

State-Based Synchronization Protocol in Sensor Networks

Shang-Chih Hsu Wei Yen¹

¹Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan, ROC
shanzihsu@yahoo.com.tw, wyen@ttu.edu.tw

Abstract

Some applications of the sensor network require precise time synchronization among local clocks of the sensors. Due to the limited resource in the sensor network, it is not trivial to maintain the said synchronization. In this paper, we propose a novel method, referred to as the state-based synchronization protocol (SBSP) to solve this issue. We create a finite state machine in the SBSP to describe the synchronization actions at each sensor node. Specifically, the SBSP makes use of the ADV, REQ and SYNC packets and the finite state machine to achieve synchronization. We will compare the performance of the proposed protocol against the popular timing-sync protocol for sensor networks (TPSN) and lightweight tree-based synchronization (LTS) through simulations. The SBSP reduces the redundant message exchanges in the TPSN. It is shown that the SBSP saves power with less message exchanges while maintaining better synchronization error.

Keywords: Drift, Sensor Networks, Synchronization Period, Time Synchronization

1. Introduction

Recently sensor networks have found applications in many areas such as battle field surveillance, healthcare, environmental observation and applications for real time tracking. Yet, it is still not easy to construct or operate a sensor network for various applications. For example, energy conservation, time synchronization, and efficient routing, all impose great challenges to the sensor network [1]. Conceptually, the sensor network is a self-organizing and cooperative network. Hundreds of thousands of sensors may be deployed in a sensing field. Each sensor exhibits limited computing and transmitting capability and cooperate with others to carry out requested tasks.

Figure 1 shows the architecture of a typical sensor network. Here every sensor node is responsible of measuring the metric of interest from its proximity.

The coverage of the sensor node is the circular area within which the metric can be collected by the sensor. In general, each sensor has its own local clock with limited accuracy. Without proper synchronization, the sensors will definitely be out of synchronization in the long run. The sink is a focal point in the sensor network. It is the bridge between the sensor network and the external networks. The user can retrieve data from the sensor network or perform other tasks by giving instructions to the sink. It is usually assumed that the sink is more powerful than the average sensor node. In order for two sensors to communicate with each other, it is evident that they must fall into each other's communication range. In Figure 1, we show that the sensor nodes *A*, *B* and *C* are cooperative with one another and relay the data they collected to the sink [1].

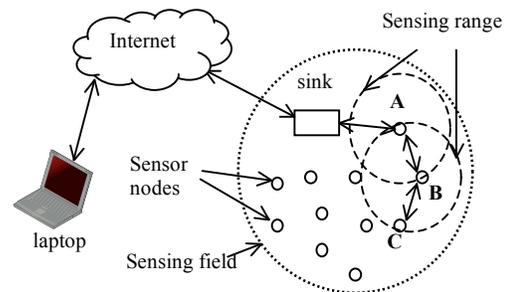


Figure 1: Sensor Network Architecture

2. Existing Works

2.1. Timing-sync Protocol for Sensor Networks

Ganeriwal *et al.* [3] proposed a time synchronization protocol, which they called the timing-sync protocol for sensor networks (TPSN). The TPSN can be divided into two phases. The initial phase is known as the level discovery phase (LDP) while the second one is called the synchronization phase (SP). The LDP starts at the root node that may be any sensor node or the sink. The root node is assigned as level 0 and

broadcast a *level discovery* packet to its neighbor nodes. This packet contains the identity and the level number of its sender. When the neighbor nodes receive this packet, they will increase the level number by 1 and assign it as their own level number. Each neighbor node, then, forwards the packets with its own ID and level number. However, a sensor ignores the *level_discover* packet whose level number is smaller than that of the sensor. The LDP continues until all nodes in this sensing field obtain their respective level numbers. It will then terminate automatically because the *level_discover* packets broadcast at the end of the LDP will be ignored by all nodes in the sensor network.

After the LDP, the synchronization phase (SP) will begin at the root node. In the LDP, a hierarchical tree structure is established. Hence, the classical approach of sender-receiver synchronization can be readily applied to perform handshakes between nodes and their children nodes in the tree. Next, we describe how to use the two-way exchange to achieve time synchronization in the SP [3, 4].

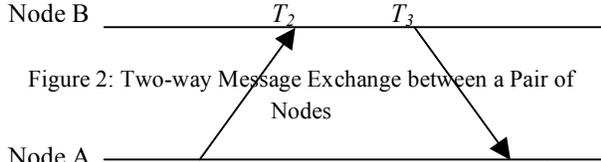


Figure 2: Two-way Message Exchange between a Pair of Nodes

Figure 2 shows the time sequence chart of the two-way exchange between nodes A and B . T_1 and T_4 are the measured by the local clock of A while T_2 and T_3 are measured by that of B . At the time T_1 , node A sends a *synchronization_pulse* packet to node B . This *synchronization_pulse* packet contains two pieces of information, (i) the level number of A and (ii) the value of T_1 . At the time T_2 , node B receives the packet and records the receiving time locally. T_2 , of the clock of node B , is equal to $T_1 + \Delta + d$, where Δ and d represent the clock drift between the pair of nodes and the propagation delay, respectively. It should be noted that Δ may not be time invariant. However, it is considered constant for the short period of the two-way exchange process. At the time T_3 , node B finishes processing the *synchronization_pulse* packet and sends back an *acknowledgement* packet to node A . This *acknowledgement* packet contains the level number of B and the values of T_1 , T_2 , and T_3 . At the time T_4 , node A receives the packet. Similarly, T_4 , of the clock of node A , is equal to $T_3 - \Delta + d$. Finally, these

parameters are used by node A to calculate Δ and d in the following equations.

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad \text{Eq (1)}$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \quad \text{Eq (2)}$$

2.2. Lightweight Time Synchronization for Sensor Networks

Greunen and Rabaey [4] proposed another scheme for time synchronization, called lightweight tree-based synchronization (LTS). The LTS has two main variations. One is known as the centralized multi-hop LTS (CLTS) and the other the distributed multi-hop LTS (DLTS). Next, we give brief introduction to the CLTS and DLTS.

In the CLTS, it works in two phases. First, it relies on the tree structure for synchronization as well. It makes use of the spanning tree algorithm which is called distributed depth-first search (DDFS) [2]. However, it does not use the level number for ordered synchronization. Another phase, the CLTS operates like the SP of the TPSN. The CLTS invokes the two-way exchange handshake between the parent and children nodes.

In the DLTS, it focuses on the resynchronization issue. Figure 3 gives a snapshot of the operations of the DLTS. When a particular node j needs resynchronization, it will send a request to the closest reference node. Along the routing path from the reference node to the node j , each node on this path cooperates with each other to relay the resynchronization request message.

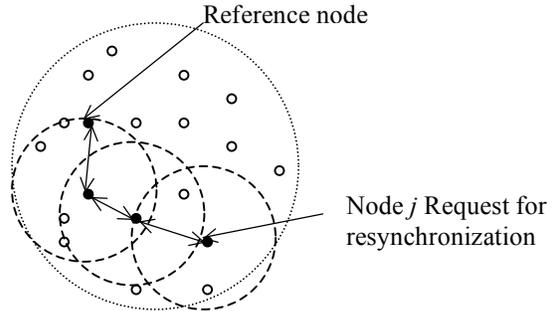


Figure 3: Distributed multi-hop Synchronization

3. State-Based Synchronization Protocol (SBSP)

In general, two smaller sub-problems are associated with the synchronization issue in the sensor

network. One is to design an efficient synchronization process [3, 4, 5, 6] while the other is to optimize the resynchronization rate [4]. In this paper, we will concentrate on the first problem and propose the state-based synchronization protocol (SBSB). Next, we introduce the three messages used in the proposed protocol.

Our protocol uses three types of messages to synchronize the network. They are briefly described in the following.

- **ADV** (advertisement) : This message sent by the initiator urges the neighbor nodes to check their synchronization states.
- **REQ** (request) : Upon receiving the ADV packet, if a node is in the unsynchronized state, it will request synchronization to the initiator.
- **SYNC** (synchronization) : When the initiator receives the REQ packet, it will calculate the delay time d and clock drift, Δ , between the initiator and its neighbor. The computed result is, then, sent in the SYNC message to instruct clock adjustment in the neighbor node.

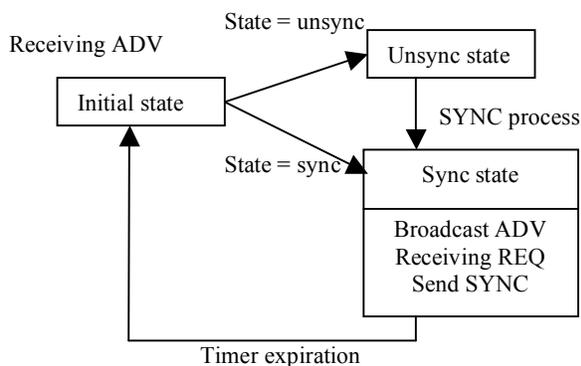


Figure 4: State Diagram

The SBSB works as follow. In the beginning, all nodes in the network do not know their state except the root node. The root node starts synchronization by broadcasting an ADV packet to its neighbor nodes. As Figure 4 shows, when a neighbor node receives the ADV packet, the node determines its state. When the node finds that it is in the synchronized state, then it ignores the ADV packet and only processes REQ packet in the future. If the node is unsynchronized, it will initiate the SYNC process with the root. Specifically, it will send a REQ packet to the root node. Note that when each node sends back REQ packet to the root node, it will wait for some random time to avoid packets collision. When receiving the REQ packet, the root node (or reference nodes) will calculate the propagation delay time and drift clock as Eq (1) and (2) definition. When the root node obtains the result, it will send it in the SYNC packet including the propagation delay and clock drift to the neighbor

node for synchronizing. After receiving the SYNC packet, the neighbor node will adjust its clock to achieve time synchronization. Once the handshake process is completed, the neighbor node becomes the reference node and acts as the role of the root node. The process of SBSB will repeat above operations until all nodes in the sensing field are synchronized.

4. Performance Evaluation

In this section, we compare the performance of the TPSN, CLTS, and SBSB through simulations. In particular, we examine three performance metrics, i.e., time efficiency, the number of message exchanges and synchronization error.

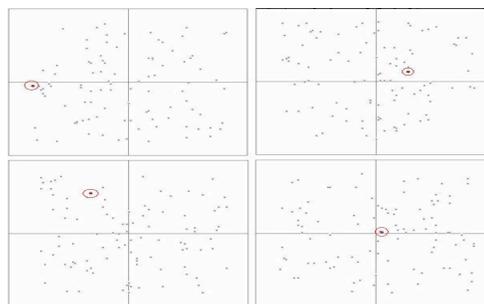


Figure 5: Sensor Nodes Configuration

Figure 5 shows the node distribution in four of the experiments we conducted. The red dots with circle are the root nodes. We build a 600*600 square plane and scatter 100, 300, 500, 700 and 1000 sensor nodes in the plane uniformly.

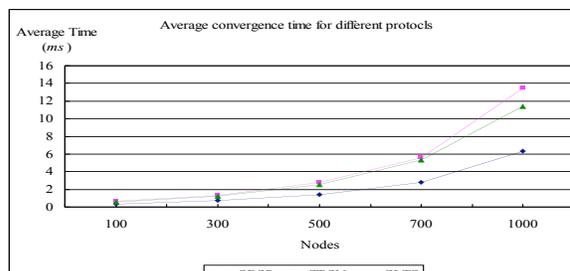


Figure 6: Average Convergence Time for Different Protocols

Figure 6 shows the convergence time that each protocol can achieve with various numbers of sensor nodes. According to Figure 6, the performance of the TPSN is the worst. And, the CLTS is slightly better than the TPSN. As we pointed out previously, both the TPSN and CLTS have two phases for synchronization. However, the spanning tree algorithm establishes the structure that is more efficient for its synchronization phase. Therefore, the

whole synchronization process of the CLTS is better than the TPSN.

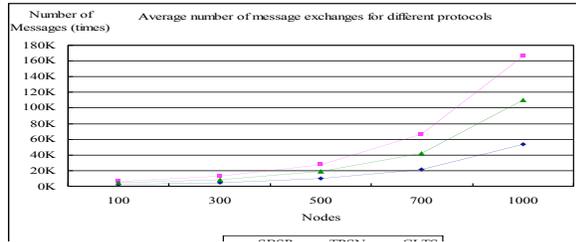


Figure 7: Average Number of Message Exchanges for Different Protocols

Figure 7 shows the simulation result for the total number of message exchanges in various synchronization protocols. The average number of message exchanges is an important performance indicator because it closely related to the power consumption demand by the synchronization. Because the TPSN may generate the duplicated packets in some occasions, the number of required message exchanges is higher. Although the CLTS has two phases to construct its synchronization process as well, it does not need the same amount of packets as the TPSN does. Because the spanning tree algorithm used by the CLTS can construct a better hierarchical tree, it will result in less message exchanges when compared with the TPSN. The SBSP makes use of the three types of packets for synchronization and reduces the duplicated packets. It is expected to save about 67% and 51% of the average amount of the message exchanges when compared with the TPSN and the CLTS protocols, respectively.

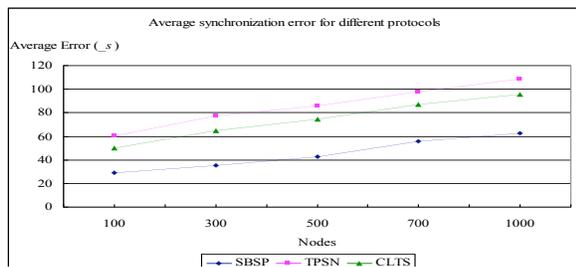


Figure 8: Average Synchronization Error for Different Protocols

Figure 8 demonstrates the synchronization error of the compared protocols. The TPSN is again the worst performance with the maximum error. It is close to 110 μ s. It is realized that the synchronization error is related to message exchanges issue. With the given data, we conclude that the SBSP is most accurate synchronization protocol among those examined. In summary, the SBSP is the best protocol in three metrics we investigate. Therefore, we believe

the SBSP is the best among the three protocols discussed in this paper.

5. Conclusions and Future Works

Time synchronization is an important issue in sensor networks. In this paper, we have introduced the state-based synchronization protocol (SBSP). It is based on the negotiation mechanism and the finite state machine to achieve synchronization in the sensor network. Through simulations, we show the SBSP is the best when compared with the existing TPSN and CLTS.

The resynchronization rate is another critical issue when network synchronization is concerned. We suspect the optimal resynchronization rate may depend on the synchronization error. A possible research direction is to identify this relationship. Furthermore, we also find it is interesting to consider synchronization in a hostile environment where there are malicious nodes try to sabotage the synchronization effort.

6. References

- [1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A Survey on Sensor Networks," *IEEE Communication Magazine*, Vol.40, No.8, p 102-116, August, 2002.
- [2] B. Awerbuch, "A New distributed depth first search algorithm," *Inf. Proc. Lett.* 20 (1985), 147-150.
- [3] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava, "Timing-sync Protocol for Sensor Networks," *Proceedings of the 1st international conference on Embedded networked sensor systems*, November 5-7, ACM 2003, Los Angeles, California, USA.
- [4] Jana van Greunen and Jan Rabaey, "Lightweight Time Synchronization for Sensor Networks," *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, September 19, 2003, San Diego, California, USA.
- [5] Wendi Rabiner Heinzelman, Joanna Kulik and Hari Balakrishnan, "Adaptive Protocol for Information Dissemination in Wireless Sensor Networks," *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOB/COM*, 1998, p 174-185.
- [6] Yao-Win Hong and Anna Scaglione, "A Scalable Synchronization Protocol for Large Scale Sensor Networks and Its Applications," *IEEE Journal on Selected Areas in*

Communications, v 23,n 5, May, 2005, p 1085-1099.