Sensor Networks Course – Student Presentations

PULSE-COUPLED CLOCKS

Niccolò Cecchinato, nicecchinato@edu.aau.at

INTRODUCTION

Nowadays there is an increasingly use of Internet of Things (IoT) devices, Cyber Physical Systems (CPSs), and distributed sensors that need to be synchronized each other, in order to maintain a coordinated, precise and efficient interaction and communication between them. The synchronization process offers new opportunities that can be used also in complex tasks like (i) *power* saving, where there is the need to have all the nodes that wake up and sleep in coordinated sequences, (ii) *sensing* tasks, for data fusion, (iii) *medium access control*, to coordinate the communications in a sensors network for Time Division Multiple Access (TDMA), cooperative communications, anti-collisions protocols, etc. In this survey the synchronization process is referred to the physical layer of wireless sensor networks and it is intended as the process of achieving and maintaining coordination among independent local clocks via the exchange of local time information.

The most used synchronization scheme sees one master clock, and the remaining clocks as slaves. They have to coordinate their clocks basing on the master's one. For a more precise time stamp, a GPS clock can be used thanks to the atomic clocks running on satellites. There can be also other type of synchronization that extends the master-slave configuration, like the receiver-receiver configuration that sees two nodes receiving the same master time stamp and then they synchronize each other directly, without any intermediate broker (Figure 1).

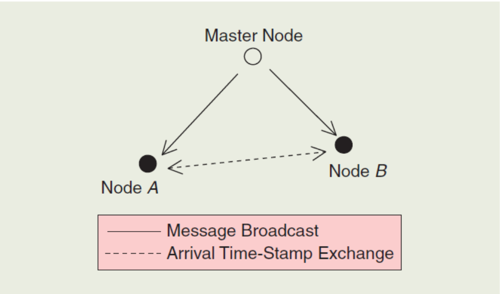

Figure 1- Receiver - Receiver synchronization [1]

However, under normal conditions, the distant clock synchronization presents two main problems: (i) the presence of random delays between the transmission (TX) and reception (RX) of timing signals and (ii) intrinsic hardware and clock inaccuracies. In addition to this, the systems can suffer interference or alterations due to malicious attacks, in order to mess up the sensor network.

The main issues that we can find in the realization of a synchronized wireless network of sensors are: (i) *energy efficiency*: it is essential to find a trade-off between energy consumption and network performance, (ii) *scalability*: performing a correct synchronization with both large networks and small networks, and then (iii) *application specificity*: to design the proper synchronization criteria basing on the application in which the sensors will be used for.

There are different approaches for mutual synchronization in wireless networks. The two main ones are: packet-coupling and pulse-coupling synchronization. The former concerns the periodic exchange of packets carrying time stamps that contain the local time at the sender node. It is very complex and requires large computational resources. The latter updates the node's clock by a periodic train of pulses corresponding to the time instants $t_i(n)$.

PULSE-COUPLING SYNCHRONIZATION

The pulse-coupling synchronization is a technique that provides synchronization by transmitting and processing pulses at the physical layer. The local timing information in encoded directly in the transmission times of given waveforms $g(t)$, as shown in Figure 2. The waveform $g(t)$ can be a band-pass pulse or have a different shape like a pseudorandom sequence. As mentioned before, the clock at each node is represented by a periodic train of pulses corresponding to time instants $t_i(n)$. According to its local clock, each node sends a periodic signal $\sum_n g(t - t_i(n))$ on a dedicated bandwidth of the communication channel or via an overlay system, like ultra-wideband (UWB). Each node's local clock is updated by the received signal that is a combination of waveforms transmitted by the neighbouring nodes (Figure 3).

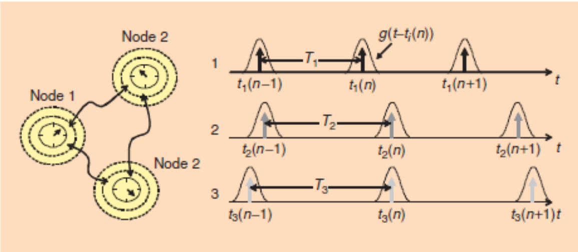

Figure 2 - Each node sends a train of waveforms g(t) for every tick of the local network [2]

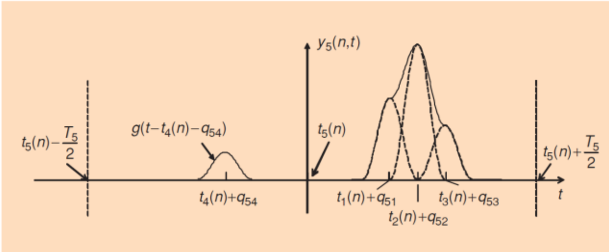

Figure 3 - signal received by a node as a combination of different waveforms [2]

This type of synchronization is naturally scalable because the operation computed and performed from each node are independent from the network composition. Unlike the packet-base technique, this one does not require high computational power, but only simple processing.

For half-duplex wireless communications there are some technological limitations inasmuch as they cannot receive and transmit at the same time. There are two solutions proposed to face this problem. The first is to choose a impulsive waveform $g(t)$ with a short duration allowing nodes to receive a signal in the refractory period before transmitting another. The second solution is to transmit a train $\sum_{n \in \mathcal{N}^*} g(t - t_i(n))$, where $\mathcal{N}^*$ is a subset of clock periods. According to this method, a node transmits its signal while the others are in listening status, to listen the other node's signals.

There are different types of coupled clocks: analog and discrete clocks. Nodes with an analog clock transmit a signal proportional to its local oscillator and updates the instantaneous phase based on the signal received from other nodes. Pulse-coupled discrete time clocks instead have two approaches: integrate-and-fire oscillators and distributed discrete time Phase Locked Loops (PLLs).

PULSE-COUPLED INTEGRATE-AND-FIRE OSCILLATORS

Each node has installed an integrate-and-fire oscillator. In these oscillators a single phaselike variable increases monotonically until it reaches a threshold (in Figure 4a it goes from 0 to 1), is thereby reset to some specific value, and simultaneously triggers the emission of a pulse that is responsible for the mutual coupling. The ticks of the clock correspond to the time instant when the phase returns. If a node detects the pulse sent by another node j at time $t_j(n)$, the node's clock modifies the state function by adding a value ε towards the goal of selecting a firing instant that is closer to that of clock j and adjusts the phase accordingly Figure 4b.

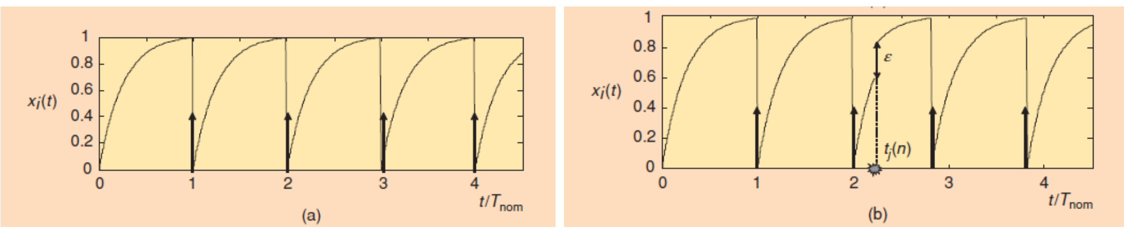

Figure 4 - Pulse-coupled integrate-and-fire clocks: (a) State function $x_i(t)$ (b) State function $x_i(t)$ behaviour in presence of a received pulse. [2]

This method has two main issues: it is difficult to implement in difficult and inaccurate scenarios and the system design is not flexible.

PULSE-COUPLED DISCRETE-TIME PLLs

In the pulse-coupled discrete-time PLLs approach the time is exchanged between neighbouring nodes. Each node calculates a convex combination of the time differences using a time difference detector (TD), as shown in Figure 5, that is fed to a loop filter $\varepsilon(z)$.

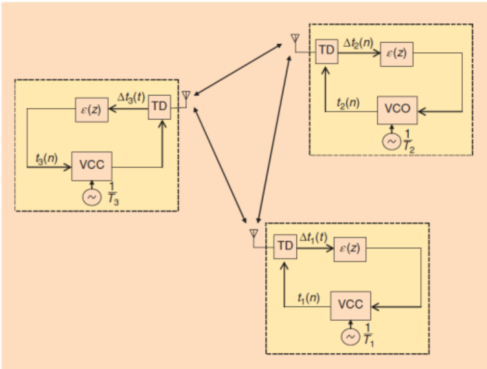

Figure 5 – Diagram of N = 3 pulse-coupled discrete time clocks [2]

The presence of the loop filter brings significant benefits, especially regarding the stability of the system. Compared to the previously described method, this is much more flexible, and can be easily implemented using standard tools. For example, adding a pole μ to the loop filter $\varepsilon(z) = \frac{\varepsilon_0}{1-\mu z^{-1}}$ will reduce the mismatch phases, but with a reduced stability margin. However, the presence of the loop filter will increase the computational complexity of the system due to the high number of operations.

PROPAGATION AND PROCESSING DELAYS

In this paragraph the effect of propagation delays in pulse-coupled PLLs will be explained.

If we have a frequency synchronous network with a common frequency of $\frac{1}{T}$ between all the nodes, there will be a finite propagation delay between two nodes, the ith and the jth. The finite propagation will be called q_{ij}. The node i will receive the impulse from the node j ($t_j(n)$) with the time taken by j to which the propagation time is added, so the time at which the impulse will be received is: $t_j(n) + q_{ij}$. Given the convex combination of the time differences between the two nodes:

$$\Delta t_i(n) = \sum_{j=1, i \neq j}^{N} \alpha_{ij} \cdot (t_j(n) - t_i(n))$$

the time detection error of the ith PPL is: $\Delta t_i(n) + \sum_{j=1, j \neq i}^{N} \alpha_{ij} \cdot q_{ij}$. Adding the local frequency $\frac{1}{T_i}$:

$$T_i = T + \varepsilon_0 \sum_{j=1, j \neq i}^{N} \alpha_{ij} q_{ij}$$

it is possible to ascertain that the propagation delays have the effect of introducing an equivalent frequency offset between the local clocks. It is possible to precompensate it only if the propagation delays are available locally.

REFERENCES

[1] Y.-C. Wu, Q. Chaudhari e E. Serpedin, «Clock synchronization of Wireless Sensor Networks,» *IEEE SIGNAL PROCESSING MAGAZINE*, pp. 124-128, January 2011.

[2] O. Simeone, U. Spagnolini, Y. Bar-Ness e S. H. Strogatz, «Distributed Synchronization in Wireless Networks,» *IEEE SIGNAL PROCESSING MAGAZINE*, pp. 81-97, SEPTEMBER 2008.