Middleware Group Communication Mechanisms in M2M environments

André Riker, Marilia Curado, and Edmundo Monteiro

University of Coimbra, Coimbra, Portugal
ariker@dei.uc.pt, marilia@dei.uc.pt, edmundo@dei.uc.pt

Abstract. Machine-to-Machine (M2M) communication is a technology that will bring new horizons for the current concept of smart systems. However, efficient M2M communication requires the design of middleware/platform components able to deal with multiple application requirements and heterogeneous wireless environments. In order to address this challenge, this paper proposes the Communication Manager Component (CMC) to integrate the M2M middleware. CMC enables the management of communication mechanisms, such as data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation, aiming to save energy and to satisfy multiple application data requests. The management is performed dynamically taking into account the applications requests, the base-station overload indicators and the M2M devices' status (e.g. energy level, location).

Keywords: Machine-to-Machine; M2M; Middleware; Data-aggregation; Sleep-schedule; Uplink-schedule; Signaling-aggregation

1 Introduction

Machine-to-Machine (M2M) communication is characterized as the autonomous information exchange between electronic devices. Although M2M communication involves any number of devices and unrestricted network technologies, a special attention has been given for M2M communication of a massive number of devices using wireless technologies. Generally, large part of the M2M devices are resource-constrained in terms of memory, Central Processing Unit (CPU) and battery, and communication is performed via mobile and capillary wireless networks.

M2M has emerged as the technology able to remotely control devices, forming a new era of smart applications and enabling new forms of services/applications (e.g. smart systems for transportation, utilities meter, surveillance and healthcare). The exploration of these M2M services has caught the attention of organizations, government and industries, since every entity in the global business wants to maximize their profits by providing better services, reducing cost and faults.

There are many M2M applications being implemented and deployed. However, the current M2M solutions require from the companies the development of comprehensive M2M solutions. For small and middle scale systems, customized solutions could have satisfactory performance. However, by extending the M2M applications for large scenarios, involving millions of devices, it is clear the need of an efficient M2M middleware/platform. The M2M middleware is defined as a service platform that enables different M2M applications to share a set of common functionalities and enables the service providers to reuse the essential functionalities of M2M communication without the need to design a complete communication architecture.

In M2M communication some problems can occur. First, the 3GPP study about M2M communications [1] shows that in M2M communication a huge number of M2M devices may try to connect at the same point in time. If this occurs, the mobile base-station will be overloaded and the communication with all requested devices will not be possible, damaging the M2M communication as well as the traditional Human-to-Human (H2H) communication. Besides the overload due to the amount of connection and signaling messages, the number of traffic sessions and the number of attached devices also can cause the base-station overload. Second, in several scenarios, M2M communication involves resource-constrained devices, which have low power resources. Without an appropriate use of the devices' energy, these devices will need human maintenance, which increases operational costs and reduce the network lifetime. Third, the M2M communication involves applications with a high level of heterogeneity in terms of amount of traffic, frequency of transmissions and delay tolerance. For example, some M2M applications require near real-time communication (e.g. tracking of objects) and other applications are delay tolerant (e.g. smart metering). The problem emerges from the fact that most of these heterogeneous applications must be able to share data. Thus, without an adequate data management, the heterogeneous M2M applications will be detached and they will not be aware about the events detected by other applications.

Knowing these problems and the M2M characteristics, a solution that aims to address these problems together should fulfill (at least) the following requirements: (i) prolong the lifetime of the constrained-resource devices; (ii) avoid the base-station overload; (iii) reduce the application programming complexity, allowing the applications to express their data interests in a high level of abstraction; (iv) manage multiple application interests; and (v) be adapted dynamically according to the level of resources available in the devices involved in the communication.

Four communication mechanisms, namely data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation, show great potential to solve the problems and to satisfy some of the mentioned requirements. Data-aggregation and sleep-schedule are well-known techniques used to extend to network lifetime. Data-aggregation mechanisms process the data gathered from the network to reduce the amount of spatio-temporal data redundancies. On the other hand, the sleep-schedule mechanism aims to keep the devices as much time in sleep mode as possible to save the devices energy. In addition, the uplink-schedule and the signaling-aggregation mechanisms, proposed in [2] and [3], aim to reduce the base-station overload. The uplink-schedule mechanism enables a negotiation between the M2M devices and the base-station to schedule the next transmission time. Performing this scheduling, the base-station provides priority transmissions for devices with lower delay tolerance. The signaling-aggregation mechanism calculates the time interval in which the base-station waits for similar signaling messages from the M2M devices, supporting the aggregation of similar signaling messages.

In addition, a solution involving these four mechanisms should consider how each mechanism affects the others. Firstly, we consider only the data-aggregation and the sleep-schedule mechanisms. In the data-aggregation schemes, the aggregation device must wait to receive data from the neighboring nodes. In the sleep-schedule schemes each device must define the time intervals in which it will be in active mode. If the aggregation device waits too much, the aggregation rate will be high, but it will damage the application requirements. On the other hand, if the sleep-schedule defines a long period of sleep-mode, the savings of energy will be high, but it will affect the application requirements. It is clear that both mechanisms, separately, need to avoid excessive delay and maintain the low energy consumption. However, it is also necessary that both mechanisms must be designed together, considering how the data-aggregation delay affects the sleep-schedule delay and vice-versa.

Besides the data-aggregation and the sleep-schedule integration problem, also there is a synchronization problem involving the sleep-schedule, uplink-schedule and signaling-aggregation mechanisms. The uplink-schedule and the signaling-aggregation mechanisms must be aware of the time that the device will be in active mode, otherwise the uplink-schedule could erroneously determine the next time transmission. Without synchronization the next transmission could be scheduled to a period that the device is in sleep mode, which means that no transmission will occur and no signaling message will be send. Therefore, only with the integrated design of these four mechanisms the M2M communication will achieve high performance, since all these four mechanisms provide performance gain to the network, but without the integration and the dynamic management of these mechanisms, the M2M communication performance is negatively affected.

In this paper, we propose the architecture of the Communication Manager Component (CMC) as part of the M2M middleware that dynamically manages the four communication mechanisms according to the applications requirements, the base-station overload conditions and the M2M devices resources.

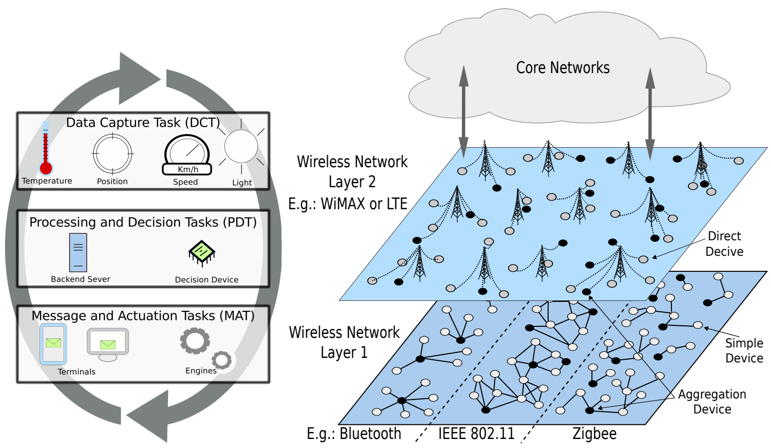
The remainder of this paper is organized as follows. Section 2 gives an overview about M2M communication and middleware. Section 3 describes the related work. Then, Section 4 describes the CMC architecture. Finally, Section 5 presents some concluding remarks and future works.

2 M2M background

In this Section we begin showing the main concepts and the network architecture of the M2M communication. Then, we show how the middleware integrates the M2M system.

2.1 M2M communication overview

To comprehend the M2M communication, Fig. 1.a shows the continuously workflow performed by the M2M system. Firstly, the machines perform the Data Capture Task (DCT), which is the data acquisition from the sensed environment. The electronic sensors conduct the DCT (e.g. temperature, humidity and flow measurement). Then, the machines perform the Processing and Decision Task (PDT), which requires computational power capabilities to manage the data received, and support decision-making functionalities. Finally, some devices perform the Message and Actuation Tasks (MAT), comprising the messages delivery and actions' execution (e.g. alerts/information, or commands to actuators, or relevant events).



a) M2M workflow b) M2M architecture

Fig. 1. M2M communication

Many communication technologies can be deployed to enable the data-flow between the machines that execute DCT, PDT and MAT functionalities. One of the most appropriate network architectures for M2M communication is the heterogeneous wireless architecture. Using this network architecture, the connections could be via short, local or wide wireless technologies, depending on the application requirements and according to the machines resources. In this direction, some of the recent research ([4], [5], [6]) has driven to a Heterogeneous Hierarchical Architecture (HHA).

HHA aims to alleviate the costs, reducing the complexity of nodes. To achieve that, HHA deploys Simple Devices (SD), which are nodes designed as simple as possible, generally, equipped with short wireless technology (see Fig. 1.b). On the other hand, the HHA concentrates some of the vital and complex services in a reduced number of nodes, called Aggregation Devices (AD). The AD nodes perform the complex tasks, such as data-aggregation, Quality of Service (QoS) management, multimedia conversion and remote access. Besides, the AD nodes could be equipped with dual network cards (e.g. short/local and wide wireless cards), acting as Gateway to forward the SD data to the core network or/and to interface the communication between the short and the wide wireless network.

In addition, some M2M scenarios (e.g. vehicular and video surveillance) require nodes with a large bandwidth capacity. Therefore, in some cases Direct Devices (DD) can be deployedq, which are nodes equipped with resources to access directly with the wireless technologies.

2.2 M2M middleware

The M2M middleware is the software component situated between the applications and the devices. Fig. 2 shows an overview of the integration of the M2M middleware within the M2M system.

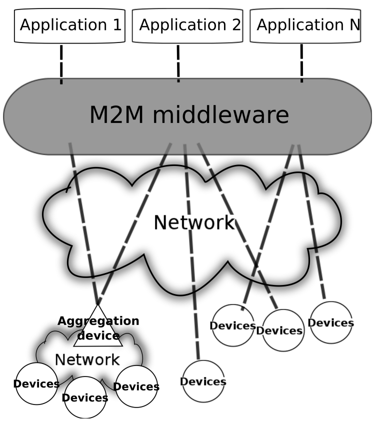


Fig. 2. The M2M middleware

The M2M middleware enables the M2M applications to access their devices using a common set of services, reducing the costs with programming and enabling the interaction of different applications from different stakeholders. For the future M2M communication it is essential the development of an efficient middleware able to support the interaction of multiple applications and to offer to them a common application infrastructure to access the communication mechanisms.

The M2M middleware receives from the applications the data requests and must answer each of these data requests with the appropriated data. To answer the data requests, the middleware can use different transmission technologies to collect the data from the devices. Besides, the M2M middleware will execute its functionalities over distributed devices (e.g. over servers, AD and DD). Then, with the M2M middleware, the development of M2M applications becomes less complex since the developer can use API's to access the middleware functionalities. The middleware approach hides from the applications the heterogeneity in terms of the communication (e.g. wired and wireless) and hardware (e.g. devices from different suppliers).

These middleware characteristics allow the applications to produce data requests with a high level of abstraction, which means that the applications are not aware of communication mechanisms, the network conditions and the devices resources involved in each specific data request. Then, the communication management is not performed at the applications level but by the M2M middleware. Therefore, a fundamental service for the future M2M communication is a management module able to deal with all these communication aspects.

3 Related work

A prominent sleep-schedule mechanism designed to M2M environments is presented in [7]. This solution considers the existence of multiple data types and devices with different sensing capabilities. This proposal defines a monitoring time for each sensing region. In each region, at least one node should transmit data to the M2M gateway during every monitoring time. This monitoring time is defined according to the data sensed. In every monitoring time, the devices have the options to transmit the data or to stay in sleep-mode. Then, this proposal schedules the sensed data transmission to save power while maintaining the gateway with freshness-sensed data.

The uplink-schedule proposed in [2] avoids that M2M devices send synchronized connection requests to the GSM base-station. In GSM networks, for a device to establish a connection, it is necessary to send to the base-station a random access burst via the Random Access Channel (RACH). After receiving the connection request, the base-station assigns a slot in the Access Grand Channel (AGCH). However, while there are approximately 217 RACH slots available per second, only 25 AGCH slots can be assigned per second. Therefore, to solve the AGCH bottleneck problem, the authors in [2] propose that in moments when the base-station is overloaded, the device and the base-station communicate in order to assign the time of the next AGCH slot. To assign an AGCH slot, the proposed mechanism uses the desired time, the report size and the delay tolerance.

The signaling-aggregation proposed in [3] avoids the signaling overload in LTE networks. Frequently, in LTE networks the devices send signaling messages to the base-station (e.g. Tracking Area Updating (TAU) request, massive attach/connect). The Base-station receives these messages and should take the appropriate operation, which in several cases is to inform the Mobility Management Entity (MME) or the Serving GPRS Support Node (SGSN). If a large number of devices sent signaling messages, the MME and the SGSN can be overloaded. Therefore, the mechanism proposed in [3] aggregates the similar signaling messages received by the base-station and informs the MME or the SGSN in a single bulk message. Thus, this mechanism reduces the traffic between the base-station and the MME/SGSN. Moreover, it reduces the resources used to open, maintain and finish MME/SGSN connections.

However, the solutions presented in [7], [2] and [3] do not address the management aspects of these mechanisms, otherwise the management do not taking into account the existence of other communication mechanisms. Moreover, to reduce the scope, most of the mechanism proposals (e.g. data-aggregation, sleep-schedule, uplink-schedule and signaling-aggregation mechanisms) assume the existence of the necessary middleware support, but to support these mechanisms is not a trivial task.

A large number of middlewares proposed to Wireless Sensor Networks (WSN) [8–12] support high level of application abstraction. These middleware solutions reduce the application programming complexity and most of them support data-aggregation schemes to reduce the data volume. Data-aggregation support means that the applications can inform the middleware about the data-aggregation function (e.g. lossy/lossless aggregation, duplicate sensitive, mathematical functions, etc) and the middleware takes the necessary decisions to delivery the requested data to the applications. However, the main drawbacks of these solutions are: (i) only a reduced number of middleware solutions, such as [13–15], support sleep-schedule mechanism. Moreover, even these solutions do not address the sleep-schedule integration with other mechanisms; (ii) except the middleware proposed in [16], none of the analyzed middleware solutions is designed to manage multiple application data requests. Generally the solutions consider a single application making data requests to the devices; and, (iii) the middleware solutions are not designed to be adjusted dynamically according to the devices' resources (e.g. the devices' mobility and energy level).

Therefore, according to the best of our knowledge, none of the middleware solutions consider the integration and the dynamic management of the four mechanisms satisfying multiple applications in a M2M environment.

4 Communication Manager Component Architecture

In this section, we propose the CMC solution, which integrates the M2M middleware and aims to dynamically adapt a set of communication mechanisms to satisfy multiple M2M applications according to the devices' resources involved and the MCN overload level. Therefore, we start this section showing an overview of the component architecture proposed. Then, we describe the input parameters of the proposed component as well as the Requests and Description Modules. Finally, we show the configuration profiles of the communication mechanisms and the Group Communication Module.

4.1 Overview

As shows Fig. 4.1, the CMC architecture is composed by the Requests and Description Modules and the Group Communication Module. The Requests and Description Modules processes the input data coming from the devices and the applications, and select a temporary set of devices that could participate in the communication.

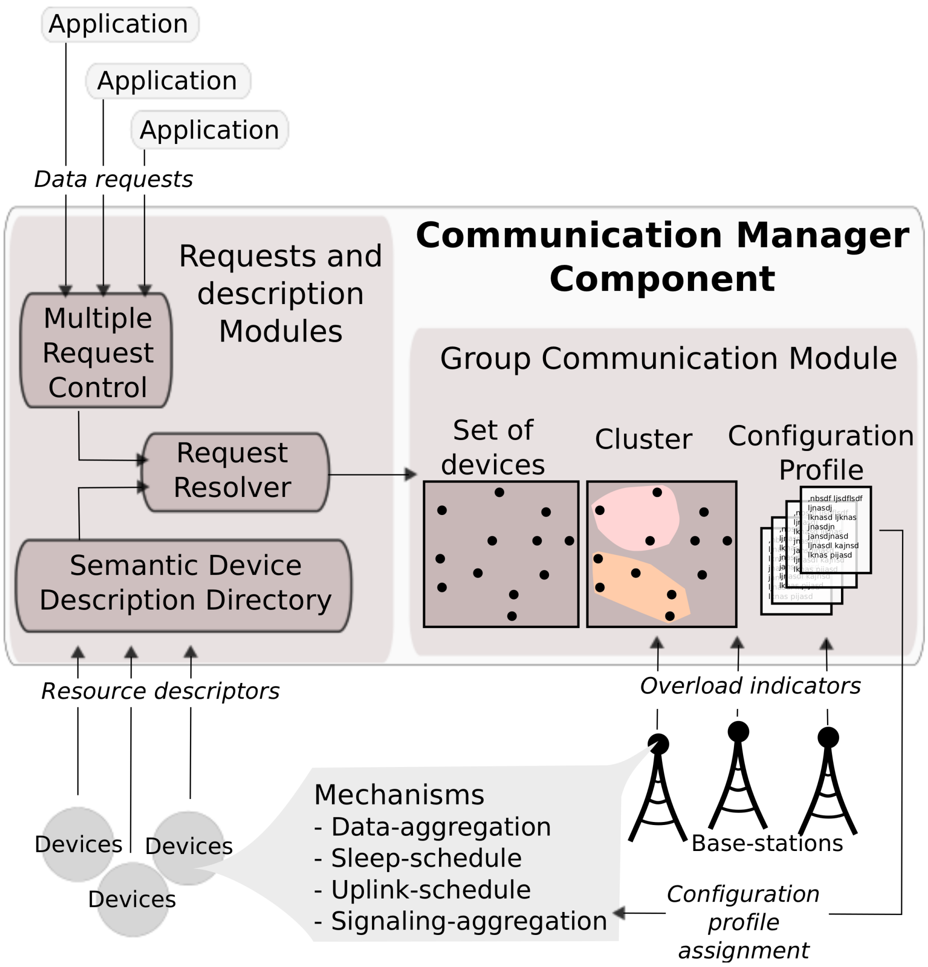


Fig. 3. CMC overview

On the other hand, the Group Communication Module (GCM) receives the temporary set of devices (provided by the Request and Description Modules) and the overload indicators (from the base-stations), and finally performs the core tasks of the CMC, which is the selection of a definitive set of devices, the clusterization of these devices and the assignment of a configuration profile for each cluster. The configuration profiles define the behavior of the four communication mechanisms.

4.2 Input parameters

The selected CMC input parameters are (i) the applications requests, (ii) the semantic resource description of the devices and (iii) the current MCN overload level.

Due to the fact that application requests are expressed in high level of abstraction, the applications do not deal with some of the network complexities, such as the IP addresses of the target group of devices and the type of network technology involved in the communication. In general, the application request denotes the application interests by specifying the data type, location, delay tolerance, level of data accuracy and the desired aggregation function (e.g. the average, min/max). In addition, these application requests could result on a single data transmission or on multiple data communication (e.g. periodical measurements) and it could involve single or multiple devices, but a special attention is given for communication of multiple devices, since data-aggregation is more relevant in this scenario.

The semantic description reveals key characteristics of the devices, such as the type of data sensed, the node position (when possible), the mobility pattern, the network interface(s), the energy level, the CPU and memory capacity. These characteristics are important information for the CMC decision. However, to maintain the consistency of some of these dynamic characteristics it is necessary to periodically access the devices and this access must be executed avoiding the excessive network overhead.

Finally, the management scheme will also receive as input the level of overload from the base-stations involved in the communication. This information gives the overall base-station capacity in terms traffic load and the number of sessions supported.

4.3 Requests and description modules

The multiple application requests received by the M2M middleware must be tested in order to verify the compatibility level between the received request and the active older ones. This functionality is performed by the Multiple Request Control (MRC) module. In the case the new request does not present related requests, it means that the new request does not have any conflict with other active requests. Otherwise, it is necessary to know the data request tolerance ranges (e.g. data accuracy and delay tolerance delay range) and verify the possibility to resolve the conflict.

Another module is the Semantic Device Description Directory (S3D). This mechanism aims to maintain the consistency of information about the M2M devices. As mentioned before, the devices information is related to static hardware characteristics (e.g. CPU, memory and network interface) and dynamic status (e.g. location, energy level and mobility pattern).

Finally, the Request Resolver (RR) module is designed to receive a data request and filter the S3D, returning a temporary set of nodes able to satisfy the data request. However, this set of devices is non-definitive, since it can be modified in case of existence of conflicting data requests or data accuracy level that do not necessitate the data collection from the whole set of devices.

4.4 Configuration profiles of the communication mechanisms

The uplink-schedule and the signaling-aggregation mechanisms actuate in the base-stations, which can have hundreds of devices attached. On other hand, the data-aggregation and the sleep-schedule mechanism actuate in the M2M devices. Therefore, in a hierarchical point of view, the uplink-schedule and the signaling-aggregation are at a higher level of actuation than the data-aggregation and the sleep-mechanism.

The data-aggregation and the sleep-schedule profiles can be configured in terms of energy consumption and communication delay. Considering these two parameters, several profiles can be composed. For example, in some situations could be a good strategy to configure the data-aggregation to have low energy consumption and high delay. Simultaneously, the sleep-schedule can be configured to generate high energy consumption and low delay. By doing this, the sleep-schedule and the data-aggregation will balance the energy consumption and the communication delay. In other situations, it could be a good strategy to configure both mechanisms to have low energy consumption and high communication delay.

According to the profile assumed by the data-aggregation and sleep-schedule, the base-station must synchronize the signaling-aggregation and the uplink-schedule mechanisms. This synchronization means update the signaling-aggregation with a new wait time and the uplink-schedule with a new priority assignment scheme.

4.5 Group communication Module

The CMC performs two decisions. The first is the definition of the definitive set of devices and the second is the selection of the configuration profile that will be assumed by the communication mechanisms. Both decisions are based on the answers provided by the RR module, the base-stations overload indicators and the data request specifications.

The task to define the definitive set of devices starts with the initial set of devices provided by the RR module. This set could be modified (delete and/or include devices) according to data-accuracy tolerance specified in the data request. For example, consider a M2M application aims to measure and control the city water consumption. The application interest is expressed by a data request which orders the measurement of the water consumption in a particular city region with a data accuracy of 90%. Initially, the RR returns the temporary set of devices that matches with the request specifications. As the data request has a data accuracy tolerance of 10%, it means that 10% of devices are not mandatory to provide data, since this data is only for estimation purposes. Therefore, 10% of devices can be deleted from the temporary set. Several metrics can be used to select the devices that must be deleted (e.g. delete the devices with lower energy level or delete devices attached to overload base-stations).

After the definitive set of devices that will participate in the communication, the second decision task is to cluster the devices and assign a configuration profile to each cluster. A cluster of devices could involve multiple aggregation devices and direct devices (See Fig. 1.b). The selected configuration profile will regulate the data-aggregation and the sleep-schedule mechanisms in order to satisfy the application delay and minimize the energy consumption. Moreover, the signaling-aggregation and the uplink-schedule will be updated in order to reduce the traffic overhead and the communication priority, respectively.

5 Conclusion and future works

In the M2M era, new forms of communication are possible and new services and applications will be available. By exploring the extensive number of services and the wide range of scenarios, an immense market potential has emerged for the M2M networks, including transportation, utilities, security, retails services and healthcare. However, to provide efficient group communication in M2M environments is a challenge because it requires the design of heterogeneous network technologies, as well as new mechanisms for efficient communication involving multiple applications. Besides, the M2M middleware should be prepared to deal with the multiple M2M applications and to manage the communication mechanisms to achieve high network performance, which includes saving energy and satisfying the application requirements.

The manager component proposed in this paper allows multiple M2M applications to use efficiently the network resources according to the applications' needs, the network overload level and the device resources. This component has importance in a scenario of massive devices like the M2M environment, since to date, there are in the world around five billion of M2M devices connected to mobile networks [17] and the M2M communication will increase this number to 50 billion by the end of this decade.

Some aspects of the proposed middleware component are under definition. The priority for future works is to design of the configuration profiles, strategies and select metrics as well as rules/polices for the decision tasks. Another aspect that will be studied is the overhead impact of the proposed component.

Acknowledgment

This work was partially funded by the Fundação para a Ciência e a Tecnologia (FCT, Portugal) by PTDC/EEA-CRO/108348/2008 MORFEU; and CAPES and CNPq (Brazil) through of the Ciência sem Fronteiras Program/2013.

References

1. 3GPP: System improvements for machine-type communications. **V0.5.1.** (July 2010) TR 23.888
2. Lioumpas, A.S., Alexiou, A.: Uplink scheduling for machine-to-machine communications in lte-based cellular systems. In: GLOBECOM Workshops (GC Wkshps), 2011 IEEE, IEEE (2011) 353–357
3. Taleb, T., Kunz, A.: Machine type communications in 3gpp networks: potential, challenges, and solutions. Communications Magazine, IEEE **50**(3) (2012) 178–184
4. Zhang, J., Shan, L., Hu, H., Yang, Y.: Mobile cellular networks and wireless sensor networks: toward convergence. Communications Magazine, IEEE **50**(3) (march 2012) 164 –169
5. Igarashi, Y., Ueno, M., Fujisaki, T.: Proposed node and network models for an m2m internet. In: World Telecommunications Congress (WTC), 2012. (march 2012) 1 –6
6. Wu, G., Talwar, S., Johnsson, K., Himayat, N., Johnson, K.: M2m: From mobile to embedded internet. Communications Magazine, IEEE **49**(4) (april 2011) 36 –43
7. Fu, H.L., Chen, H.C., Lin, P., Fang, Y.: Energy-efficient reporting mechanisms for multi-type real-time monitoring in machine-to-machine communications networks. In: INFOCOM, 2012 Proceedings IEEE, IEEE (2012) 136–144
8. Kumar, M., Schwiebert, L., Brockmeyer, M.: Efficient data aggregation middleware for wireless sensor networks. In: Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on. (2004) 579–581
9. Manjhi, A., Nath, S., Gibbons, P.B.: Tributaries and deltas: efficient and robust aggregation in sensor network streams. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ACM (2005) 287–298
10. Nath, S., Gibbons, P.B., Seshan, S., Anderson, Z.R.: Synopsis diffusion for robust aggregation in sensor networks. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM (2004) 250–262
11. Lindsey, S., Raghavendra, C., Sivalingam, K.: Data gathering algorithms in sensor networks using energy metrics. Parallel and Distributed Systems, IEEE Transactions on **13**(9) (2002) 924–935
12. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: A tiny aggregation service for ad-hoc sensor networks. ACM SIGOPS Operating Systems Review **36**(SI) (2002) 131–146
13. Schiele, G., Becker, C.: Experiences in designing an energy-aware middleware for pervasive computing. In: Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on. (2008) 504–508
14. Vasanthi, N.A., Annadurai, S.: Sleep schedule for fast and efficient control of parameters in wireless sensor-actor networks. In: Communication System Software and Middleware, 2006. Comsware 2006. First International Conference on. (2006) 1–6
15. Xiong, F., Bai, L.: Interoperable wireless sensor network model using multi-agent-based middleware. In: Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on. (2010) 1–4
16. Majeed, A., Zia, T.: Multi-set architecture for multi-applications running on wireless sensor networks. In: Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. (2010) 299–304
17. OECD: Machine-to-machine communications: Connecting billions of devices. OECD Digital Economy Paper (192) (2011)