A Study on Clock Synchronization Methods in Wired Networks and WSN

Seyed Kazem Kazeminezhad

Abstract—In this paper we study time synchronization protocols in wireless sensor networks and wired networks. There are some synchronous and asynchronous methods. Some are based on remote clock reading and others are based on source clock frequency recovery. Energy consumption is critical in WSN. Minimizing the number of message transmissions from sensor nodes to the head node or from the head to the source nodes is the best way to save energy. We will discuss some of the methods of time synchronization in WSN and other networks. We will discuss NTP, RBS, TPSN, FTSP, RSTP and SCFR methods. NTP is appropriate for wired networks like internet and others for wireless networks. Each method has its own characteristics such as power consumption and accuracy and so on. Study results shows that each network should have its own operational time synchronization protocol due to its own needs.

Index Terms—Time synchronization, wireless sensor network, symmetric, asymmetric.

I. INTRODUCTION

Time synchronization is one of critical components in WSN operation, as it provides a common time frame among different nodes. It supports functions such as fusing data from different sensor nodes, time-based channel sharing and media access control (MAC) protocols, and coordinated sleep wake-up node scheduling mechanisms [1]. There are symmetric synchronization against Asymmetric synchronization. In computer networking, unicast transmission refers to sending messages to a single network destination identified by a unique address. However, the term unicast contrasts with the term broadcast which refers to transmitting the same data to all possible destinations. One more multi-destination distribution method is multicasting which sends data only to interested destinations by using special address assignments [2]. The old synchronization method in wired networks is NTP which is designed to distribute time information in a large, diverse internet system operating at speeds from mundane to light wave. It uses a symmetric architecture in which a distributed subnet of time servers operating in a self-organizing, hierarchical configuration synchronizes local clocks within the subnet and to national time standards via wire, radio, or calibrated atomic clock [3]. Node A will receive messages sent by node B if and only if node B can receive messages sent by node A. In this case, it is mentioned that the link between these two nodes is symmetrical. Otherwise, it is asymmetrical [4]. In Master–slave synchronization, a master–slave protocol assigns one node as the master and the other nodes as the slaves. The slave nodes consider the local clock reading of the master at the reference time and attempt to synchronize their times with the master. In general, the master node requires CPU resources proportional to the number of slaves and nodes with powerful processors or lighter loads are assigned as the master node [5]. In contrast, in external synchronization, a standard time source such as Universal Time is provided. Thanks to an atomic clock which calls reference time, there is no need for a global time base. Reference time provides the actual real-world time. In WSNs, most protocols do not use external synchronization unless the application demands it [5]. WSNs are used in diverse applications such as the measurement of humidity, temperature, etc, in which time synchronization is a critical requirement. Since each event should be labeled exactly with respect to the reference time, maintenance and repair of a real-time clock is essential for data fusion in WSNs. Based on the conditions of the environment, the reference time may be made local or global. The potential applications of wireless sensor networks (WSN) include:

1) Non-intrusive and non-disruptive environmental monitoring helps biologists to study sensitive wildlife habitats and people with certain medical conditions can receive constant monitoring through sensors [6]. Sensor networks monitor the structural health of the Golden Gate Bridge in San Francisco and the micro climate son Great Duck Island, Maine [7].

2) Mobile commerce, inventory management: by measuring continuously changing conditions, WSN will influence them movement of commodities to locations where the need exists.

3) Smart office, kindergarten: systems containing wireless sensors will be an integral part of our office space. They would improve the education process by tailoring it to the individual needs of a child [8], adapt to context, and coordinate activities of multiple children.

4) Military applications: potential applications include surveillance, target tracking [9], counter sniper systems [9] or battlefield monitoring that propagates information to the soldiers and vehicles involved in combat.

WSN are large-scale distributed systems, yet their unique characteristics, especially the severe resource constraints, require the reevaluation of traditional distributed algorithms for problems once considered to be solved. One of the basic middleware services of sensor networks is time synchronization. Time synchronization is required for consistent distributed sensing and control. Furthermore, common services in WSN, such as coordination, communication, security, power management or distributed logging also depend on the existence of global time.

Manuscript received August 23, 2015; revised December 1, 2015.
S. K. Kazeminezhad is with Computer Engineering department of Azad University of Tabriz, Iran (e-mail: kkazeminezhad@yahoo.com).

II. METHODS

A. NTP Method

NTP method is used in weird networks e.g. internet and based on remote clock reading. The principal features of NTP design can be summarized as follows.

1) The synchronization subnet consists of a self-organizing, hierarchical network of time servers configured on the basis of estimated accuracy, precision and reliability.

2) The synchronization protocol operates in connectionless mode in order to minimize latencies, simplify implementations and provide ubiquitous interworking.

3) The synchronization mechanism uses a symmetric design which tolerates packet loss, duplication and misordering, together with filtering, selection and combining algorithms based on maximum-likelihood principles.

4) The local-clock design is based on a type 11, adaptive-parameter, phase-lock loop with corrections computed using timestamps exchanged along the arcs of the synchronization subnet.

5) Multiply redundant time servers and multiply diverse transmission paths are used in the synchronization subnet, as well as engineered algorithms which select the most reliable synchronization source(s) and path@ using weighted-voting procedures.

6) System overhead is reduced through the use of dynamic control of phase-lock loop bandwidths, poll intervals and association management[3]. When you submit your final version, after your paper has been accepted, prepare it in one-column format, including figures and tables.

B. RBS Method

Elson, Girod, and Estrin [2] proposed a synchronization scheme for sensor networks, RBS, where their simple yet novel idea is to use a “third party” for synchronization: instead of synchronizing the sender with a receiver (as in most of the previous work), their scheme synchronizes a set of receivers with one another (although its application in sensor networks is novel, the idea of receiver-receiver synchronization was previously proposed for synchronization in broadcast environments). In the RBS scheme, nodes send reference beacons to their neighbors. A reference beacon does not include a timestamp; instead, its time of arrival is used by receiving nodes as a reference point for comparing clocks. The authors argue that by removing the sender’s non determinism from the critical path, RBS achieves much better precision than traditional synchronization methods that use two-way message exchanges between synchronizing nodes.

As the sender’s non determinism has no effect on RBS precision, the only sources of error can be the non determinism in propagation time and receive time. The authors claim that a single broadcast will propagate to all receivers at essentially the same time; hence, the propagation error is negligible. This is especially true when the radio ranges are relatively small (compared to speed of light times the required synchronization precision), as is the case for sensor networks. So they only account for the receive time errors when analyzing the accuracy of their model. In the simplest form of RBS, a node broadcasts a single pulse to two receivers. The receivers, upon receiving the pulse, exchange their receiving times of the pulse and try to estimate their relative phase offsets. This basic RBS scheme can be extended in two ways:

1) Allowing synchronization between n receivers by a single pulse, where n may be larger than two

2) Increasing the number of reference pulses to achieve higher precision

The authors show by simulation that 30 reference broadcasts (for a single synchronization in time) can improve the precision from 11 µ s to 1.6 µ s when synchronizing a pair of nodes. They also make use of this redundancy for estimating clock skews: instead of averaging the phase offsets from multiple observations (e.g., each of 30 reference pulses), they propose to perform a least squares line regression to this data. Then the frequency and phase of the local node’s clock with respect to the remote node can be recovered from the slope and intercept of the line [2].

C. TPSN Method

Timing-sync Protocol for Sensor Networks (TPSN) that aims at providing network-wide time synchronization in a sensor network. The algorithm works in two steps. In the first step, a hierarchical structure is established in the network and then a pair wise synchronization is performed along the edges of this structure to establish a global timescale throughout the network. Eventually all nodes in the network synchronize their clocks to a reference node. Implemented algorithm on Berkeley motes and show that it can synchronize a pair of neighboring motes to an average accuracy of less than 20ms [8].

TPSN is based on the simplistic approach of conventional sender-receiver time synchronization. We argue that unlike traditional wireless networks, for sensor networks classical approach of sender-receiver synchronization is better than receiver-receiver synchronization. We demonstrate our claim by comparing TPSN with an algorithm based on receiver-receiver synchronization, RBS. TPSN roughly gives a 2x better performance than RBS through analysis verifying the efficacy of this analysis by implementing TPSN and RBS on Berkeley motes. TPSN is completely flexible to the model used for time synchronization. An integration of TPSN with post facto synchronization and use this to obtain results on multi hop clock synchronization on a network of motes. The efficacy of TPSN for large-scale networks is verified via simulations in PARSEC. The results clearly show that TPSN is completely scalable. Thus, TPSN is a simple, efficient, scalable and a comprehensive solution to the problem of time synchronization in sensor networks [6].

D. FTSP Method

Flooding Time Synchronization Protocol (FTSP), especially tailored for applications requiring stringent precision on resource limited wireless platforms. The proposed time synchronization protocol uses low communication bandwidth and it is robust against node and link failures. The FTSP achieves its robustness by utilizing periodic flooding of synchronization messages, and implicit dynamic topology update. The unique high precision
performance is reached by utilizing MAC-layer time-stamping and comprehensive error compensation including clock skew estimation. The sources of delays and uncertainties in message transmission are analyzed in detail and techniques are presented to mitigate their effects. The FTSP was implemented on the Berkeley Mica2 platform and evaluated in a 60-node, multi-hop setup. The average per-hop synchronization error was in the one microsecond range, which is markedly better than that of the existing RBS and TPSN algorithms. The proposed FTSP algorithm uses a fine-grained clock, MAC-layer time-stamping with several jitter reducing techniques to achieve high precision. This approach eliminates the send, access, interrupt handling, encoding, decoding and receive time errors, but does not compensate for the propagation time [10].

Multiple time-stamps with linear regression are used to estimate clock skew and offset. The average error of the algorithm for a single hop case 47 using two nodes was 1.48\mu s, according to measurements described in Section 5.2. In the multi-hop case, the average error was 3\mu s in a 6-hop network, resulting in a 0.5\mu s per hop accuracy. The applied flood-based communication protocol in FTSP provides a very robust network, and still induces only small network traffic. The network hierarchy is maintained using the time synchronization messages, without additional message passing, as opposed to the solution in TPSN.

FTSP also utilizes less network resources than either RBS or TPSN. If the resynchronization period is T seconds, then each node sends 1message per T seconds in FTSP, 2 messages per T seconds in TPSN (1 message to parent and 1 response) and 1.5 message per T seconds in RBS (0.5 for a reference broadcast and 1 for a time stamp exchange message). Since FTSP does not rely on a fixed network hierarchy but updates it continuously, it supports network topology changes including mobile nodes. The robustness of the protocol was demonstrated by the harsh experiment described in Section 5.4. Unfortunately, no similar data is readily available for TPSN or RBS for comparison [11].

<table>
<thead>
<tr>
<th>NAME</th>
<th>External/Internal</th>
<th>Time Stamp</th>
<th>Wired/Wireless</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP</td>
<td>External/Internal</td>
<td>Yes</td>
<td>Wired</td>
</tr>
<tr>
<td>RBS</td>
<td>Internal</td>
<td>No</td>
<td>Wired-Wireless</td>
</tr>
<tr>
<td>TPSN</td>
<td>External/Internal</td>
<td>Yes</td>
<td>Wired-Wireless</td>
</tr>
<tr>
<td>FTSP</td>
<td>External/Internal</td>
<td>Yes</td>
<td>Wired-Wireless</td>
</tr>
<tr>
<td>RSTP</td>
<td>External/Internal</td>
<td>No</td>
<td>Wired-Wireless</td>
</tr>
<tr>
<td>SCFR</td>
<td>External/Internal</td>
<td>No</td>
<td>Wired-Wireless</td>
</tr>
</tbody>
</table>

### E. RSTP and SCFR Method

In a typical WSN, a master/head node is equipped with a powerful processor, connected to both wired & wireless networks, and supplied power from outlet because it serves as a gateway between the WSN & a backbone and a center for fusion of sensory data from sensor nodes, which are limited in processing and electrical power because they are connected only through wireless channels and battery-powered. It is this asymmetry that we focus our study on; unlike existing schemes which save the power of all WSN nodes including the head (e.g., [12] and [13]), we concentrate on battery-powered sensor nodes, which are many in numbers, in minimizing energy consumption for synchronization. Specifically, RTSP and SCFR discuss a time synchronization scheme based on the source clock frequency recovery (SCFR) [13], where it minimizes the number of message transmissions at sensor nodes because the energy for packet transmission is typically higher than that for packet reception [13]. It also discusses its extension to network-wide, multi-hop synchronization through gateway nodes.

### III. Conclusion

Time synchronization protocols are presented in many details. But the specifications of each are different due to their usage. Some act better in wired networks like NTP and others provide better performance in wireless sensor networks. No algorithm has been presented to do the exact time synchronizations because of non-deterministic errors. NTP is designed to distribute time information in a large, diverse internet system operating at speeds from mundane to light wave. It uses a symmetric architecture. RBS novel idea is to use a “third party” for synchronization instead of synchronizing the sender with a receiver. This scheme synchronizes a set of receivers with one another. TPSN provides network-wide time synchronization in a sensor network. The algorithm works in two steps. In the first step, a hierarchical structure is established in the network and then a pair wise synchronization is performed along the edges of this structure to establish a global timescale throughout the network. Eventually all nodes in the network synchronize their clocks to a reference node. FTSP especially tailored for applications requiring stringent precision on resource limited wireless platforms. The proposed time synchronization protocol uses low communication bandwidth and it is robust against node and link failures. RTSP and SCFR discuss a time synchronization scheme based on the source clock frequency recovery and also concentrate on battery-powered sensor nodes, which are many in numbers, in minimizing energy consumption for synchronization. Specifically, RTSP and SCFR discuss a time synchronization scheme based on the source clock frequency recovery. We will merge some of the methods and try to eliminate non-deterministic errors to achieve a new method in future. Table I presents a comparison between the ways discussed in this article.

### References

Seyed Kazem Kazeminezhad was born in Tabriz, Iran, in 1984. He received his B.Sc. in computer engineering from Islamic Azad University, Tabriz, Iran, in 2007. He received his M.Sc. degree in architecture of computer from Islamic Azad University, Tabriz, Iran, in 2013. His research interests include wireless sensor networks and its sub fields of study.

Seyed Kazem has worked in Shohada hospital as computer administrator from 2005 to 2008. He worked in computer and information center of Sepah bank from 2008 till now. The recent publication is “Proposing a novel method for clock synchronization by reducing the Number of Synchronization Messages and Eliminating Non-Deterministic Errors in Wireless Sensor Networks”, ICEEE2015 Islamic Azad University Gonabad Branch Aug 19,20 - 2015.