

Synchronizing smart city nodes using Skew Integrated Timestamp (SIT)

Muhammad Usman Hashmi¹, Muntazir Hussain², Asghar Ali Shah¹,
Muhammad Babar³ and Basit Qureshi⁴

¹ Department of Computer Science, Bahria University, Islamabad, Islamabad, Islamabad, Pakistan

² Department of Electrical and Computer Engineering, Air University Islamabad, Islamabad, Islamabad, Pakistan

³ Robotics and Internet of Things Lab, Prince Sultan University (PSU), Riyadh, Riyadh, Saudi Arabia

⁴ College of Computer and Information Sciences, Prince Sultan University, Riyadh, Riyadh, Saudi Arabia

ABSTRACT

Time synchronization among smart city nodes is critical for proper functioning and coordinating various smart city systems and applications. It ensures that different devices and systems in the smart city network are synchronized and all the data generated by these devices is consistent and accurate. Synchronization methods in smart cities use multiple timestamp exchanges for time skew correction. The Skew Integrated Timestamp (SIT) proposed here uses a timestamp, which has time skew calculated from the physical layer and uses just one timestamp to synchronize. The result from the experiment suggests that SIT can be used in place of multiple timestamp exchanges, which saves computational resources and energy.

Subjects Computer Networks and Communications, Emerging Technologies

Keywords Smart cities, Time synchronization, Skew correction, Time offset

INTRODUCTION

A smart city uses state-of-the-art technology for the improvement and enhancement of various urban services and provides a better lifestyle to the residents. Smart city technology typically includes the Internet of Things (IoT), which enables devices and systems to collect and share data in real-time. In smart cities advanced analytics and machine learning algorithms helps to optimize decision-making (*Hassan et al., 2021; Arasteh et al., 2016; Ghazal et al., 2023; Ghazal et al., 2021; Koubaa et al., 2020; Babar et al., 2021; Umer et al., 2023*). IoT forms the backbone of many smart city initiatives by connecting various devices, sensors, and systems to gather data, facilitate communication, and enable intelligent decision-making.

The Internet of Things (IoT) uses smart devices to collect data from various sources in a smart city, including traffic sensors, environmental sensors, smart meters, surveillance cameras, etc., (*Gaur et al., 2015; Aditya et al., 2023; Bellini, Nesi & Pantaleo, 2022; Poongodi et al., 2021; Babar, Arif & Irfan, 2019; Alshathri et al., 2023; Pranto et al., 2021*). These smart devices generate data communicated over the network, which needs to be processed and analyzed in real-time. The IoT technology integrates various city systems, e.g., energy, water, transportation, and waste management. It connects people to infrastructure and even vehicles. Data collected from these various IoT devices can be analyzed by city

Submitted 31 August 2023

Accepted 19 April 2024

Published 24 May 2024

Corresponding author
Muntazir Hussain,
muntazir_hussain14@yahoo.com

Academic editor
Ahyoung Lee

Additional Information and
Declarations can be found on
page 14

DOI 10.7717/peerj-cs.2049

© Copyright
2024 Hashmi et al.

Distributed under
Creative Commons CC-BY 4.0

OPEN ACCESS

administrators, who can make guided decisions to optimize the allocation of resources. In this way, city administrators can improve their operational efficiency. The use of applications and services by smart city citizens allows them to remain updated in real-time. Their participation through their response to the community is increased and encouraged. Citizens have updated information about public services, and they can actively contribute to development and management by providing feedback. Smart cities aim to enhance the quality of life in various ways, e.g., smart parks can sense air quality and inform the citizens. Hence, IoT is a main technology that forms the basis of smart cities as IoT enables data collection and analysis to create more sustainable and livable urban environments (Bellini, Nesi & Pantaleo, 2022; Cvar et al., 2020; Hassan et al., 2021).

IoT-based smart city offers various benefits, for instance, improved efficiency and quality of life, better resource management, economic development, and increased sustainability. To achieve these benefits coordination between IoT nodes is required (Arasteh et al., 2016; Gaur et al., 2015; Ghazal et al., 2023; Ali et al., 2023; Cvar et al., 2020; Hassan et al., 2021; Saba, Sadad & Hong, 2022). IoT nodes are smart devices typically equipped with sensing, communication capabilities, and computing power to collect and exchange data with other smart devices within a smart city environment. When these IoT nodes are synchronized in time, they have the capability to timestamp events, providing a chronological reference for a more comprehensive understanding of the overall scenario. Synchronized nodes work together in a coordinated and efficient manner (Yiğitler, Badihi & Jäntti, 2020; Tirado-Andrés, Rozas & Araujo, 2019; Hashmi et al., 2023; Hashmi et al., 2021a; Hashmi et al., 2021b). Time synchronization enhances predictive maintenance, disaster response, and real-time decision-making, creating a more sustainable and efficient urban environment. Time synchronization is crucial for effectively coordinating IoT nodes in a smart city (Fan et al., 2019; Skiadopoulos et al., 2019; Fan et al., 2022).

There are several solutions to synchronization problem (Yiğitler, Badihi & Jäntti, 2020; Tirado-Andrés, Rozas & Araujo, 2019; Yoo, Park & Lee, 2020; Skiadopoulos et al., 2019), including the Network Time Protocol (NTP), Precision Time Protocol (PTP), Global Navigation Satellite Systems (GNSS), and the Distributed Time Synchronization Protocol (DTSP). Based on the available information and the author's understanding, every method uses several exchanges of timestamps for the estimation of clock frequency offset (skew) among synchronizing nodes. The SIT method proposed here requires one timestamp and incorporates skew correction information with a time phase value. Reducing timestamps increases the energy efficiency of smart cities, which can bring several benefits, including cost savings, environmental benefits, increased reliability and comfort, enhanced resilience, and economic growth.

The organization of the article is made in six sections. 'Literature Review' elaborates on the literature available for time synchronization among smart city nodes. 'Methodology' describes the proposed Skew Integrated Timestamp (SIT) synchronization methodology. 'Mathematical Analysis' provides mathematical analysis and details. 'Experimentation and Results' discusses the experimental details and results, showing that skew can be corrected with a single timestamp by using the SIT method. In light of the results, 'Conclusions' concludes that the proposed SIT method can lead to efficient time synchronization.

LITERATURE REVIEW

A synchronization error of a clock occurs when the clock's time is not aligned with the actual time. In a smart city network, every node's clock can drift away in time due to various factors, including clock skew, network delays or even temperature variations. Overall network efficiency is reduced because of non-synchronized nodes resulting in incorrect timing of events. Synchronization errors can be prevented by periodically synchronizing the whole network using any reliable time synchronization method. Various time synchronization mechanisms are in place for smart cities. The Network Time Protocol (NTP) (Mills et al., 2010), Precision Time Protocol (PTP) (Eidson, Fischer & White, 2002), Flooding Time Synchronization Protocol (FTSP) (Maróti et al., 2004), Blockchain-based time synchronization (Fan et al., 2019; Fan et al., 2022) and the Distributed Time Synchronization Protocol (DTSP) are few state of the art methods.

Network Time Protocol (NTP) is a widely used protocol for time synchronization of nodes in many networks, including smart city network (Mills et al., 2010; Hou et al., 2022; Sommars, 2017). NTP creates a hierarchical structure of nodes for synchronization. A tree-like structure of smart city nodes is created. The highest-level nodes have master clocks (nodes with external time references such as atomic clocks or GPS). Lower-level nodes need to synchronize with the higher-level nodes. With NTP, millisecond accuracy can be achieved by using a combination of intersection and clustering algorithms, internal clocks, and master clocks.

The Precision Time Protocol (PTP), also known as IEEE 1588, uses a peer-to-peer architecture to achieve microsecond accuracy (Eidson, Fischer & White, 2002; Watt et al., 2015; Resner, Fröhlich & Wanner, 2016). It is considered a more advanced protocol that uses timestamps and delay measurement techniques. PTP architecture is master-slave architecture: the master clock sends time synchronization messages to the slave clocks. Upon receiving of synchronization messages, the slave clocks synchronize their time by adjusting the clock phase and skew. PTPv1 and PTPv2 are two versions of PTP. PTPv2 is more promising with better accuracy if network infrastructure and configuration are designed to minimize network delays.

The Flooding Time Synchronization Protocol (FTSP) is a widely deployed protocol in wireless sensor networks (WSNs) and also has its application in smart cities (Maróti et al., 2004; Gheorghe, Rughinis & Tapus, 2010). FTSP is a distributed protocol central time server or clock. Flooding of time synchronization messages is done periodically among neighboring sensor nodes. Each node broadcasts its local time, and neighbors adjust their clocks based on the received timestamp. The protocol employs techniques such as timestamp re-synchronization and clock rate estimation to mitigate clock skew and achieve synchronization accuracy.

Blockchain-based time synchronization refers to utilizing blockchain technology to establish a trusted and decentralized mechanism for synchronizing time among various devices and systems in a smart city environment (Fan et al., 2019; Fan et al., 2022). Traditional time synchronization methods, such as NTP, PTP, and FTSP, rely on centralized time servers or hierarchical structures, which may introduce vulnerabilities

or single points of failure. Instead of relying on a central time server, blockchain-based time synchronization distributes the time authority across multiple nodes in the network. These nodes reach a consensus on the accurate time through a consensus mechanism. Each event or transaction in the smart city network is assigned a timestamp recorded on the blockchain. The timestamp represents the agreed-upon time established by the distributed time authority. This allows for the chronological ordering and verification of events across the network. Once a timestamp is recorded on the blockchain, it becomes immutable and transparent. It cannot be altered retroactively, providing a tamper-proof record of time synchronization. This enhances the integrity and trustworthiness of the time information in the smart city network.

The Global Navigation Satellite System (GNSS) is a constellation of satellites that provide positioning, navigation, and timing information to users on Earth (*Zaidi & Suddle, 2006*). The most well-known GNSS system is the Global Positioning System (GPS) operated by the United States, but there are other systems such as GLONASS (Russia), Galileo (European Union), and BeiDou (China). In the context of smart cities, GNSS plays a crucial role in enabling location-based services, accurate positioning, and synchronization of devices and systems. GNSS/GPS provides highly accurate time signals, which are essential for time synchronization in smart city networks (*Zaidi & Suddle, 2006; Dana, 1997*). Devices can receive GNSS signals to synchronize their internal clocks, ensuring coordination and accurate timing in applications. However, GNSS signals can be affected by environmental factors like tall buildings, dense urban environments, or signal interference. Advancements in GNSS technology, including multi-constellation support and enhanced signal processing techniques, are addressing these challenges and improving accuracy in urban areas.

Any of the above-mentioned methods can be used for time synchronization in smart cities. However, they all require several exchanges of timestamps for the estimation of skew. The proposed SIT method uses one timestamp to estimate skew.

The main contributions of this article are:

1. In contrast to the existing methods (*Maróti et al., 2004; Gheorghe, Rughinis & Tapus, 2010; Zaidi & Suddle, 2006; Dana, 1997; Fan et al., 2019; Fan et al., 2022*), the proposed Skew Integrated Timestamp technique is more efficient. Because the existing methods, for instance, (*Maróti et al., 2004; Gheorghe, Rughinis & Tapus, 2010; Zaidi & Suddle, 2006; Dana, 1997; Fan et al., 2019; Fan et al., 2022*), require several exchanges of timestamps for the estimation of skew. Whereas the proposed SIT method uses one timestamp to estimate skew.
2. Energy efficiency is improved for the reason that multiple timestamps are no longer required to extract the clock skew of the application layer clock.
3. SIT method is experimentally implemented to show that skew correction can be done using one timestamp. TMS320C6713 (Texas Instruments, Dallas, TX, USA) are DSP Starter Kits (DSKs) and are used as nodes in the experiment transmitting node A (DSK A) and receiving node B (DSK B).

The following section shows the details of the proposed methodology.

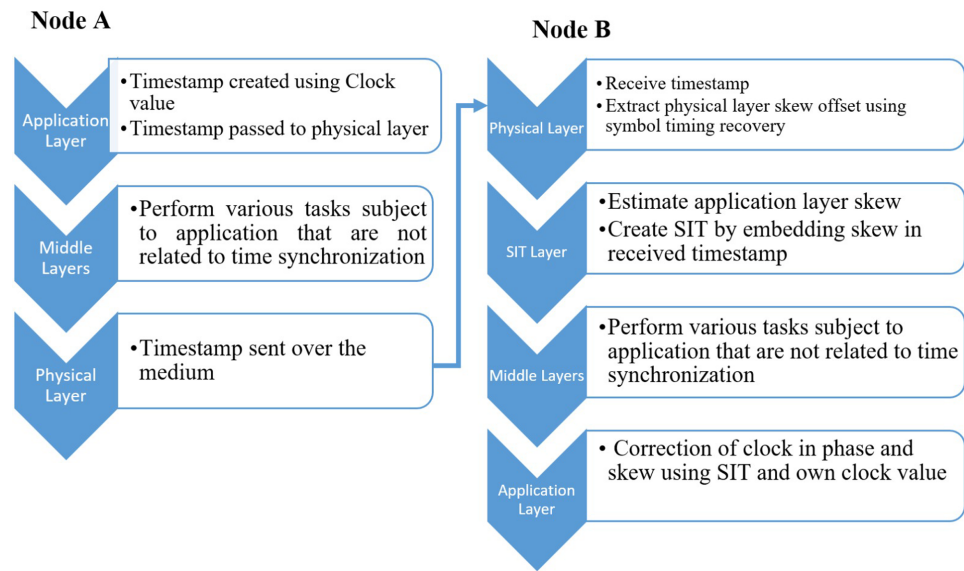


Figure 1 Node synchronization using SIT.

Full-size DOI: [10.7717/peerjcs.2049/fig-1](https://doi.org/10.7717/peerjcs.2049/fig-1)

METHODOLOGY

Application layer clocks typically function as timers, tracking the number of oscillations of a crystal and utilizing two registers to determine the equivalence between the oscillations and a single tick of clock running at application layer (Sundararaman, Buy & Kshemkalyani, 2005). Additionally, the clock running at the application layer and the clock responsible for symbol timing synchronization at the physical layer of the same node also rely on counting the oscillations of the quartz crystal of the hardware clock. To elaborate on how the proposed Skew Integrated Timestamp (SIT) method works, consider that two nodes (Node A and B) need time synchronization as shown in Fig. 1. Node A acts as a node with accurate time information and node B needs to be aligned with node A by correcting its time phase and skew.

Steps in node synchronization using SIT

The steps of the method are as follows.

1. Node A will create a timestamp using its application layer master clock value and transfer it to receiving node B.
2. Node B will receive the timestamp and implement symbol timing recovery at its physical layer for the estimation of the skew.
3. This computed skew of the physical layer is used to estimate the application layer skew. **Since both the physical and application layer clocks are derived from the hardware clock within the same node, it is essential for their skew to be identical.**
4. Now estimated application layer skew is embedded within the received timestamp, creating the Skew Integrated Timestamp (SIT).

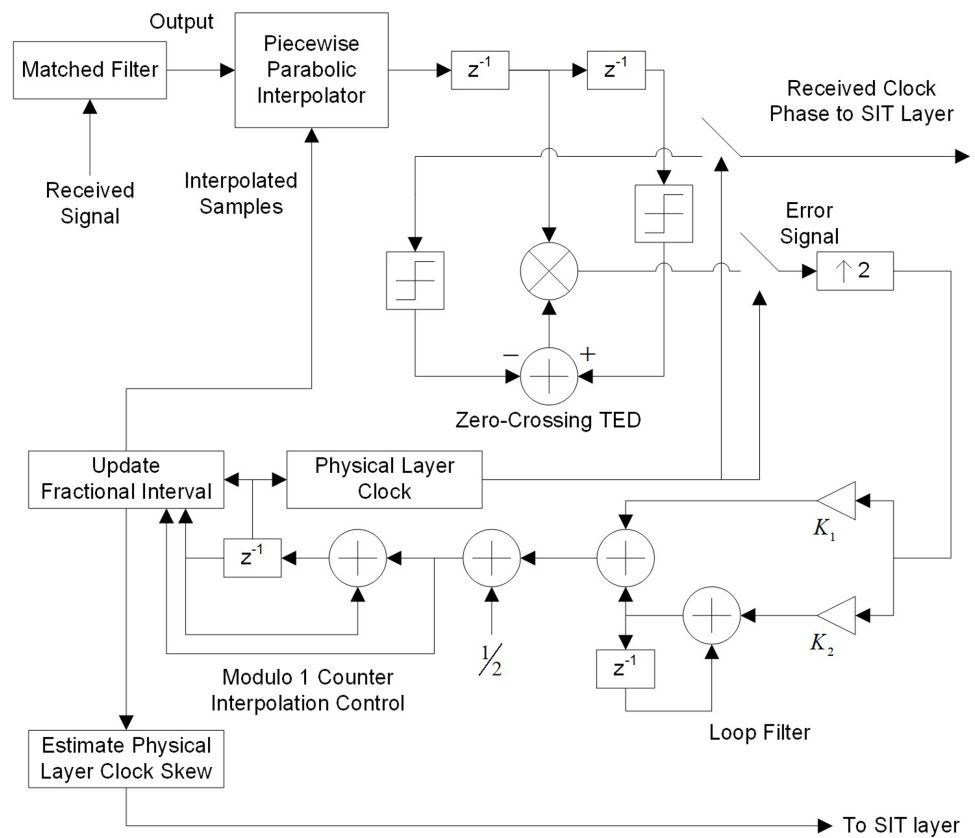


Figure 2 Symbol timing recovery for skew estimate.

Full-size DOI: [10.7717/peerjcs.2049/fig-2](https://doi.org/10.7717/peerjcs.2049/fig-2)

5. SIT will now be used by the application layer of node B to correct its phase and skew offset.
6. Node A and Node B get synchronized in time by using one timestamp by inserting SIT method at receiving node B.

[Figure 1](#) shows the process of synchronization among two nodes using SIT.

To further elaborate the steps involved in receiving node B symbol timing recovery are discussed next. Detailed discussions of various synchronization systems at the physical layer can be found in the works of [Rice \(2008\)](#) and [Meyr, Moeneclaey & Fechtel \(1998\)](#). These synchronization systems have different modulation schemes, implementation of various interpolators, special types of timing error detectors, multiple loop filters, and diverse interpolation controls. [Figure 2](#) presents one such system, modified for SIT.

Physical layer configuration for skew estimate

The physical layer configuration of the receiving node is as follows,

- Modulation scheme: Binary Pulse Amplitude Modulation (PAM).
- Interpolator: Piecewise Parabolic.
- Timing Error Detector: Zero Crossing Timing Error Detector (TED).
- Loop Filter: Proportional plus Integrator.

- Interpolation Control: Mod-1 Counter.

This system effectively traces and amends for phase and frequency errors (skew) in the physical layer clock.

MATHEMATICAL ANALYSIS

The received signal at node B is passed to matched filter which processes the signal. The processed outcome $x(nT)$ is fed to piecewise parabolic interpolator for p -th interpolant and can be written as,

$$x((m(p) + \mu(p))T) = \left(\frac{\mu(p)^3}{6} - \frac{\mu(p)}{6} \right) x((m(p) + 2)T) - \left(\frac{\mu(p)^3}{2} - \frac{\mu(p)^2}{2} - \mu(p) \right) x((m(p) + 1)T) + \left(\frac{\mu(p)^3}{2} - \mu(p)^2 - \frac{\mu(p)}{2} + 1 \right) x(m(p)T) - \left(\frac{\mu(p)^3}{6} - \frac{\mu(p)^2}{2} + \frac{\mu(p)}{3} \right) x((m(p) - 1)T). \quad (1)$$

Here, p is the p -th interpolant which is the new sample which reduces the error margin each time. The value is between $+1$ and -1 . $m(p)$ represents the basepoint index, and $\mu(p)$ is the fractional change. This is the output of the Mod-1 counter which will be used by the interpolator for the computation of the next interpolant. Zero Crossing TED attempts to adopt the signal of interpolator for tracking of error and when perfectly aligned, it produces a zero error (Rice, 2008). The error signal $e(p)$ in p -th interpolant can be determined from TED by Eq. (2).

$$e(p) = x((p - 1/2)T_s + \tau) [sgnx((p - 1)T_s + \tau) - sgnx(pT_s + \tau)], \quad (2)$$

where $sgnx$ represents the sign of x . The loop filter, with constant coefficients K_1 and K_2 , generates an output that is passed to the Mod-1 counter to estimate $\mu(p)$ and $m(p)$. K_1 and K_2 is carefully calculated using sample time T , symbol time T_s and noise bandwidth (Rice, 2008). The performance of the synchronization system in terms of tracking and acquisition time relies on these loop filter parameters. The Mod-1 counter takes advantage of the signal from loop filter $v(n)$, to determine $m(p)$ and $\mu(p)$. The values of Mod-1 counter η are computed as,

$$\eta(n+1) = (\eta(n) - T/T_s - v(n)) \quad (3)$$

when η underflows, n is equal to basepoint index $m(p)$. To find the fractional interval at the basepoint index, Eq. (4) is used,

$$\mu(m(p)) = \eta(m(p)) / (T/T_s + v(m(p))). \quad (4)$$

The values of $\mu(m(p))$ and $m(p)$ are then employed to evaluate the value of the upcoming interpolant.

By employing the aforementioned iterative mechanism, the variations of $\mu(p)$ can be used to estimate skew. Various mechanisms for computing the skew are described in the works of Rice (2008), Mengali & D'Andrea (1997) and Meyr, Moeneclaey & Fechtel (1998).

Rice (2008) proposes that skew ξ_P is the slope of the fractional change that can be computed using any method including least square. Now the new sampling rate f_s can be written with the help of symbol rate f_y as Eq. (5), to achieve symbol timing recovery at the physical layer.

$$f_s = (2 + \xi)f_y. \quad (5)$$

A simple calculation of skew can be done by making a line between the first point $k=1$ and the last point $k=p$ of $\mu(k)$ as given by using Eq. (6).

$$\xi_P = \frac{\mu(p) - \mu(1)}{p - 1}. \quad (6)$$

SIT layer is next responsible for mapping physical layer clock skew ξ_P to application layer clock skew ξ_A . Keeping the fact in mind that the hardware clock is driving both clocks of the physical and application layer. This means that ξ_P is actually the skew of hardware clock ξ_H and ξ_H must be the same as of ξ_A can be related by Eq. (7) and as shown by the experimentation results in experiment section.

$$\xi_H = \xi_P = \xi_A. \quad (7)$$

Now ξ_A is appended in the timestamp received with the sender's clock phase value $\phi(S)$ and passed to the application layer of receiving node B. This appended timestamp is the Skew Integrated Timestamp (SIT). This SIT is now used at the application layer for rectifying application layer clock skew using ξ_A and also phase such that the clock value of the node B $\phi(R)$ is available at the application layer and can be used to compute phase offset as $\phi(R) - \phi(S)$ and synchronize in phase along with skew. Now Node A and Node B clocks are synchronized using one timestamp by implementing SIT at Node B.

EXPERIMENTATION AND RESULTS

An experimentation setup is prepared to prove that the physical layer clock skew and application layer clock skew are the same since both are derived from the same hardware clock within the node. Once the fact is established, the SIT method can be used to synchronize the two nodes using one timestamp.

Experimental objectives

The objectives of the experiment are as follows:

1. Verify equivalence of physical and application layer clock skew: Demonstrate that the physical and application layers clock skew are identical, since both the physical and application layer clocks are derived from the hardware clock within the same node.
2. Establish the basis for synchronization using the SIT method: Once the equivalence of clock skew is achieved, then the SIT method is employed to synchronize two nodes using a single timestamp.
3. Utilize TMS320C6713 DSP Starter Kits: Implement the experiment using TMS320C6713 DSP Starter Kits (DSKs) (Texas Instruments, Dallas, TX, USA) as the transmitting node A (DSK A) and receiving node B (DSK B).

4. Include physical and application layer synchronization systems: Set up both a physical layer synchronization system and an application layer synchronization system to ensure a comprehensive evaluation.
 5. Designate DSK A clock as the reference: Consider the clock of DSK A as having accurate clock information, and aim to synchronize DSK B with the clock information from DSK A.
 6. Configure DSKs for implementation of communication system: Configure DSK A as transmitter and DSK B as receiver in order to implement communication system.
- These objectives collectively aim to validate the synchronization approach.

Experimental configuration

The physical layer configuration of used DSKs are as follows:

1. Configuration of physical layer of DSK A for transmission: Configuration of the physical layer with specific parameters required for communication system are as follows:
 - Modulation: Binary PAM
 - Symbols: +1, -1
 - Transmitted symbols: 10,000
 - Symbol rate: 4,000 symbols/s
 - Sampling rate: 16,000 samples/s
 - Samples/Symbol: 4
 - Samples: 40,000
 - Transmission time: 2.5 s
 - Upsampling factor: 4
 - Pulse Shaping filter: Square Root Raised Cosine (SRRC) with 50% excess bandwidth.
2. Configuration of physical layer of DSK B for reception: At the receiving end (DSK B), initiate the physical layer processing using the symbol timing recovery system, mirroring the configuration outlined in the experiment. Additional physical layer configuration and details are as follows:
 - Received symbols undergo matched filtering with SRRC.
 - Downsample to two samples/symbol.
 - Apply a piecewise parabolic interpolator to downsampled symbols.
 - Pass interpolated symbols through a Zero Crossing TED.
 - Use the error signal from TED in the proportional plus integrator loop filter.
 - Utilize the loop filter output for Mod-1 counter interpolation control.
 - The loop filter parameters required for efficient acquisition time and tracking time is set using noise bandwidth = $1/\sqrt{2}$, $K_p = 2.7$, $K_0 = -1$, and $N = 2$. Where K_p and K_0 are used for loop gain calculations K_1 and K_2 .

Experiment methodology

System presented in Fig. 2 is implemented using the configuration discussed above. The error signal and fractional interval are obtained and depicted in Fig. 3. The fractional

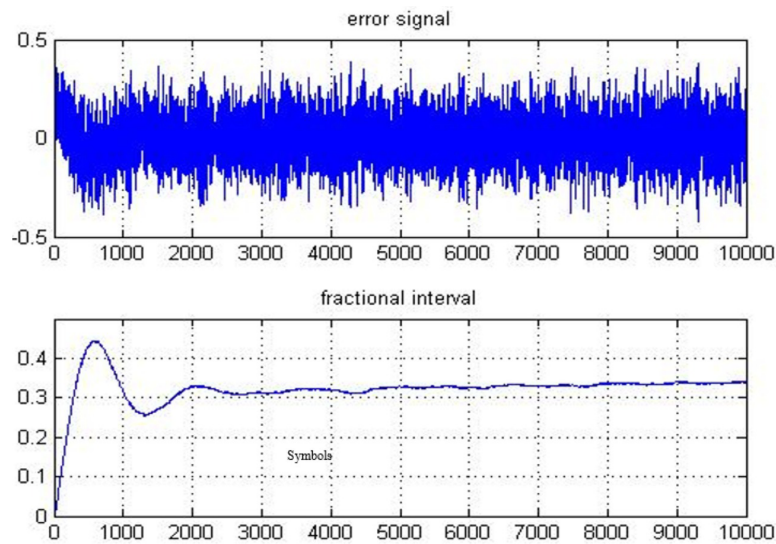


Figure 3 Error signal and fractional interval examined for received symbols.

Full-size [DOI: 10.7717/peerjcs.2049/fig-3](https://doi.org/10.7717/peerjcs.2049/fig-3)

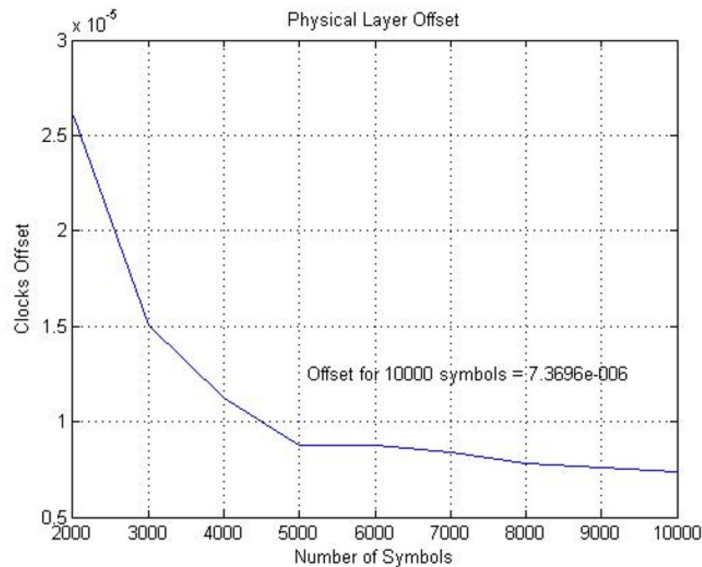


Figure 4 Physical layer skew computed from fractional interval.

Full-size [DOI: 10.7717/peerjcs.2049/fig-4](https://doi.org/10.7717/peerjcs.2049/fig-4)

interval slope is the skew among DSK A and B clocks. The least squares (LS) estimation approach is employed to determine the slope. The accuracy of the frequency offset estimate improves with an increasing number of transmitted symbols, as demonstrated in Fig. 4.

On application of LS on the fractional intervals with various transmitted symbols, ranging from 2,000 to 10,000, gives the skew estimate using a specified number of symbols. The results show that the estimate of the skew becomes more accurate with a larger number

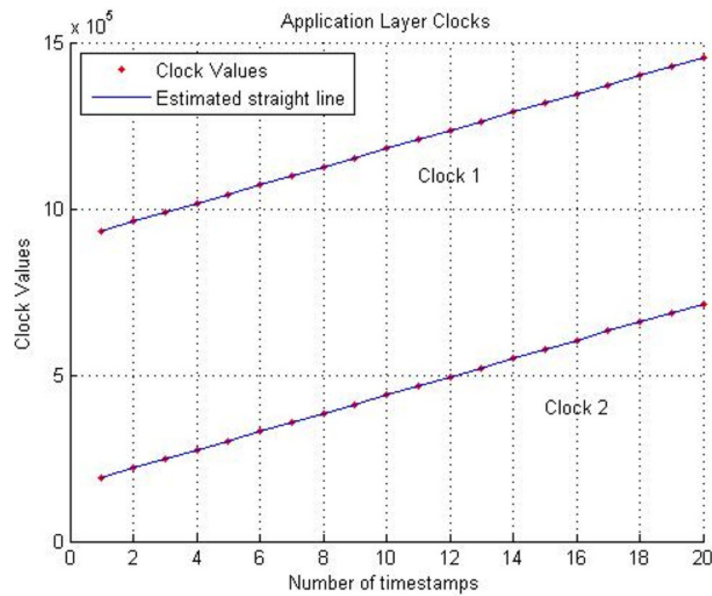


Figure 5 Application layer clocks. Clock 1:DSK A and Clock 2:DSK B.

Full-size DOI: [10.7717/peerjcs.2049/fig-5](https://doi.org/10.7717/peerjcs.2049/fig-5)

of transmitted symbols. Over 10,000 symbols that were originally transmitted from DSK A, the estimated physical layer clock skew is determined to be 7.3696 ppm.

For the skew estimation of the application layer clock of the same DSKs that are used above, multiple timestamps are exchanged. DSK A transmits twenty timestamps over the same transmission time *i.e.*, 2.5 s. DSK B records their respective clock values upon reception of timestamps. This synchronization mechanism at the application layer uses LS estimation on these timestamps for the computation of the skew. In Fig. 5, “Clock 1” belongs to DSK A, and “Clock 2” belongs to DSK B.

LS estimation method is applied to values of Clock 1 and Clock 2. The slope of the line representing Clock 1 and Clock 2 is the frequency of DSK A’s clock and DSK B’s clock. The offset among slopes represents the skew (frequency offset) between Clock 1 and Clock 2. Figure 6 illustrates the application layer skew computed against a different number of timestamps. It is observed that increasing the number of timestamps enhances the accuracy of the computed skew. For twenty timestamps, the skew is determined to be 7.3637 ppm, which is approximately identical to the physical layer clock skew of 7.3696 ppm.

Figure 7 provides a comparison between the skew (frequency offset) at the application and physical layer for 2.5 s of transmission time. Twenty transmitted timestamps correspond to 2.5 s of transmission time and with 4,000 symbols/s, the transmission of 10,000 symbols also corresponds to 2.5 s of physical layer communication. From Fig. 7, it can be observed that the clock skew at both layers converges after one second of transmission, equivalent to the transmission of 4,000 symbols. This demonstrates that the skew is the same at both layers as expected.

SIT exploits this fact established from the above experiment and uses one timestamp, instead of multiple. The process of synchronization among two nodes using SIT method is

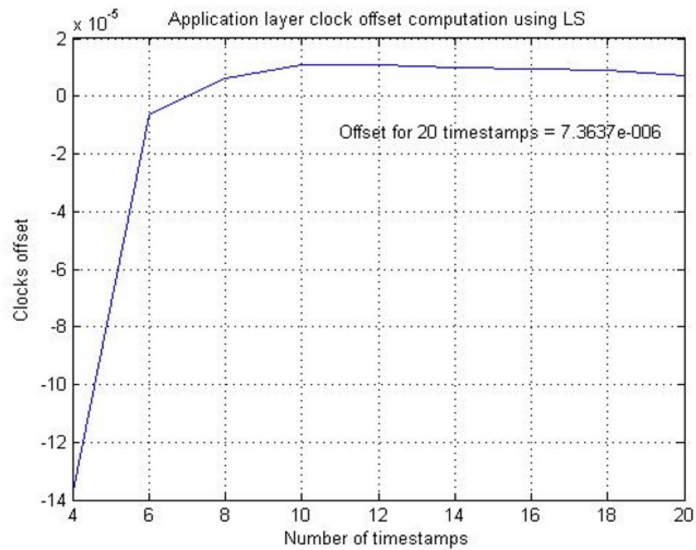


Figure 6 Application layer skew computed using various timestamps.

Full-size [DOI: 10.7717/peerjcs.2049/fig-6](https://doi.org/10.7717/peerjcs.2049/fig-6)

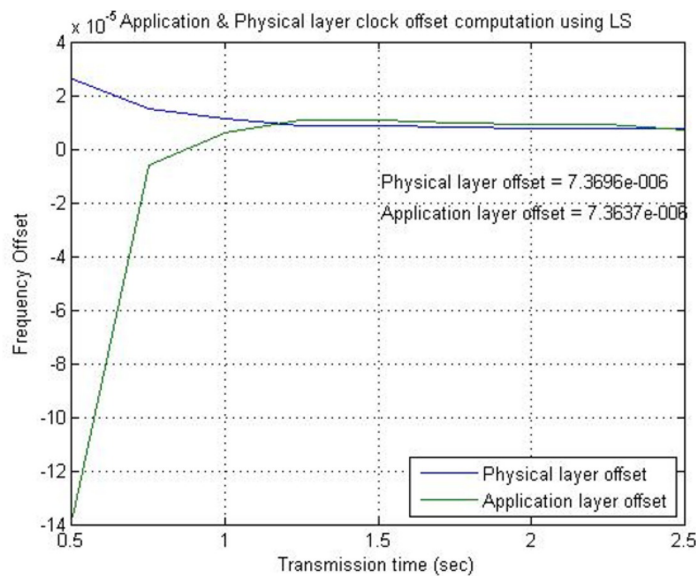


Figure 7 Skew (frequency offset) comparison of physical layer clock and application layer clock.

Full-size [DOI: 10.7717/peerjcs.2049/fig-7](https://doi.org/10.7717/peerjcs.2049/fig-7)

explained in Figs. 1 and 2. The following procedure is followed in SIT to synchronize DSK A and DSK B:

- DSK A creates a timestamp with its application layer clock value 23.
- Timestamp is converted to symbols and transmitted over the medium to DSK B.
- Physical layer of Node B receives the symbols and passes them from the system model shown in Fig. 2.

- Physical layer skew of node B is computed to be 7.3696 ppm and passed to SIT layer and symbols are converted back to received timestamp value 23.
- SIT layer equates the application layer clock skew with the physical layer clock skew *i.e.*, 7.3696 ppm.
- SIT layer creates SIT by appending the received timestamp value with application layer skew. SIT has two values 23 and 7.3696 ppm and passes them to the application layer.
- Node B application layer clock value is 59 which is used to compute clock phase offset as $59 - 23 = 36$. By knowing the phase offset Node A and Node B synchronizes in the clock phase.
- Node B uses SIT (skew 7.3696 ppm) to correct application layer clock skew and hence synchronize in frequency.
- By using SIT method, both nodes are synchronized in clock phase and frequency by using just one-timestamp.

Number of symbols required to estimate skew at the application layer have a direct relation with accuracy. If these symbols are below a certain number then precise synchronization may not be achieved which may be a requirement in some areas of smart city applications.

CONCLUSIONS

The Skew Integrated Timestamp (SIT) method presented in this article offers a highly promising approach to enhance the efficiency of smart city networks. Through experimentation and demonstrated results, it has been shown that the SIT approach significantly reduces the resources needed for distributing multiple timestamps. By implementing SIT at the receiving node, a single timestamp is sufficient to synchronize two nodes in time, resulting in computational and energy savings. SIT proves to be particularly advantageous for nodes with limited battery power in smart cities, as it can be applied to all synchronization-dependent nodes. Its utilization among all nodes can exponentially improve the overall energy efficiency of smart city networks. In comparison to existing methods, the proposed SIT technique stands out for its efficiency. While existing methods require multiple exchanges of timestamps to estimate skew, the SIT method accomplishes this with just one timestamp. This streamlined process eliminates the need for multiple timestamps to extract the clock skew of the application layer clock, further enhancing energy efficiency. The experimental implementation of the SIT method confirms that skew correction can indeed be achieved using a single timestamp. Overall, the findings suggest that the SIT method holds great promise for optimizing the efficiency and performance of smart city networks, making it a compelling solution for synchronization challenges.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Prince Sultan University paid the Article Processing Charges (APC) for this publication. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
Prince Sultan University paid the Article Processing Charges (APC).

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Muhammad Usman Hashmi conceived and designed the experiments, prepared figures and/or tables, and approved the final draft.
- Muntazir Hussain conceived and designed the experiments, performed the experiments, prepared figures and/or tables, and approved the final draft.
- Asghar Ali Shah analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Muhammad Babar performed the computation work, authored or reviewed drafts of the article, and approved the final draft.
- Basit Qureshi analyzed the data, performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:
The raw data and code are available in the [Supplemental Files](#).

Supplemental Information

Supplemental information for this article can be found online at <http://dx.doi.org/10.7717/peerj-cs.2049#supplemental-information>.

REFERENCES

- Aditya A, Anwarul S, Tanwar R, Koneru SKV. 2023.** An IoT assisted Intelligent Parking System (IPS) for smart cities. *Procedia Computer Science* **218**:1045–1054 DOI [10.1016/j.procs.2023.01.084](https://doi.org/10.1016/j.procs.2023.01.084).
- Ali Z, Mahmood A, Khatoon S, Alhakami W, Ullah SS, Iqbal J, Hussain S. 2023.** A Generic Internet of Things (IoT) middleware for smart city applications. *Sustainability* **15**(1):743 DOI [10.3390/su15010743](https://doi.org/10.3390/su15010743).
- Alshathri S, Hemdan EED, El-Shafai W, Sayed A. 2023.** Digital twin-based automated fault diagnosis in industrial IoT applications. *Computers, Materials & Continua* **75**(1):183–196 DOI [10.32604/cmc.2023.034048](https://doi.org/10.32604/cmc.2023.034048).

- Arasteh H, Hosseinneshad V, Loia V, Tommasetti A, Troisi O, Shafie-khah M, Siano P.** 2016. IoT-based smart cities: a survey. In: *2016 IEEE 16th international conference on environment and electrical engineering (EEEIC)*. Piscataway: IEEE, 1–6.
- Babar M, Arif F, Irfan M.** 2019. Internet of things–based smart city environments using big data analytics: a survey. *Recent Trends and Advances in Wireless and IoT-Enabled Networks* 129–138.
- Babar M, Khattak AS, Jan MA, Tariq MU.** 2021. Energy aware smart city management system using data analytics and Internet of Things. *Sustainable Energy Technologies and Assessments* 44:100992 DOI 10.1016/j.seta.2021.100992.
- Bellini P, Nesi P, Pantaleo G.** 2022. IoT-enabled smart cities: a review of concepts, frameworks and key technologies. *Applied Sciences* 12(3):1607 DOI 10.3390/app12031607.
- Cvar N, Trilar J, Kos A, Volk M, Stojmenova Duh E.** 2020. The use of IoT technology in smart cities and smart villages: similarities, differences, and future prospects. *Sensors* 20(14):3897 DOI 10.3390/s20143897.
- Dana PH.** 1997. Global Positioning System (GPS) time dissemination for real-time applications. *Real-Time Systems* 12(1):9–40 DOI 10.1023/A:1007906014916.
- Eidson JC, Fischer M, White J.** 2002. IEEE-1588™ Standard for a precision clock synchronization protocol for networked measurement and control systems. In: *Proceedings of the 34th annual precise time and time interval systems and applications meeting*. Piscataway: IEEE, 243–254.
- Fan K, Shi Z, Su R, Bai Y, Huang P, Zhang K, Li H, Yang Y.** 2022. Blockchain-based trust management for verifiable time synchronization service in IoT. *Peer-To-Peer Networking and Applications* 15(2):1152–1162 DOI 10.1007/s12083-021-01276-2.
- Fan K, Sun S, Yan Z, Pan Q, Li H, Yang Y.** 2019. A blockchain-based clock synchronization scheme in IoT. *Future Generation Computer Systems* 101:524–533 DOI 10.1016/j.future.2019.06.007.
- Gaur A, Scotney B, Parr G, McClean S.** 2015. Smart city architecture and its applications based on IoT. *Procedia Computer Science* 52:1089–1094 DOI 10.1016/j.procs.2015.05.122.
- Ghazal TM, Hasan MK, Alshurideh MT, Alzoubi HM, Ahmad M, Akbar SS, Kurdi BA, Akour IA.** 2021. IoT for smart cities: machine learning approaches in smart healthcare—a review. *Future Internet* 13(8):218 DOI 10.3390/fi13080218.
- Ghazal TM, Hasan MK, Alzoubi HM, Alshurideh M, Ahmad M, Akbar SS.** 2023. Internet of things connected wireless sensor networks for smart cities. In: *The effect of information technology on business and marketing intelligence systems*. Cham: Springer International Publishing, 1953–1968.
- Gheorghe L, Rughinis R, Tapus N.** 2010. Fault-tolerant flooding time synchronization protocol for wireless sensor networks. In: *2010 Sixth international conference on networking and services*. Piscataway: IEEE, 143–149.

- Hashmi SU, Hussain M, Arshad SN, Inayat K, Hwang SO. 2021a.** Energy efficient cross layer time synchronisation in cognitive radio networks. *International Journal of Internet Technology and Secured Transactions* 11(4):329–340 DOI 10.1504/IJITST.2021.116743.
- Hashmi MU, Hussain M, Babar M, Qureshi B. 2023.** Single-Timestamp Skew Correction (STSC) in V2X networks. *Electronics* 12(6):1276 DOI 10.3390/electronics12061276.
- Hashmi SU, Hussain M, Muslim FB, Inayat K, Hwang SO. 2021b.** Implementation of symbol timing recovery for estimation of clock skew. *International Journal of Internet Technology and Secured Transactions* 11(3):241–268 DOI 10.1504/IJITST.2021.114928.
- Hassan RJ, Zeebaree SR, Ameen SY, Kak SF, Sadeeq MA, Ageed ZS, AL-Zebari A, Salih AA. 2021.** State of art survey for IoT effects on smart city technology: challenges, opportunities, and solutions. *Asian Journal of Research in Computer Science* 8(3):32–48.
- Hou TC, Liu LH, Lan YK, Chen YT, Chu YS. 2022.** An improved network time protocol for industrial internet of things. *Sensors* 22(13):5021 DOI 10.3390/s22135021.
- Koubaa A, Aldawood A, Saeed B, Hadid A, Ahmed M, Saad A, Alkanhal M. 2020.** Smart Palm: an IoT framework for red palm weevil early detection. *Agronomy* 10(7):987 DOI 10.3390/agronomy10070987.
- Maróti M, Kusy B, Simon G, Lédeczi A. 2004.** The flooding time synchronization protocol. In: *Proceedings of the 2nd international conference on embedded networked sensor systems*. 39–49.
- Mengali U, D’Andrea AN. 1997.** Timing recovery with linear modulations. In: *Synchronization techniques for digital receivers. Applications of communications theory*, Boston: Springer DOI 10.1007/978-1-4899-1807-9_8.
- Meyr H, Moeneclaey M, Fechtel SA. 1998.** *Digital Communication Receivers: synchronization, channel estimation and signal processing. Volume 2 ed.* Hoboken: Wiley International.
- Mills D, Martin J, Burbank J, Kasch W. 2010.** Network time protocol version 4: protocol and algorithms specification (No. rfc5905). Available at <https://www.rfc-editor.org/rfc/rfc5905.txt>.
- Poongodi M, Sharma A, Hamdi M, Maode M, Chilamkurti N. 2021.** Smart healthcare in smart cities: wireless patient monitoring system using IoT. *The Journal of Supercomputing* 77:12230–12255 DOI 10.1007/s11227-021-03765-w.
- Pranto TH, Noman AA, Mahmud A, Haque AB. 2021.** Blockchain and smart contract for IoT enabled smart agriculture. *PeerJ Computer Science* 7:e407 DOI 10.7717/peerj-cs.407.
- Resner D, Fröhlich AA, Wanner LF. 2016.** Speculative precision time protocol: sub-microsecond clock synchronization for the IoT. In: *2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA)*. Piscataway: IEEE, 1–8.
- Rice M. 2008.** *Digital communication: a discreet time approach. s.l.* Prentice Hall.
- Saba T, Khan AR, Sadad T, Hong SP. 2022.** Securing the IoT system of smart city against cyber threats using deep learning. *Discrete Dynamics in Nature and Society* 2022:1241122 DOI 10.1155/2022/1241122.

- Skiadopoulos K, Tsipis A, Giannakis K, Koufoudakis G, Christopoulou E, Oikonomou K, Kormentzas G, Stavrakakis I. 2019.** Synchronization of data measurements in wireless sensor networks for IoT applications. *Ad Hoc Networks* **89**:47–57 DOI [10.1016/j.adhoc.2019.03.002](https://doi.org/10.1016/j.adhoc.2019.03.002).
- Sommars SE. 2017.** Challenges in time transfer using the network time protocol (NTP). In: *Proceedings of the 48th annual precise time and time interval systems and applications meeting*. 271–290.
- Sundararaman B, Buy U, Kshemkalyani AD. 2005.** Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks* **3**(3):281–323 DOI [10.1016/j.adhoc.2005.01.002](https://doi.org/10.1016/j.adhoc.2005.01.002).
- Tirado-Andrés F, Rozas A, Araujo A. 2019.** A methodology for choosing time synchronization strategies for wireless IoT networks. *Sensors* **19**(16):3476 DOI [10.3390/s19163476](https://doi.org/10.3390/s19163476).
- Umer M, Sadiq S, Alhebshi RM, Sabir MF, Alsubai S, Al Hejaili A, Mohamed A. 2023.** IoT based smart home automation using blockchain and deep learning models. *PeerJ Computer Science* **9**:e1332 DOI [10.7717/peerj-cs.1332](https://doi.org/10.7717/peerj-cs.1332).
- Watt ST, Achanta S, Abubakari H, Sagen E, Korkmaz Z, Ahmed H. 2015.** Understanding and applying precision time protocol. In: *2015 Saudi Arabia Smart Grid (SASG)*. Piscataway: IEEE, 1–7.
- Yiğitler H, Badihi B, Jäntti R. 2020.** Overview of time synchronization for IoT deployments: clock discipline algorithms and protocols. *Sensors* **20**(20):5928 DOI [10.3390/s20205928](https://doi.org/10.3390/s20205928).
- Yoo SG, Park S, Lee WY. 2020.** A study of time synchronization methods for IoT network nodes. *The International Journal of Advanced Smart Convergence* **9**(1):109–112.
- Zaidi AS, Suddle MR. 2006.** Global navigation satellite systems: a survey. In: *2006 International conference on advances in space technologies*. 84–87.