IFERP

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

A Comparative Analysis of Various Time Synchronization Protocols in Wireless Sensor Networks

[1] R. Arthi, [2] M. Muthumalathi, [3] Dr P.B. Pankajavalli

[1] PG student, Department of Computer Science, Bharathiar University, Coimbatore, India
[2] Assistant Professor, Department of Computer Applications, Nirmala College for Women, Coimbatore, India
[3] Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, India
Email: [1] arthisrii2461@gmail.com, [2] muthumalathi3004@gmail.com, [3] pankajavalli@buc.edu.in

Abstract— ***Wireless Sensor Networks (WSNs) are small battery-powered devices that detect and respond to changes in their surroundings. Sensor networks are collection of wirelessly connected sensors that are used in the real world. The aggregated data of a network is produced by analyzing the local information of all sensor nodes. Time synchronization is an important aspect of wireless networks. In this paper, we examine the various existing time synchronization protocols and compare them using various criteria to demonstrate the need for a new protocol that is scalable, secure, energy efficient, fast convergent, and less complex.***

Keywords— ***Sensor Networks, Energy Consumption, Time Synchronization, Node Deployment***

I. INTRODUCTION

Time synchronization is essential for Sensor network, both wired and wireless. This allows for efficient communication between nodes in the network. However, this is especially important in WSNs. Synchronization of wireless nodes allows the use of TDMA rule sets in multi-hop wireless networks. Wireless time synchronization is used for many unique purposes, including area, proximity, power efficiency, and call mobility[1]. In a sensor network, a node's specific proximity is not recognized when it is placed, so time synchronization is used to determine its area. You can also send timestamped messages to some nodes to determine their relative proximity to each other. Time synchronization is used to save power. This allows the node to sleep for a set amount of time and then wake up periodically to receive a beacon signal. Many wireless nodes are battery powered and require power efficient protocols. After all, the absence of abnormal synchronization between nodes can determine the speed of a moving node. The need for synchronization is clear. Besides many uses like determining proximity or speed, it's desirable because of the fact that the hardware clock isn't the best. There are versions of oscillators where the clock can also slip and the time period is no longer evenly spaced between the nodes. Especially in Wi-Fi networks, the concept of time and time synchronization is required [2].

II. CHALLENGES IN WIRELESS SENSOR NETWORKS

1) Limited bandwidth

In a WSNs, information analysis consumes significantly less power than transmission. Wireless communications are currently limited to an actual speed of 10100 kilobits per second. Delay-free message transmission between sensors is recommended over bandwidth barriers, without which synchronization is not possible. Sensor networks often use multi-hop radio environments for bandwidth-constrained conversations and overall performance. These wireless links communicate using radio, infrared, or optical frequencies.

2) Energy Efficiency -

The strength characteristic is the first and most important issue when designing a WSNs. The three functional areas of power consumption are perception, dialogue and fact processing, all of which require optimization. The lifetime of the sensor node is generally highly dependent on the lifetime of the battery [3]. Sensor nodes have a limited power budget, and sensors are typically powered by batteries that need to be replaced or recharged when the battery runs out. The sensor node must be able to run on a non-rechargeable battery until the task timer expires or the battery is replaced. The duration of the test is determined by the software [4].

3) Cost-effectiveness -

A WSNs consists of many sensor nodes. As a result, the fee per node is very important to the overall economic indicators of the sensor network. Simply put, for the global figures to be accurate, the cost of each sensor node must be kept to a minimum. Relying on the use of the sensor community, such as climate monitoring, vast arrays of sensors can be randomly distributed throughout the environment. Changing the overall price of the sensor network to a reasonable price will be more appropriate and beneficial to customers who need close attention [5].

4) Node Placement -

One of the important issues to deal with in WSNs is node placement. The Awesome Node deployment approach can

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

reduce the complexity of the problem. Deploying and managing a large number of nodes in a confined environment requires the use of a unique methodology. A sensor area can have a maximum number of sensor masses. On the other hand, deployment models include (i)static deployment and (ii)dynamic deployment. Static Deployment chooses an entirely appropriate location based on the optimization method, and the area of the sensor node remains unchanged at some point in the lifetime of the WSNs. Nodes are randomly selected from dynamic deployment for optimization[6].

5) Layout development -

The main purpose of the Wi-Fi sensor layout is to make devices smaller, cheaper and more efficient. Deploying sensor nodes and wireless sensor networks can be challenging due to a number of challenges. With limited limitations, WSNs is problematic for all software and hardware models [7].

6) Trouble -

One of the challenges of WSNs is the high security requirements of assemblies when working with limited sources. An extensive network of wireless sensors collects sensitive data. Remotely operated and unmanaged sensor nodes are more vulnerable to hostile intrusions and attacks. Besides data privacy, host authentication is the two security standards of WSNs[8]. A distribution sensor must be a node authenticated by its supervisor node or cluster head to identify both honest and untrusted nodes from a security point of view, and at certain points in the node authentication method WSNs can remove rogue nodes. As a result, sensor networks require new responses for key entry and distribution, node authentication, and privacy [9].

III. WIRELESS SENSORE NETWORK SYNCHRONIZATION

- Determining time synchronization does not necessarily mean that all clocks on the network are perfectly aligned. This is the strictest form of synchronization and the most difficult to implement.
- Accurate clock synchronization is usually not always required, so you should use protocols ranging from weak to strict protocols to meet your requirements.
- There are three main types of synchronization strategies for Wi-Fi networks. The default time is relative time and is the most efficient. Miles depend on the sequence of messages and events. The basic idea is that you can determine whether event 1 happened before event 2 or not. Comparing community clocks is sufficient to determine the order [10].
- The next way is the relative time at which the network clock has no offsets and the nodes keep the music in waves and offsets. In general, a node maintains statistics about its floating position and offset according to its neighbors. A node has the ability to synchronize its nearest time with the nearest time of other nodes in place. Maximum timing protocols use this approach.
- The last method is a global sync with a regular global timeline for the entire community. This is obviously the most difficult and most difficult to implement. Few synchronization algorithms specifically use this method because this form of synchronization is generally not always necessary[11].

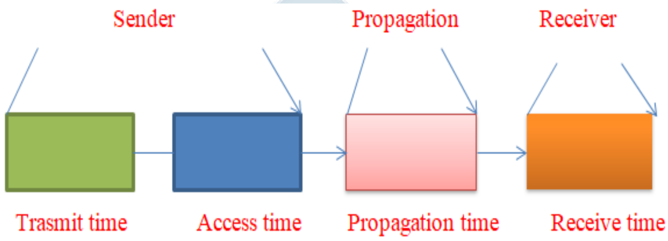

Figure 1 – Time Synchronization of Wireless Communication

As in figure 1, all wireless synchronization schemes have four main packet delay factors: transmit time, access time, propagation time, and receive time. Transmit time is the time the sender creates a time message for transmission over the network. Access time is Mac-level latency when accessing the network. It waits for a TDMA transfer. The time it takes for a bit to physically move through the medium is called propagation time. Ultimately, the acquisition time is the receiving node processing the message and moving it to the host. The important hassle of time synchronization isn't the simplest that this packet delay exists, but also being capable of predicting the time spent on each may be tough. Getting rid of any of those will greatly increase the overall performance of the synchronization approach[12].

IV. TIME SYNCHRONIZATION PROTOCOLS

S.No	Time Sync Protocol name	Developed Author name and year	Description
1	**Network Time Protocol**	It was developed by David mills in 1985	It was used as a communication mechanism so that a seamless connection is present between the computers.it uses coordinated universal time(UTC) to sync CPU clock time

2	**Simple Network Time Protocol**	It was developed by David mills in 1992	SNTP is a subset of NTP. It describes the protocol as a simplified version of the Network Time Protocol and is also used to set the system time on networks.
3	**Precision Time Protocol**	It was developed by John Edison in 2002	The PTP protocol is used to synchronize clocks on computer networks.

Table 1 – Time synchronization protocol list

1. Network Time Protocol (NTP)

Network Time Protocol (NTP) can be a protocol that helps to keep instances of computer clocks on a physical network synchronized. This protocol is the application protocol responsible for host synchronization in the TCP/IP community. Table 1 shows how NTP became cutting edge in 1985 thanks to David Mills of the University of Delaware. This is often required in real-world communication mechanisms simply to ensure smooth communication between computer systems. NTP is implemented in most Linux and Windows-based operating structures that are widely used in control systems [13].

NTP has improved for nearly 30 years, but the paint hasn't dried out yet. This release of NTP Version 4 (ntpv4) distribution for UNIX, VMS, and Windows contains new features and enhancements, but retains backward compatibility with previous versions, including ntpv3 and ntpv2, but is no longer ntpv1. Help for ntpv1 has been discontinued due to a specific security vulnerability [13].

1.1 Architecture of Network Time Protocol

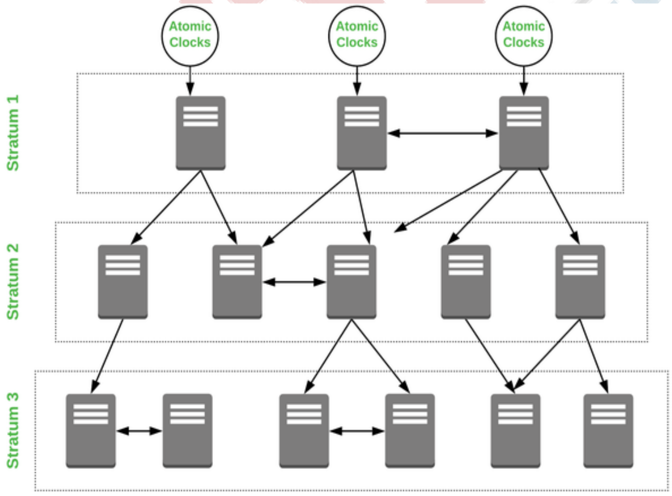

Figure 2-Working State of Network Time Protocol (NTP)

1.2 Working of NTP

As in figure 2, that the network time protocol is a protocol that operates at the software level, uses a hierarchical system of time sources, and provides synchronization at the server level. First, for example, the highest level has an exceptionally accurate time source. Atomic or GPS watch. These timing sources are referred to as tier 0 servers and are connected to lower NTP servers such as tier 1, 2, or 3. These servers then provide the exact date and time so that the communicating hosts can keep them in sync with each other[14].

1.3 Features of NTP

1. NTP servers have access to amazingly specific atomic and GPU clocks.
2. Synchronize processor time using Coordinated Universal Time (UTC).
3. Make sure that even some of the vulnerabilities don't exist in the recording of the alternate conversation.
4. Provide regular timing for recording servers.

2. Simple Network Time Protocol (SNTP)

The SNTP approach is much simpler than NTP because it skips various steps and periodically adjusts the time, which is much less accurate. The contents of the packets communicating with the server carry many functions of the NTP procedure. Table 1 above shows that using SNTP as the primary source or clock is not always recommended. The latest model for this protocol is SNTPv4. The main problem with this protocol is low security. Since there is no encryption method, there is a risk of time-changing attacks.

2.1 Architecture of Simple Network Time Protocol

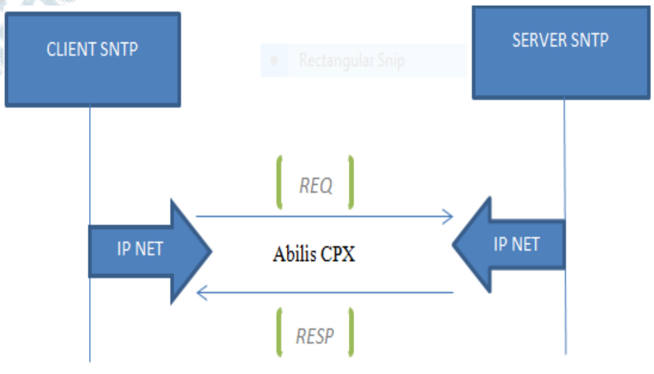

Figure 3 –SNTP Client and SNTP Server

2.2 Working of SNTP

The SNTP protocol can be used when the ultimate performance of the full NTP implementation is not needed or justified. It allows the automatic management of clock synchronization of any equipment by recovering information about the date and current time from one or more SNTP server stations available in the Internet[17]. The above figure 3 mentions that SNTP has got a client/server implementation. The Abilis CPX SNTP port, by using the Connection-Less

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

transport service provided by the UDP protocol, can:

- behave as SNTP Client, that is to synchronize the internal abilis clock with data provided by an SNTP Server available on the Internet;
- behave as SNTP Server, that is to provide the clock synchronization to any requesting SNTP Client[15].

The use of the SNTP protocol is interesting in the following cases:

- Simple devices, which include microcontrollers and small computer systems, with little reminiscence.
- Equipment where time synchronization is not critical.
- Manage gadgets such as interest, remote gadgets, and built-in gadgets.

2.3 Features of SNTP

1. SNTP is a subset of Network Time Protocol (NTP).
2. The latest version is SNTP v4.
3. Easy to synchronize with a full-fledged NTP server. It was initially designed for small computer systems and microcontrollers.
4. It requires less memory and consumes less power than NTP.
5. This time is used in applications where specific clock timing is not critical.
6. It uses the TCP/IP protocol suite, UDP port 123.

3. Precision Time Protocol (PTP)

Precision Time Protocol (PTP) is a protocol that provides clock synchronization in computer networks. This protocol is used to synchronize the clocks of various devices. The PTP was led by John Edison in 1588 for standardization and was published within 12 months of 2002 (see Table 1). This is necessary to achieve messaging over the communication medium in a synchronous environment. PTPs are particularly used for manipulating and generating strength, controlling parts manufacturing systems, transacting financial institutions, matching the values of measurements under test, and controlling robots [16].

3.1 Architecture of Precision Time Protocol

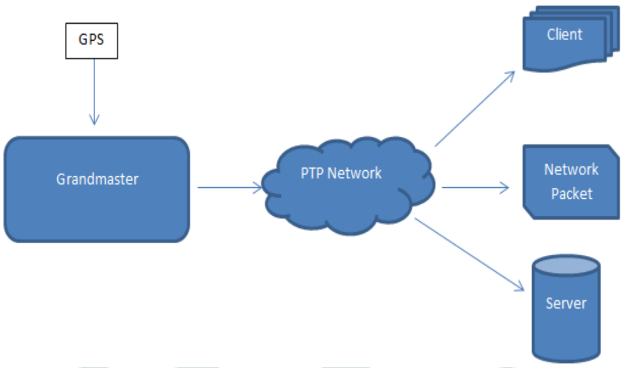

Figure 4 – Precision Time Protocol networking (PTP)

3.2 Working of precision time protocol

PTP is a protocol that works to seamlessly exchange data between different devices. It uses the master-slave device of the time source and provides synchronization. The system includes one or more communication gadgets and a single connection to the community through basic tools. In figure 4- This grandmaster is responsible for the basic timing. The grandmaster transmits the synchronized data to the devices inside the telecommunication environment. This association scheme compensates for community transfer volatility to ensure accurate distribution [17].

3.2 Features of PTP

Some of the features of PTP are –

1. It has an change time-scale capability.
2. It uses a grand grasp clock to synchronize the conversation.
3. It works on grasp-slave architecture.
4. It makes the direction of communique traceable.

V. COMPARSION CHART OF TIME SYNCHRONIZATION PROTOCOL

Table 2 – Comparative Table for Existing Time Synchronization Protocols [18, 19]

Criteria	NTP	SNTP	PTP
Abbreviation	Network Time Protocol	Simple Network Time Protocol	Precision Time Protocol
Version Available	NTPv1, 2,3, & 4	SNTPv1, 2,3, & 4	PTPv1 & 2
Year of Publication	NTP published in a year of 1985	SNTP published in a year of 1992	PTP published in a year of 2002
Nature	It used to synchronize the time of a computer that is located in a different geographical location on the internet in order to keep the same time.	The nature of SNTP is same as NTP. SNTP is simplest form of NTP	It used to synchronize clocks of different types of devices
Implementation	Hardware or Software Servers & Clients.	Hardware or Software Servers &Clients.	Hardware masters, hardware or software clients.

Accuracy	It can achieve better than one millisecond in local area network	It achieve accuracy within 100 millisecond of the time source	It achieve accuracy to sub microsecond (10 ms)
Complexity	High	Average	High
Achieve	It can achieve sub-microsecond accuracy by using hardware implementation.	It can achieve every 64 seconds to refreshes IP timing for the SNTP client to NTP server.	It can achieve nanosecond accuracy using a better oscillator.
Level	NTP used for application level synchronization.	SNTP is same as NTP. It was most vast used for application level.	PTP used for precision level synchronization.
Operate in	Master and slave (or client – server)mode.	Master and slave (or client – server)mode.	Master and slave (or client – server)mode.
Cost of Solution	Inexpensive	Inexpensive	More expensive
Security	Through Hash codes and improved clock selection.	This protocol does not present security measures, so they should be taken into account when it is implies any device.	Through cryptography security mechanisms.

VI. CONCLUSION

This paper summarizes several key elements of various time synchronization protocols such as NTP, SNTP and PTP. As in all dispensed systems, time synchronization is an important factor in WSNs because the design of many protocols and implementation of applications require precise time, accuracy, cost and complexity. This paper provides an comparison of new and existing clock synchronization protocols, which is useful for researchers in developing a new proposed protocol that is appropriate for sensor networks.

REFERENCES

[1] Fikret Sivrikaya, Bulent Yener – Time Synchronization in sensor networks : A Survey, August 2004, IEEE network 18(4):45 – 50.
[2] D.L. Mills, University of Delaware – Measured performance of the network time protocol in the internet system.
[3] M.Muthumalathi and P.B.Pankajavalli, Survey on Clock Synchronization in Wireless Sensor Networks, International Conference on Research Trends in Computing Technologies, ICRTCT- N- 23, ISBN: 978-93-87862-01-2, pp. 470-476.
[4] Tian, Z., Luo, X., Giannakis, G.B., "Cross-layer sensor network synchronization.", Signals, Systems and Computers. Conference Record of the Thirty-Eighth Asilomar Conference, Volume 1, 7-10, p. 1276 - 1280.
[5] Sheu, J. Chao, C. Sun "A Clock Synchronization Algorithm for Multi-hop Wireless Ad Hoc Networks.", Distributed Computing Systems, Proceedings. 24th International Conference p. 574 – 581
[6] Apoorvi Dwivedi , Katyayni Rastogi , Aparna Tiwari , Manish Kumar, 2019, Survey of Time Synchronization Protocols in Wireless Sensor Networks, International journal of engineering research & technology(IJERT) Volume 08, Issue 05 (May 2019),
[7] C. Liu, H. Pang and N. Cao, "Research on Time Synchronization Technology of Wireless Sensor Network," *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2017, pp. 391-394, doi: 10.1109/CyberC.2017.67.
[8] Cox, D., Jovanov, E., Milenkovic, A., "Time Synchronization for ZigBee Networks.", System Theory, SSST '05. Proceedings of the Thirty-Seventh Southwestern Symposium p. 135 – 138
[9] Sitharama Iyengar-Time Synchronization in Wireless Sensor Network based on Level Discovery Phase-International Journal of Computer Application
[10] Maroti, M., Kusy, B., Simon, G., Ledeczi, A. "The Time Synchronization Protocol.", Proc. Of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys).
[11] Aakvaag, N., Mathiesen, M., Thonet, G. "Timing and Power Issues in Wireless Sensor Networks - An industrial Test Case.", Parallel Processing. ICPP 2005 Workshops Page(s):419 - 426.
[12] Michael Roche, Time Synchronization in Wireless Networks.
[13] Linh-An Phan, Taejoon Kim, Taehong Kim, JaeSeang Lee, Jae-Hyun Ham - Performance Analysis of Time Synchronization Protocols in Wireless Sensor Networks- Published: 9 July 2019
[14] Shanta Rangasamy, Sourabha Murthy, Rashtreeya vidyalaya college of engineering – An overview of network time protocol, luly 2021.
[15] Ionos digital guide, Simple network time protocol: the stripped-back protocol for time synchronization, 24.09.18.
[16] Dell technologies, Timing and Synchronization in Networks - Introduction to Precision Time Protocol and Synchronous Ethernet.
[17] Paul Krzyzanowski, Precision time protocol High precision clock synchronization that computes latency and offset. February 15,2021.
[18] Sanath kumar nadiger, senior software engineer, Precision time protocol-Debugging in Ethernet driver.
[19] Arun Karri, AB Volvo,Ethernet time synchronization over NTP vs PTP, January 2021.