Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 321 – 329

Procedia Computer Science

17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

Rich Presence Information in Agent Based Machine-to-Machine Communication

Mario Kusek*, Ignac Lovrek, Hrvoje Maracic

University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia

Abstract

Machine-to-Machine (M2M) communication is characterized by communication between two or more entities that do not need human intervention in a heterogeneous environment. M2M entities in some cases need to turn off communication in order to preserve energy in which cases other entities can not communicate with them. The M2M entities can be aware of such context information and utilize it to perform optimisation in communication. Such information is defined as rich presence information and is used in this paper. The paper extends the mobile agent network with presence context information and applies it in experiments.

Selection and peer-review under responsibility of KES International

Keywords: machine-to-machine communication; presence; agent; multi-agent system

1. Introduction

Machine-to-Machine (M2M) communication is communication between machines that do not necessarily need any direct human intervention. Machines can be devices such as: computers, mobile phones, tablets, sensors, smart grid network elements, embedded processors, cars, industrial and medical equipment, and countless other everyday devices. In such an environment, some devices are powered by batteries and need to run for a long period of time without charging or battery change. Consequently, machines must efficiently use energy by turning off communication when it is not necessary. To address this power efficiency challenge and enable machines to regulate their power consumption, it is necessary to have available the information about a machine's state and context information from environment. By introducing intelligence into a decision making process machine will further improve energy efficiency. We evaluate rich presence as a tool to address the availability of context information, and program agents as a way to enable intelligent decision making.

This paper is organized as follows. In Section 2, rich presence information is introduced as a type of context information that will be used in the M2M system (Section 3). Section 4 reports on related work, while the formal Mobile Agent Network Model is explained in Section 5 and extended with presence information as context. Section 6 presents a case study based on the Mobile Agent Network Model extension and conducted experiments. Section 7 concludes the paper.

*Corresponding author. Mario Kusek Tel.: +385-1-6129-801 ; fax: +385-1-6129-832.
E-mail address: mario.kusek@fer.hr.

Selection and peer-review under responsibility of KES International
doi:10.1016/j.procs.2013.09.109

2. Rich Presence Information

Presence information in general represents information indicating the willingness or ability of a user to communicate with the other users in a communication network. In order to use presence information for and when establishing communication, i.e., placing a call or sending a message, a presence service is required. A presence service mediates between users that provide presence information and users that act upon presence information. An initial model for presence and instant messaging has introduced the notion of "presentity", a communicating entity that provides presence information for notifying its state and changes in its state, and for communicating according to the state. The roles of the presence service are to receive presence information from presentities, store it, and distribute it towards entities that will used it, called "watchers" [1].

Presence information can be provided explicitly by the user or derived automatically from situational information or activity of a user. Proposed formats for representing presence information are based on presence tuples that include status and additional presence attributes. The basic format defined as Presence Information Data Format (PIDF) [2] is extended to the Rich Presence Information Data (RPID) [3] in order to introduce context-related presence information.

There are many definitions and interpretations of context, such as ones presented in early works on context and context-awareness [4, 5]. The basic context classification separates human factors (users, their tasks and social environment) from the physical environment (location, communication and computing infrastructure and physical conditions) [6]. By introducing the context, the presence service is opened to different explicitly or implicitly defined or derived presence attributes. RPID defines presence attributes to associate persons, services and devices with information describing activities (what the person is doing), mood (the mood of the person), relationship (how user relates to the person), place-type (the type of place the person is located in), sphere (current role of the presentity), status (current status of the presentity), class (grouping of similar person elements, devices, or services), device identification (relationship device - service), privacy (observability of the service by other parties) and others. Some attributes relate to each type of presentity, while some relate to human users.

For the purpose of relating rich presence information to M2M communication, the definition given by A.K. Dey will be applied: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [7]. Therefore, "machine" will be treated as an entity that is considered relevant to the interaction with other "machines", performing roles of a presentity or watcher in an agent-based M2M presence service.

3. Machine-to-Machine Communication

Machine-to-Machine (M2M) communication is communication between two or more entities that do not need human intervention [8] [9]. M2M Communication represents a heterogeneous environment that includes devices such as computers, mobile phones, sensors, embedded processors, cars, industrial and medical equipment, etc. The European Telecommunications Standards Institute (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 partnership project: oneM2M. The goal is to develop technical specifications which will be accepted worldwide [10].

The high level M2M functional architecture consists of M2M Devices, M2M Gateways, and M2M Servers [11] which are put into 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.

The problem with M2M devices is that in some cases those devices are dependent on the energy from batteries. In such cases energy must be efficiently used. That is the reason why M2M devices turn off communication and

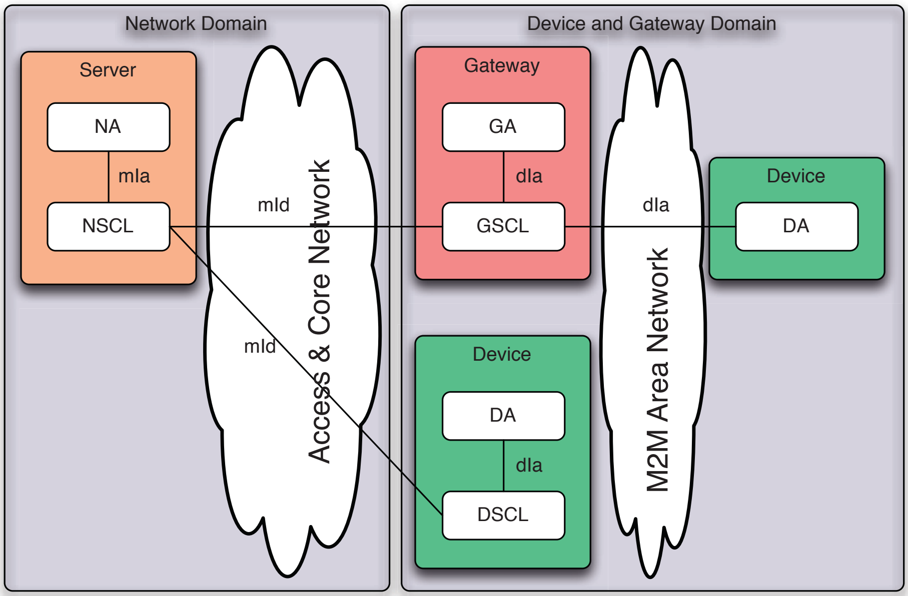

Fig. 1. High Level M2M Functional Architecture

go to hibernate mode for some period of time (inactive state) [12]. In inactive state an M2M device can not communicate and can not be woken up. Each device can autonomously define how much time that device will be inactive and how much it will be active.

If an M2M gateway wants to communicate with an M2M device, it needs to send data in the time-frame when the M2M device is active. For that reason, it is very important that the M2M gateway knows presence information regarding active and inactive state and their future active schedule. The M2M gateway will need to be in watcher role and M2M device in presentity role in the presence domain. The presence information relevant for M2M system is the schedule of the active/inactive time period for each M2M device.

4. Related work

Presence in the context of presence services, being defined as the willingness and ability of a user to communicate across a set of devices with other users, can be applied to a number of different types of users. The research and applications of presence information have so far been focused on human users and their interaction. There are a number of industry standards covering the subject [1], [2], [13]. Paper [14] deals with challenges imposed on the network infrastructure by simple presence services, due to the volume of the messages and related signalling. In [15] the authors propose a novel context-aware approach to presence services. [16] introduces an ecosystem based on rich presence. The common thing in all these applications is that any sensors, if used, are presence enablers, and not presentities. They are used to provide presence information about humans, and not as the subjects of the presence information [17]. To reduce power consumption of machines running in power constrained environments, a method has been used by Texas Instruments in some of their products [18], but it does not use presence information, it is a scanning method instead. To the best of the authors' knowledge, work has not been done on using Rich Presence Information in the context of M2M communication. Our approach is to use stationary agents that will be located on each node. Each node will have one agent that is responsible for managing communication and energy consumption. Agents need to make decisions based on context information which includes rich presence information.

5. Mobile Agent Network Model

A mobile agent network is a formal model of a multi-agent system residing in networked nodes, which host agents allowing their operation, communication and migration. It is introduced in [19] and further developed having in mind functionality, as well as performance issues related to multi-agent systems which include mobile agents [20]. A mobile agent network is represented by the following triple:

MAN = {*A, S, N*}

where *A* represents a multi-agent system consisting of co-operating and communicating agents operating in an environment defined by *S* and *N*. *S* denotes a set of processing nodes in which the agents perform services, and *N* a network that connects processing nodes and allows agent communication and agent mobility. The functionality of *A* is defined by a set of elementary services $ES = \{es_1, es_2, \ldots, es_j, \ldots, es_{nes}\}$. Each elementary service can be provided by multiple competing or collaborating agents. Agents are considered as multiservice ones, i.e., a single agent can support multiple elementary services. An extension towards a context-aware mobile agent network is elaborated in [21]:

CA-MAN = {*A, S, N, C*}

where *C* represents a set of context types handled by the agents. Context types are derived from generic context categories that include human-related and environment-related context. Elaboration of presence information for machines as M2M entities requires two additional steps as follows: i) definition of machine-related context, and ii) definition of machine-specific presence information.

M2M entities are modelled as agents in CA-MAN which means that there are two type of agents: M2M device agent and M2M gateway agent. The context *C* of M2M entity is $C = \{ctx_1, ctx_2, \ldots, ctx_j, \ldots, ctx_{nctx}\}$ where ctx_j is defined as $ctx_j = \{c_{state}, c_{ta}, c_{dc}, c_w, c_p\}$. c_{state} is the state of an entity and can have two values: *active* and *inactive*. c_{ta} is the absolute time when an entity will become active. c_{dc} is the time period of an entity in active state. c_w is a set of watchers that this entity will inform before going to inactive state. The information that is sent to the watcher is c_{ta} and c_{dc}. c_p is a set of presentities that the watcher receives information from. The special case is when an M2M entity (gateway in our case) is always active. In that case, context information is following $c_{state} = active$, $c_{ta} = 0$ and $c_{dc} = \infty$.

6. Case Study: Influence of Rich Presence Information on Energy Efficiency in M2M Communication

Energy consumption of machines is most influenced by communication. Therefore, reduced network traffic results in better energy efficiency. In this case study generated network traffic is measured as indirect measure of energy efficient communication. This is accomplished in a simulated environment using the M2M Network Simulator. The M2M Server tries to retrieve a single sensor reading from all connected M2M Devices. To determine the effect of Rich Presence Information (RPI) on the amount of network traffic generated, this task is performed in two environments, an environment without RPI and an environment with this information available to Gateways. The RPI available in this Case Study is the availability schedule of the M2M Devices, which M2M Gateways use for defining the time when to contact M2M Devices.

6.1. M2M Network Simulator and network architecture

The experiment is simulated using the M2M Network Simulator. It is a derivative work of the Mobile Agent Network Simulator (MAN Simulator) [20]. The MAN Simulator was used since it implements real-world models of network elements such as network switches, links and hosts. The modification of the Simulator was adding the ability to measure traffic between elements in addition to the elapsed time of the simulation.

In the experiment there are three types of nodes: M2M Server node, M2M Gateway node and M2M Device node, which contain one agent on each node: Server Agent, Gateway Agent and Device Agent respectively. There is only one M2M Server node in the system and it is connected to all M2M Gateway nodes. Each M2M Gateway node is connected to M2M Server node and to a set of its M2M Device nodes. Each M2M Device node is connected to only one M2M Gateway node.

The Server Agent initiates the communication by sending a request to each Gateway Agent. A Gateway Agent is an intermediate agent between the Server Agent and Device Agents. A Gateway Agent receives a single,

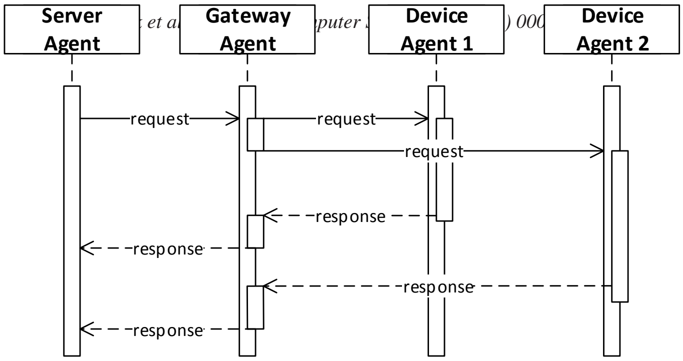

Fig. 2. Communication sequence diagram

aggregated request for all the Device Agents of M2M Device nodes connected to that M2M Gateway node. The Gateway Agent forwards the request to each of its Device Agents. The Device Agent waits for a request and replies to it. When a Device Agent replies, it forwards the response back to the Server Agent (Figure 2).

6.2. Experiment design

The experiment consists of several cases, depending on which of the experiment parameter's value has been varied. The common part is the network topology, as specified before. There is one central M2M Server Agent. It is connected to one or more M2M Gateway Agents. Each Gateway Agent is also connected to one or more M2M Device Agents. Parameters of the experiment are:

- Number of M2M Gateways
- Number of M2M Devices per Gateway
- M2M Device duty cycle

Initially, the Server Agent sends a request to all Device Agents. The number of requests that a Server Agent sends depends on the number of Gateway Agents since the Server Agent sends one request to each Gateway Agent

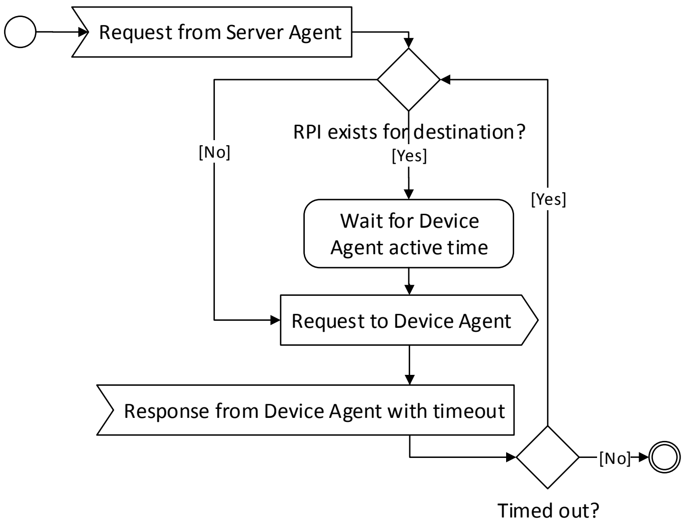

Fig. 3. Request processing activity diagram for Gateway Agent

specifying in it the Gateway Agent's Device Agents. Immediately after a Gateway Agent receives the request it acknowledges the request to the Server Agent. The Gateway Agent then checks whether it has RPI data for each Device Agent to which it needs to forward the request. If the RPI data is available, it schedules the transmission, according to the RPI, for the next scheduled availability. If the RPI data is not available for the requested Device Agent, the request is scheduled for immediate transmission. The Gateway Agent then awaits for the response. If the response is not received in a defined time-out period, regardless of the availability of the RPI data, the Gateway Agent resends the request and the process repeats as if it were a new request for that Device Agent as shown in Figure 3. M2M Device duty cycle is the period, out of a pre-defined time window, when the Device Agent will reply to a request (i.e., a Device Agent with a 5% duty cycle and a 1000 time-step time window will successfully reply to all requests which come during 50 time-steps every 1000 time-steps). The period is continuous and its position inside the time window is determined randomly upon the start of each Device Agent. If the request comes outside of this active period, the Device Agent will receive it and simply ignore it. The time-out values were set sufficiently high to allow for successful response to a request, considering the link speed, if the Device Agent received it during its active period and replied to it.

6.3. Simulation results

For each of the cases, a simulation has been run 20 times. The displayed results are the minimum, the average and the maximum of the traffic values of those 20 runs.

- Case 1: 5 Device Agents per Gateway Agent, 1-10 Gateway Agents per Server Agent, 5% Device Agent duty cycle
- Case 2: 1-10 Device Agents per Gateway Agent, 5 Gateway Agents per Server Agent, 5% Device Agent duty cycle
- Case 3: 5 Device Agents per Gateway Agent, 5 Gateway Agents per Server Agent, 1-99% Device Agent duty cycle

Results for Case 1 (Figure 4) show a dependence of generated traffic on the number of Gateway Agents in the network. Compared to the environment with RPI, the average value of the traffic generated in an environment without RPI is more than a magnitude bigger, with an example of 8738 data units with RPI versus 122383 data units without RPI for 10 Gateway Agents. In the environment with RPI, the results were equal for all the runs with the same number of Gateway Agents. This is due to the fact that the Gateway Agent in each run had to send only one request per Device Agent to receive a response, since it knew when the Device Agent would be available. The minimum and maximum values for the environment without RPI are also generally increasing functions.

Given a large enough number of iterations, the minimum function should equal the environment with RPI since there is some small probability that all Device Agents will receive their requests at the optimal (active) time.

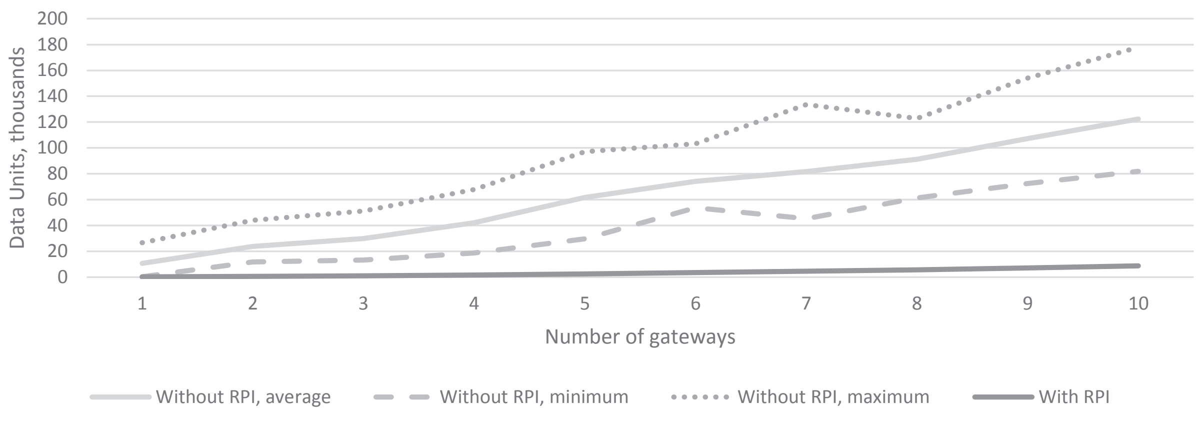

Fig. 4. Case 1, traffic generated for different number of Gateway Agents. 5 Device Agents per Gateway Agent, 5% Device Agent duty cycle.

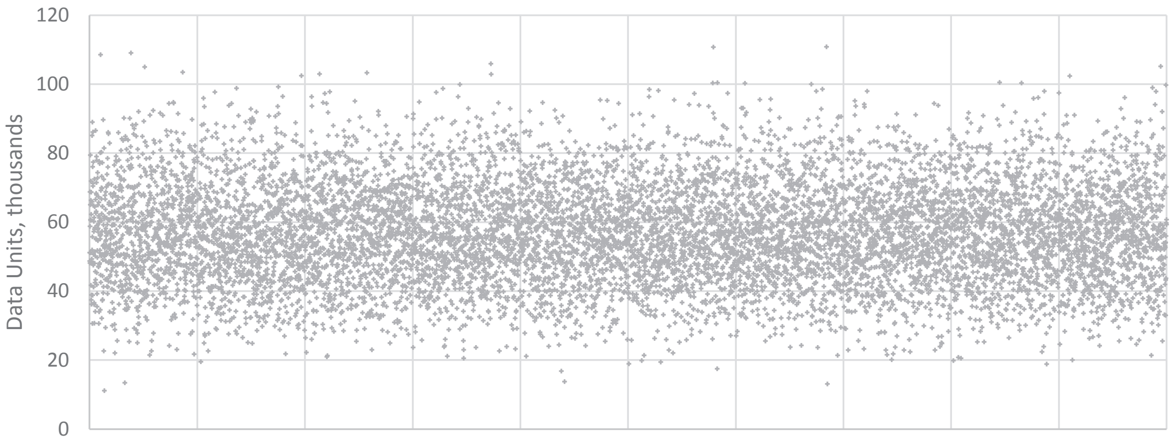

Fig. 5. Dispersion of results for 5 Gateway Agents with 5 Device Agents each, 5% duty cycle.

Figure 5 shows dispersion of results for 5 Gateway Agents with 5 Device Agents each and duty cycle set to 5%. Each mark represents one of 10000 performed iterations. It can be observed that most results are concentrated in the middle, average, area of the chart, and the intensity of the extreme cases decreases with the expected decrease in outcome probability.

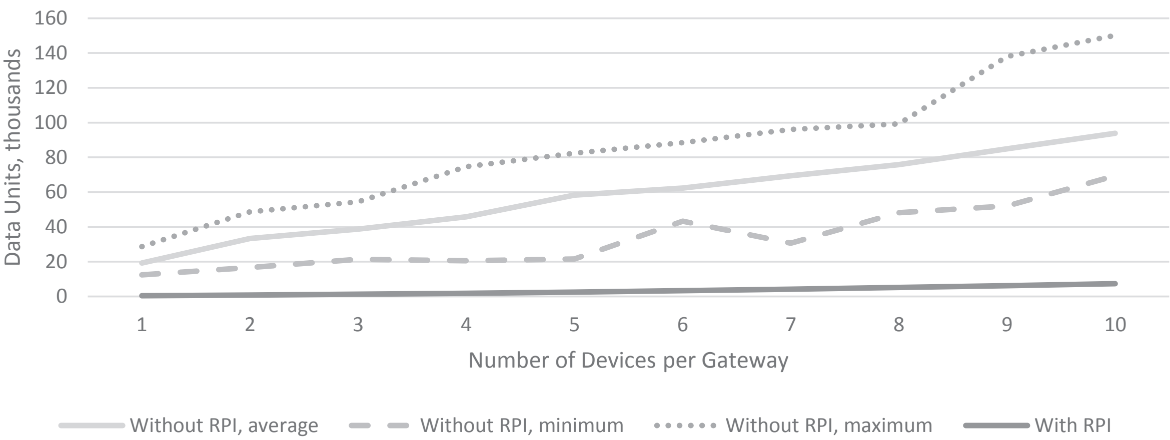

Fig. 6. Case 2, traffic generated for different number of Device Agents per Gateway Agent. 5 Gateway Agents, 5% Device Agent duty cycle.

Results for Case 2 (Figure 6) show the same trends. RPI environment values are more than an order of magnitude less than for the non-RPI environment. The minimum and maximum values for the non-RPI environment follow the average value, but with more discrepancies due to the randomness of the individual runs and the relatively small number of iterations. When comparing the values for the same number of Device Agents, fifty, in Case 1 (10 Gateway Agents with 5 Device Agents each) to the values in Case 2 (5 Gateway Agents with 10 Device Agents each) we can see that the second configuration with more Device Agents per Gateway Agent is more efficient. It is easily explainable due to the fact that it requires less Server Agent to Gateway Agent messages for the same number of Device Agents (as the number of those messages depends solely on the number of Gateway Agents in the network, not on the number of Device Agents).

In Case 3 (Figure 7), for the non Rich Presence Information environment we can notice the trend of roughly linear decrease of generated network traffic as a consequence of the duty cycle increase. This is in line with expectations, since the probability of the Gateway Agent transmitting during the active period of the Device Agent increases with the increase of the duty cycle, therefore lowering the number of retransmissions and generated

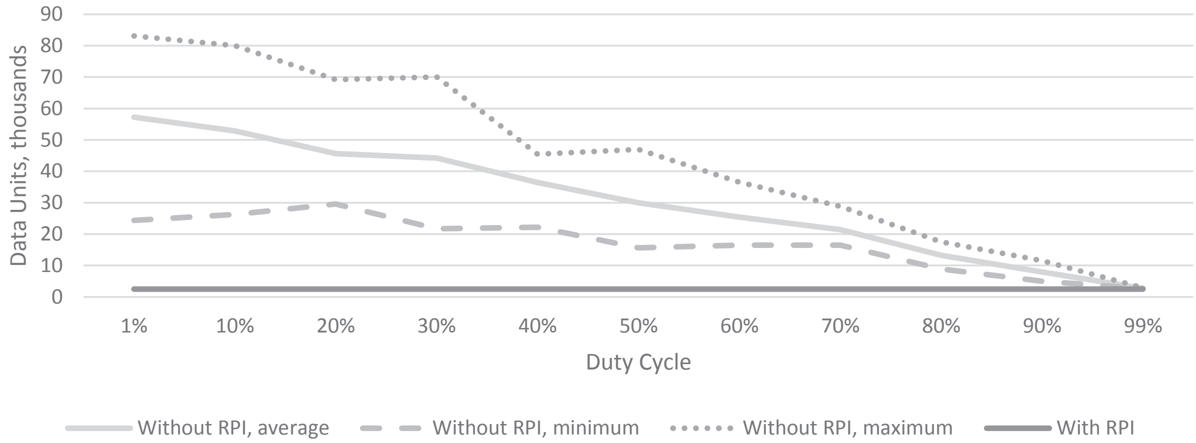

Fig. 7. Case 3, traffic generated for different Device Agent duty cycles. 5 Gateway Agents, 5 Device Agents per Gateway Agent.

traffic. In the Rich Presence Information environment, the value is constant and it does not depend on the duty cycle. This is the expected behavior as long as the length of the active period is at least equal to the length of time needed to transmit the request from Gateway Agent to Device Agent. This is because Gateway Agent's scheduling does not take into the account the delay of data propagation across the link.

7. Conclusion and Future Work

The paper introduced presence information in a Machine-to-Machine Communication System. The presence information is included in a CA-MAN (Context-Aware Mobile Agent Network) formal model and it is used in the case study. Energy consumption in M2M Devices is most influenced by communication. Therefore, that is the reason why reducing network traffic is influencing reduction of energy consumption and increase of energy efficiency. The case study shows that the introduction of presence information and subscription mechanism reduces the number of messages sent in the system. Future work will introduce gateways that can change state, multi level hierarchy of M2M devices, and peer-to-peer presence subscription.

Acknowledgements

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 "Energy Efficient M2M Device Communication" in cooperation with FTW Forschungszentrum Telekommunikation Wien GmbH, Austria.

References

[1] M. Day, J. Rosenberg, H. Sugano, A Model for Presence and Instant Messaging (2000).
URL `http://portal.acm.org/citation.cfm?id=RFC2778`

[2] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, J. Peterson, Presence information data format (pidf), Internet RFC 3863 (August 2004).

[3] H. Schulzrinne, V. Gurbani, P. Kyzivat, J. Rosenberg, RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF), RFC 4480 (Proposed Standard) (July 2006).
URL `http://www.ietf.org/rfc/rfc4480.txt`

[4] B. Schilit, N. Adams, R. Want, Context-aware computing applications, in: Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, WMCSA '94, IEEE Computer Society, Washington, DC, USA, 1994, pp. 85–90. doi:10.1109/WMCSA.1994.16.
URL `http://dx.doi.org/10.1109/WMCSA.1994.16`

[5] M. J. Pascoe, Adding generic contextual capabilities to wearable computers, in: Proceedings of the 2nd IEEE International Symposium on Wearable Computers, ISWC '98, IEEE Computer Society, Washington, DC, USA, 1998, pp. 92–.
URL http://dl.acm.org/citation.cfm?id=857199.858020
[6] A. Schmidt, M. Beigl, H. w. Gellersen, There is more to context than location, Computers and Graphics 23 (1998) 893–901.
[7] A. K. Dey, Understanding and using context, Personal Ubiquitous Comput. 5 (1) (2001) 4–7. doi:10.1007/s007790170019.
URL http://dx.doi.org/10.1007/s007790170019
[8] ETSI TS 102 689 M2M service requirements, http://www.etsi.org/deliver/etsi-ts/102600-102699/102689/01.01.01-60/ts-102689v010101p.pdf.
[9] 3GPP TR 22.868 Study on facilitating machine to machine communication in 3GPP systems, http://www.3gpp.org/ftp/Specs/html-info/22868.htm.
[10] Welcome to onem2m (2013).
URL http://www.onem2m.org/
[11] ETSI TS 102 690 M2M functional architecture, http://www.etsi.org/deliver/etsi-ts/102600-102699/102690/01.01.01-60/ts-102690v010101p.pdf.
[12] I. Bojic, Application of self-organized software agents in telecommunication service provisioning, Ph.D. thesis, University of Zagreb, Faculty of Electrical Engineering and Computing (2013).
[13] P. Saint-Andre, Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence (March 2011).
URL http://tools.ietf.org/rfc/rfc6121.txt
[14] P. Bellavista, A. Corradi, L. Foschini, Ims-based presence service with enhanced scalability and guaranteed qos for interdomain enterprise mobility, Wireless Commun. 16 (3) (2009) 16–23. doi:10.1109/MWC.2009.5109460.
URL http://dx.doi.org/10.1109/MWC.2009.5109460
[15] M. Hauswirth, J. Euzenat, O. Friel, K. Griffin, P. Hession, B. Jennings, T. Groza, S. Handschuh, I. P. Zarko, A. Polleres, A. Zimmermann, Towards consolidated presence., in: CollaborateCom, IEEE, 2010, pp. 1–10.
URL http://dblp.uni-trier.de/db/conf/colcom/colcom2010.html
[16] I. Zarko, M. Kusek, K. Pripuzic, A. Antonic, Presence@fer: An ecosystem for rich presence, in: Telecommunications (ConTEL), Proceedings of the 2011 11th International Conference on, 2011, pp. 133–140.
[17] D. J. Patterson, X. Ding, S. J. Kaufman, K. Liu, A. Zaldivar, An ecosystem for learning and using sensor-driven im status messages, IEEE Pervasive Computing 8 (4) (2009) 42–49. doi:10.1109/MPRV.2009.67.
URL http://dx.doi.org/10.1109/MPRV.2009.67
[18] Application note an047 cc1100/cc2500 - wake-on-radio (2013).
URL http://www.ti.com/lit/an/swra126b/swra126b.pdf
[19] V. Sinkovic, I. Lovrek, Generic model of a mobile agent network suitable for performance evaluation, in: R. J. Howlett, L. C. Jain (Eds.), KES, IEEE, 2000, pp. 675–678.
[20] M. Kusek, K. Jurasovic, G. Jezic, Verification of the mobile agent network simulator - a tool for simulating multi-agent systems, International Journal of Software Engineering and Knowledge Engineering 18 (5) (2008) 651–682.
[21] I. Lovrek, Context awareness in mobile software agent network, RAD (513) (2012) 7–28.