M2M Device Management in LTE Networks

Nenad Gligorić, Srdan Krčo, Senior Member, IEEE, Dejan Drajić, Member, IEEE, Stevan Jokić, and Bojana Jakovljević

Abstract — The driving concept of device management in M2M systems is establishment of a comprehensive end-to-end resource oriented design. In this paper, OMA-DM protocol for device management of M2M devices in 3GPP LTE networks is proposed and discussed. The OMA-DM represents a stable starting point for device management. Utilization of this protocol in M2M requires efficient message formats and transport replacement such as CoAP, Core Link Format, EXI, etc. The additional wake up mechanisms should also be considered for managing the standby policies.

Keywords — Device Management, LTE, M2M, OMA-DM.

I. INTRODUCTION

The Machine-to-Machine (M2M) concept represents the state of the art ecosystem composed of existing worldwide standards, industry forums and emerging protocols, data formats and technology. This ecosystem requires the processes and procedures between the system components as every other architecture. Important component of every system that enables monitoring, remote configuration, software updates, and application distribution is device management.

However, the device management of the M2M devices is a demanding process primarily due to the processing and battery restrictions of the end devices and differences in IP address assignment and communication schemas. A new device management perspective should provide efficient and functional device management within a capillary network as well as on a direct link between the server and the end-point. Managing a large number of devices in the M2M networks is one of the key future tasks. The new gateway design specific to the M2M is reasonable way to overcome these challenges, but this is not the answer for all requirements. The Open Mobile Alliance Device Management protocol (OMA-DM) [1] provides a reliable device management (DM) entity and service mechanism for mobile devices. Although, not made for the M2M, it presents a good basis that can be extended for the M2M domain. Utilization of the OMA-DM in the M2M requires modifications to include efficient protocols and message format replacements and additional gateway policies for interaction with the end points.

The SyncML (Synchronization Markup Language) and Hypertext Transfer Protocol (HTTP) transport are currently utilized in OMA-DM. The HTTP protocol, although often considered as a simple protocol is actually complex from the low-power devices perspective. To overcome this issue, the Constrained Application Protocol (CoAP) [2], CoRE Link format [5], Protocol Buffers [10] as well as EXI (Efficient XML Interchange) [8] are discussed as possible replacements.

Therefore, in this research the following two approaches for M2M device management in LTE networks are discussed: utilization of OMA-DM for device management and efficient protocols and message formats replacement.

In section II an overview of the OMA Device Management protocol is presented together with the requirements for its usage in M2M networks. Wake-Up/paging mechanisms for management of the end M2M devices are discussed in section III. The analysis and evaluation of the CoAP protocol and different message formats in the OMA-DM context are given in sections IV and V.

II. OMA-DM

The OMA Device Management is a protocol developed for managing large sets of mobile devices, offering platform scalability and horizontality. The first step before a device can communicate with an OMA-DM server is a bootstrap configuration called provisioning. OMA Client Provisioning specifications defines the OMA client provisioning object as an XML document containing the initial provisioning parameters for the end device. This XML document includes configuration parameters for proxy servers, network access points and access rules. Once the device is provisioned, the OMA DM server can manage it remotely due to the device being configured using a trusted and verified relationship with the management servers [3].

OMA-DM is designed considering security and bandwidth requirements of mobile phones, PDA, etc; but for low-powered devices that are even more constrained, the complexity of protocol messages is high and new
transport and messages formats are required to ensure efficient execution of management activities. OMA-DM utilizes SyncML protocol based on the XML markup for device management and device synchronization. SyncML message consists of a root SyncML element, a header (SyncHdr) and a body element (SyncBody) [7]. The OMA-DM XML documents need not to be well-formed, but only with properly identified name space element types from the OMA DM Device Description Framework (DDF) information DTD. The messages in SyncML are verbose due the nested types and large number of namespaces.

In order to replace the XML with other message formats, existing data types must be mapped to a new message format.

Development of the client for the M2M devices that will communicate with a server is a must, in order to enable end-to-end device management.

The detailed description and evaluation of mentioned protocols and formats is presented in sections IV and V. Next section is discussing the mechanisms for addressing the M2M devices in stand-by mode. These mechanisms are important in order to realize available options in LTE networks that can be employed for new device management use cases.

III. WAKE-UP / PAGING MECHANISMS

Management of a large number of M2M devices expected within the LTE networks is one of the main challenges ahead. As power consumption becomes a critical issue for standalone M2M devices, it is necessary to maximize operation life time of the devices. This implies that the device should go to sleep mode at earliest opportunity and wake-up only when necessary.

In the networks that are supporting circuit switched communication, like GSM/GPRS/WCDMA networks, SMS (Short Message Service) and USSD (Unstructured Supplementary Service Data) are often used for communication with M2M devices and could be used also for wake-up procedures of the end points.

SMS is developed for the circuit switched (CS) domain. LTE and EPS (Evolved Packet System) will support a rich variety of messaging applications including SMS. The solution is twofold, covering both the full IMS (IP Multimedia Subsystem) case and a transition solution for those networks that do not support IMS: A solution for SMS over IP was specified in Rel-7; SMS over SGs (Signaling Gateway): this solution requires the SGs interface introduced during the work on CSFB (Circuit Switch Fall Back). SMS are delivered in the Non Access Stratum over LTE. There is no inter-system change for sending or receiving SMS. This feature was specified in Rel-8.

SMS over IP was fully specified in 3GPP Rel 7 [11]. It depends on IMS and it is intended to provide compatibility between the existing cellular legacy and the implementations with more elaborate messaging capabilities via SMS and IMS interworking. The newest achievements related to SMS over IP, i.e. requirements for sender and receiver of messages can be found in [12].

For environments without IMS, a transition solution was specified. This is called SMS over SGs (previously misleadingly referred to as SMS over CS). It is a hybrid approach that allows the transmission of native SMS from CS infrastructure over the LTE radio network. SMS over SGs was specified as part of 3GPP Rel 8 [13]. It provides SMS service for mobiles in LTE and since it requires also CS domain infrastructure for the SMS transmission, it is intended to be a transition solution. The SG is a reference point between the MME (Mobility Management Entity) of the evolved packet system and the MSC (Mobile Switching Center) server. The protocol used to connect an MME to an MSC server is SGsAP (Signalling Gateway Application Part) [14]. SMS over SGs is seen as interim solutions, the long term being IMS.

USSD based configuration and control of services are being used widely in GSM networks today [15]. USSD services are triggered by the user, who dials a special feature code. After entering a USSD code on UE, the USSD messages are routed by the MSC to the HLR (Home Location Register) which proxies it over MAP (Mobile Application Part) to a Service Node (called the USSD server). The USSD server responds to the request. USSD works in two modes: MMI Mode (Man Machine Interface Mode) initiated by the UE (User Equipment) and Application Mode initiated by the Network.

The MMI mode is like a "pull" mode where the user pulls data from the network using USSD. The Application mode is like a "push" mode, where the network pushes information to the registered UE using USSD. USSD is designed for the circuit switch domain. LTE is fully packet switch oriented and there is no support for circuit switch services like USSD is. So far, USSD based service configuration was missing in the LTE standards. This has been mainly due to the presence of XCAP (XML Configuration Access Protocol) which provides user controlled service configuration. However, there are a large number of users who would still like to use USSD for controlling their legacy services even when they move to LTE to be able to use a single mechanism to control and configure their legacy as well as new LTE services. Due to these reasons, work item 480031 (USSI) "Unstructured Supplementary Service Data (USSD) simulation service in IMS" is being developed and is under standardization at 3GPP [16]. This work item is still under study, and various options are being evaluated.

As it currently stands, it will not be reintroduced in its current form. The 3GPP has noted that the implementation of USSD in legacy networks is quite an overhead, and therefore other implementation options are being considered:

- Re-using XCAP for USSD based control (by introducing new application use cases for the XDM -XML Document Management-). This option at first seems to be the most natural fit for implementing USSI as in IMS, all service configuration procedures for MMTEL (Multi Media Telephony), PoC (Push to talk Over Cellular) and Presence are already standardized using XCAP. Hence, USSI can also be accommodated using this option.
However, using XCAP for USSI will only solve half of the problem as only USSD’s MMI mode can be implemented. The application mode (push mode) will in this case still remain non-standardized.

• Other option being explored is the possibility of UE encapsulating the USSD codes in IP messages and forwarding them to the network when being attached via the EPS to the IMS. This could happen either via the Ut interface as XCAP data using HTTP or in a SIP (Session Initiation Protocol) message.

Contrary to the above mentioned wake-up methods which are using simulation of SMS and USSD services in LTE network via IMS, other approach could be to consider possibilities to include wake-up messages in the existing network signaling/procedures that are regularly exchanged between the network and the end devices, but it seems to lead to an unnecessary high signalization traffic load (in the IDLE mode the following procedures are performed: paging, tracking area update), because these procedures are performed simultaneously in the whole tracking area, thus multiplying necessary traffic load for one device to the all devices in the observed area.

IV. CoAP

The Constrained Application Protocol (CoAP) is a light-weight application protocol that supports multicast requests and REST web services between the end-points. It is designed for low-powered devices and it fulfils M2M requirements.

A. CoAP utilization in OMA-DM

Device Management is a process of communication with the end device within a capillary network or over a direct link. Therefore, options for CoAP utilization in OMA-DM are replacing the HTTP with CoAP in general or translation from HTTP to CoAP with HTTP/CoAP (HC) proxy [4]. OMA-DM protocol enables multiple SyncML messages per SyncML package and each message is transferred in a separate HTTP request or response. In order to employ CoAP, the CoAP binding for the OMA-DM protocol must be defined. This binding will provide the fixed set of OMA-DM operation for device management procedures. Since CoAP is not session oriented protocol, the CoAP transport layer may split the messages into many requests, followed by a Final ending request. Accordingly, if there are multiple SyncML messages per SyncML package to transfer, each message is transferred in a separate set of PUT/GET commands; depending on whether it is a SyncML request or response. The device management procedure is finalized with Final element in the last received SyncML message. For required request header options that CoAP does not have (Accept-Charset, and User-Agent) appropriate mapping must be used i.e. should be overridden within CoAP options in header.

The HTTP methods GET, PUT, POST, and DELETE are mapped directly by CoAP. Other methods that are not supported will return a response “Method not allowed”. In the case when the end device is behind the gateway, HTTP transport can be used from the server to the gateway, and then translation from HTTP to CoAP on the gateway itself. In order to device communicate with the server a bidirectional proxy is required to allow flow of information in both directions. Translation also involves TCP to UDP mapping as the CoAP utilizes UDP.

The impact of these changes will have next characteristics and outcomes on OMA-DM protocol:

• Compliant with OMA-DM Management Object
• Fixed set of operation (single message per SyncML package)
• Not compliant with OMA-DM security model (CoAP security with IPsec and DTLS can be imposed)
• only GET, PUT, POST and DELETE methods for DM procedures can be used
• Single message per SyncML package
• Light-weight protocol based on OMA-DM with fundamental concept change (Session-less)

The CoAP is light-weight and resource oriented, but without efficient message format used instead of the XML there is no expected gains from the protocol replacement. In next section efficient message formats are considered.

V. EFFICIENT MESSAGE FORMATS FOR OMA DM

A. EXI (Efficient XML Interchange)

The EXI format is grammar-driven, meaning that processing is based on regular grammars derived from XML Schema constraints. The EXI standard specifies how information can be passed as a stream of events, thus largely reducing the overhead caused by representing that data as a stream of plaintext characters. The EXI specification allows for a simplified mode of operation called schema-informed mode. Here a schema is used to form a grammar and state-machine limited to that schema, enabling an extremely efficient encoding. The result of the simple state machine is that even the most minimal embedded devices can work directly with the encoding without the need to work with a full XML parser.

B. CoRE Link Format

CoRE Link Format extends Web Links [6] as a payload with additional resource description functionalities for M2M networks using APIs. It does not require any kind of parsing process unlike XML and other binary message formats. A new Internet media type is assigned for this format. This format must be mapped from XML syntax in order to be used with the OMA-DM. This process requires rewriting the messages adding additional burden to implementation complexity.

C. Protocol Buffers

Protocol Buffers (Protobuf or PB) [10] is a Google mechanism for serializing structured data that is being released to wider community. Google uses Protobuf in many projects, so it is tested and stable. This protocol uses a binary encoding which makes the serialized data a bit
smaller and does not require binary data to be encoded before. The data-structures have to be described before serialization by creating a .proto file, compiling it and including the header files in the project. Implementation of PB would consist of defining message formats in a .proto file, using the protocol buffer compiler, using the Java protocol buffer API to write and read messages. In [9], Protobuf was evaluated and compared against the XML. Result shows that the Protobuf in the worst case is ten times more efficient than XML for parsing, message size and power consumption processes.

D. The comparison of proposed formats

Implementation of the proposed formats is different as expected, and the lowest implementation complexity is anticipated with EXI as it “compress” the XML by using binary methods. Protocol Buffers is also a binary protocol, but it is not based on the encoding of the XML, rather on the raw encoded messages that must be predefined and compiled before usage. This message format is highly efficient concerning the energy and processing requirement of the M2M devices.

<table>
<thead>
<tr>
<th></th>
<th>EXI</th>
<th>CoRE Link Format</th>
<th>Protobuf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Low</td>
<td>High</td>
<td>Hard</td>
</tr>
<tr>
<td>Library</td>
<td>Open Source</td>
<td>None</td>
<td>Open source</td>
</tr>
<tr>
<td>Battery</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Processing</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The CoRE Link Format are more complex in term of implementation, and rewriting of all messages is required. Implementation of EXI for device management is relatively easy as opposed to converting an XML information sharing architecture with a binary format, since existing parsers can handle EXI with only a few adjustments. For scalability metrics in large device networks like in case of the LTE, the EXI is actually the plug’n’play solution for OMA-DM that offers the best features. In table 1 all features are summarized.

VI. CONCLUSION

The utilization of OMA-DM protocol for device management in M2M can be considered as an efficient way for exploiting the existing DM protocol that can provide starting point for device management of the low-powered devices. Due the verbosity of XML messages and HTTP transport that are commonly used in OMA-DM, light-weight version of existing formats and protocols for the OMA-DM are considered: the CoAP protocol and efficient payload formats. Since the CoAP is not session oriented protocol, the CoAP transport layer may split the messages into many requests, followed by a Final ending request. These changes will not influence the OMA DM Management Object (MO), but fundamental concept of the OMA-DM is affected: the leveraged protocol will be session-less, it will utilize security on transport level provided by the CoAP (not HTTP), and device management procedures will be available through single SyncML message per request. Replacement of the HTTP with CoAP is not efficient if the message sent over the network is in the XML format. Evaluation of proposed payload formats shows that the CoRE Link Format and Protobuf’s mapping complexity for SyncML messages exceeds the implementation complexity of the EXI. In addition, EXI is easy to implement with only few extra parser adjustments and represents the plug’n’play solution. The standby policy of the M2M devices can be addressed with SMS over SGs, as currently the most promising (interim) wake-up mechanism, but the long term being SMS over IMS.

ACKNOWLEDGMENT

This paper describes work undertaken in the context of the EXALTED project - Expanding LTE for Devices (http://www.ict-exalted.eu). The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° ICT-5-258512.

REFERENCES

