facts.aspx

[18] C.-M. Fan, S. Shieh, and B.-H. Li, "On the security of password-based pairing protocol in bluetooth," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, 2011, pp. 1–4.

[19] (2013) Discover and learn. [Online]. Available: http://www.wi-fi.org/discover-and-learn

[20] M. Starsinic, "System architecture challenges in the home m2m network," in *Applications and Technology Conference (LISAT), 2010 Long Island Systems*, May, pp. 1–7.

[21] (2013) Wireless m-bus. [Online]. Available: http://www.silabs.com/products/wireless/Pages/Wireless-M-Bus.aspx

[22] "Wireless m-bus software implementaion," Silicon labs, Tech. Rep. AN451, 2010.

[23] "Wireless m-bus," Infineon Technologies, Tech. Rep. TDA5340, 2012.

[24] (2013) About nfc. [Online]. Available: http://www.nfc-forum.org/aboutnfc/

[25] (2013) Nfc/rfid fundamentals. [Online]. Available: http://www.rohde-schwarz.com/en/technologies/wireless-connectivity/rfid-nfc/

[26] Libelium Waspmote sensors, *http://www.libelium.com/*.

[27] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[28] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M Communications: A Systems Approach*, 1st ed. Wiley Publishing, 2012.