Online Charging Based on Machine Context for M2M Communication in LTE

Ranko Maric, Tomislav Grgic, Maja Matijasevic, Ignac Lovrek

Online Charging Based on Machine Context for M2M Communication in LTE

Ranko Maric, Tomislav Grgic, Maja Matijasevic, and Ignac Lovrek

University of Zagreb, Faculty of Electrical Engineering and Computing,
Unska 3, 10000 Zagreb, Croatia
{ranko.maric,tomislav.grgic,
maja.matijasevic,ignac.lovrek}@fer.hr
http://www.fer.unizg.hr

Abstract. Efficient management of scarce access network resources for growing volume of Machine-to-Machine (M2M) communications play an important role in a Long Term Evolution (LTE) network. Understanding the communication requirements of machine-to-machine (M2M) services, and linking them to technical, as well as economic aspects, is a crucial step towards "smarter" charging of such services. We discuss the capabilities of M2M services to postpone their communication in LTE's core network, called the Evolved Packet Core (EPC), to avoid times when usage of network resources would be expensive (e.g., while the network is congested). We introduce a context of a group of machines, which describes the postponement capabilities of M2M communication, which is used as an input to the online charging process. We illustrate the proposed approach and its benefits using a smart home M2M service as an example.

1 Introduction

The range of potential applications for machine-to-machine (M2M) communications is huge – from wireless sensors and phones to emerging smart infrastructures for transport, utilities, health, and smart cities. The main elements of M2M communication include a set of "(M)achines", a wireless network, and an M2M application entity (AE), typically a remote server, where all the data gathered by the machine are stored and processed. The amount of data globally transmitted between the machines and respective AEs is steadily increasing. By the year 2018, global mobile M2M traffic is expected to reach 900 TB per month (compared to 50 TB per month in 2013) [2]. With other mobile traffic also growing fast, at an estimated annual rate of 61%, mobile network operators (MNOs) will continue to struggle to use (and to monetize as well) the available capacity as efficiently as possible. This includes two parallel and interrelated goals: first, to alleviate peak network traffic during "rush hours", and second, to keep their customers happy by providing an adequate quality of service (QoS) to M2M users and M2M service providers, even if/when the network is congested.

For M2M services, such as smart metering or home energy consumption management, understanding the communication requirements, and linking them to

technical, as well as economic aspects, is a must. One mechanism that does that is based on dynamic congestion-based pricing plans, meaning that the network traffic sent through the mobile network while the network is congested is charged at a higher rate [9, 18]. Congestion-based pricing plans are particularly suitable for M2M services (or classes of services) that have deterministic or foreseeable communication requirements [19]. What the current 3GPP online charging system (OCS), which is responsible for real-time cost calculation and service authorization, does not take into account (by design) are the communication requirements and the contextual situation of M2M services that are being charged [13], nor do the OCS implements the mechanisms for "smarter" authorization of services given their context.

In this paper, we aim to explore the idea of online charging based on machine context for M2M communications in LTE Evolved Packet Core (EPC) network, and probe further by proposing a (very preliminary) specification for an application-specific context of a (set of) machine(s) involved in M2M service provisioning. For example, we envision that smart meters that periodically measure some non-critical environmental parameters, and send their readings to an AE regardless of network conditions (high/low tariff), could be "instructed" by the M2M application to postpone sending their readings to a time when the network is no longer congested (at low tariff). In this scenario, the desired outcome for an MNO would be to reduce the peak network load during critical times (with M2M service providers tending to minimize own cost and avoid sending data at high tariff, given that choice), as well as to better utilize its own scarce resources while accommodating the needs of M2M service providers (and end-users). As opposed to existing scheduling schemes in LTE that manage network resources at physical and data link layers [7, 14], in our approach the application decides whether to postpone the communication or not.

While this high-level concept is quite clear, there are many open issues that are not covered by related work, nor emerging standards (specifically, the 3GPP Machine-Type Communications [1] and the recent oneM2M Candidate Release, August 2014 [3][4]). The M2M service context proposed in this paper is a possible first step. It refers to a group of machines used jointly for a certain purpose (i.e., M2M service), wherein the communication requirements and context of each machine in the group (e.g., above-mentioned postponement capability) affect the context of the whole group. The online charging system considers the context of a group, and determines whether to authorize the communication or not for each machine in the group. A context monitoring mechanism is also introduced, which notifies the charging process on M2M communication postponement capabilities in a given point of time.

The rest of the paper is structured as follows. Section II summarizes the related work, and Section III presents the communication requirements of M2M services. Section IV introduces the idea of M2M service context and describes its use in online charging. Section V presents an example, and Section VI concludes the paper.

2 Related work

A large-scale measurement and characterization of cellular M2M traffic [19] shows that the M2M traffic exhibits diurnal patterns in which M2M traffic peaks correspond to working hours (and thus to intensive smartphone usage), and that the machines are more likely to generate synchronized traffic resulting in bursty aggregate traffic volumes. To alleviate traffic peaks, many M2M scheduling mechanisms have been proposed in the physical layer, such as, a protocol extension of the Random Access Procedure in LTE [8], Physical Resource Block-based scheduling framework [10], or a scattering-based load balancing technique [5]. Although these solutions improve the performance of an LTE access network in cases of M2M communication during congestion times, and may reduce the price of the network resources used, they do not consider a broader context in which the M2M communication takes place. In our approach, however, we consider the context of the respective M2M communications, and represent it as a dynamically changing set of QoS requirements. For the purposes of this work, we use the QoS-based classification of machines that has been proposed in [16].

A key benefit of our approach, compared to the approaches focused only on the physical layer in LTE, is twofold. First, the postponement time of the M2M communication is not limited by the technical constraints of the physical layer in LTE, and second, the goal of postponement (in general case) may be not only to alleviate traffic peaks, but also to achieve more complex application-level goals (e.g., to allow the machines to communicate only during the cheapest daily tariff). Existing literature already offers solutions that partially consider M2M application requirements in charging (e.g., an auction-based traffic management scheme [15], or the Smart Data Pricing approach [18]), but they require an additional software to be installed on the machines (which our approach does not).

This work utilizes the well-known concept of congestion-based pricing [9, 18], which enables charging of network resources by using a higher tariff when the network is congested, and by using a normal tariff otherwise. An example is the Network Load Based Pricing Scheme [17] in LTE, in which network congestion is determined by a threshold parameter, expressed as a certain percentage of the maximum network load. Although congestion-based pricing schemes have shown to help alleviate peaks in network traffic, they also must be followed with charging mechanisms that would (for the benefit of the user) in some cases reject new connections in congestion times to save users' budget. An example of such mechanism is proposed in the charging context model, which we proposed in our previous work [11, 12]. The model of charging context is in this work further extended to encompass the specific requirements of an M2M communication (e.g., a discussion on which information encompasses the charging context in case of M2M communication).

To summarize, this work fills the gaps which are missing in the related work: it relates each machine with a maximum postponement time that matches the machine's communication context; it manages the network congestion by postponing the traffic generated by the machines (when possible, and if economi-

cally justified), without the need to install any additional software on the machines; and it combines the characteristics of the known congestion based pricing schemes with the M2M communication characteristics, which ultimately enables "smarter" online charging of M2M services.

3 M2M communication requirements

For the purposes of this paper, the term *M2M communication* refers to a data transmission that takes place between a respective machine and an M2M Application Entity (AE), which represents an entity (usually a remote server) where all the data gathered by the machine are stored and processed.

3.1 Description of communication requirements of machines

We adopt a QoS-based classification of communication requirements of machines, initially proposed in [16]. The parameters used for classification are the following:

- A requirement for *real-time communication*, which denotes whether an M2M communication should be carried out in real time or it may be postponed;
- A requirement for *accuracy*, which denotes the tolerance for maximum packet loss rate in the M2M communication observed; and,
- A requirement for *priority*, which marks how packets belonging to the M2M communication should be handled in MNO nodes in periods of network congestion.

A combination of those parameters determines communication requirements of a machine at a certain point of time. Based on those parameters, communication requirements of machines are categorized into four classes (Table 1): *Mobile Streaming*, *Smart Metering*, *Regular Monitoring*, and *Emergency Alerting*, as described next.

Mobile Streaming class communication requirements require low packet delay, have high demand for priority, and have low requirement for accuracy. Typical machines that have such communication requirements are audio and/or video streaming devices (e.g., surveillance cameras). They usually consume most of the bandwidth for data transmission, compared to other types of machines. For example, data transmission of a high definition video stream would require up to several Mbits per second data rate. Smart Metering class communication requirements have high demand for accuracy, and have low requirements in terms of real-time communication and priority. High demand for accuracy is needed as any data transmission errors could result in delivering incorrect data to the AE. Typical machines that have such communication requirements include smart metering machines (e.g., for electricity, for water, or for gas). Most smart metering machines communicate on an on-demand basis. Data does not need to be sent immediately upon request, although the metered value may be associated with a fixed timestamp and stored locally in the machine, waiting to be

sent. Therefore, fairly large delays in data transmission can be tolerated. Regular Monitoring class communication requirements have low demands for all the observed parameters. Machines with such communication requirements are usually used for remote control, as well as for monitoring and automation of remote processes. Such machines usually send/receive very small amounts of data, such as, short control and signaling messages. The data may be transmitted either periodically or on demand. Since the transmitted data is usually several tens of bytes in size, loss rate of one in thousand packets is tolerable. Finally, in this class, the M2M communication may be executed in near real-time. Emergency Alerting class communication requirements have high demands for all the observed parameters. Machines with such communication requirements are used for detection of and alerting in emergency situations (security cameras, gas leaking sensors, etc.). Therefore, these machines may generate varying amounts of data (e.g. alarm device would typically send short signaling messages, while a security camera would typically transmit a continuous video signal). Priority of such communication is high, as data needs to be transmitted within a shortest possible time frame, and this has to be done as accurately as possible.

Table 1. Classes of communication requirements of machines

	Mobile Streaming	Smart Metering	Regular Monitoring	Emergency Alerting
Real-time communication	High	Low	Low	High
Accuracy	Low	High	Low	High
Priority	High	Low	Low	High

To summarize, it should be noted that any given machine may have different communication requirements in different contexts. For example, a gas meter would have *Smart Metering* communication requirement if the consumption of gas is within a predefined "normal" range, but it may switch to *Emergency Alerting* if the gas consumption exceeds a certain value.

3.2 Discussion on M2M communication postponement capabilities

In general, a potential capability to postpone M2M communication depends on the technical, economic, and contextual constraints (Fig. 1). Technical constraints are determined by a machine's capability to locally save the data to be transmitted, that is, while the M2M communication is delayed. We state t_t as a maximum time the data may be stored in the machine's local memory. For example, t_t for a video camera would be determined by buffer size, and that of a metering machine by its memory size, both referring to the maximum recording time that can be stored on a machine. Economic constraints relate to the

question of whether it would pay off (either to an MNO or to a person/entity who pays for the M2M communication) to postpone the M2M communication. It is expressed as t_e, which is a maximum time an M2M communication may be postponed given the economic constraints. For example, if the M2M communication is to be initiated while the network is congested, meaning at a higher price per bit (in congestion based tariff model), it could be estimated that postponing the M2M communication (until the congestion is over) by a certain time interval would be economically justified. Contextual constraint is determined by the very communication requirements of the machine observed, in a given situation. It is expressed as t_c, which is a maximum time to postpone the M2M communication given the machine's current communication requirements. For example, a surveillance camera in a smart home would normally have *Mobile Streaming* communication requirements, which would allow fairly large delays, but if there is an emergency situation at home (e.g., a fire), the camera would switch to *Emergency Alerting* communication requirements, which would require the lowest delay possible.

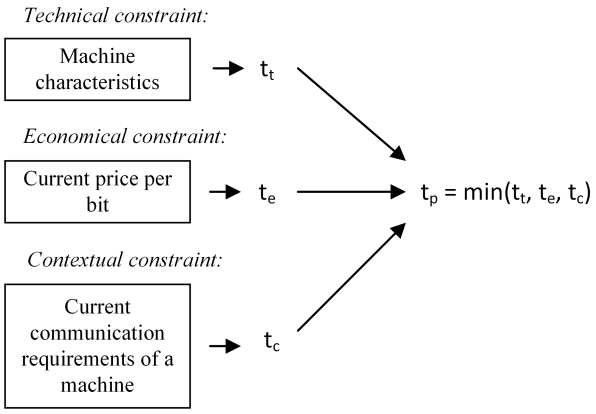

Fig. 1. Technical, economic, and contextual constraints of M2M communication

In general, each machine may be associated with its own t_t, t_e, and t_c parameters, while t_e and t_c may dynamically change while the machine is in use (t_t is in this model considered as a constant for each machine). At the observed point of time, a maximum time to postpone an M2M communication t_p is the minimum value of the t_t, t_e, and t_p current values, $min(t_t, t_e, t_c)$ (Fig. 1). This may result in dynamic changes of the t_p parameter for each machine as well, which would result in different delays (or no delay at all) to be used to postpone the machine's M2M communication at different points of time.

The next step is to determine a relation between the online charging process and the "smart" communication postponement. We elaborate on this issue in the next section.

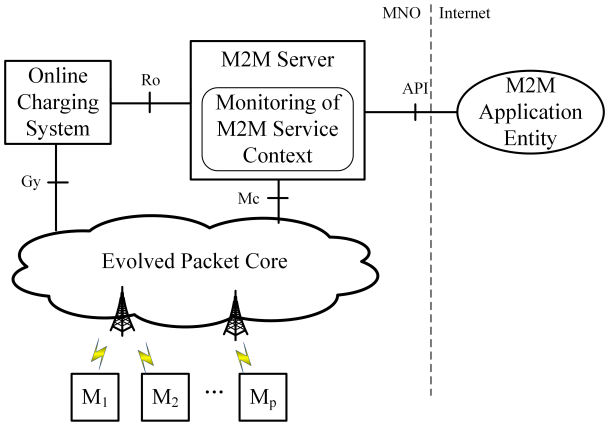

Fig. 2. A reference architecture for M2M service context model

4 A model of an M2M Service Context for online charging

The term "M2M Service" is used to describe an application logic for providing a certain service to a user, by using a group of machines $m_1, m_2, ..., m_p$ and an AE. Machines use MNO's access network to connect to the AE through the EPC (assuming that the AE is situated outside of the MNO's domain), as shown in Fig. 2. The observed M2M communications are established between each machine and the AE. Next, the MNO runs an M2M Server which is responsible for monitoring the M2M Service Context and for coordinating processes of online charging with the OCS, as described later. AE communicates with the M2M Server by using an Application Programming Interface (API), while other functions communicate by using standard Diameter-based reference points (Gy, Ro, and Mc). This functional architecture is in accordance with the standard oneM2M architecture [4, 3]. M2M server in this case would stand as an M2M Common Service in a standard architecture, having the operator-controlled communication model [1].

4.1 Specification of M2M Service Context

The M2M Service Context refers to the capabilities of machines to postpone the respective M2M communications, given the contextual constraints explained earlier. In general, t_c of each machine may dynamically change due to the changes in communication requirements of other machines. Therefore, we represent the M2M Service Context as a Moore Finite-State-Machine (FSM), where each FSM-state represents an unique combination of communication requirements of machines, and hence represents a unique set of machines' t_c values. A transition between two states is triggered by a change of communication requirements of one or more machines. The formal notation of the FSM, used to model the M2M Service Context, is given next:

- M: $\{m_1, m_2, \ldots, m_p\}$ – a group of p machines;
- S: $\{s_1, s_2, \ldots, s_n\}$ – a finite set of n FSM-states, where s_1 is the initial FSM-state;
- Σ: $\{\sigma_{kl}\}, k \in [1, 2, \ldots, x], l \in [1, 2, \ldots, p]$ – input alphabet, where each letter represents one communication requirement of one machine;
- x – a number of communication requirements of each machine;
- T: $S \times \Sigma \rightarrow S$ – a transition function, which switches to the next FSM-state given the previous FSM-state and the input letter;
- Λ: $\{t_{cij}\}, i \in [1, 2, \ldots, n], j \in [1, 2, \ldots, p]$ – output alphabet, where each letter represents current t_c of each machine; and,
- G: $S \rightarrow \Lambda$ – a group of output functions that map FSM-states to outputs as follows:
  g_1: $s_1 \rightarrow \{t_{c11}, t_{c12}, \ldots, t_{c1p}\}$ – output function for s_1;
  $\vdots$
  g_n: $s_n \rightarrow \{t_{cn1}, t_{cn2}, \ldots, t_{cnp}\}$ – output function for s_n.

As each machine establishes an M2M communication with the AE regardless of other machines, each machine will be charged by using a separate online charging process. However, the proposed model of M2M Service Context allows t_c for each machine to be determined based on the current FSM-state (and thus based on the communication requirements of other machines).

4.2 Using M2M Service Context in Online Charging

The OCS could use the M2M Service Context to decide whether to authorize an M2M communication or not (e.g., the OCS might reject M2M communication if current communication requirement of a machine would allow postponement, and if price per bit is currently above the acceptable level from the perspective of a user of the M2M service). To do so, M2M Service Context has to be monitored while the M2M Service is in use (and while the respective M2M communications are being charged). We consider the monitoring process of M2M Service Context to run at the M2M Server (Fig. 2). (The monitoring process for context-based charging has been introduced as a concept in our past works as a part of the context-based charging model. For further information, an interested reader is referred to [13] and [11].) The monitoring process is responsible for the following:

- Maintaining the current FSM-state of a group of machines;
- Changing to a new state if communication requirements of at least one machine in the group have changed; and,
- Informing the online charging process (upon request) whether to authorize an M2M communication or not.

Based on the current communications requirements of machines and on the analysis of the data that the machines send, the M2M AE maintains and regularly updates the M2M Service Context and the t_t values for each machine. The

MNO's M2M Server is updated by the M2M AE in real time with t_t parameters and any changes in the M2M Service Context.

Two key interaction scenarios between the respective entities are shown in Fig. 3 and Fig. 4. Fig. 3 shows a situation in which a machine requests an M2M communication from a relevant Service Provisioning Function (SPF) in EPC (1), and the M2M communication is granted on request (7). The M2M communication request (1) triggers the SPF to request authorization of the M2M communication (2). This would next trigger the OCS to indicate to the M2M Server the current value of t_e (if such value is available or may be determined at the OCS), and to request the M2M communication postponement (3). The monitoring process at the M2M Server determines that no postponement is needed or possible (based on t_p calculation, Fig. 1) (4, 5). Following this, the OCS authorizes the communication (6, 7).

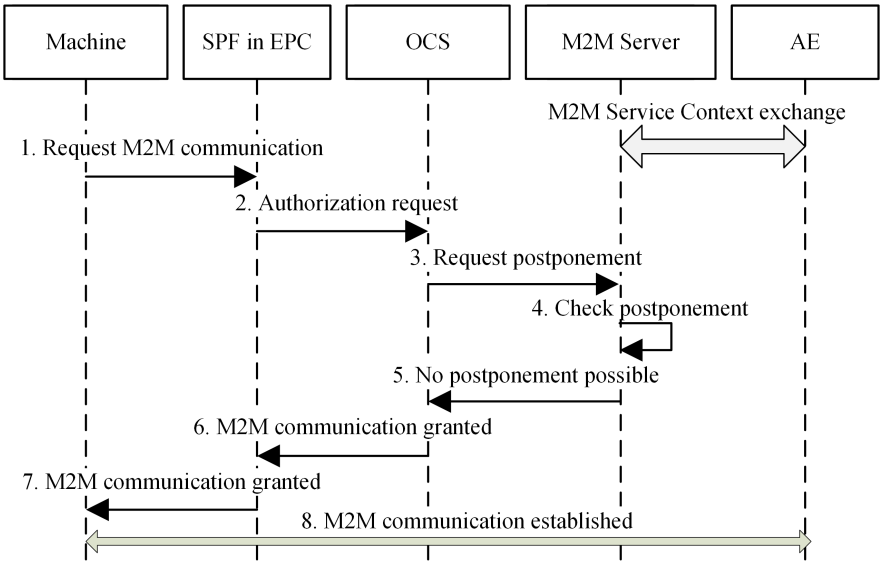

Fig. 3. Interaction between entities in case the requested M2M communication is not postponed

Fig. 4 shows a situation in which a machine requests an M2M communication (1), which is followed by authorization requests that are performed in the same way as in the previous case (2, 3), but the M2M communication is denied (6, 7) based on a postponement decision at the M2M Server (3-5). After the M2M communication is denied, the monitoring process sends the current t_p value to the machine (8). This value is used to initiate a timer at the machine. Once the timer expires, the machine would attempt to establish the M2M communication

again, which results in granting the M2M communication (assuming that the reason for the delay is no longer there).

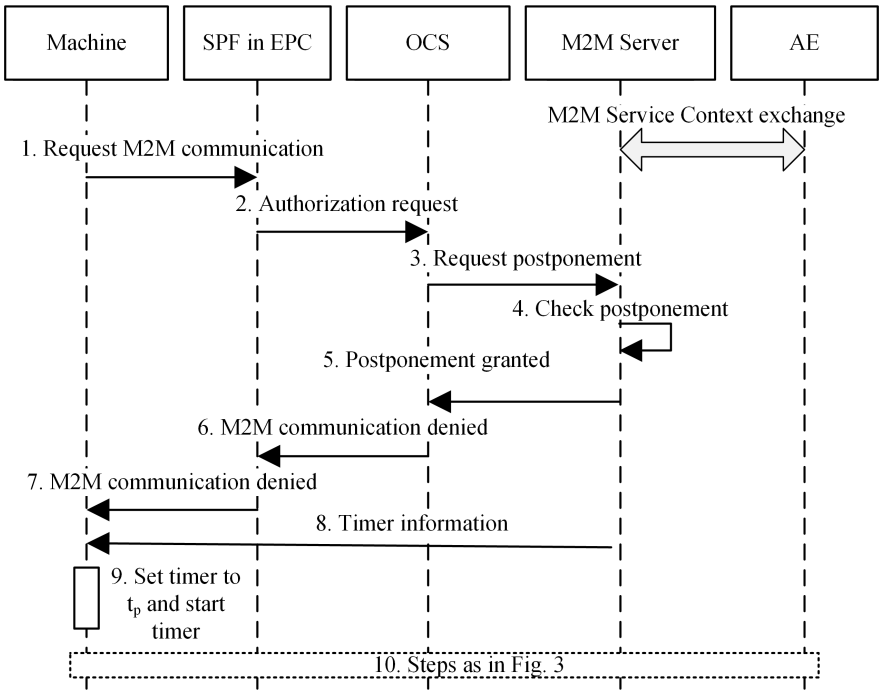

Fig. 4. Interaction between entities in case the requested M2M communication is postponed

5 Example: M2M Service for smart home

We illustrate the proposed model using a smart home heating M2M service as an example. A user has subscribed to a "smart home" M2M service, which allows the service provider to remotely control and operate the central heating equipment installed in the user's home. The following machines are considered (Fig. 5): 1) a gas meter, since in this example gas is used for heating; 2) a heater actuator, which is used to remotely turn the heater on and off; 3) a video streaming machine, which is used for home security video surveillance (of, e.g., home entrance); and 4) a thermometer, which measures current temperature in the home. All the above machines are remotely operated by an AE located in the Internet. All the machines have Internet access enabled over an LTE EPC network, which charges the M2M service provider for the traffic generated by each machine, by using a congestion-based tariff model. (In our initial laboratory

prototype we use emulated sensors and emulated EPC network with periods of network congestion set to 5-second intervals that occur every 30 seconds). The "smart home" M2M service maintains a set temperature in the home, but is also designed to recognize potentially dangerous situations, like gas leak (or, gas flow above acceptable level, which would, for example, trigger the M2M service to shut down the heater).

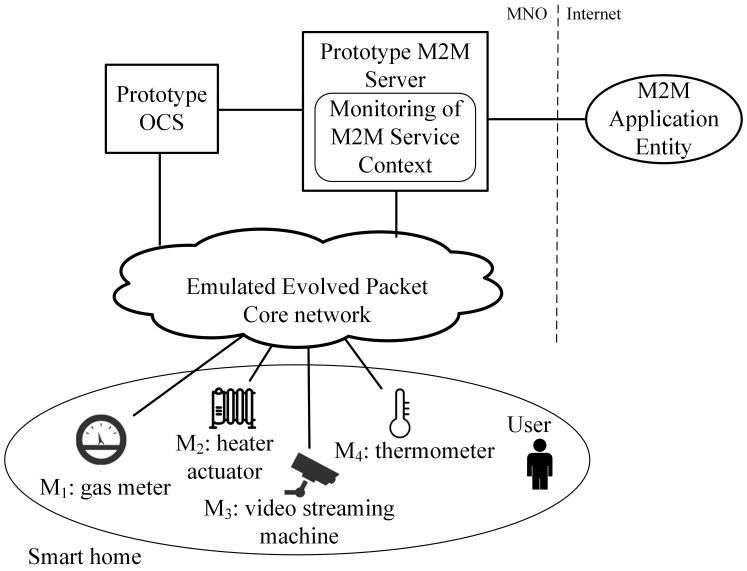

Fig. 5. Group of machines for remote surveillance in an LTE EPC network (emulation)

In the above scenario, it is of interest of the M2M service provider to pay as little as possible for the M2M communication, which occurs between the machines and the AE. Therefore, the M2M communications should be postponed during network congestion when possible. To do so, the M2M service utilizes the M2M Service Context. The M2M Service Context encompasses three FSM-states (s_1, s_2, and s_3), as depicted in Fig. 6. Each FSM-state represents one combination of communication requirements of the group of machines. FSM-states s_2 and s_1 distinguish between the situations in which the user is at home or he/she is not, respectively, but in both cases gas readings are normal. (The M2M Service may ascertain user's absence by, e.g., analyzing the readings from the video streaming machine). FSM-state s_3 describes a situation in which the gas reading is above a normal level. In each FSM-state, each machine is associated with a t_{cij} value, where i represents the state and j represents the machine. For example, thermometer readings may be postponed for 30 minutes if the user is absent, for 1 minute if the user is at home, or the reading could not be postponed at all in s_3 (gas leak – immediate response is required). (It should be noted that the t_c

values in this example are for illustration purposes only.) A transition from s_1 to s_2 (and vice versa) is triggered by a change in communication requirements of M_3 (i.e., the M2M service concludes the user has come home, and increases M_3's t_c value from 1 minute to 10 minutes due to the fact that there are less demands for real-time surveillance when the user is at home). Similarly, transitions from s_1 to s_3 and from s_2 to s_3 are triggered by a change in communication requirements of M_1. In this example, transition from s_3 to s_2 is triggered by a manual system reset, e.g., by a gas technician after fixing the gas system. (Transition from s_3 to s_1 is not allowed for the sake of safety - potential gas leakage must be inspected on-site by a technician, which is possible only if the user is at home.)

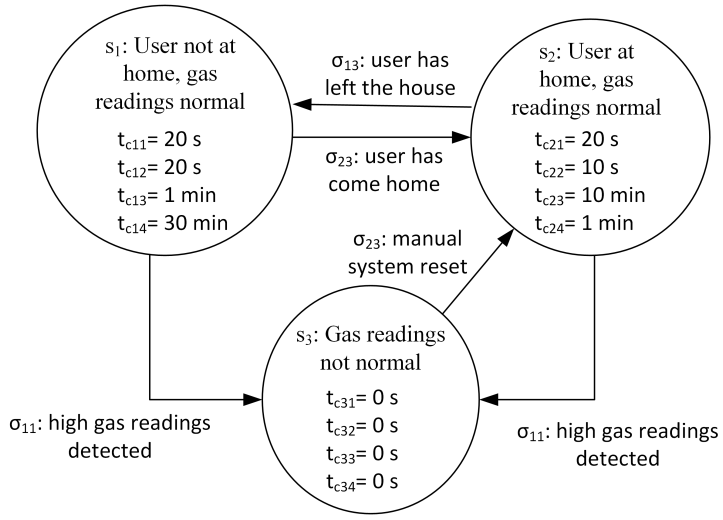

Fig. 6. M2M Service Context of the group of machines: states and transitions

Our setup includes a prototype SPF in EPC, which is responsible for policy-based resource management and charging, a prototype OCS, which performs online charging, and a prototype M2M server, which monitors the M2M Service Context. The emulated machines are implemented in Java, and they are integrated with Policy Control 2.0, a software that emulates policy-based resource reservation mechanisms in LTE and provides a prototype functionality of the OCS and the M2M Server. Policy Control 2.0 has already been used in our previous works [13, 11].

In our demonstration scenario we control the M2M communication of machines from the M2M Server, by granting the M2M communication or postponing it, based on the calculation of the t_p value given the current FSM-state and the t_e (technical constraints of machines were not considered in the demonstration). We use signaling scenarios presented in Fig 3 and Fig 4. In the cases when the communication is requested when network resources are expensive, the commu-

nication is postponed and attempted later (after the congestion is over or when t_p expires).

The demonstration shows the key benefit of utilizing the M2M Service Context in online charging: the decision whether to allow the communication or to postpone it, is made based on both the communication requirements of the group of machines, and on the economic constraint posed by the congestion-based tariff scheme. Therefore, it is left to the M2M Service provider to decide which M2M communication could be postponed and in which contextual situations it could be done, and in times when the network resources are expensive. Moreover, the machines do not have to use the same sink to establish a communication with the M2M AE, nor does their relative geographical locations compromise the ability to maintain the M2M Service Context of the entire group of machines. Finally, the model provides the M2M AE with an ability to generate and change the M2M Service Context based on the application-level decisions of the M2M service, but the full utilization of the model can only be done within the MNO's network (that is, at the M2M Server), where economical constraint is added up to the postponement decision making process.

6 Conclusion

Any given machine may have different communication requirements in different contextual situations. Such situations mostly depend on the application logic of an M2M Service, which would in some cases tolerate postponing a certain M2M communication. By combining the contextual constraint with other relevant constraints of M2M communication (that is, technical and economic), we used a resource authorization capability of an online charging system to "smartly" manage M2M communications in cases when postponement would pay off for both the MNO and the M2M service user. The M2M service management has been realized on a group of machines by monitoring the M2M Service Context (which is represented as a FSM), where each FSM-state reflects a unique combination of postponement capabilities of the machines in the group.

A next step in our research will be to design a simulation model based on the M2M service context, in order to evaluate model performance. We plan to utilize one of the existing M2M context management platforms (e.g., [6]) in the standard M2M architecture [1], and compare the performance of our approach with the related work [15, 18].

Acknowledgment

The research leading to these results has received funding from the project "Information and communication technology for generic and energy-efficient communication solutions with application in e-/m health (ICTGEN)" co-financed by the European Union from the European Regional Development Fund.

References

1. 3GPP 22.368: Service requirements for Machine-Type Communications (MTC) (2014)
2. Cisco visual networking index: Global mobile data traffic forecast update, 2013 - 2018 (Feb 2014)
3. oneM2M TS 0001: oneM2M functional architecture (2015)
4. oneM2M TS 0002: oneM2M requirements (2015)
5. Chang, C.W., Chen, J.C., Chen, C., Jan, R.H.: Scattering random-access intensity in LTE Machine-to-Machine (M2M) communications. In: IEEE Globecom Workshops. pp. 4729–4734 (Dec 2013)
6. Chihani, B., Bertin, E., Crespi, N.: Enhancing M2M communication with cloud-based context management. In: 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST). pp. 36–41 (Sept 2012)
7. Ghavimi, F., Chen, H.H.: M2M communications in 3GPP LTE/LTE-A networks: Architectures, service requirements, challenges and applications. IEEE Communications Surveys Tutorials to appear (2014)
8. Giluka, M., Prasannakumar, A., Rajoria, N., Tamma, B.: Adaptive RACH congestion management to support M2M communication in 4G LTE networks. In: IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS). pp. 1–6 (Dec 2013)
9. Gizelis, C., Vergados, D.: A survey of pricing schemes in wireless networks. IEEE Communications Surveys Tutorials 13(1), 126 –145 (2011)
10. Gotsis, A., Lioumpas, A., Alexiou, A.: M2M scheduling over LTE: Challenges and new perspectives. IEEE Vehicular Technology Magazine 7(3), 34–39 (Sept 2012)
11. Grgic, T.: Online Charging for Services in Communication Network based on User-related Context. Ph.D. thesis, University of Zagreb, Faculty of Electrical Engineering and Computing (Apr 2013)
12. Grgic, T., Matijasevic, M.: 'Smarter' online charging for over-the-top services by introducing user context. In: Proc. 5th Joint IFIP Wireless and Mobile Networking Conference. pp. 81–87 (September 2012)
13. Grgic, T., Matijasevic, M.: An overview of online charging in 3GPP networks: New ways of utilizing user, network, and service-related information. International Journal of Network Management 23, 81–100 (2013)
14. Lien, S.Y., Chen, K.C., Lin, Y.: Toward ubiquitous massive accesses in 3GPP machine-to-machine communications. IEEE Communications Magazine 49(4), 66–74 (April 2011)
15. Lin, G.Y., Wei, H.Y.: A multi-period resource auction scheme for machine-to-machine communications. In: IEEE International Conference on Communication Systems (ICCS). pp. 177–181 (Nov 2014)
16. Liu, R., Wu, W., Zhu, H., Yang, D.: M2M-oriented QoS categorization in cellular network. In: 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM). pp. 1–5 (Sept 2011)
17. Mir, U., Nuaymi, L.: LTE pricing strategies. In: 77th IEEE Vehicular Technology Conference. pp. 1–6 (June 2013)
18. Sen, S., Joe-Wong, C., Ha, S., Chiang, M.: Smart data pricing (SDP): Economic solutions to network congestion. In: Haddadi, H., Bonaventure, O. (eds.) Recent Advances in Networking, vol. 1, pp. 221–274 (2013)
19. Shafiq, M., Ji, L., Liu, A., Pang, J., Wang, J.: Large-scale measurement and characterization of cellular Machine-to-Machine traffic. IEEE/ACM Transactions on Networking 21(6), 1960–1973 (Dec 2013)