Self-organizing Synchronization in Machine-to-Machine Communication Systems Based on a Pulse-Coupled Oscillators Model

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

Introduction
In distributed systems, very important information is that concerning physical time. 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. Communication in an M2M system refers to the communication among different machines (e.g. personal computers, mobile phones, embedded processors and smart sensors) that communicate using different types of communication technologies without, or with limited human intervention (Watson and et al. 2004), (Darmois and Elloumi 2012).

In my thesis I have proposed a taxonomy for time synchronization mechanisms and classified existing mechanisms using the aforementioned taxonomy. Since the M2M concept implies a highly automated usage of a set of machines, I concluded that a self-organizing synchronization mechanism based on a pulse-coupled oscillators model has the best properties to be used in such systems. In their seminal work Mirollo and Strogatz modeled firefly-inspired synchronization with pulse-coupled oscillators model (Mirollo and Strogatz 1990). They proved 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. A graph is connected if there exists 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.

Pulse-Coupled Oscillators model
In an M2M system, each machine $i$ can be described as:

$$z_i = f(\phi_i) + \sum_{j=1}^{N} \varepsilon_{ij}(t) g_{ij}(t),$$

where $z_i$ is machine’s $i$ state variable, $f(\phi_i)$ describes the excitation evolution of the machine’s oscillator (i.e. its intrinsic frequency), $N$ denotes the number of machines in the system, $\varepsilon_{ij}$ is a coupling constant, while $g_{ij}(t)$ is a coupling function between machines $i$ and $j$ (Bojic and et al. 2012). 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 message to its neighbors. The time between two "flashes" when no other "flashes" are received is called time synchronization cycle, which consists of a finite number of steps.

Research challenges
My research objective is to answer the following question: can different machines in a heterogeneous M2M system synchronize themselves using a self-organizing principle inspired by fireflies, despite of limitations of the Mirollo and Strogatz model? And if so, what is the cost? Time synchronization of which precision can be achieved? Time synchronization precision is defined with the length of 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.

Nonidentical machines
In related work, it is 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 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.

In my thesis, I conducted experiments in a real-world environment using Libelium Waspmote sensors. Measurements showed that time needed for a message to be sent using Bluetooth communication technology is 300 ms longer than when using XBee. That results in different time synchronization cycles lengths, since steps (in which messages are sent) of different machines are not equal, and yet every machine’s oscillator has exactly the same number of steps within one time synchronization cycle.
I implemented the Mirollo and Strogatz model on heterogeneous Libelium Waspmote sensors platform and concluded that time synchronization cannot be achieved. Thus I extended their model with a mechanism for dynamic frequency adaptation and showed that time synchronization can be achieved with a precision of 69 ms which corresponds to 1.5% of the time synchronization cycle length.

**Nondeterministic message delay**

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. Many scientists (e.g. (Tyrrell and et al. 2008)) investigated how delays affected the synchronization process, but they did not take network heterogeneity into consideration.

I have proposed a mechanism based on a simple ping-pong messages exchange during which average time delay for each communication technology can be calculated and afterwards taken into consideration every time when a new message is received. The problem is that each machine adjusts its state variable when it receives a message from its neighbor, and that happens some time after the message was sent. That means that the first machine has the wrong perception about the state variable value of the other machine. Results show that when using ping-pong messages exchange for nondeterministic message delay compensation, the time synchronization window becomes 30% shorter.

**Faulty machines**

When talking about robustness, we can identify two situations which have negative effects on the robustness of the time synchronization. In the first situation, a machine can change its behavior by accident (e.g. due to occurrence of malfunction) and in the second situation, machines are intentionally designed with faulty behaviors which classify them as attackers. In thesis I model a faulty machine as follows:

- within one time synchronization cycle a faulty machine cannot send more than one message to the same machine,
- no non faulty machine can become a faulty one.

The first closure condition ensures that even a faulty machine cannot send more than one message to every other machine in the M2M system during one time synchronization cycle. This is feasible since I assume that faulty machines do not perform jamming or Sybil attacks. Because of the second condition, I propose a mechanism where exchanged messages are protected with logical operation exclusive disjunction (XOR). Namely, since every machine that knows a secret key, exchanged before the time synchronization process, cannot become a faulty one, the only way for a faulty machine to find a secret key is to break a code.

Since messages exchanged between machines are short (80 bits long) and the secret key is the same length as exchanged messages, a faulty machine cannot use frequency analysis to break a code. Moreover, since time synchronization process lasts for only 30 seconds, a faulty machine cannot use brute force to break a code either. I also showed that the usage of XOR operation has a negligible negative effect on the time needed for time synchronization to be achieved.

**Conclusions and future work**

In this thesis summary, I have identified some of the research challenges that emerge when using fireflies as a role model for the time synchronization process in heterogeneous M2M systems. I have conducted research that has tackled the aforementioned research challenges. A result of my doctoral research is an algorithm in which nonidentical machines, as well as machines that use different communication technologies are able to synchronize themselves. Moreover, machines can identify other machines in an M2M system that have faulty behaviors and ignore their messages. Consequently, this increases the robustness of the synchronization process without affecting its scalability.

However, there are some limitations of my work. Namely, in order to allow different machines to use the same time synchronization algorithm, no matter which communication technology they use, my algorithm is implemented on the application layer. A trade off is time synchronization of less precision. Other time synchronization mechanisms leverage information available on lower layers and thus have higher precision. For example, precision of my algorithm is a few milliseconds, while when using NTP one can achieve precision of a few microseconds (NTP v4 2010). Future research questions to be addressed include:

- what happens with the precision of time synchronization when a faulty machine replicates messages,
- what is the scalability of my algorithm and can it be improved by extending the proposed algorithm, and
- how different machines can be, while still being able to synchronize themselves.

**References**


