A Global Synchronization Scheme for Clustered Wireless Ad-hoc/Sensor Networks

Chao Gao[†] and Riku Jäntti[‡]
[†]Department of Information Technology
Vaasa Polytechnic, Vaasa, Finland (65100)
Email: `gc@puv.fi`
[‡]Department of Computer Science
University of Vaasa, Vaasa, Finland (65101)
Email: `riku.jantti@uvasa.fi`

***Abstract*— In this paper a global synchronization scheme is proposed for large-scale two-layer clustered wireless ad hoc/sensor networks. The scheme shows a very high efficiency to synchronize the cluster heads throughout the whole network because collision of synchronization packets is avoided.**

I. INTRODUCTION

Synchronization is an important issue for many applications in wireless ad-hoc/sensor networks. A synchronization scheme helps the nodes coordinate the transmission/reception of data, avoid collisions, analyse and track events correctly, and perform a more efficient sleeping policy for energy saving.

Synchronization in wireless ad-hoc/sensor networks can be done either globally (i.e., all the nodes in the network agree a common clock time) or locally (i.e., a set of nodes geographically located close agree a common clock time). On the other hand, synchronization can be also classified as external (i.e., there is an external clock reference such as GPS) or internal (i.e., nodes agree a common clock among themselves). In [1], the authors claim that energy efficiency, scalability, robustness, and ad hoc deployment are the key principles of time synchronization design in wireless sensor networks.

In wireless ad-hoc/sensor networks, synchronization is usually performed by passing a time-stamped message to the nodes that need to be synchronized. Two basic ways can be used to perform synchronization between two nodes: *sender-receiver synchronization* and *receiver-receiver synchronization*. In sender-receiver synchronization a pair of nodes adjust their clock by exchanging a synchronization frame (e.g., NTP in [2]); in receiver-receiver synchronization two or more receivers adjust their clock by receiving a (broadcast) synchronization frame from a common reference node (e.g., RBS in [3]).

Authors in [4] proposes a global clock synchronization scheme that a time-stamped message is passed through a loop of nodes. The scheme can be applied to both flat topology and clustered topology. Research of [5] deals with synchronization problem in a way called *post-facto* sychronization. In post-facto scheme, the synchronization is proceeded after some nodes in the network have detected an event. Post-facto synchronization is more energy efficient comparing to the traditional ones.

Authors in [6] proposed a global synchronization protocol denoted as TPSN—Time-sync Protocol for Sensor Networks. In TPSN, first the network runs into a “level discovery phase” by a root node broadcasting *level-discovery* message. Once the level-discovery phase is done. The whole network is synchronized by sender-receiver pair-wise synchronization initiated by the root node. However, this scheme will encounter heavy collision of level-discovery messages when the node density increases, because level-discovery messages are broadcast.

On the other hand, due to the scalability problem of large scale sensor/ad-hoc networks, it is desired to organize large-scale networks into clustered architecture. A clustered ad-hoc/sensor network can greatly reduce the network overhead cost in routing and medium access control [7], [8]. In this paper a global synchronization scheme for clustered ad-hoc/sensor networks is proposed. The clustering protocol is a cluster-head-based scheme and cluster forming (cluster head election) is proceeded by beacon broadcasting periodically sent by cluster heads. The proposed synchronization is globally initiated by a default cluster head called *synchronizer* and spread out through the cluster heads in the network. Synchronization is periodically repeated in order to cope with node mobility and clock driftings.

The rest of the paper is organized as follows: Section II describes the global synchronization scheme. A brief introduction of random competetion clustering using link state is given in this section as well. Section III gives performance analysis of our scheme. Time accuracy of the scheme is also discussed. Section IV presents the settings and results of simulation. Section V concludes the paper.

II. SYNCHRONIZATION SCHEME

A. Link State Clustering Algorithm

The synchronization scheme proposed in this paper is based on the LSCA—*Link State Clustering Algorithm* proposed in [9]. LSCA can be briefly described as follows:

- **Random Competetion Clustering** When a network is deployed, each node differs a random time to compete for

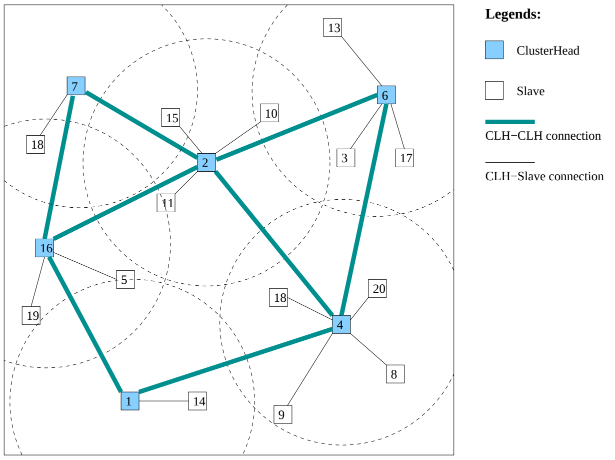

Fig. 1. Dynamic cluster forming scenario (The number on each node is the time sequence that the node's timer expires)

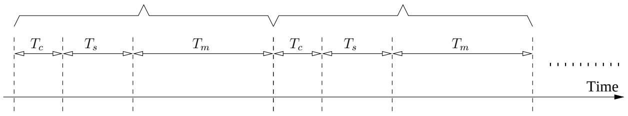

Fig. 2. Network timing of LSCA

cluster head (CLH) role. When the timer expires and the node has not received any Clustering Beacon (CB), the node marks itself as a CLH and broadcast a CB package immediately.

- **Link State** All the nodes that hear a CB will mark themselve as slave of the corresponding cluster head. The slaves keep the Link Qaulity Indicator (LQI[1]) of received CB frame. A slave node should join a cluster that CB from the CLH has best LQI. An cluster forming example is shown in Fig. 1.
- **Virtual Backbone** The CLHs form a virtual backbone for inter-cluster communications. To achieve this, each CLH uses two transmission power levels: P_L for intra-cluster communications and P_H for inter-cluster communications.
- **Phased Operation** The network runs periodically in rounds and each round alternatively consists of three phases 1) CLH election (C-phase), 2) Synchronization (S-phase), and 3) Network operation (O-phase), denoted as T_c, T_s, and T_m respectively, as shown in Fig. 2.

Since LSCA is based on random competetion, it clusterizes a network within $O(1)$ time, thus a fast and simple scheme. Link state monitoring results in a high stability for cluster maintenance [9].

B. Global Synchronization Scheme

Once the network is clustered, it runs into the synchronization phase. Our synchronization scheme is receiver-receiver-based. In order to synchronize the whole network, a *network synchronizer* (denoted as *S-node*) is introduced. The S-node is a CLH by default. To guarantee this, the differ timer of S-node in C-phase is minimized so that it will always win the contention.

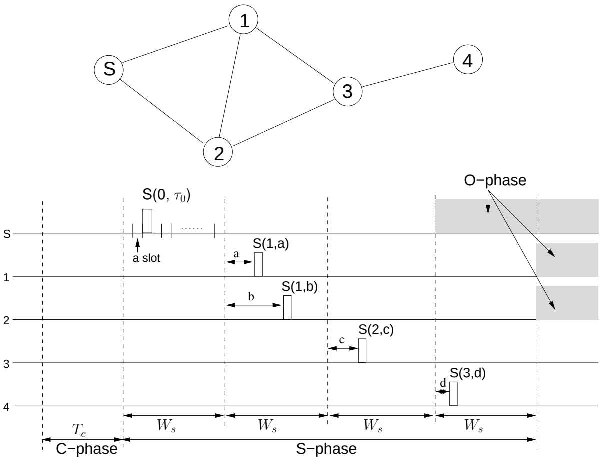

Fig. 3. Synchronization scheme for beacon-based clustering (only CLHs are shown in the figure)

After a C-phase, the S-node starts to broadcast a *Synchronization Beacon* (SB) using inter-cluster communication power P_H. A SB frame is marked as S(l, τ_d), where l is called *synchronization layer* (for the S-node $l = 0$) and τ_d is a random differ time as a number of time slots. We set $0 \leq \tau_d \leq W_s$, where W_s is the duration of *synchronization window* (SWIN). Each time slot interval, denoted as t_s, is set long enough so that the radio propagation time t_p is negligible, i.e., $t_p \ll t_s$.

The CLHs of the first synchronization layer that have received S(0, τ_d) will align their clock to start next SWIN at the same time as they know τ_d from the received SB. Again, each CLH differs the rebroadcasting of SB by a random time τ_d. The SB sent by the i-th layer CLH is marked as S$(i\%N+1, \tau_d)$, where N is a constant that prevents co-channel interference. Typically $N = 3$ or 4.

An example of this synchronization scheme is shown in Fig. 3. The idea of random delay of SB is similar to CSMA/CA, in which a random backoff is used to avoid collision. For example in Fig. 3, nodes 1 and 2 ($l = 1$) will likely broadcast SB at different time slots, thus collision of SB on node 3 is avoided. The idea that the CLHs having received a SB frame will be aligned to start next layer synchronization is similar to IEEE 802.11 DCF function [11], in which *network allocation vector* (NAV) is used to align the neighbouring nodes for the next channel contention.

A CLH may not be synchronized due to the collision of SB frames. In this case, the CLH will run in unsynchronized mode until it receives a SB in next round. However, the probability that a CLH runs in unsynchronized mode is very small, as analysed in Section III.

---

[1] LQI is a standard feature defined in IEEE 802.15.4[10].

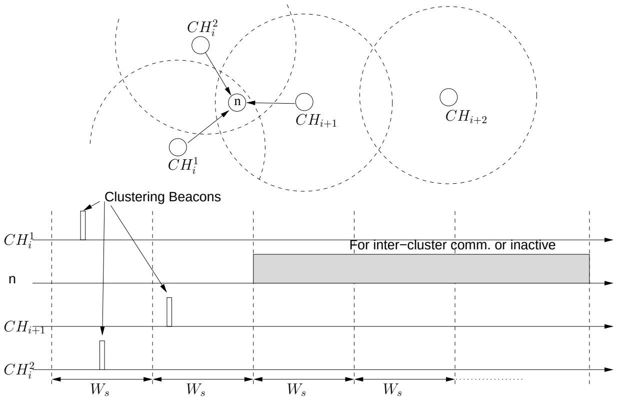

Fig. 4. The slave node n can receive CB from CLHs at same and/or different layers without collision.

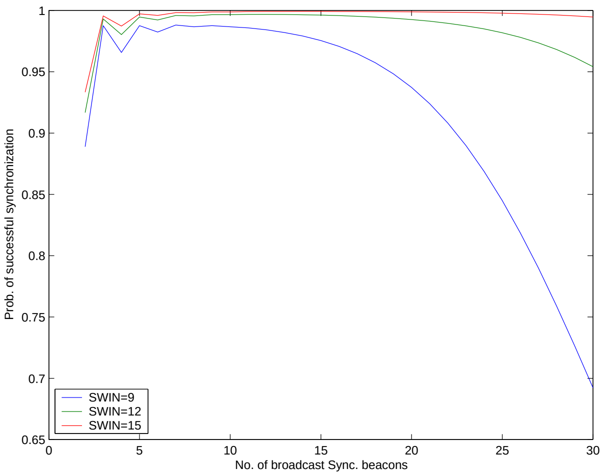

Fig. 5. The probability of receiving an SB without collision

C. Network operation

A sychronized CLH starts O-phase to maintain the cluster by sending CB frames after $N \times W_s$ time, as shown in Fig. 3. Between two consecutive CB frames the time is equally divided into a number of windows. Here we still use W_s as window duration so that the synchronized CLHs can coordinate with each other. CB frames sent by a synchronized CLH is now denoted as CB(i, τ_d), where i is the synchronization layer of the CLH and τ_d is the random delay in term of t_s. The random delay of CB is used to avoid collision on those slave nodes that can hear CB frames from different CLHs at the same synchronization layer.

A slave node that has received a CB$(i, *)$ should keep awaken in next W_s for the case that CB$(i+1, *)$ may be heard. Because the CLHs are synchronized in layers, CB$(i, *)$ should always come before CB$(i+1, *)$. This is illustrated as an example in Fig. 4.

O-phase duration is $M \times N \times W_s$, where M, denoted as *O-phase duration*, is an integer constant dependent on the nodes' clock precision and network topology change caused by mobility.

III. ANALYSIS

A. Synchronization Accuracy

In Section II-B it is mentioned that the signal propagation time t_p is negligible, i.e., $t_p \ll t_s$. If $t_s = 1ms$, which is close to the slot duration defined in IEEE 802.15.4 standard[2], and usually the radio range of sensor nodes is shorter than 200m, we have the radio propagation time $t_p = \frac{200}{3 \times 10^8} = 0.67\mu s$, which is much less than t_s. This synchronization scheme can make the network synchronized at 1ms level.

Nowadays for embedded sensors, the clock accuracy is achieved to 10^{-6} [12]. It means that the clock drifting of two sensor nodes is around $60\mu s$ after one minute. If in O-phase it is chosen $M = 100$ and $N = 4$, then synchronization will be re-proceeded after $M \times N \times W_s = 6.4$ seconds. The synchronization scheme in this paper keeps an accuracy of 1ms level, which is much greater than the clock drifting in 6.4 seconds.

B. Synchronization probability

The synchronization scheme proposed in this paper gives high probability to synchronize the whole network, because only CLHs are involved in synchronization phase. Furthermore, each CLH will have maximally 6 neighbouring CLHs (as the optimal case in cellular system). The synchronization phase is initiated from the S-node, therefore these neighbouring CLHs will belong to different synchronization layers on a planar area[3].

Suppose that there are n i-th layer CLHs neighbouring to an unsynchronized CLH. The probability that this node will be synchronized is: at least one i-th layer CLH sends SB in a time slot different from other $n-1$ CLHs, denoted as

$$\Pr(n, W_s) = \begin{cases} \dfrac{\sum\limits_{k=1}^{n} (-1)^{k+1} P_k^n C_k^{W_s} (W_s - k)^{n-k}}{W_s^n}, & W_s \geq n \\ \dfrac{\sum\limits_{k=1}^{W_s} (-1)^{k+1} P_k^{W_s} C_k^n (W_s - k)^{n-k}}{W_s^n}, & W_s < n \end{cases} \tag{1}$$

It can be seen that the probability a CLH to be synchronized is very high as $W_s \geq n$ holds. With $W_s = 16$ and $n = 6$, $\Pr(6, 16) = 0.9966$. Fig. 5 shows $\Pr(n, W_s)$ at different n and W_s.

IV. SIMULATION RESULTS

We set a simulation scenario of 200 nodes uniformly and randomly distributed in a square of 600×600 sq.meters. Mobility is considered and a random way-point model is deployed with a nonzero minimum speed. The total simulation time is 300 sec. and statistics are collected every 10 sec. We set

---

[2] In IEEE 802.15.4 15ms time is equally divided into 16 slots between two consecutive CBs. A beacon frame takes one slot.

[3] The case that all the neighbours belong to the same synchronization layer happens when the network is deployed on the surface of a sphere. The CLH on the opposite pole to the S-node will encounter such a case.

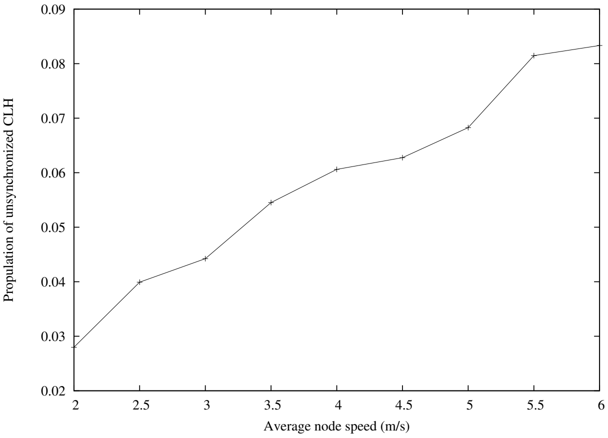

Fig. 6. Population of unsynchronized CLHs vs. mobility

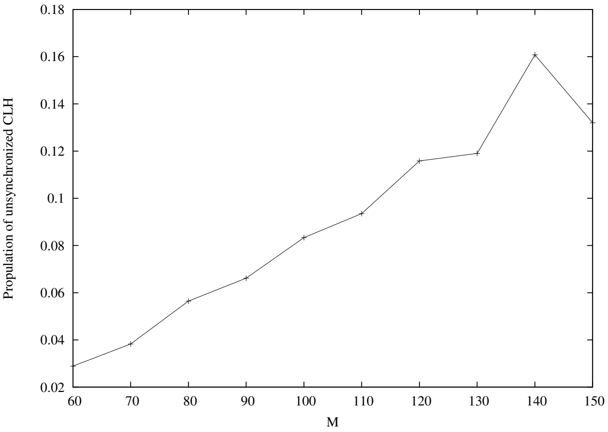

Fig. 7. Population of unsynchronized CLHs vs. M (average node speed: 6m/s)

t_s = 1ms, T_c = 1sec, $W_s = 16$, $N = 4$, and $M = 100$ as default. The default value of beacon range R_b = 100m and inter-cluster communication range R_c = 200m.

Since in LSCA, mobility of nodes will make a CLH (except the S-node) be canceled by hearing a CB from another CLH and join the second cluster as a slave, and a slave node may rise up to be a CLH when it doesn't hear CB for a certain period, the number of unsynchronized CLHs will increase as mobility increases. Fig. 6 shows the population of unsynchronized CLHs at different mobility settings. From the figure one can see that the chance that most CLHs are synchronized is rather high.

To cope with the increasing unsynchronized CLHs caused by mobility, a shorter O-phase duration is desired. Fig. 7 shows the population of unsynchronized CLHs as M increases. Fig. 8 shows the ratio of collided S-frames to the total number of sent S-frames. From the figure we can also assert that the main fact that causes CLHs unsynchronized is mobility, because the ratio of collided S-frames is independent of M and node speed.

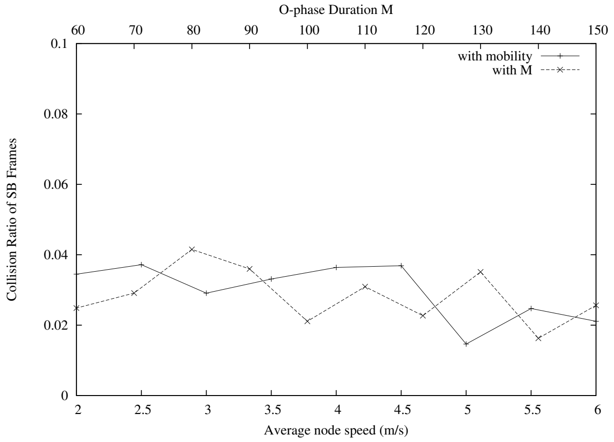

Fig. 8. SB collision ratio statistics at different M and mobility settings

V. CONCLUSION

In this paper we proposed a global time synchronization scheme for clustered wireless ad-hoc/sensor networks. The synchronization is proceeded right after the cluster-forming phase and continued by network operation phase. In order to cope with mobility of nodes and clock drifting, synchronization must be periodically repeated. Because only cluster heads are involved in the synchronization and a collision-avoidance mechanism is introduced when broadcasting synchronization frames, this scheme presents a high probability that the whole network is synchronized.

REFERENCES

[1] J. Elson and K. Römer, "Wireless sensor networks: A new regime for time synchronization," in *proc. of 1st workshop on hot topics in networks (HotNets-I)*, (Princeton, New Jersy), Oct 2002.

[2] D. L. Mills, "Internet time synchronization: The network time protocol," in *Global States and Time in Distributed Systems* (Z. Yang and T. Marsland, eds.), IEEE Computer Society Press, 1994.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating System Review*, vol. 36, pp. 147–163, 2002.

[4] Q. Li and D. Rus, "Global clock synchronization in sensor networks," in *proc. of IEEE INFOCOM'04*, 2004.

[5] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in *proc. of International Paralle and Distributed Processing Symposium (IPDPS)*, (San Francisco, USA), April 2001.

[6] S. Ganeriwal, R. Kumar, and M. Srivastava, "Time-sync protocol for sensor networks," in *in Proc. 1st International Conference of Embedded Networked Sensor Systems (SenSys)*, 2003.

[7] K. Xu and M. Gerla, "A heterogeneous routing protocol based on a new stable clustering scheme," in *MILCOM 2002, Proceedings*, vol. 2, pp. 838–843, Oct. 2002.

[8] Z. Cai, M. Lu, and X. Wang, "Channel access-based self-organized clustering in ad hoc networks," *Mobile Computing, IEEE Trans*, vol. 2, pp. 102–113, Apr.-Jun. 2003.

[9] C. Gao and R. Jäntti, "Link-state clustering based on ieee 802.15.4 mac for wireless ad-hoc/sensor networks," in *Proc. of IEEE Wireless Communication and Networking Conference (WCNC'06)*, (Las Vegas), April 2006.

[10] IEEE, "Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans)," tech. rep., IEEE802.15.4 WorkGroup, 2003.

[11] W. G. IEEE, "Ieee std 802.11b-1999: Wireless lan medium access control (mac) and physical layer (phy) specifications: Higher- speed physical layer extension in the 2.4 ghz band.," tech. rep., IEEE, 1999.

[12] K. Römer, “Time synchronization in ad hoc networks,” in *MobiHoc ’01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 173–182, ACM Press, 2001.