

Cluster-based Hierarchical Time Synchronization for Multi-hop Wireless Sensor Networks

Hyunhak Kim, Daeyoung Kim and Seong-eun Yoo
School of Engineering
Information and Communications University
119, Munjiro, Yuseong-gu, Daejeon, South Korea
{overflow, kimd, seyoo}@icu.ac.kr

Abstract

In this paper, we propose the cluster-based hierarchical time synchronization protocol(CHTS) for wireless sensor networks(WSN). To couple the network with real world tightly, we provide a network-wide time synchronization through the introduction of abstractions for cluster and hierarchy. Generally the error of time synchronization has the tendency of being accumulated as the hop count along the synchronization path increases in multi-hop wireless sensor network. Thus we cluster the randomly deployed nodes to decrease the hop count along the synchronization path. We then apply different optimal synchronization mechanisms to the cluster heads and cluster members for the sake of power efficiency through an explicit broadcast manner. We prove the performance enhancements with simulation comparisons.

1. Introduction

Advances in recent technologies such as MEMS, low cost wireless communications, micro sensors and embedded systems, increasingly realize the feasibility of WSN research. In the era of ubiquitous computing, the need for WSN is significantly rising.

As examples of real world adoption of WSN, there are many applications such as intrusion detection system, monitoring system for animal[1], volcano[2] and surveillance systems for military[3].

A WSN is typically made up of numerous sensors which do collaborative work over the network. Thus each node needs common norms to explain and express their own information. As a norm, time information is one of the most necessary infrastructures to any collaborative system. Maintaining accurate time information in each system not only provides meaningful information to the others, but also combines the system with the real world well.

Also time synchronization is a critical piece of infrastructure for sensor networks to suppress dupli-

cated detection of redundant messages from different systems and to reduce power consumption by precisely timed wakeups, overcoming the problem of battery capacity.

Internal oscillator has inherent error elements such as skew and offset, which make periodical synchronization necessary. For example, one of Berkeley MICA's internal oscillators has ± 50 p.p.m. error, which results in an offset of 0.1 ms/sec if we apply the theoretical worst-case relative offset between two nodes and [4] commented the experimental worst-case relative offset is about 4.75 μ s/s. Also, the error elements vary due to temperature, aging and so on.

We carefully focus on the tendency that the synchronization error through the synchronization procedure is typically accumulated as hop count increases [4][8]. Thus in order to reduce the error, we approach the problem by decreasing hop counts through the clustering of nodes. Then, the clusters synchronize with two different methods for inter-clusters and intra-cluster. Also we introduce a novel way to calculate synchronization period to meet the required synchronization error level by using clock drift value, deterministic delays and nondeterministic offset. Using the ns2, we evaluate the performance and effectiveness.

The remaining section is as follows; In section 2, we refer to the related works. In section 3, we illustrate the characteristics of the WSN and describe the system model and problems and then, we present the details of the proposed protocol in section 4. In section 5, we evaluate the performance of the proposed algorithm with comparisons. Finally in section 6 we conclude this paper and suggest future work.

2. Related works

RBS[5] follows 'Receiver-Receiver' model, which synchronizes a group of receivers. The advantage of RBS is to remove the uncertainty from 'sender' side with explicit broadcast manner and to estimate the clock drift of nodes with linear regression. The dis-

advantage of RBS is the requirement of additional message exchange for estimation.

TPSN[4] follows ‘Sender-Receiver’ model, a sender synchronizes a receiver. The advantage of TPSN is to remove access delay and byte alignment delay by timestamping at the low level, and the propagation delay by the pair wise packet exchange. The disadvantage is the overhead due to node-to-node synchronization mechanism for each link.

Tsync[6] uses the broadcast manner, however it has less overhead than RBS, though the number of neighbor node increases. The advantage of Tsync is to reduce overhead by using the broadcast nature explicitly and to synchronize the overhearing nodes implicitly without consideration of the propagation delay. The disadvantage of Tsync is to use multi-channel facility and weak synchronization.

There are no approaches which consider multi-hop. Thus in order to cope with the accumulating tendency of synchronization error in multi-hop networks, we propose an approach to reduce the error by decreasing the average hop count through the abstractions of cluster and hierarchy.

3. System model and problems

In our paper, we assume the WSN is deployed densely and randomly. The WSN is used for the fine-grain monitoring of phenomena which can not be observed nearby such as military activities[3], volcanic eruption[2] and wildlife ecology[1]. That implies that the deployment position can not be designated and the density of deployment for the target can be quite high[8].

Also, we assume the performance is strongly affected by collision due to the nature of wireless communication and by overhead due to the limited battery capacity as mentioned in RBS[7] and TPSN. Furthermore, the bandwidth of WSN transceiver is low such as 38.4kbps, 76.8kbps and 250kbps[10]. It can worsen the occurrence of collision in dense situations compared with high bandwidth communication.

We classify nodes into HPNs(High Performance oscillator equipped Node) and LPNs(Low Performance oscillator equipped Node). The percentage of HPNs which will be deployed is much smaller than the percentage of LPNs such as 5% or 10%.

The network is composed of cluster heads(CH), cluster members(CM) and a cluster head reference(CHR), which is the reference clock node of cluster heads. The CHs are chosen among the HPNs, which have a connection with CHR directly or indirectly through other CHs and make their radio range longer by changing their radio output signal strength[9][10].

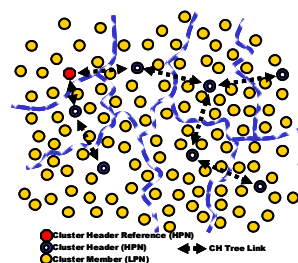


Figure 1. Synchronization model

As we mentioned, synchronization error shows a tendency to accumulate as hop count increases[4][8]. Thus, we explicitly use long radio range and clustering to decrease the average hop counts in the synchronization path. If we cluster all the nodes uniformly, the average hop counts will be remarkably reduced.

CHs are synchronized with CHR by pair-wise packet exchange and CMs are synchronized with CHs by using the overhearing manner of broadcasting. Since the number of CMs is much larger than the number of CHs, applying the overhearing manner from long range CHs to CMs gives a great benefit to the network in the aspects of lower collision and overhead. If we apply the implicit synchronization manner to CHs, we cannot assure that all the CHs are synchronized relative to explicit manner. Thus we apply the explicit manner to the main synchronization path between CHR and CHs and the implicit manner to the branches path between CHs and CMs.

4. Cluster-based Hierarchical Time Synchronization protocol(CHTS)

We first build the main synchronization path across the area between CHs as depicted in Figure 1. Then, the other nodes join as CMs hierarchically as shown in the Figure 1. After the synchronization path construction, the CHR starts to synchronize along the CHs' paths and each CH starts to synchronize its own CM. The synchronization is activated periodically according to the synchronization error requirement.

4.1. Cluster head tree construction

The CHR broadcasts ‘CH discovery’ packet which contains the broadcaster’s hop count. The receiving HPNs join the cluster head tree by setting their own hop count as the received hopcount+1, and re-broadcast the packet after replacing the hop count number with their own hop count number. Then they keep two information, the sender’s ID as parent ID and the power level of the received packet detected with mechanism such as RSSI[10].

During the CH tree construction, CHs check all the packets they receive and if they receive a packet with lower hop count number from the CHR or with a stronger power level but with the same hop count, they change their parent. When their own hop count has changed, they re-broadcast a discovery packet. The above mechanism decreases the overall CHs' hopcounts if the nodes are deployed uniformly.

Meanwhile some HPNs, which don't have any connections with CH, change themselves to LPN by lowering radio output signal.

4.2. Cluster member tree construction

There are three groups of CM. The first group(e.g., Figure 2. A, B and C) are the first children of each CH and the only nodes reachable to the CH directly. The second group(e.g., D, E, F, G and H) are the nodes within the CH's radio range but are not reachable to CH. The third group(e.g., I, J, K and L) are the nodes out of the CH's radio range but are the CH's CM.

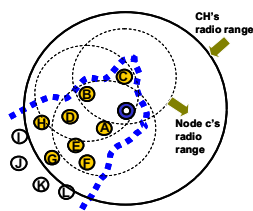


Figure 2. Composition of the cluster

When CHs set their own hop count during CH tree construction, all the CHs, including the CHR, set a timer to broadcast the '*1st CM discovery*' packet which contains their own ID. The receiving LPNs check all the packets during the constant time and choose the CH which sent the packet with the strongest power level as its CH before replying to the chosen CH. Some of nodes'(i.e., the second group) packets cannot be received by the chosen CH because the radio range of LPN is shorter than that of HPN. After the constant duration, CHs collect the replies and broadcast a list of the first group nodes in a packet as the acknowledgement, thus the first and second group nodes are determined.

After receiving the acknowledgment from the CH, the first group's CMs broadcast the '*2nd CM discovery*' packet which contains the broadcaster's hop count from CH, CH ID, the broadcaster's group type and parent ID. From now it is similar to the construction of CH tree, the receiving LPNs join the cluster member tree by setting their own hop count as the received hopcount+1, and re-broadcast the packet after replacing the hop count number with their own hop count number. Then, they keep the sender's ID as

the parent ID and the power level of the received packet.

During the above second CM tree construction, CMs check all the received packets and if they receive a packet with lower hop count from the CHs or with a stronger power level but with the same hop count, they change their parent. When their own hop count has changed, they re-broadcast a discovery packet. The above mechanism decreases the overall CMs' hop counts if the nodes are deployed uniformly. Two differences between CM tree and CH tree construction are that the second group does not change their parent by the power strength comparison for the sake of the overhearing synchronization and all the nodes check whether they have the third group's nodes as their children for the final synchronization procedure.

If any node excluding the CHR does not synchronize during the constant duration, they broadcast the packet '*orphan request to join*', then, the parent re-confirms or a new parent reforms the linkage.

4.3. Time synchronization

At the startup of the CH tree construction, the CHR sets a timer to start the synchronization of CHs by broadcasting the '*CH synch start*' packet. The receiving CHs back-off a random time and broadcast a timestamped packet(i.e., T1) to the CHR and the CHR replies by inserting two timestamps(i.e., T2, T3) to the packet. Following this, the requested node timestamps (i.e., T4) on reception of the "reply" packet and calculate its own offset from the T1,T2,T3 and T4 by the following Eq.(1), correcting its own clock.

$$offset = \{(T2 - T1) - (T4 - T3)\} / 2 \quad (1)$$

Lower level nodes(i.e., toward the terminal node) can overhear its parent CH's broadcast of T1. When this occurs, they back-off some predefined amount of time plus a random time and then broadcast a timestamped packet(i.e., T1) to its parent CH. From now on, this same procedure, which was already done between this node's parent CH and its grandparent, will be repeated until the terminal CH node.

After CH synchronization, each CH backs-off a constant duration and broadcasts a timestamped packet(i.e., T1) to a chosen node on the first group, which replies by adding two timestamps(i.e., T2, T3) to the packet. Following, the CH can timestamp(i.e., T4) on reception of the "reply" packet and calculates the chosen node's offset from the (-1)*Eq.(1) and broadcasts its offset and T2. The chosen node can then correct its own clock from the delivered offset. Moreover, all other first and second group CM in the same cluster can overhear the mentioned values and

can calculate their own offset by the following Eq.(2) [6] and correct their own clock.

$$\text{offset} = \text{offset}_{\text{chosen}} + T2_{\text{chosen}} - T2_{\text{own}} \quad (2)$$

The nodes on the first or second group, which have the third group's children, perform the same procedures as the CH's synchronization by broadcasting a '3rd CM synch start' packet. Further procedures are the same as the described CH's synchronization.

When we maintain a system with a required maximum error level of time synchronization, we can apply theoretical worst-case drift and periodical synchronization to meet the required level as follows.

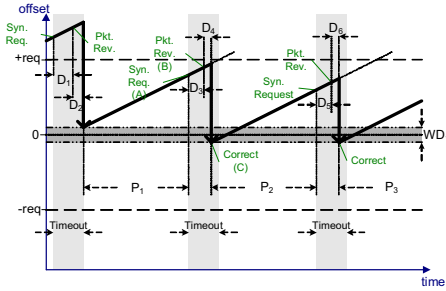


Figure 3. Synchronization error and period

$$P = \frac{|req - WD1|}{2} - \text{Timeout} \quad (3)$$

Table 1. The element of the Figure 3.

- A : Timer P expire, request the sync. to the upper node.
- D1, D3, D5 : Delay of roundtrip packet between two nodes.
- B : When sender receives the response packet.
- D2, D4, D6 : Delay for calculating offset and correcting local clock.
- C : On completion of the correction of the local clock.
- P1, P2, P3 : After C, re-sync. interval should be calculated in order to meet the sync. error requirement not to exceed the line, 'req' or '-req'.
- G : Constant gradient, the worst-case offset from the oscillator manual.
- WD1 : Nondeterministic offset during the sync. procedures.
- Timeout : Time-out duration to re-request sync.

5. Performance evaluation

The performance is evaluated using ns 2.28 simulator. Since TPSN[4] is the same 'Sender-Receiver' model and hierarchical structure explicitly and it does not use linear regression estimation neither, we select TPSN[4] for the performance comparison. For the fair comparison, we use the same oscillator model for all nodes in both schemes, TPSN and CHTS.

5.1. Simulation setup

We use the two-ray ground propagation model, omni antenna model and IEEE 802.11 MAC, and modify some parts of the ns2 simulator such as radio signal power output on the operation, timestamping at MAC layer, small random jittering at application layer,

and clock modeling using normal distribution with 99% confidence interval.

We use Mica2[11] node hardware platform specification and place the nodes on a 50mx50m grid in a uniformly random fashion. The radio range of the nodes, LPN is 17.9m and HPN is 31.8m, determined from the ns2 propagation model by the parameters of 0.2m height on the ground and Tx power output of 0dBm(30.4e-3W) for LPN, 10dBm(80.1e-3W) for HPN. Rx consumes 22.2e-3W and we use the maximum data rate 76.8kbps[10][11]. For each scenario, we generate fifty random cases of all random variables such as node position, clock drift, jittering and random back-off, and apply these set equally to each algorithm during 500 seconds simulation time.

5.2. Simulation results

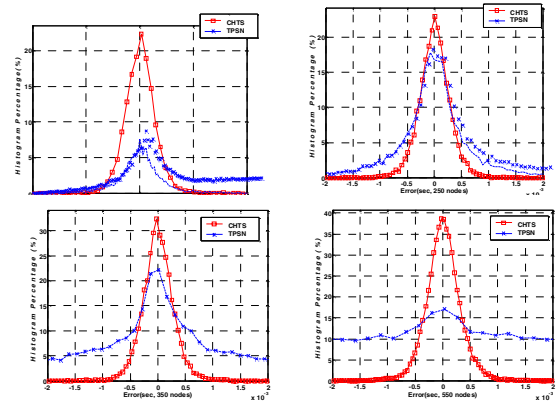


Figure 4. Distribution of sync. error

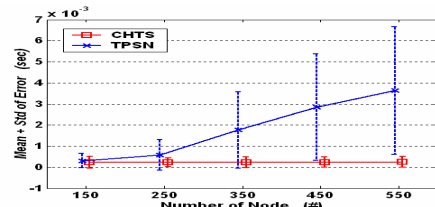


Figure 5. Distribution bar of sync. error

Table 2. The number of collision in Figure 4, 5

Nodes#	150	250	350	450	550
CHTS	0.81	4.76	14.38	28.62	49.38
TPSN	1.79	6.37	13.70	23.55	37.13

(unit : thousand)

Figure 4 and 5 show the error under the different density of nodes with fixed 10% of HPN. We can observe that the distribution of the synchronization error is significantly affected by the network density. CHTS shows better accuracy and deviation. From the third scenario(i.e., 350, 450 and 550), TPSN suffers from the density due to the collisions, but CHTS does not suffer with more collisions as shown in Table 2.

Figure 6 shows the error under different synchronization periods. Due to the collisions, TPSN shows

remarkably lower performance than that of CHTS at 10 sec and 20 sec period.

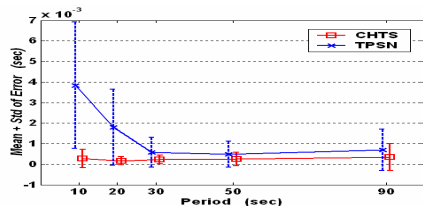


Figure 6. Mean and std. dev. of sync. error

Table 3. The number of collision in Figure 6

Period(sec)	10	20	30	50	90
CHTS	8.72	5.12	4.76	4.90	4.44
TPSN	11.43	7.17	6.38	5.54	4.87

(unit : thousand)

Figure 7 shows the error under the different ratios of HPN. Note that in TPSN there are no HPNs and we depicted the data of it just for comparison. To deploy too many HPN degrades the performance, since the long range causes collisions. A HPN ratio of 4% is a desirable choice as it gives a good performance while reducing the cost because it uses a more expensive oscillator.

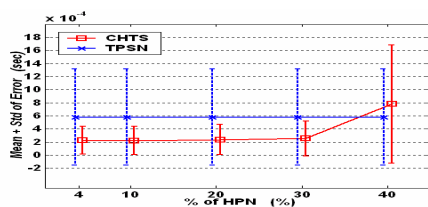


Figure 7. Mean and std. dev. of sync. error

Table 4. The number of collision in Figure 7

HPN %	4	10	20	30	40
CHTS	2.82	4.76	10.05	18.43	26.55
TPSN	6.38	6.38	6.38	6.38	6.38

(unit : thousand)

Figure 8 shows the observations of synchronization errors and the remaining energy while varying the hop count (250 nodes, 10% HPN and 50 sec period). As we mentioned, the error tends to accumulate as the hop count increases. Note the remaining energy of CHTS is much larger than that of TPSN.

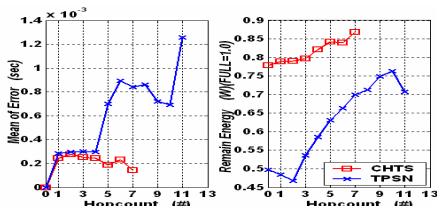


Figure 8. Mean of error and remained energy

6. Conclusion

WSN is generally constructed as multi-hop, and synchronization error has a tendency to accumulate as the average hop count increases[4][8]. Thus we intro-

duced the hierarchical clustering protocol to decrease the synchronization error by reducing the average hop count from the reference node to each node. Then, we applied the different synchronization methods to each hierarchy. Also we presented an equation calculating synchronization period to meet the required synchronization error based on the characteristic of oscillator and the required synchronization error. Finally, we proved the proposed algorithm performed better through simulation. We plan to add a dynamic cluster head election algorithm in order to cope with the topology changes for future work.

References

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", *The First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, Georgia, September 2002.
- [2] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring Volcanic Eruptions with a Wireless Sensor Network", *In Second European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, January 2005.
- [3] J. Nemeroff, L. Garcia, D. Hampel, S. DiPierro, "Application of sensor network communications", *IEEE Military Communications Conference 2001, vol. 1*, October 2001, pp. 336-341.
- [4] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-Sync Protocol for Sensor Networks", *The First Int. Conf. on Embedded Networked Sensor Systems*, Los Angeles, California, November 2003.
- [5] J. Elson, L. Girod and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts", May 17, 2002, University of California, Los Angeles.
- [6] H. Dai, R. Han, "TSync : A Lightweight Bidirectional Time Synchronization Service for Wireless Sensor Networks", *ACM SIGMOBILE Mobile Computing and Communications Review, Special Issue on Wireless PAN & Sensor Networks, vol. 8, no. 1*, University of Colorado, Boulder, January 2004, pp. 125-139.
- [7] R. Karp, J. Elson, D. Estrin, and S. Shenker, "Optimal and Global Time Synchronization in Sensornets", *Technical Report CENS 0012, Center for Embedded Networked Sensing*, University of California, Los Angeles, April 2003.
- [8] A. Hu, S.D. Servetto, "Asymptotically optimal time synchronization in dense sensor networks", *International Workshop on Wireless Sensor Networks and Applications*, San Diego, California, September 2003, pp. 1-10.
- [9] G. Anastasi, A. Falchi, A. Passarella, M. Conti, E. Gregori, "Performance measurements of motes sensor networks", *ACM 7th International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, University of Pisa, 2004, pp. 174-181.
- [10] Chipcon RF-IC company, <http://www.chipcon.com>.
- [11] Crossbow Technology, <http://www.xbow.com>.