Position Papers of the 2013 Federated Conference on
Computer Science and Information Systems pp. 83–88

Time synchronization in Wireless Sensor Networks

Krzysztof Daniluk
Warsaw University of Technology, Institute of Control and Computation Engineering
Email: K.Daniluk@stud.elka.pw.edu.pl

Abstract—**In this paper a survey of selected topics concerning development of wireless sensor network systems, from the point of view of time synchronization algorithms, is presented and discussed. The focus is on currently used time synchronization techniques. Finally is presented a novel concept for time synchronization scheme.**

I. Introduction

AN accurate and consistent sense of time is essential in sensor networks, especially in wireless sensor networks. Distributed wireless sensor networks need time synchronization to coordinate the communication, computation, sensing and actuation of distributed nodes. Used techniques for time synchronization, as well as the newest solution in this field are discussed. The remainder of this paper is organized as follows. The reasons for using time synchronization are described in **chapter II,** clock and frequency issues, as well as the fundamental technique for clock synchronization are presented in **chapter III**, review of time synchronization algorithms for wireless sensor networks are discussed in **chapter IV**, packet delay model is described in **chapter V**, novel concept in time synchronization for wireless sensor networks is presented in **chapter VI.** The paper concludes in **chapter VII.**

II. Reasons for time synchronization

There can be distinguished different reasons to use time synchronization, the most crucial are presented below.

A. Sleep scheduling

One of the most significant sources of energy savings is turning off the radios of sensor devices in the situation when they are not active. It means, that without proper synchronization such technique cannot exist and work correctly and efficiently.

B. Medium-access

TDMA-based medium-access schemes require that nodes are synchronized. There is a need to assign distinct slots for collision-free communication.

C. Coordinated actuation

Distributed actuators require synchronization in order to coordinate the actuators through distributed control algorithms. Time synchronization is again a crucial element in this process.

D. Coordinated signal processing

Time stamps are needed to determine which information from different sources can be fused/aggregated within the network.

E. Multi-node cooperative communication

Multi-node cooperative communication techniques involve transmitting in-phase signals to a given receiver. Such techniques [1] have the potential to provide significant energy saving and robustness, but again, there is required synchronization, as key element of the communication process.

III. Clock and frequency

The clock at each node consists of timer circuitry, often based on quartz crystal oscillators. After each K ticks/interrupts of the timer the clock is incremented.

In real-life conditions, timer circuits, especially in low-end devices, are unstable and error prone.

Let assume following notation:
fo is the ideal frequency,
deltaF – the frequency offset,
Df – the drift in the frequency,
Rf(t) an additional random error process.

It means, that instantaneous oscillator frequency *Fi(t)* of oscillator *i* at time *t* can be modeled as follows:

Fi(t) = fo + deltaF + Dft + Rf(t) [2]

Frequency drift and the random error term may be neglected, so can have a simpler linear model for clock non-ideality, which is following.

Ci(t) = (Alfa_i+ Beta_i)(t)

Where *Alfa_i* is the clock offset at the reference time t=0 and *Beta_i* the clock drift (rate of change with respect to the ideal clock).

It means, that the more stable and accurate the clock, the closer *Alfa_i* is to 0, and the closer *Beta_i* is to 1.

A clock is said to be fast if *Beta i* is greater than 1, and slow otherwise.

It is good to underline, that typical sensor nodes have drift rate of +- 40microseconds per second.

This work was partially supported by National Science Centre grant NN514 672940

Very often are specified manufactured clocks with a maximum drift rate parameter *p*, such that *1-p <= Beta_i <= 1+p*.

Motes, so typical sensor nodes, have *p* values on the order of 40 ppm (parts per million), which corresponds to a drift rate of +-40 microseconds/second.

That means, that any two clocks that are synchronized once may drift from each other at a rate at most *2p*.

On the other hand, to keep their relative offset bounded by gamma seconds at all times, the interval *Tsync* corresponding to successive synchronization events between these clocks must be kept bounded using following formula

Tsync <= gamma/2p

where relative offset is bounded by *gamma* seconds at all times

A. Fundamental technique – Cristian's algorithm

Clock synchronization is the synchronization of both clock drift and clock offset.

The two clocks are synchronized if they are running on the same frequency and showing similar time.

Clock Drift: it is the difference in frequency of the clocks at which they are ticking

Clock Offset: it is the difference of time between two clocks

Accuracy: Accuracy of a clock is how well its time compares with global time

Efficiency: The time and energy needed to achieve synchronization

One of the most significant sources of energy savings is well known mechanism - turning off the radio.

A fundamental technique for two-node clock synchronization is known as Cristian's algorithm [3].

Let assume following, node A sends a request to node B (which has the reference clock) and receives back the value of B's clock, *Tb*. Node A records locally both the transmission time T1 and the reception time T2.

In Cristian's time-synchronization algorithm, there are many sources of uncertainty and delay, which have big influence on its accuracy.

An important issue in synchronization processes is message latency.

Message latency can be decomposed into four components:

- Send time: includes processing time, as well as time taken to assemble and move the message to the link layer
- Access time: includes random delays while the message is buffered at the link layer caused by contention and collisions
- Propagation time: time taken for point-to-point message travel.
- Receive time: time taken to process the message and record its arrival

Approximation of the message propagation time can be done with formula *(T2-T1)/2*.

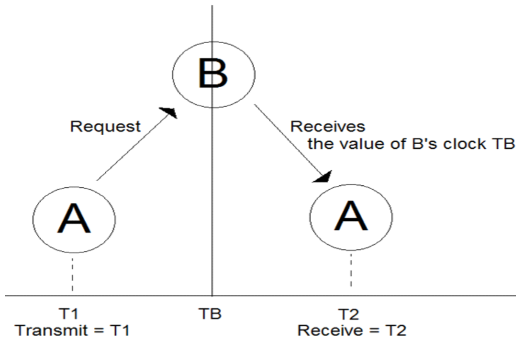

Fig 1. Cristian's synchronization algorithm

If the processing delay is known to be *I*, then it is better to estimate as *(T2-T1-I)/2*.

There can be found more complex approaches, where are taken several round-trip delay samples and are used minimum or mean delays.

B. Network time protocol used on the Internet for time synchronization, based on Cristian's algorithm

The network time protocol (NTP) [4] is used widely on the Internet for time synchronization. The key feature is, that it uses a hierarchy of reference time servers providing synchronization to querying clients, essentially using Cristian's algorithm.

IV. Review of Algorithms for time synchronization in wireless sensor networks

In following chapter are presented and discussed time synchronization algorithms used in Wireless Sensor Networks.

A. RBS – Reference Broadcast Synchronization algorithm

Let assume, that three nodes A, B, C are in the same broadcast domain.

Knowing, that B is a beacon node, i.e. broadcasts the reference signal, that is received by both A and C simultaneously, the two receivers record the local time when the reference signal is received.

Receivers, i.e. nodes A and C then exchange this local time stamp through separate messages.

It is enough for the two receivers to determine their relative offsets at the time of reference message reception.

Presented scheme can be extended to use in greater numbers of receivers.

The key feature of RBS is that it eliminates completely sender-side uncertainty.

There can be seen improved synchronization in scenarios, where are significant sender delays, especially when time stamping has to be performed at the application layer instead of the link layer.

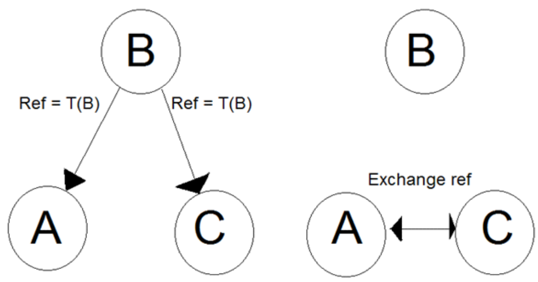

Fig. 2. Time synchronization in RBS algorithm

B. Pair-wise sender-receiver synchronization – timing-sync protocol for sensor networks (TPSN)

The timing-sync protocol for sensor networks (TPSN) [5] is classical sender-receiver synchronization, similar to Cristian's algorithm.

Node A transmits a message, which is locally stamped as T1.

Next, it is received by node B, which stamps the reception time as its local time T2. Then, node B sends the packet back to node A, marking the transmission time locally at B as T3.

At the end the message is received at node A, which marks the reception time as T4.

Assumptions:
Offset between nodes A and B is defined as *DELTA*.
Propagation delay between them is ***d.***

Then,
$T2 = T1 + DELTA + \mathbf{d}$
$T4 = T3 - DELTA + \mathbf{d}$

Result looks following:
$DELTA = [(T2 - T4) - (T1 - T3)]/2$
$\mathbf{d} = [(T2+T4) - (T1+T3)]/2$

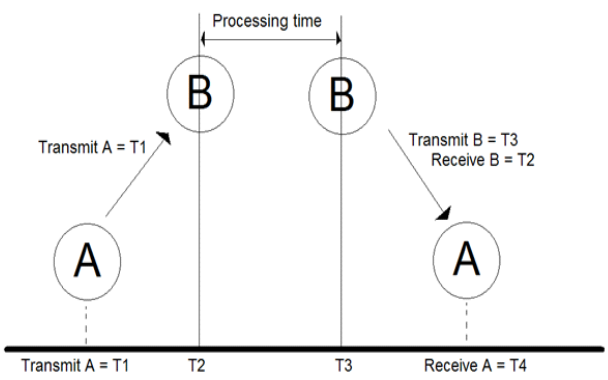

Fig. 3. Time synchronization in TPSN algorithm

The key fact in network-wide time synchronization in TPSN is that time synchronization is obtained level-by-level on a tree structure. For example, nodes at level 1 first synchronize with the root node, then nodes at level 2 synchronize with one node at level 1, it happens until the moment, when all nodes in the network are synchronized with respect to the root.

C. Flooding time synchronization protocol (FTSP)

The flooding time synchronization protocol (FTSP) [6] has big advantage over already presented algorithms, it reduces following sources of uncertainties, which exist in RBS and TPSN algorithms:
- Interrupt handling time
- Modulation/encoding time

Interrupt handling time
Delay in waiting for the processor to complete its current instruction before transferring the message in parts to the radio.
Modulation/encoding time
Time taken by the radio to perform modulation and encoding at the transmitter, as well as the corresponding demodulation and decoding at the receiver.

FTSP uses a broadcast signal from a single sender to synchronize multiple receivers.

Different than in RBS, the sender actually broadcasts a time measurement, and the receivers do not exchange messages among themselves.

Each broadcast provides a synchronization point (a global-local time pair) to each receiver.

In FTSP there are distinguished two main components:
- Multiple time measurements
- Flooded messaging

Multiple time measurements

The sender takes several time stamp measurements during transmission, one after a set of SYNC bytes used for byte alignment. The measurements are normalized by subtracting an appropriate multiple of the byte transmission time, and only the minimum of these multiple measurements is embedded into the message. On the receiver side, multiple time measurements are taken and the minimum of those is used as the receiver time. This is very important, because serves to reduce the jitter in interrupt handling and the (de)coding and (de)modulation time. With few time stamps, improvement in precision can be obtained - in a Mica Mote platform - from the order of tens of microseconds to the order of about one microsecond.

Flooded messaging

To propagate the synchronization information, a flooding approach is used. An important fact, is that a single node in the network provides the global clock. The receiver gets a broadcast message, i.e. a reference synchronization point.

When a receiver has several reference points, then it becomes synchronized itself.

Nodes can collect reference points either from the global reference node. Reference point can be also from other nodes that are already synchronized. The frequency of the flooding provides a tradeoff between synchronization accuracy and overhead.

D. Predictive time synchronization

Predictive time synchronization is only reasonable for very short time intervals. In the real-life conditions, clock drift can vary over time quite a lot because of environmental temperature and humidity changes.

It means, that clock drifts must by continually reassessed.

An simple approach to this measurement is to resynchronize nodes periodically at the same interval.

It may cause unnecessarily high overhead in many cases.

This problem is addressed by the predictive synchronization mechanism [7], where the frequency of inter-node time sampling is adaptively adjusted.

Environments are characterized by a time constant *T* over which drift rates are highly correlated, which can be determined through a learning phase.

Assuming that the time sampling period is *S*, a window of T/S sample measurements is used in this technique not only to predict the clock drift (through linear regression), but also to estimate the error in the prediction.

A technique is used to adapt the sampling period: if the prediction error is above a desirable threshold, the sampling period S is reduced multiplicatively; and if it is below threshold, the sampling period is increased accordingly. Such adaptive scheme provides long-term synchronization in a self-configuring manner.

E. A wireless sensor network system for structural-response data acquisition

A wireless sensor network system for structural-response data acquisition [8] presents an interesting lightweight alternative to clock-synchronization approaches.

The approach is to collect and record latency measurements within each packet.

Only the base station is obligatory to have an accurate reference clock.

Since the radio propagation delays are insignificant, what is measured at each hop is actually the time that the packet spends at each node – which can be of the order of milliseconds due to queuing and processing delays.

Presented approach assumes that time stamps can be added as close to the packet transmission and reception as possible at the link layer. It means, that it is robust to many of the sources of latency uncertainty that cause errors in other synchronization techniques.

An important fact is that it is vulnerable to varying clock drifts at the intermediate nodes.

F. Time Synchronization Protocol for Wireless Sensor Networks using Clustering

Time synchronization means bringing all the sensor nodes to a common notion of time [10]. It is very important to know the order of events that has been sensed by the nodes for data fusion.

In presented algorithm, the root node (RN) initiates the synchronization process by multicasting the Syn_start packet to level-1 cluster heads (CHs), fig. 4.

The Syn_start packet contains packet sending timestamp t1.

After receiving Syn_start packet by all the level-1 CHs, the CH with CH_ID=1 responds back by sending the Syn_ack packet which contain t1, t2 and t3, where t1 is RN send time, t2 is the CH packet receive time and t3 is CH packet send time.

Now RN calculates the propagation delay (d) and multicasts Syn_pkt that contains (d, t) the delay *d* and global time *t*.

When the level-1 CHs receive Syn_pkt with d and t then each CH will compute its drift and set the local clock according to the global time.

When all the level-1 CHs are synchronized, then these CHs become RN for level-2 CHs and the same process will be repeated in the next step.

The purpose of this algorithm is to set the logical clock of the CHs and cluster nodes with global time. The delay (d) and offset can be defined as

$d=[(t2-t1) + (t3-t4)]/2$

$Offset = t + d - LocalTime$

Where *t* is global time, *d* is the delay which will be constant for single hop communication, Offset is the time deviation of the two nodes (i.e. offset), LocalTime is CHs/nodes local time.

In fig.4 is presented message delay estimation process...

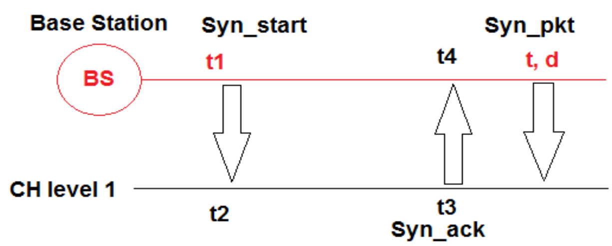

Fig. 4. Message Delay Estimation

… and in fig. 5 multilevel clustering scheme.

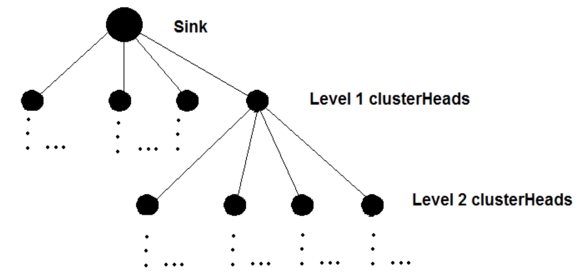

Fig. 5 Multilevel clustering

V. Packet delay model

Packet delay may be decomposed into the following six components.

A. Send Time

The time required by the sender to construct the packet and deliver to the MAC layer. Due to processor load it is non-deterministic.

B. Access Time

The waiting time for packets to get access to the wireless channel, also non-deterministic.

C. Transmission Time

The time that the sender takes to transmit the packet bit by bit at the physical layer, depends on the length of the packet and transmission baud rate.

D. Propagation Time

The time taken by the packet from the sender to receiver on a wireless link. It is deterministic and depends on the distance between sender and receiver.

E. Reception Time

The delay it takes for the receiver to receive the packet. It depends on the packet length and the transmission baud rate.

F. Receive Time

It is the time taken by the receiver to process the incoming packet

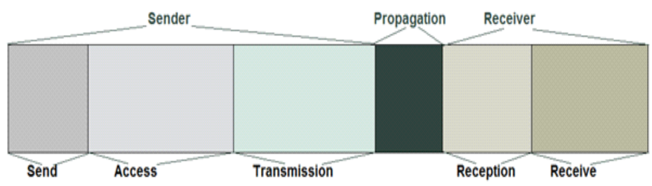

Fig. 6 Packet delay components

VI. Concept of time synchronization in wireless sensor network for health monitoring

The concept for time synchronization in wireless sensor network for health monitoring is presented. The focus is on efficient, reliable synchronization technique.

Network requirements, like traffic loads and latency may be different at different time. Increasing traffic loads cause unacceptable latency, because of data overload. These facts may be dangerous for patients, who e.g. are not yet fully diagnosed.

It is very important, that real-time transmission of life-critical data is guaranteed and latencies are minimum.

An interesting approach seems to be switching between normal state and emergency state, which means dynamically changing of data rate in order to minimize the latencies.

Novel synchronization scheme is proposed to decrease the overhead.

In fig. 7 all data from implantable sensors are within range of synchronization point, which has global clock. All data between implantable sensors go through synchronization point, which is also a kind of base station, covering all implantable sensors on the patient body. When implantable sensor wants to send data to external device, like doctor's computer, these data also go through synchronization point. Suggested synchronization point is following: synchronization with global clock situated in synchronization point is done, when implantable sensors send its readings to synchronization point (which is also a base station collecting all data from sensors and forwarding data further). Synchronization process has to be simple, because implantable sensors have limited resources (limited battery), so it seems to have no way to enforce them to calculate the clock drifts and delays.

All calculations, needed for time synchronization are done on synchronization point side, i.e. the most suitable way seems to be a scenario, when implantable sensor sends its data to synchronization point together with its radio signal strength and timestamp when packet was sent on the sensor side. Synchronization point having these information, can calculate estimated delay between implantable sensor's clock and global clock located in synchronization point. This solution is probably not perfect, but is limiting energy usage on the implantable sensor's side, because only sensing data together with value of radio signal strength of such sensor with time-stamp of sent time on the sensor side are sent and nothing more. The estimated clock difference is sent back to implantable sensor. It is good to underline, that this process occurs only during first communication process between sensor and synchronization point and later only randomly,

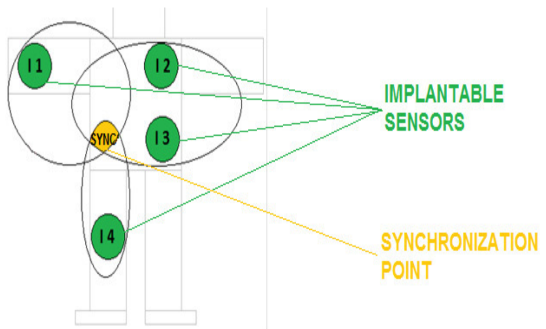

Fig. 7 Synchronization in health monitoring

not during each sending/receiving process. Synchronization time is important for health monitoring, because order of sensed data has to be the same as order of sent and received data between interested endpoints. This is crucial for patient's health.

VII. Conclusion

Synchronization belongs to core configuration problems in WSNs.

It is a fundamental service building block useful for many network functions, including time stamping of sensor measurements, coherent distributed signal processing, cooperative communication, medium-access, and sleep scheduling. Synchronization is necessitated by the random clock drifts that vary depending on hardware and environmental conditions.

Two approaches can be distinguished for time synchronization:

- Receiver-receiver synchronization technique of RBS
- Sender-receiver approach of TPSN

RBS has the advantage that multiple receivers can be synchronized with fewer messages.

It has been shown that these can provide synchronization of the order of tens of micro-seconds.

The flooding-time synchronization protocol improves performance by another order of magnitude by reducing uncertainties due to jitter in interrupt handling and coding/modulation.

An interesting direction for further research seems to be analyzing presented concept for time synchronization in wireless sensor networks in the scenario of use in health monitoring process. In such scenario are used implantable medical devices, which have to synchronize their clocks with one chosen point called synchronization point. There is presented an idea, where each sensor node sends together with sensed data – only during first communication process and later in a random mode - the radio signal strength and time-stamp of sent data on the sensor's side. The synchronization point can then estimate the delay, based on radio signal strength and time-stamp when packet was sent.

References

[1] E.Khandani, J. Abounadi, E. Modiano, L. Zheng "Cooperative Routing in Wireless Networks" Proceedings of Allerton Conference, March 2004

[2] A. Hu, S.D. Servetto "AsymptoticallyOptimal Time Synchronization in Dense Sensor Networks" Proceedings of ACM WSNA, September 2003

[3] F. Cristian "A Probabilistic Approach to Distributed Clock Synchronization" Distributed Computing, 3, 1989, 146-158

[4] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol" IEEE Transactions on Communications, 39, 10, October 1991, 1482-1493

[5] S. Ganeriwal, R. Kumar, M. B. Srivastava „Timing-Sync Protocol for Sensor networks" Proceedings of ACM SenSys'03, November 2003

[6] M. Maroti, B. Kusky, G. Simon, A. Ledeczi, "The Flooding Time Synchronization Protocol", Proceedings of ACM SenSys, November 2004.

[7] S. Ganeriwal, D. Ganesan, M. Hansen, M. B. Srivastava, D. Estrin "Predictive Time Synchronization for Long-Lived Sensor Networks", University of Massachusetts, Amherst, Technical Report TR-04-97, 2004

[8] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, D. Estrin "A Wireless Sensor Network for Structural Monitoring" Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2004

[9] J. Elson, L. Girod, and D. Estring, "Fine-grained network time synchronization using reference broadcasts", Proceedings of Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Vol. 36, pp. 147-163, 2002

[10] G. C. Gautam, T.P. Sharma, V. Katiyar, A. Kumar „Time Synchronization Protocol for Wireless Sensor Networks using Clustering", IEEE International Conference on Recent Trends in Information Technology ICRTIT 2011, pp. 417-422, 2011