Wireless Pers Commun
DOI 10.1007/s11277-016-3377-7

CrossMark

Soft Computing Based Fine Tuning and Clock Synchronization in Wireless Sensor Networks

N. Saravanaselvam[1] · C. Devipriya[1] · S. Janakiraman[2]

Abstract In this paper, we focus on the various aspects related to synchronization in wireless sensor networks, their features and how to increase the lifetime of the sensor and network nodes. The focus is about clock synchronization, delay measurement in time synchronization and fine grained network time synchronization. The recent advancements in the sensor technology has led to the miniaturization of the sensors and its related components, and similarly reduced the cost to a larger extent. This enabled an active rise in the research of large scale networks of sensors. The synchronization of sensor nodes is critical to its effective operation especially the time synchronization, for the data fusion, duty cycling and so on. Most of the algorithms for synchronization that were proposed recently do the same work. The server periodically sends a piece of information or message that contains the current value of clock to the client. This happens in case of time synchronization of the sensor nodes. The proper definition for sensor networks comes as follows. It is a special type of ad-hoc network, in which the nodes, also referred to ass wireless devices get together and the form a network without any need for underlying infrastructure spontaneously. They can cooperate only by forwarding packets to each other since there are no routers in an environment without any infrastructure. When the environment is distributed in nature, then more complex is the scenario. But time synchronization is an important component of such distributed environments. It aims to provide a common time for the local clocks in their network (Sivrikaya and Yener in IEEE Netw 18(4):45–50, 2004). For applications of networking protocols, at any particular instant, there should be a common view of the clock in the network.

✉ N. Saravanaselvam
saravanaselvam.n@sece.ac.in

C. Devipriya
devipriya.c@sece.ac.in

S. Janakiraman
sj.dbt@pondiuni.edu.in

1 Department of CSE, Sri Eshwar College of Engineering, Coimbatore, India

2 Department of Banking Technology, Pondicherry University, Pondicherry, India

Published online: 14 June 2016

Keywords Wireless sensor networks (WSN) · Time synchronization · Quality of Service (QoS) · Clock time · Routing · Soft computing · Evolutionary computing

1 Introduction

Wireless sensor networks (WSNs) have received a huge attention and focus among the emerging areas of the researches due to its application in wide areas like battle fields, habitat monitoring, underwater and underground monitoring, mobile robots, etc. There can be several environmental conditions which may make a sensor node to die and can also be due to depletion of battery power. The basic composition of a WSN is motes which pose only scarce resources like limited computational capabilities and memory features. The first challenge in the implementation of WSNs is source location privacy [2]. The other design challenge in designing a (WSN) is to maximize the network lifetime, since each sensor node of the network is equipped with a limited power battery. To overcome this, different methods were proposed like network protocols, data fusion algorithms using low power, energy efficient routing, and locating optimal sink position. We focus on finding the optimal sink position. To improve WSNs' reliability, we should consider that a sensor covers targets with users' satisfied probability. To overcome this, introduce a failure probability into the target coverage problem to improve and control the system reliability. Energy savings optimization becomes one of the major concerns in WSN routing protocol design, due to the fact that most sensor nodes are equipped with the limited non rechargeable battery power. Using the principle of opportunistic routing theory, multi hop relay decision to optimize the network energy efficiency can be made based on the differences among sensor nodes, in terms of both their distance to sink and the residual energy of each other.

A wireless sensor network comes under Chip based technology where storage, sensing, computation, communication are all in a single chip. Furthermore, these nodes have to be minimized in terms of size and increased in terms of battery life and node lifetime. There are strict constraints on the battery and processing power, memory needed for programming and the bandwidth available for usage. As of now, the advances in technology has brought out improvements in manufacture of minimized and cheap nodes which are capable to communicate wirelessly with other systems and also be able to sense and perform computations on its own. The components of a sensor node [3] include transceiver, micro controller, sensing unit, Analog to Digital converter (ADC) and most importantly a battery and memory part. A transceiver is a unit which does the job of transmitting the signals to other sensor nodes or the station controller and receiving the signals from other devices which are transmitting signals that are useful for the operating sensor node. The Micro controller unit is the master part which controls the entire operations of the node and it operates at low frequency compared to traditional contemporary processing units. The sensing unit is responsible for the gathering of information related to various environmental aspects in which it is deployed like temperature, humidity, climatic changes, and pressure variations and so on. The information gathered by the sensing unit will be in the form of analog signals which needs to be converted into digital signals to be able to transmit to other receiving nodes. This task is done by the ADC and the converted signals are fed into the master unit of the node which is the micro controller. Then either continuously or at periodic times this information is transmitted by the transceiver based on the control signals issued by the micro controller.

2 Synchronization Methods

The methods for synchronization are based on exchange of some form of messages among the nodes of the network. There can be some characteristics in the underlying network such as the propagation time and the channel access time that makes these synchronization methods to be non-productive. The synchronization in our aspect works with timestamp generation and encapsulate that timestamp as a packet and pass it to other nodes in the network, so that others can check with the clock view of the nearby node. When one of the nodes in the network generates a packet containing the timestamp, it is sent to the other node for synchronization [4]. That packet which is indented for immediate transmission will actually face a varying amount of delay till it reaches the receiver and is decoded by the receiver node intended for the same. So this delay in receiving of the packet prevents the receiver node from comparing exactly the local clocks of the nodes in the network and be able to synchronize with the sender node of the network. There are four basic components for the source of the error in the synchronization of the time in the network [5]. They are send time, access time, propagation time and the receive time. The send time refers to the time which is spent by the sender to construct the message at the sender site. This may cause interrupts and context switching overhead of the OS, and also the time which is taken to transfer the message to the transmission network interface of the system. The packets face some delay at the Medium Access Control (MAC) layer before they are transmitted. The access time delay depends on the scheme used in MAC, waiting for the channel to be free for transmission, and the TDMA slots. The propagation delay is the time which is spent for the propagation of the packets of the message between the network interface of the sender and the network interface of the receiver. The receive time refers to the time which is spent for the interface of the receiver to receive it and transfer the same to the host.

3 Necessity of Synchronization

The reason for addressing the problem of synchronization in the sensor networks extends to some number [6, 7]. The first and the foremost reason is that the sensor nodes need to organize in such a way that their operations are coordinated and collaborated to achieve a good performance sensing. The second reason is that the synchronization can be used to achieve power saving of the nodes which will increase the lifetime of the network. This is the case of sleep and awake characteristics of the sensor nodes. The sensor nodes go to sleep mode when there is no work to perform. This should be coordinated by the precise timing between the sensor nodes. The traditional synchronization schemes will not be suitable for sensor networks because of the energy issues and the complex nature of the sensor nodes. Those schemes will only work better under the networked systems.

4 Requirements of the Scheme

The metrics for synchronization schemes on the sensor nodes are presented by [1]. But there is a need for trade-offs among the following requirements. One of the requirements is energy efficiency. The synchronization schemes designed for the sensor nodes should consider the energy resources which are limited in the nodes. The next comes the scalability. The sensor nodes may be increased with the increase in requirements of the sensor network. So the designed synchronization scheme should fit to consider the deployment of any number of nodes in the network.

The other factor is precision or the accuracy [8], which may depend on the purpose of the synchronization scheme. Then the next attribute is robustness of the scheme. Even if there is a failure in one node, the remaining of the nodes should be able to function effectively for the rest of the network and remain valid. The other requirement is related to lifetime of the scheme. The synchronization time provided by the algorithm may last as long as the operation of the network time. The next is the scope of the synchronization scheme. The scheme may give a global base time or a local time for the spatially close nodes. This should also provide a global synchronization. Then comes the cost and size of the scheme. The sensor nodes are expensive and they are small in size. So we cannot attach a large and expensive hardware on such a device. So the method should be developed with these factors in consideration, so that it is with limited size and cost issues.

In the following section, we will focus on some of the proposals in the same nature by the eminent researchers. After various proposals, some are considered to be proven for the sensor networks.

5 Methods for Synchronization

5.1 Reference Broadcast (RBS)

The synchronization scheme proposed by Elson et al. [9] was Reference Broadcast Scheme (RBS). Their idea was to use a third party for synchronization. In the mentioned scheme, there is no need to synchronize the sender system with the receiver system. Instead, the scheme is meant in synchronizing the set of receivers with one another, that is receiver–receiver synchronization. The nodes are insisted in sending beacon signals to their neighboring nodes. The reference beacon is sent with the intention not to include a timestamp in it, but the time of arrival of the reference beacon is taken as a reference point for the receiving nodes to compare the clocks.

The argument is that, by removing the nondeterminism of the sender from the critical path, this scheme can achieve better precision than the other synchronization scheme. As there is no effect of nondeterminism on the sender synchronization, the only possible source of error is the receive time and propagation time nondeterminism.

5.2 Lightweight Tree-Based Synchronization

The scheme was proposed by Greunen and Rabaey [10]. This scheme is different from the traditional schemes, where its intention is not to increase the accuracy, but to minimize the synchronization complexity. The needed measure of synchronization is given as the constraint and the aim is to develop an algorithm with minimum level of complexity to achieve the constrained precision. This scheme is sufficient to use as a lightweight synch scheme in the sensor networks. This is proposed for multihop synchronization, based on the pairwise synchronization. This requires nodes to be synchronized at a reference point which can be considered as a sink node in the network. The first algorithm is a centralized one, it needs the construction of spanning tree as a prior step. Then the synchronization is pairwise, which is done along all the n − 1 edges. The reference node is taken to be the root node of the tree. It has the responsibility to initiate resyncronization whenever it is needed in the sensor network [11, 12].

The second algorithm performs a distributed type of network-wide synchronization. In this case, each node decides by itself, the time for its synchronization. When a node say x, needs to synchronize, then it sends a request for synch to all the reference nodes which are close in the network. Then all the nodes in that path from node x to that reference node needs to be synchronized, before the way x is synchronized. This type presents the advantage that there is no need for frequent synchronization. This saves unnecessary efforts taken by the sensor nodes to synchronize, since each node is given the opportunity to decide its own synchronization point. The overall idea is to aggregate the synchronization requests. If there is any request for synchronization, then the nodes check for any pending request already available. If there is any one pending already, then it is aggregated with the pending ones.

5.3 Timing-Sync Protocol

A network wide synchronization scheme for sensor network was proposed by Ganeriwal [13] and that was called as timing synch protocol for sensor networks. This works in two different phases. They are level discovery and synchronization. The first phase aims in creating a leveled arrangement of nodes, which is in the form of a hierarchical topology. There can be only one node at the level zero, which is given the position as the root of the tree. In the subsequent phase, a particular node at level I synchronize with a node in the previous level. There are two phases in the working of this protocol [14]. The first phase is referred to as the level discovery phase. This step is done only one at the time of deployment of the network. The foremost step in this phase is to find a node to be considered as the root node in the network tree. This considered node can even be a sink node in the network and it can even possess GPS receiver. If it is the case, the node possesses a GPS receiver, and then the algorithm can be able to synchronize all the nodes in the network to an external time in the physical world. In the other case, the nodes in the network can periodically take over the functionality of the root node.

The root node at level 0, takes the work of initiating the first phase, that is, the level discovery phase [15]. It starts by sending a level discover packet. It contains the level of the sensor node and its identity in the network. If this packet is being received by some other node which is the neighbor of the root node, then it assigns itself at level 1, after which it discards any further incoming such packets. Hence all nodes are assigned a level. The next phase is the synchronization phase whose building block is the exchange of messages between nodes. This involves sending of time_synch packets and the propagation delay is considered to be constant in both directions.

6 Clock Synchronization

The clock synchronization protocols can be evaluated based on cost, accuracy, complexity and precision. The next part describes an existing clock synchronization protocols. The requirements of the clock synch protocols [16] are as follows.

- Protocol must be able cope with unbounded latencies in message and unreliability in network transmission.
- Each node should be able to estimate other node clock's local time.
- Time should always be forwarded and not back warded, that is, the clock time should gradually advance.
- It should not degrade system performance.

6.1 Remote Clock Reading Method

Clock synchronization is accomplished by exchange of messages. This allows one node to estimate other node's clock. Once the difference in time between the clocks is estimated, then the clocks can be corrected accordingly. But this is not so easy with unbounded message delays. This issue is handled by Remote Clock Reading method [17]. When one process wants to estimate the time of a remote process, then it sends a time request and waits for the response. If it receives the response, the sender of the request calculates the round trip time of the packet. This response contains an estimate of the time on the remote process. On this response, the sender process corrects the local time to the total of the estimate and the half round trip time. This step can be done multiple times and its average is taken in a non-deterministic message delay environment [18].

6.2 Challenges in Sensor Networks

In order to exploit the fullest potential of the sensor nodes, the limitations and technical issues of the same has to be addressed. The synchronization protocols must address the following features of the network [19].

- Limited energy
- Limited bandwidth
- Limited hardware
- Unstable network connections
- Tight coupling between sensors and physical world

6.3 Design Principles of Clock Synchronization

Due to the above limitations and peculiar features of WSN, the synchronization protocols cannot be directly applied. The design considerations are [16]:

- Energy efficiency
- Infrastructure
- End-to-end latency
- Message loss and message delivery
- Network dynamics.

6.4 Application Dependent Features of Synchronization Protocols

Single-Hop Communication A sensor node can directly exchange messages with any other sensor node in the network. Mostly, the network is large, hence impossible to communicate directly.

Multi-Hop Communication The increase in network size leads to multi-hop communication. The sensors in one domain communicate with one in another domain via an intermediate node that relates both domains.

Stationary Networks In this scenario, the sensors do not move. Once sensors are deployed, the topology remains unchanged.

Mobile Networks In a mobile network, the sensors can move. The sensors within a scope can only communicate with each other.

7 Conclusions

The sensor networks are widely applicable in a diverse set of applications. Even though there are many issues and difficulties in building such networks, the main challenges are clock synchronization and time synchronization.

In order for the physically distributed processors to have common concept of time, clock synchronization is critical. Only by ensuring clock synchronization, we can ensure time synchronization in the network. In this paper, we have addressed several critical issues related to clock and time synchronization for the operation of the sensor networks. The clocks need to be synchronized to an accurate real time standard not only with each other, but also with the physical clock. There is a lack of analytical models for time synchronization for multi-hop networks.

References

1. Sivrikaya, F., & Yener, B. (2004). Time synchronization in sensor networks: A survey. *IEEE Network, 18*(4), 45–50.
2. Devipriya, C., Saravanaselvam, N., & Saradha, S. (2015). Achieving increased QoS, maximized lifetime, better routing and target coverage in wireless sensor networks. *ARPN Journal of Engineering and Applied Sciences, 10*(13), 5642–5645.
3. Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5th symposium on operating systems design and implementation.*
4. Ping, S. (2003). Delay measurement time synchronization for wireless sensor networks. *Intel Research Berkeley Lab 6,* 1–10.
5. Sundararaman, B., Buy, U., & Kshemkalyani, A. D. (2005). Clock synchronization for wireless sensor networks: A survey. *Ad hoc Networks, 3*(3), 281–323.
6. Reddy, V., Kumar, P., Janakiram, D., & Kumar, G. A. (2009). Operating systems for wireless sensor networks: A survey. *International Journal of Sensor Networks, 5*, 236–255.
7. Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks, 52*(12), 2292–2330.
8. Greunen, J. V., & Rabaey, J. (2003). Lightweight time synchronization using reference broadcasts. In *Proceedings of the 2nd ACM international conference on WSNA*. San Diego, CA.
9. Li, Y., Chen, C.-S., Song, Y.-Q. et al. (2007). RealtimeQoS support in wireless sensor networks: A survey. In *Proceedings of the 7th IFAC international conference on fieldbuses & networks in industrial &embedded systems (FET'07)* (pp. 373–380). Toulouse, France.
10. Cristian, F. (1989). Probablistic clock synchronization. *Distributed Computing, 3*, 146–158.

11. Devipriya, C., Mohanram, S., & Saradha, S. (2015). Design aspects of operating systems used in wireless sensor networks. *International Journal of Applied Engineering Research, 10*(20), 18570–18579.
12. Son, D., Krishnamachari, B., & Heidemann, J. (2006). Experimental study of concurrent transmission in wireless sensor networks. In *Proceedings of the 4th international conference on embedded networked sensor systems (SenSys'06)* (pp. 237–250). Boulder, Colorado, USA.
13. Li, J., Blake, C., Couto, D. S. J. D. et al. (2001). Capacity of adhoc wireless networks. In *Proceedings of the 7th annual international conference on mobile computing and networking (MobiCom'01)* (pp. 61–69). Rome, Italy.
14. Radi, M., Dezfouli, B., Bakar, K. A., Razak, S. A., & Nematbakhsh, M. A. (2011). Interference-aware multipath routing protocol for QoS improvement in event-driven wireless sensor networks. *Tsinghua Science and Technology, 16*(5), 475–490.
15. Rahman, M. N., & Matin, M. A. (2011). Efficient algorithm for prolonging network lifetime of wireless sensor networks. *Tsinghua Science And Technology, 16*(6), 561–568.
16. Wang, W., Srinivasan, V., & Chua, K.-C. (2008). Extending the lifetime of wireless sensor networks through mobile relays. *IEEE Transactions on Networking, 16*(5), 1108–1120.
17. Hou, Y. T., Shi, Y., & Sherali, H. D. (2006). Optimal base-station selection for anycast routing in wireless sensor networks. *IEEE Transaction on Vehicular Technology, 55*(3), 813–821.
18. Showkat, M. J., Paul, B., Matin, M. A., et al. (2009). Optimal sink location in wireless sensor networks using particle swarm optimization. In *Proceedings of the 2009 IEEE international conference on antennas, propagation and systems (INAS 2009)* (pp. 1–4). Johor Bahru, Malaysia.
19. Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *IEEE International conference on neural networks* (Vol. 4, pp. 1942–1948). Perth, Australia.

Dr. N. Saravanaselvam has obtained his Ph.D. in Computer Science and Engineering from Anna University, Chennai in the year 2013. He has obtained both of his Post Graduate degree, M.E. (Computer Science and Engineering) and Graduate degree B.E. (Electronics and Communication Engineering) from Madurai Kamaraj University (Tamilnadu, India). He is currently serving as Professor and Head of the Department of Computer Science and Engineering at Sri Eshwar College of Engineering, Coimbatore, Tamilnadu. During his 16 years of teaching profession, he shouldered a member of teaching, administrative and societal based assignments. He is a Life Member of ISTE, IAEng and IACSIT. Currently, he is specializing in the area of Network Engineering. He has published more than 30 papers in international journals.

Ms. C. Devipriya has obtained her Masters in Computer Science and Engineering at Sri Eshwar College of Engineering, Coimbatore, India and Bachelor of Engineering from Dr. Mahalingam College of Engineering and Technology, Pollachi, India. She is currently working as an Assistant Professor in the department of Computer Science and Engineering at Sri Eshwar College of Engineering, Coimbatore, India with an experience of about 2 years. Her fields of specialization are Network Security, Wireless Sensor Networks, Operating systems and Data structures.

Dr. S. Janakiraman has obtained his Ph.D. in Computer Science and Engineering from Anna University, Chennai in the year 2009. He has obtained both of his Post Graduate degree, M.E. (Computer Science and Engineering) and Graduate degree B.E., (Electrical and Electronics Engineering) from Madurai Kamaraj University (Tamilnadu, India). He is currently serving as Assistant Professor, Department of Banking Technology at Pondicherry University, Pondicherry. He has 15 years of teaching and research experience. He is a Life Member of ISTE, Institution of Engineers (India). His area of research interest is Machine Learning and pattern recognition, Computer Networks, Security, and Image Processing. He has published more than 22 papers in international journals and presented 44 papers in international and national conferences.