Brief Reviews: Time Synchronization Protocols in Wireless Sensor Network—Centralized Versus Distributed

Yeong Chin Koo and Muhammad Nasiruddin Mahyuddin

Abstract Rapid developments in the areas of sensor design, information technologies, and wireless networks have caused the wireless sensor network (WSN) emerges as a key role in many fields of science and industry. Time synchronization is essential in WSN to ensure all nodes in the WSN are synchronized and coordinated. In this paper, a brief introduction of WSN is given. Then, this paper reviews the widely discussed time synchronization protocols developed in the past and compares the protocols in two group, centralized and distributed.

1 Introduction

Recently, there has been a growing interest in wireless sensor networks (WSNs) [1]. A WSN consists of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it. The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disable relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities. Another unique feature of sensor networks is the cooperative effort of sensor nodes.

Every sensor node can perform tasks of sensing and computation. However, each sensor node is resource constrained; it operates under a short communication range, has limited memory and computational power. These sensor nodes form an ad hoc wireless network for communication. Although each node has limited capabilities, they can perform various tasks as a group, at a level which is compatible to a small number of high-end sensor nodes. Applications of WSN include,

Y.C. Koo (✉) · M.N. Mahyuddin (✉)
School of Electrical and Electronics Engineering, Universiti Sains Malaysia,
14300 Nibong Tebal, Pulau Pinang, Malaysia
e-mail: kooyeongchin@yahoo.com

M.N. Mahyuddin
e-mail: nasiruddin@usm.my

H. Ibrahim et al. (eds.), *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*, Lecture Notes in Electrical Engineering 398,
DOI 10.1007/978-981-10-1721-6_47

but not limited to, military application, environment monitoring, health monitoring, building comfort control, smart home, traffic control, manufacturing and plant automation, and surveillance systems [2].

To transfer data among the WSN, it requires gateways with application specific knowledge such as ZigBee, Z-Wave, INSTEON, Wavenis, Internet Protocol (IP) and etc. Through the Internet Protocol, native connectivity between WSN and the Internet is achieved and this is also known as Internet of Things (IoT).

The rest of the paper is organized as follows. Section 2 presents challenges regards to the WSN. Section 3 discusses the existing protocols that popularly discussed in time synchronization of WSN. Last but not least, Sect. 4 presents the authors' conclusions.

2 Challenges of WSN

The challenges faced in building a WSN are vary depending on the system requirement. The issues most systems will need to handle are as follow [3]: heterogeneity, transparency, fault tolerance and failure management, scalability, concurrency, openness and extensibility, migration and load balancing, security and etc.

Although sensor networks share many similarities with other distributed systems, they are subject to a variety of unique challenges and constraints [4]: energy efficiency, self-management, decentralized management, design constraints, privacy and security, connectivity, time synchronization, localization, and wireless reprogramming.

Among the challenges mentioned, time synchronization should be paid the most attention. In distributed systems, every nodes have their own clocks and their own notion of time. Since each node in a sensor network operates independently and relies on its own clock, the clock readings of different sensor nodes will also differ [4]. Through time (or clock) synchronization, a common time reference is provided to the local nodes in WSN to ensure that sensing times can be compared in an orderly and meaningful way [5]. The common time reference is essential as temporal coordination is required for tasks executed by the WSN. Although many time synchronization techniques for wired networks have been developed throughout the years, these techniques are unsuitable for WSN because of the unique challenges posed by wireless sensing environments.

3 Previous Works

Traditionally, Network Time Protocol (NTP) [6] is a popular protocol used for time synchronization. NTP can synchronize effectively the computer on the internet, however it is computationally intensive and requires a lot of energy. Thus, it is not suitable for use in sensor network [5]. Therefore, a new kind of methodology/protocol has been proposed through the years to achieve time synchronization in WSN which particularly fulfill the requirements of WSN. These time synchronization protocols can be divided into two categories: centralized synchronization and distributed synchronization.

3.1 Centralized Time Synchronization

Reference Broadcast Synchronization (RBS) [7] provides synchronization amongst a set of client nodes located within the single-hop broadcast range of a beacon node (Fig. 1a). RBS used least square linear regression to compensate for the clock drift. The drawback of RBS is this protocol is vulnerable to node failure and node mobility by using the beacon node.

Timing-sync Protocol for Sensor Networks (TPSN) [8] is the flexible extension of NTP. TPSN consists of two phases: level discovery phase and synchronization phase. The level discovery phase arranges the whole network into a hierarchical tree topology with the master node at its root (Fig. 1b). The synchronization phase is achieved by pair wise synchronization along the branches of the hierarchical tree structure through the classical sender-receiver synchronization handshake exchange. However, TPSN has two limitations: (1) the root node and nodes that responsible for synchronization may fail often, which may cause substantial overhead to the code and desynchronization, (2) geographically near nodes might have significant differences in clock readings as they might be far in terms of the tree distance (nodes J and K in Fig. 1b).

Flooding Time Synchronization Protocol (FTSP) [9] achieves high accuracy by utilizing timestamping of radio messages in low layers of radio stack. Compensate clock skews is done via linear regression. A leader or root is elected through

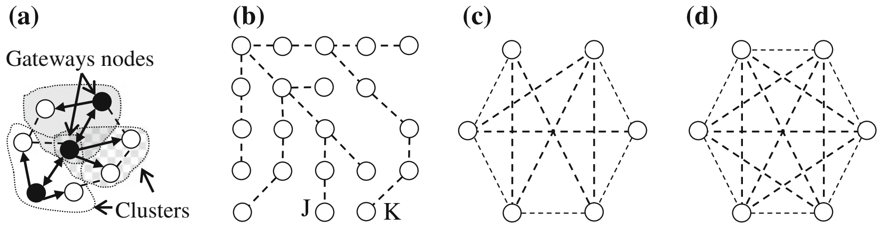

Fig. 1 Network topology. **a** Cluster topology, **b** tree topology, **c** partially connected mesh topology, **d** fully connected mesh topology

message exchanges, and the global time is passed from the root to all nodes through flooding. Although this protocol had better performance and robustness against node and link failures, it still does not completely solve the limitations mentioned before in TPSN.

3.2 Distributed Time Synchronization

In Time Diffusion Protocol (TDP) [10], the nodes periodically self-determine to become master/diffusion leader nodes using an election/re-election procedure. Then, neighboring nodes are engaged by master nodes in a Peer Evaluation Procedure to isolate problem nodes. Timing information messages are broadcast from master nodes and then re-broadcast by the diffused-leader nodes for a fixed number of hops, forming a radial tree structure. TDP will increase the synchronization frequency if small tolerance time is needed.

Reachback Firefly Algorithm (RFA) [11] is inspired by firefly synchronization mechanism. Every node periodically broadcasts a synchronization message, and if they hear a message, they will advance of a small quantity of their internal clock that schedules the periodic message broadcasting. Eventually, all nodes will advance their phase until they are all synchronized. RFA can operate at lower frequency to have higher accuracy. However, this algorithm does not compensate for clock drift.

In Global Clock Synchronization (GCS) [12], nodes take turns to broadcast a synchronization request to their neighbors who each respond with a message containing their local time. The receiving node that located at the center of this exchanges averages the received timestamp and broadcasts the averaged value back to its neighbors which adopt this value as their new time. This process is repeated until network wide synchronization is achieved. The time synchronization period will be decreased if higher time accuracy is needed.

In Distributed Time Synchronization Protocol (DTSP) [13], neighboring nodes exchange timestamp packets on a one-to-one basis. The receiving node then estimates the best-fit offset line between timestamp packets using a recursive least squares estimate.

Average Time Synchronization (ATS) [14] uses a cascade of two consensus algorithms to tune compensation parameters and converge nodes to a steady state virtual clock. Nodes broadcast local timestamps in the first stage in order to estimate the clock skew rates compare to each other. Then, the nodes broadcasts their current estimate of the virtual clock skew rate and receiving nodes combine this with their relative skew estimates to adjust their own virtual clock estimates. These two stages are repeated again to remove the offset errors.

Consensus Clock Synchronization (CCS) [15] is like an enhancement of ATS, it utilizes average consensus algorithm to compensate the clock offset. Nodes can learn how much their own clocks drift away from the consensus time through the accumulated offset error removed during each round of offset compensation. Then, the information is used to compensate the clock drift.

Time Synchronization using Max and Average consensus protocol (TSMA) [16] is similar to ATS and CCS but it uses max consensus method to compensate the clock drift. To compensate clock offset, TSMA uses average consensus algorithm, like ATS and CCS. The main idea is to achieve global synchronization by only using local information.

Time Synchronization using Distributed Observer algorithm with Sliding mode control element (TSDOS) [17] is an extended algorithm to enhance ATS by using the sliding-mode control term. In WSN, every nodes try to estimate the relative skew rate with respect to their neighboring node. By introduces a sliding-mode like term at the relative skew rate, the relative skew rate can be enhanced.

3.3 Comparisons Between Centralized and Distributed Protocol

Table 1 displays the comparisons in general for the protocols in the two categories.

Table 2 compares the protocols in 6 categories: centralized/distributed, clock skew compensation, MAC timestamp consideration, network communication topology, convergence speed, and average accuracy. The average accuracy stated in Table 2 are as reported by the respective authors for reference purpose. However, it is hard to make a direct comparison of these protocols in terms of synchronization error. This is because each protocols has their own evaluations and differences in hardware specification [11].

Table 1 Comparisons of the protocols in two categories

	Centralized	Distributed
Convergence speed	Fast	Slow, relating to the network topology
Special/reference node	Yes	No
Node failure robustness	Low	High
Other advantages	Little synchronization error	Improved scalability, load balancing, and robustness to new node appearance [16]
Disadvantages	Synchronization errors grow with the increase of network hops [16]	Do not synchronize to an external source, will require some form of post-facto synchronization if the network partitioned into multiple isolated islands [15]

Table 2 Comparisons of time synchronization protocols in 6 categories

	Dist.	Skew Comp.	MAC	Topology	Convergence speed	Average accuracy
RBS [7]	No	Yes	Yes	Cluster	Fast	29.1 μs
TPSN [8]	No	No	No	Tree	Slow	16.9 μs
FTSP [9]	No	Yes	Yes	Tree + Mesh	Fast	1.48 μs
TDP [10]	Yes	No	No	Mesh	Slow (~34 s)[a]	–
RFA [11]	Yes	No	No	Mesh	Slow	100 μs
GCS [12]	Yes	No	No	Mesh	Slow	–
DTSP [13]	Yes	Yes	Yes	Mesh	Slow	2 μs
ATS [14]	Yes	Yes	Yes	Mesh	Slow (~20 s)[b]	–
CCS [15]	Yes	Yes	Yes	Mesh	Slow (~13 s)[c]	–
TSMA [16]	Yes	Yes	Yes	Mesh	Slow (~5 s)[c]	±10 ticks
TSDOS [17]	Yes	Yes	Yes	Tree	Fast	–

[a]Specifications = 200 simulated nodes deployed randomly in a 80 m by 80 m sensor field
[b]Specifications = 4 nodes with synchronization period of 3 min
[c]Specifications = 100 simulated node in a 10 * 10 network with synchronization period of 1 min in transmission ranges of 3 hops

4 Conclusion

In this paper, brief introductions of the wireless sensor network and the time synchronization protocols are given. WSN is getting more and more popular due to its suitability to be applied in different areas of applications. However, there are still many constraints have to be solved for the WSN to be fully utilized. Among the constraints, the most influential constraint is time synchronization as it can be easily caused by others constraints and itself will also affect the functionality of the WSN. Many techniques and protocols have been developed by researchers to solve the issue of time synchronization. However, there is still do not have the best solution for the issue of time synchronization. Through this paper, some insights into the issue of time synchronization and the protocols used in time synchronization are provided. It can be noticed that although distributed approaches has slower convergence speed, they have received many attentions in past few years. This is because the advantages of distributed approach such as improved scalability and robustness to node failure and new node appearance. These key features are very important as in real world, the sensor nodes may malfunction unexpectedly.

References

1. Culler D, Estrin D, Srivastava M (2004) Guest editors' introduction: overview of sensor networks. Comput (Long Beach Calif) 37(8):41–49
2. Akyildiz IF, Vuran MC (2010) Wireless sensor networks, 1st edn. Wiley, Chichester

3. Nadiminti K, De Assunção MD, Buyya R (2006) Distributed systems and recent innovations: challenges and benefits. InfoNet Mag 16(3). Victorian Information Technology Teachers Association (VITTA) Inc., Melbourne
4. Dargie W, Poellabauer C (2010) Fundamentals of wireless sensor networks: theory and practice. Wiley, West Sussex
5. Sivrikaya F, Yener B (2004) Time synchronization in sensor networks: a survey. IEEE Netw. 18(4):45–50
6. Mills DL (1991) Internet time synchronization: the network time protocol. IEEE Trans Commun 39(10)
7. Elson J, Girod L, Estrin D (2002) Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Oper Syst Rev 36(SI):147–163
8. Ganeriwal S, Kumar R, Srivastava MB (2003) Timing-sync protocol for sensor networks. In: Proceedings of the first international conference on embedded networked sensor systems SenSys 03, pp 138–149
9. Maróti M, Kusy B, Simon G, Lédeczi Á (2004) The flooding time synchronization protocol. In: ACM international conference on embedded networked sensor systems, 39–49
10. Su W, Akyildiz IF (2005) Time-diffusion synchronization protocol for wireless sensor networks. IEEE/ACM Trans Netw 13(2):384–397
11. Werner-Allen G, Tewari G, Patel A, Welsh M, Nagpal R (2005) Firefly-inspired sensor network synchronicity with realistic radio effects. In: Proceedings of the 3rd international conference on Embedded networked sensor systems—SenSys'05, p 142
12. Li Q, Rus D (2006) Global clock synchronization in sensor networks. IEEE Trans Comput 55 (2):214–226
13. Solis R, Borkar VS, Kumar PR (2006) A new distributed time synchronization protocol for multihop wireless networks. In: 45th IEEE conference on decision and control, pp 2734–2739
14. Schenato L, Gamba G (2007) A distributed consensus protocol for clock synchronization in wireless sensor network. In: 46th IEEE conference on decision and control, pp 2289–2294
15. Maggs MK, O'Keefe SG, Thiel DV (2012) Consensus clock synchronization for wireless sensor networks. IEEE Sens J 12(6):2269–2277
16. Dengchang Z, Zhulin A, Yongjun X (2013) Time synchronization in wireless sensor networks using max and average consensus protocol. Int J Distrib Sens Netw 2013:10
17. Hui YT, Mahyuddin MN (2015) Time synchronisation using distributed observer algorithm with sliding-mode element in wireless sensor network. In: 9th IEEE international conference on sensor and technology, Auckland, New Zealand, 8–10 Dec 2015