Self-synchronization of nonidentical machines in Machine-to-Machine Systems

Iva Bojic
University of Zagreb,
Faculty of Electrical Engineering and Computing,
Unska 3, 10000 Zagreb, Croatia
Email: iva.bojic@fer.hr

Mario Kusek
University of Zagreb,
Faculty of Electrical Engineering and Computing,
Unska 3, 10000 Zagreb, Croatia
Email: mario.kusek@fer.hr

I. INTRODUCTION

The notion of common time is very important information in distributed systems (e.g., for maintaining the consistency of distributed data). Namely, due to the lack of global time and imperfections (e.g., skew) of physical clocks, in order to agree on common time, distributed nodes have to synchronize themselves. Imperfections of physical clocks are even more emphasized in heterogeneous distributed systems called Machine-to-Machine (M2M) systems in which communication refers to the communication among nonidentical machines that communicate using different types of communication technologies without, or with limited human intervention [1].

Since the M2M is a concept that implies a high level of independence among communicating machines, it is necessary to use time synchronization mechanisms that are scalable, robust and adaptable. From the taxonomy proposed in [2], it can be concluded that a self-synchronization mechanism based on a Pulse-Coupled Oscillator (PCO) model has the best properties to be used in such systems. The PCO model was proposed by Mirollo and Strogatz who modeled firefly-inspired synchronization [3]. They showed that synchronization can be achieved in fully-meshed systems in which physical connectivity exists among all components in a system. Lucarelli and Wang proved that synchronization can also be achieved within meshed systems where connections among components are described with a connected graph in which its edges join only neighboring components [4]. A graph is connected if there is a path between any two vertices in it.

Although Mirollo and Strogatz proved that when using their model time synchronization can be achieved, their model has several limitations, and therefore cannot be directly applied in M2M systems. Some limitations of the model stem from following assumptions:

- oscillators are the same (i.e., have same frequencies),
- no delays occur in the message exchange among oscillators,
- oscillators cannot join or leave the network nor change their positions in the network (i.e., no mobility), and
- none of oscillators have a faulty behavior that desynchronizes the network.

In this paper we will concentrate on a problem of nonidentical oscillators, i.e., nonidentical machines in M2M systems.

II. NONIDENTICAL MACHINES

In an M2M system, each machine $\imath$ can be described as: $z_\imath = f(\varphi_\imath) + \sum_{\jmath=1}^{N} \varepsilon_{\imath\jmath}(t)\ g_{\imath\jmath}(t)$, where $z_\imath$ is machine's $\imath$ state variable, $f(\varphi_\imath)$ describes the *excitation* evolution of the machine's oscillator (i.e., its intrinsic frequency), N denotes the number of machines in the system, $\varepsilon_{\imath\jmath}$ is a coupling constant, while $g_{\imath\jmath}(t)$ is a coupling function between machines $\imath$ and $\jmath$ [5]. The value of the state variable is between 0 and 1, and each time when it becomes 1, the machine's oscillator "fires", i.e., the machine sends *time synchronization messages* to its neighbors. The time between two "flashes", when no other "flashes" are received, is called a *time synchronization cycle*, which consists of a finite number of *steps*.

Research objective is to answer the following question: can nonidentical machines in a heterogeneous M2M system self-synchronize themselves using a self-organizing principle inspired by fireflies, despite of limitations of the Mirollo and Strogatz model? And if so, time synchronization of which precision can be achieved? Time synchronization precision is defined with the length of the *time synchronization window*, a maximal difference between machines' state variables values once when machines are synchronized, compared to the length of the time synchronization cycle. Required precision of the time synchronization process depends on the type of application or process for which synchronization is needed.

In related work, it is mostly assumed that all oscillators have the same intrinsic frequencies, i.e., the time synchronization cycle length of all machines is constant. However, in M2M systems generally it is not true. Different lengths of time synchronization cycles not only are a result of physical imperfections, but also depend on different time duration of sending and receiving messages. Experiments, conducted in a real-world environment using Libelium Waspmote sensors, showed that time needed for a message to be sent using Bluetooth communication technology is 300 ms longer than when using XBee. This results in different time synchronization cycles lengths, since time synchronization steps (in which messages are sent) of nonidentical machines are not equal, and yet every machine's oscillator has exactly the same number of steps within one time synchronization cycle. After the Mirollo and Strogatz model was implemented on heterogeneous Libelium platform, the conclusion was that time synchronization cannot be achieved. Thus their model was extended with a mechanism for a dynamic frequency adjustment.

III. Dynamic frequency adjustment mechanism

Figure 1 shows: a) time synchronization cycle length in theory; b) time synchronization cycle length in reality; and c) time synchronization cycle length after the proposed mechanism for dynamic frequency adjustment was used [2]. After the dynamic frequency adjustment (Figure 1 c)), the length of the time synchronization cycle was approximately equal to the cycle length in theory (Figure 1 a)), i.e., time synchronization can be achieved. The idea of the dynamic frequency adjustment is based on a model proposed by Ermentrout in 1991 [6]. In this model oscillators have the ability to modify their intrinsic frequencies in order to attain time synchronization. Otherwise, if the frequency adaption cannot be achieved, then, under the assumption that oscillators have different frequencies, it is impossible to ensure that any two oscillators maintain the same phase at all time, i.e., are synchronized [7].

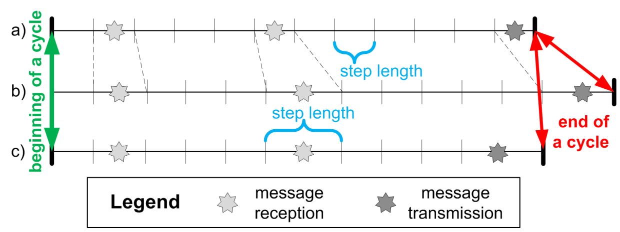

Fig. 1. Length of the time synchronization cycle: a) in theory, b) in reality c) after dynamic frequency adjustment

The proposed dynamic frequency adjustment mechanism is based on the usage of a different number of steps within one time synchronization cycle. For example, in Figure 1 a) time synchronization cycle has 12 steps, while in Figure 1 c) only 10 steps are used. The idea is to constantly measure the length of each time synchronization step, and then at the end of the cycle, if previous steps lasted too long, omit some steps. Even more fine-grained tuning can be made if the length of each step within one time synchronization cycle is adjusted. Namely, each step has to be adjusted in a way that it is longer than the longest time needed for messages reception for all used communication technologies.

This can be done if each machine measures the time needed for a message reception for each communication technology. Then after some time (e.g., three time synchronization cycles) it calculates average time and sends it within a time synchronization message. Finally, each machine, after it receives the message, adjusts the length of its time synchronization step if it is shorter than the received number. Otherwise, the length of the step is not changed. Using that simple rule, after some time every step within every cycle on every machine is approximately the same. Specially, the same rule of the dynamic frequency adjustment is applied to the last step (i.e., step in which synchronization messages are sent).

When implementing the proposed mechanism, two functionalities must be achieved: one that measures the time and the other that suspends the execution of the program. In Waspmote API, *millis()* function counts the number of clock cycles of a Waspmote sensor and returns unsigned long number that denotes the number of milliseconds since the beginning of the program, while *delay (unsigned long ms)* function suspends the execution of the program for ms milliseconds.

IV. Discussion

Testing environment consisted of ten Waspmote sensors: two of them communicated using Bluetooth, two of them had both Bluetooth and XBee communication modules, while others had only XBees. After the proposed dynamic frequency adjustment mechanism was implemented on this heterogeneous sensors platform, it was showed that time synchronization can be achieved with a precision of 69 ms which corresponds to 1.5% of the time synchronization cycle length. Although, this precision cannot be compared with the precision of Network Time Protocol (NTP) in which one can achieve precision of a few microseconds [8], for some application this precision is sufficient (e.g., non-critical distributed data collection). Namely, in order to allow nonidentical machines to use the same time synchronization mechanism, no matter which communication technology they use, the proposed mechanism is implemented on the application layer. A trade off is time synchronization of less precision since other time synchronization mechanisms (e.g., NTP) leverage information available on lower layers and thus have a higher precision.

The proposed dynamic frequency adjustment mechanism enables time synchronization of nonidentical machines in an M2M system. However, since in M2M systems machines communicate using different types of communication technologies, not only do delays appear within one network, but also between different networks. Future research challenges will thus include the problem of nondeterministic message delays. Moreover, another research challenge is to improve the robustness of the self-synchronization mechanism in respect of intentionally designed machines with faulty behaviors which classify them as attackers. Finally, the last open issue is the problem of the scalability of the proposed mechanism.

Acknowledgment

This work was supported by research project: ”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.

References

[1] D. S. Watson and et al., “Machine-to-Machine Technology in Demand Responsive Commercial Buildings,” in *Proceedings of ACEEE Summer Study on Energy Efficiency in Buildings*, 2004, pp. 1–14.

[2] I. Bojic, “Self-organizing Synchronization in Machine-to-Machine Communication Systems Based on a Pulse-Coupled Oscillators Model,” Ph.D. dissertation, University of Zagreb, 2013.

[3] R. E. Mirollo and S. H. Strogatz, “Synchronization of Pulse-coupled Biological Oscillators,” *SIAM J. on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.

[4] D. Lucarelli and I.-J. Wang, “Decentralized Synchronization Protocols with Nearest Neighbor Communication,” in *Proceedings of Embedded Networked Sensor Systems Conference*, 2004, pp. 62–68.

[5] I. Bojic and et al., “A Self-Optimizing Mobile Network: Auto-Tuning the Network with Firefly-Synchronized Agents,” *Information Sciences*, vol. 182, no. 1, pp. 77–92, 2012.

[6] B. Ermentrout, “An adaptive model for synchrony in the firefly Pteroptyx malacae,” *J. of Mathematical Biology*, vol. 29, no. 6, pp. 571–585, 1991.

[7] Z. An and et al., “Nonidentical linear pulse-coupled oscillators model with application to time synchronization in wireless sensor networks,” *Tran. on industrial electronics*, vol. 58, no. 6, pp. 2205–2215, 2011.

[8] NTP v4, *Network Time Protocol Version 4: Protocol and Algorithms Specification*, 2010.