Adaptive Time Synchronization Protocol for BANs

Paulo Vilares   Pedro Brandão

Instituto de Telecomunicações, Faculdade Ciências, Universidade do Porto

Rua do Campo Alegre, 1021/1055, 4169 – 007 Porto, Portugal

up090316161@alunos.dcc.fc.up.pt   pbrandao@dcc.fc.up.pt

ABSTRACT

We propose a time synchronization protocol specific for Body Area Networks (BAN). The diversity of hardware on which sensors will be deployed, the need for energy efficiency and the degree of accuracy required by different applications, are constraints that are not always addressed in current synchronization schemes. Using a broadcast message scheme we introduce the ability for the node to decide when to resynchronize, based on a maximum error and the clock drift, which can differ for each type of sensor. The node can therefore, dynamically adapt its synchronization period thus saving energy while keeping accurate. The protocol is designed to support the IEEE 802.15 TG6 guidelines.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: *Network Architecture and Design, Network Protocols*

1. INTRODUCTION

BANs enable a variety of applications both medical and non-medical. In addition to monitoring and data collection, it is necessary to correlate the different data collected from different nodes. For this correlation to be useful and correctly related, timing synchronization between the different nodes is of essence. The protocol needs to take into account (i) the need for energy efficiency, (ii) the diversity of sensor hardware deployed and (iii) the degree of accuracy required by different application.

We will focus on time synchronization protocols proposed for networks that have similarities with BANs, namely protocols for WSNs. We will base our approach in FTSP [1] which uses periodic flooding of synchronization messages

2. TIME SYNCHRONIZATION

Our synchronization protocol is based on the message exchange scheme of FTSP. We will assume: (i) a star network topology, (ii) the base station (BS) is the reference clock, (iii) the hardware on each node can differ, and (iv) the required accuracy (*Emax*) for each sensor can be different and dependent on the application requirements. In a first phase, the BS informs the nodes about the resynchronization interval (*Tsync*). Different hardware clocks will imply different drift rates (ρ), thus to limit the clock offset to the required *Emax* for each type of sensor, the *Tsync* interval must meet the requirement: $T_{sync} = \frac{Emax_i}{\rho max_{bs} + \rho max_i}$.

The BS calculates *Tsync* based on the interval needed by the most stringent requirement for accuracy and the worst clock in the system. The synchronization phase is based on unidirectional broadcast synchronization. Using the transmitted (T_{bs}) and received (T_{rcv}) time the offset between the node's local time and the reference time can be determined by: $offset = T_{rcv} - T_{bs}$.

For the accuracy of time synchronization, the time stamping of beacons and received messages is crucial. The time stamp is done immediately after the SFD byte of the synchronization message that is being transmitted to the radio as suggested in [2]. This approach eliminates most delay times associated with sending and receiving, namely send, access, reception and receive time.

Using two values of the clock offset at time t_1 and t_2, a node i can calculate the clock drift relatively to the BS as: $\rho_{i\to bs} = \frac{offset_{t2} - offset_{t1}}{t_2 - t_1}$.

We compensate for clock drift using a weighted moving average filter, it improves the clock drift compensation based on the clock offset estimation ($offset_{avg}$) along the synchronization process $offset_{avg(t)} = \alpha \cdot offset_t + (1-\alpha) \cdot offset_{avg(t-1)}$.

We introduce the ability for the node to change α during the synchronization process. External factors, like temperature, can introduce clock drift variations and the current estimation should produce more effect on the clock drift estimation than the previous ones.

Based on the current offset, the clock drift and the *Emax* the node decides when to resynchronize (*nextSync*) so not to exceed *Emax*: $nextSync \leq \frac{Emax - offset}{\rho_{i\to bs}}$. This gives an upper bound for the next synchronization interval. As drift varies, the node may need to wake up more often or may sleep during longer times.

The efficiency of our protocol depends on the *Tsync*, at which the BS sends the synchronization messages, and on the application's requirements. Since the nodes do not respond to synchronization messages, the communication cost can be seen as 1 message per *Tsync*. Moreover, a node can dynamically adapt its synchronization period thus saving energy while keeping accurate.

3. CONCLUSION AND FUTURE WORK

We presented a time synchronization protocol specific for BANs. Currently we provide synchronization between the BS and the nodes, nodes do not synchronize between each other. We intend to further investigate the feasibility and costs of using pair-wise synchronization (similar to RBS). The proposed time synchronization protocol is being implemented in the Castalia simulator (part of OMNET++ framework) for its evaluation.

4. REFERENCES

[1] Maróti, M., Kusy, B., Simon, G., and Lédeczi, Á. 2004. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (SenSys '04). ACM, New York, NY, USA, 39-49.

[2] Cox, D., Jovanov, and Milenkovic, A. 2005 .Time synchronization for zigbee networks. *In Proceedings of the Thirty-Seventh Southeastern Symposium*, 2005. 135–138.

This work was partially funded by the VitalResponder project CMU-PT/CPS/0046/2008

BodyNets'12, September 24-26, 2012, Olso, Norway.

BODYNETS 2012, September 24-26, Oslo, Norway

DOI 10.4108/icst.bodynets.2012.249971