Wireless Pers Commun (2017) 97:6469–6481
https://doi.org/10.1007/s11277-017-4848-1

# Clock Synchronization in IoT Network Using Cloud Computing

**Ravi Shankar Jha**[1,2] · **Punit Gupta**[1,3]

Published online: 17 August 2017

**Abstract** The Internet of Things is a trending future technological revolution that emerging distributed computing and real-time based application and its development depends on dynamic technical innovation in a number of important fields from wireless sensors to nanotechnology. Cloud integration with IoT made things more convenient and easy. Clock synchronization between systems in a distributed network is the complex and tedious job. It is mandatory to get sync with other and source as well. There are many proposed ways of data synchronization protocols like NTP, global position system to maintain sync process between Systems or IoT devices network. In this paper, we are proposing a clock synchronization approach to sync clocks between IoTs and Cloud which are connected with each other in distributed network. Here we have used cloud service Software as a Service to collect the information to analysis and trigger the results action to IoTs. In this paper, we present the Clock Synchronization based on precision time protocol between IoTs simulation model for Omnet++ with INET framework, which allows us to check clock sync accuracy with a different network configured topologies. We have tried to minimize the clock drift with clock offset updating and with master–slave phenomena also minimize the master–slave delay. To show our simulation results we have used chunks nodes of IoTs in distributed manner placed in different places with different clock values. The simulation result shows that proposed approach works in the desired manner within ideal and network delay symmetry.

✉ Punit Gupta
punitg07@gmail.com

Ravi Shankar Jha
ravijha65@outlook.com

[1] Department of Computer Science Engineering, Jaypee University of Information Technology, Himachal Pradesh, India

[2] House No. 26, Shoghi Housing Board, Shoghi, Shimla, Himachal Pradesh 171219, India

[3] Waknaghat, P.O. Waknaghat Teh Kandaghat, Dist. Solan, Solan, Himachal Pradesh 173215, India

**Keywords** Internet of Things · Precision time protocol (PTP) · Clock drift · Time deviation · Synchronization

## 1 Introduction

The Internet of Things is now trending and future of new era of technology. Smart device smart enough to deal at its end to preprocess data and sense the data and automatically transfer data over cable or wireless network for further process. Data transmission is happening at the both end IoTs with sensing and collection of data but its perform action or action can be trigger at the main server where data analyses and action taken against the analysis of data which is transferred to these smart devices. Data transfer mode is may wireless or wired but keep data updated is more important than sensing, if data arrived late or ambiguously then analysis or further all process with be affected and that is a big problem. Synchronization [24] is key word which implies that data, and device to work parallel. IoT devices enabled with PTP which works with packet switching. In this paper we are proposing Devices (IoTs) to Server synchronization which lied on basis synchronization concept of clock which examined at the end of the device and IoTs which is more precise than any other aspect. If clock tick is different at both end then data never going to be received on real time which change the whole things.

As shown in above figure clearly we can see that clock value in real time system make a big impact on network, either its homogeneous or heterogeneous. To be on time in system dt/dc must be 1 then only data will transmit perfectly on time but in real network it's never happen that's why we need some adjustment in clock. If $dt/dc > 1$ then clock is fast means data is reached before it sends and $dt/dc < 1$ in this case clock slow means data is received at end side after short time of period. Clock skew show up here with difference between ideal and fast clock which can be adjust using logical clock that adjust when system lie on network while data being transmitted.

*Clock Drift* A computation device has clock which run on oscillator frequency which is run at different frequency than other device causing the clocks to tick at different rate, due to different tick rate creates a wide gap in perceived time is known as clock drift.

*Skew* Difference between two clocks at any point is called as clock skew and it is coming due to both clock drift and possibility that clocks might been set different at both end devices.

*Status of Events* As the events generates with time at both devices during the communication via message, at both end events status being capture with respect to time get the real time track. In this paper we are focusing on both clock drift and skew values at global network of IoTs. Device clock synchronization really on concept of system clock time which has to be keep track and adjustment of clock drift which need to be maintain at both end i.e., Device as well server end. As IoTs are trending and giving their part in each area which play a vital role in human being life, if this is the future technology then must be sure that data being processed is updated and error free. In this paper we are proposing a simulation model for IoTs and cloud for data synchronization at both end using Precision Time Protocol. Here we are examined the clock errors while data being served and shred over server and devices.

## 2 Motivation

IoTs are new trending technology, as per survey there will be 24 billion devices by 2020. These devices interconnect people and taking place of sensor with their additional capabilities of computation, and as far we are look for future these devices also comes up with new way of data collection like sensor and with their additional computation flexibility they also able to take action. Synchronization of these device also required to make collected data more sensible with keeping their clock sync with each IoTs and other network (computer system), big data set being collected by these device analyses at the other end. Recently cloud computing is more popular to do such computation with huge data set. Our aim is to simulate IoTs distributed network in such way that these devices collect data and send data to cloud node which will do further process and generate the action against analysis. To get the accurate result from collected data, these device need to be synchronize with each other without any conjunction and collision. Here we are proposing a clock synchronization algorithm for network of IoTs to keep them sync with cloud as well with other IoTs.

## 3 Related Work

Bojic [1] investigate scalability issues of self-synchronization emergent properties, described with the pulse coupled oscillator model to synchronize the devices using pulse coupled oscillator model the information propagation process, which is selective coupling synchronization sender side to reduce network traffic and minimize the time to sync process time in M2M (machine to machine) communication.

A hybrid-MAC protocol strategy proposed by Verma [2] to keep data synchronization between M2M. Device to device synchronization play a vital role which keep maintain data integrity updated information, keep thing rely in synchronize. Carrier sense multiple access (CSMA)-based contention period, and a Time division multiple access (TDMA)-based transmission period sync approach proposed here.

IEEE 802.15.4 is network standard proposed by Dressler [3] proposed on physical layer to control media access for low rate for wireless connection which show that data correction and flow control is required to maintain data synchronized and errors free. To synchronize IoTs some use agent base clock synchronization like Kubler [4] proposed an agent based peer to peer synchronization approach to synchronize all smart thing to maintain synchronization. Peer to peer device can be synchronizing with agent system strategies, agents keep track device and keep them synchronize which lead to data impotency and robust ness of data but if agent got distortion or affected by somehow them there might be big problem. Some other method came in existence to synchronize the node clock like using precisions Time protocol (PTP) proposed by Martin [5] clock skew and drift can find at logical and can be adjust the clock according to this. In Network there are traffic lie in real network which lead to propagation delay and packet delay and many more network conjunction instead of ideal condition. Author proposing Average TimeSyncprotocol [6] to keep sync wireless sensor in distributed environment, which is based on consensus, agreement, gossip or rendezvous whose main idea is averaging local information data and compute the average value and trigger all sensor to update their value according to that value or time. Due to lite computation this approach is frequent which is good because these are small devices having less computing component.

Kopetz [7] presents upper bounds on the achievable synchronization strategy for external and internal synchronization of processing unit in a distributed real-time system. Which lead to process synchronization but what about data being served by distributed devices. Roberto [8] also proposed an approach to sync distributed network of wireless sensor network. The concept of continuous versus instantaneous synchronization is introduced in order to generate a uniform common time base for local, global, and external time measurements. New trending technology Wireless sensor network is contribution of sensible devices which collect information is form of measurements, infer and understand environmental situation, all these devices in communicating actuating network made Internet of Things, and integration with cloud make it worldwide centric vision. Gong [9, 11] shows during fault in clock sync their security issues rise even after faulty clock has been synchronization and solution for that which is seems works fine. Such issue is more critical when we take scenario of global network, Choi [19, 20] proposed Distributed Asynchronous Clock Synchronization (DCS) and Distributed Synchronous Clock Synchronization algorithm to sync distributed network which is delay tolerant networks, key idea to global synchronize all node which are being stablished in tolerant networks. Li [10] also given his contribution to synchronize the distributed network of wireless sensor. Some network which are follow IEEE standard like IEEE 802.15.3 standard being used for wired and for wireless 1394 network, Lee [12] proposed an approach new clock time synchronization method for wireless 1394 heterogeneous networks between 1394 and 802.15.3. To keep the synchronize all node in distributed network is difficult and not easy to implement, there are large variety of protocols middleware which are rely on proper time synchronization across the computational infrastructure and depend on the clock accuracy. The Network Time Protocol (NTP) is more popular and current widely accepted standard for synchronizing clocks over the Internet. Gurewitz et al. [13] proposed NTP based non-hierarchical peer to peer approach for time synchronization. Giridhar and Kumar [14] show that with high probability the error variance is O(1) as the number of nodes increase in the wireless network. Mutual network synchronization [15] is a key for distributed environment to keep sync network in which geographically separated clocks align their times to one another without the need of reference or master clocks in other word with mutual network synchronization the timing information is exchanged explicitly by using periodic time stamp packets to send theses packet can be send by same beacons used in the IEEE 802.11 or IEEE 802.15.4 Standards. Average Time Sync (ATS), for synchronizing a wireless sensor, this algorithm is based on a popular distributed algorithms known as consensus, agreement, gossip which implies with idea of averaging local information. Rentel and Kunz [16] proposed algorithm which extend ATS with three feature (1) It is fully distributed without node failure and to new node appearance. (2) It compensation clock skew, thus network synchronized for long periods than using simple clock offset compensation. (3) Less computation makes it lite. Clock synchronization in real time network more important when in automation functions required to be synchronization with their computer clock. Frank et al. [17] proposed new methodology with this scenario to improve the performance with telegram service and alternative current (AC) power supply. Estimation of the clock offsets of the slave node and other nodes located in the common region of master and slave paradigm is used for synchronizing in clock synchronization [18], given approach to improvising the estimation of clock offset for wireless sense network. Leng and Wu [21] proposed belief propagation based algorithm to synchronization for fully distributed wireless sensor network, with results show that overhead low and we can achieve scalable synchronization. Wireless node's clock change with environment behavior, due to dynamic and unpredictable environment of sensors network. In such environment its big task to accurately estimate clock skew, and to get the reason behind this change. To deal with non-stationary situation Leng and Wu [22] proposed additional information aided multi-model Kalman filter (AMKF) algorithm, temperature measurements to assist clock skew estimation to get the clock offset to resynchronize clocks in dynamic network. With assumption of two-way message exchange mechanism Zennaro et al. [23] proposed clock synchronization algorithm with max-product algorithm adaptation to achieve sync global network of wireless sensor network.

## 4 Organization of Paper

In proposed work Sect. 5 we will discuss event variations between IoT nodes then we will also discuss more on clock distribution. In this section we will talk about Quality of Service (QoS) of synchronization. In Sect. 6 simulation and results of proposed model is discussed. Section 7 discusses conclusion and future works.

## 5 Proposed Work

IoTs and cloud are new era of technology and growing rapidly positively effecting human life and playing a big role in current computation with embed processing unit, parallel cloud computing is based on pay per service buy or rent the component and use them as per user required, best part of cloud is feasibility when required its one click away.

Cloud computation is more reliable and efficient likewise, in this paper we are proposing data and clock synchronization between IoTs and Cloud, here we are using cloud SaaS (Software as a Service) which enable devices as sensing and acting devices, which collect the information via their sensing components and send this data to cloud for further process Here's the thing happens, due to clock differentiation between clock of these devices which are lied out at same place with some distance need to be synchronized and updated and if there is clock difference between cloud and IoTs clock then data never going to reached at destination which occurs a fault of data processing i.e., due to lake updated data computation or any other process happening at the cloud end going totally wrong which lead to wrong decision making process, if this process still accepted at the end of cloud and resultant will be wrong and action taken against the result will be completely wrong (Fig. 1). Figure 1 shows daviation between system clock time and real clock time under slow, fast and ideal clock.

In Fig. 2, let's take an example of two Node A and Node B in a network, are taking participation in network to communicate with each other, here Node B generate a message frame and transmitted to Node A at time t1 and at the other end A received it at time t2, after receiving message at A, its take little bit time tp to process (after receiving message to extract data from frame) during this process tp elapse time, now thing is time difference at Node TB = (t4 − t1), similarly at Node A time TA = (t3 − t2). if this time difference will grow and clock skew between both end increase rapidly which is big problem, with respect to clock skew, clock drift will also increase which is not good.

With respective of previous scenario let's take a big scenario of big network of IoTs where there are multiple and heterogeneous devices are connected with different topology like some are in star and some are in ring and any other topology, some are connected via wire and some are wireless. Some of them are connected with high link channel and some

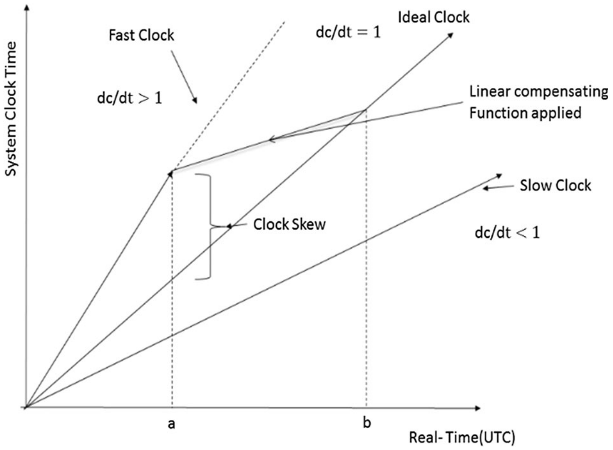


**Fig. 1** Clock error and compensations for drift with linear compensating function

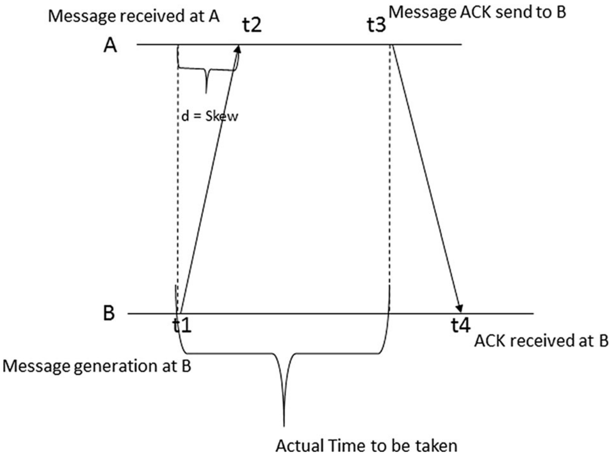


**Fig. 2** Event variation between nodes A and B

are using less bandwidth connection, in such situation data may not be transferred in time and somehow its reached at receiver side then it would be no longer useful because data keeping sensed by IoTs and its real time value required to take a real time decision to trigger these IoTs. Let's take an another example of health care IoTs network, IoTs sensing and monitoring the patient body and keep updating physical condition of patient and reporting to doctor or hospital. In IoTs-network these devices directly uploading data to hospital's server where all these data lied and analyses of further process happen.

If data is wrongly received or data being received is not real time data is useless and may cause big problem to take the right decision.

In This paper, we are proposing an approach to synchronize all node in network where they are connected within a homogenous or heterogeneous way.

To simulate the scenario, we have taken chunks of node (IoTs) which are placed in heterogeneous way and are connected with each other. As shown in Fig. 3. In each local

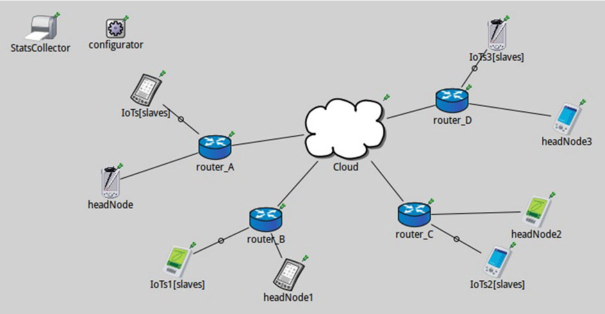


**Fig. 3** IoTs network used for simulation using Omnet++

network we are placed a head node which is lead the network to keep sync with each node or in other words head node will monitor the clock sync process to do so. Each Node contains logical and hardware clock, logical or software clock track and adjust its clock value and updates as required to sync. Hardware clock and software clock play a vital role in clock synchronization.

*Hardware Clock* Each system contains a system hardware clock which runs on oscillator which runs the system clock. These oscillator frequencies keep change with atmosphere, it may get high or low with temperature. A quartz crystal on one system will oscillate at a slightly different frequency than on another computer, causing the clocks to tick at different rates.

*Software Clock* This is also known as logical clock which can be adjusted, means clock value can be alter with respect to requirement. Due to flexibility of clock value it keeps updated with network to make real time transmission.

As in above figure we have taken this scenario to simulate our algorithm. As in above figure in each local networks of IoTs we taken a head node which keep track other IoTs clocks. Here we have taken cloud module (SaaS) to keep data which is collected by these IoTs and for further analysis. As part of synchronization process cloud also play a head node of all local network which keep sync all local network head node.

Each node configured with two types of clocks software clock and hardware clock as in below Fig. 4.

Each contain both clocks, routing table, battery and follow all properties of IoT device.

All node base synchronization is algorithm which we are using to synchronize network of IoTs, here head node starts with a sync process message generation which is call for clock time value of all devices.

To sync the network of IoTs we are proposing an approach which works on all node based clock synchronization algorithm. Pseudo code for the algorithm is shown in Fig. 5.

All nodes are in network participate in this process with network initialization. As per shown in above pseudo code execution of process start with hops registration in network which initiate with message generation with head node, after receiving member devices will send back acknowledge to head node with their clock value. This takes place and execute with pseudo code as shown in Fig. 6.

As in above pseudo code initiate the local network sync process with registration of devices, here might be possible that device may leave the network due to some problem that's why we are keeping update network with the registration of device process. After registration we use algorithm 1 (All node base synchronization), to synchronize the network.

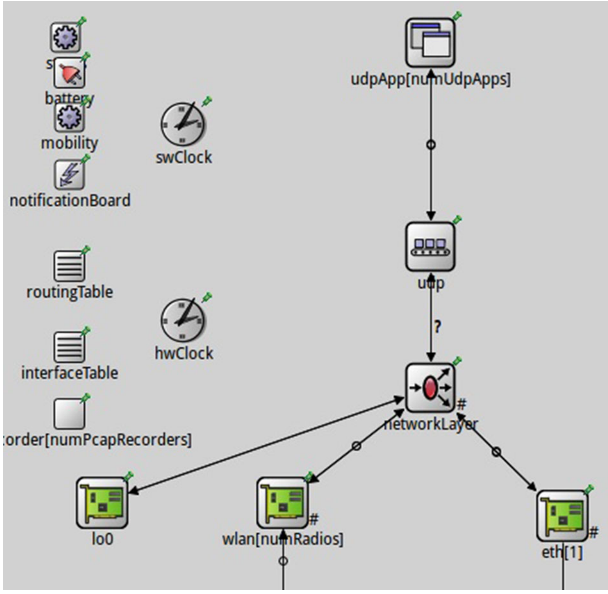
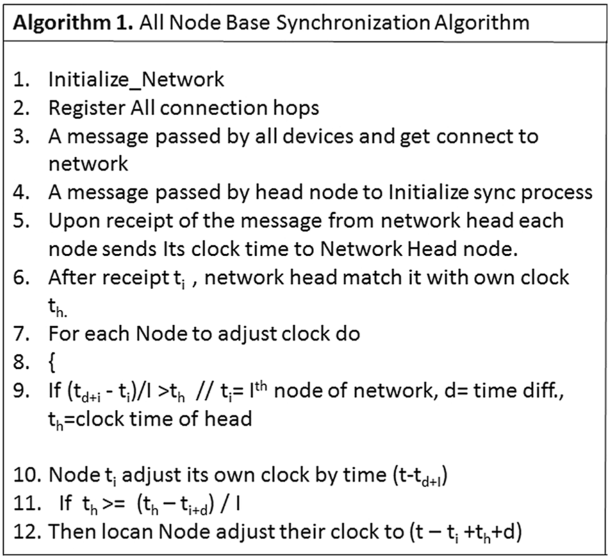


**Fig. 4** IoT node configuration



**Algorithm 1.** All Node Base Synchronization Algorithm

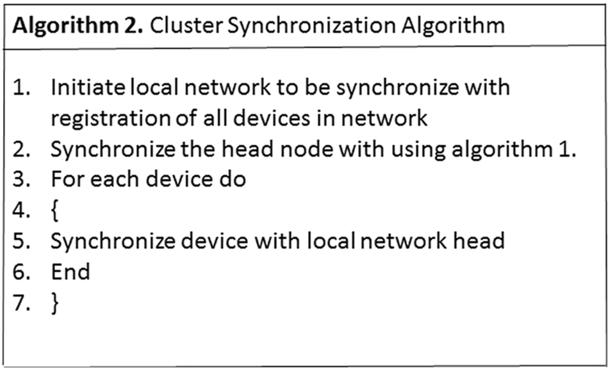
```
Initialize_Network
Register All connection hops
A message passed by all devices and get connect to
network
A message passed by head node to Initialize sync process
Upon receipt of the message from network head each
node sends Its clock time to Network Head node.
After receipt $t_i$ , network head match it with own clock
$t_{h.}$
For each Node to adjust clock do
{
If $(t_{d+i} - t_i)/I > t_h$ // $t_i$= $I^{th}$ node of network, d= time diff.,
$t_h$=clock time of head

Node $t_i$ adjust its own clock by time $(t-t_{d+I})$
If $t_h >= (t_h - t_{i+d}) / I$
Then locan Node adjust their clock to $(t - t_i + t_h + d)$
```

**Fig. 5** Pseudo code of all node base synchronization

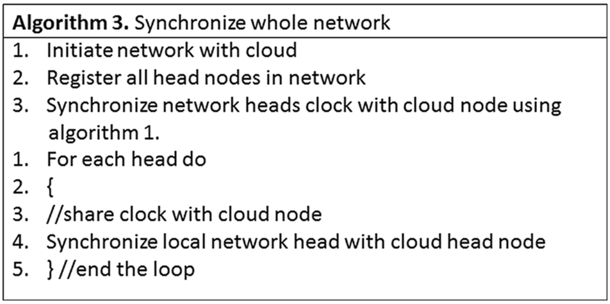
To sync the whole network of IoTs completed by algorithm 3, which sync the cluster head by adjusting their clock respective to their cloud head node. Pseudo code for algorithm 3 is shown in Fig. 7.

Algorithm 3 will take place when global sync happen it goes like this way all local network head node participate in global synchronization with cloud node which is more



```
Algorithm 2. Cluster Synchronization Algorithm
Initiate local network to be synchronize with
  registration of all devices in network
Synchronize the head node with using algorithm 1.
For each device do
{
Synchronize device with local network head
End
}
```

**Fig. 6** Pseudo code of local network synchronization



```
Algorithm 3. Synchronize whole network
Initiate network with cloud
Register all head nodes in network
Synchronize network heads clock with cloud node using
  algorithm 1.
For each head do
{
//share clock with cloud node
Synchronize local network head with cloud head node
} //end the loop
```

**Fig. 7** Pseudo code of head node synchronization

important and major part of the whole synchronization process. It starts with the initialization of network of head node of local networks including cloud node, Synchronization of all head node synchronized with using algorithm 1, which synchronize the clock of the head node with respective to cloud node.

Here our assumption is all clock start with same time initially and there is no fault in master–slave coordination i.e., there is no fault in the master selection, here we are assigning the mater and slave node. Here master node in the local network is stated as the head node of the network and it initiates the clock synchronization in that network with local other devices with sending a message which contains a request for clock time.

## 6 Performance Analyses

A For simulation we have taken a scenario of a distributed network of 16 IoTs in which all these devices connected to cloud. Here two network base approached we have considered first in a local network of IoTs we have taken some master–slave phenomena to sync rest devices with master or head node. Here for simulation, we have divided sync process in two part first local synchronization in which local network of IoT will sync and in second part all head nodes of local network will sync their clock with cloud node.

To show our work we have calculated drift, master–slave delay, and time deviation between head node of local network of rest IoTs same for all head node with respective to cloud node. Simulation result is in below figures.

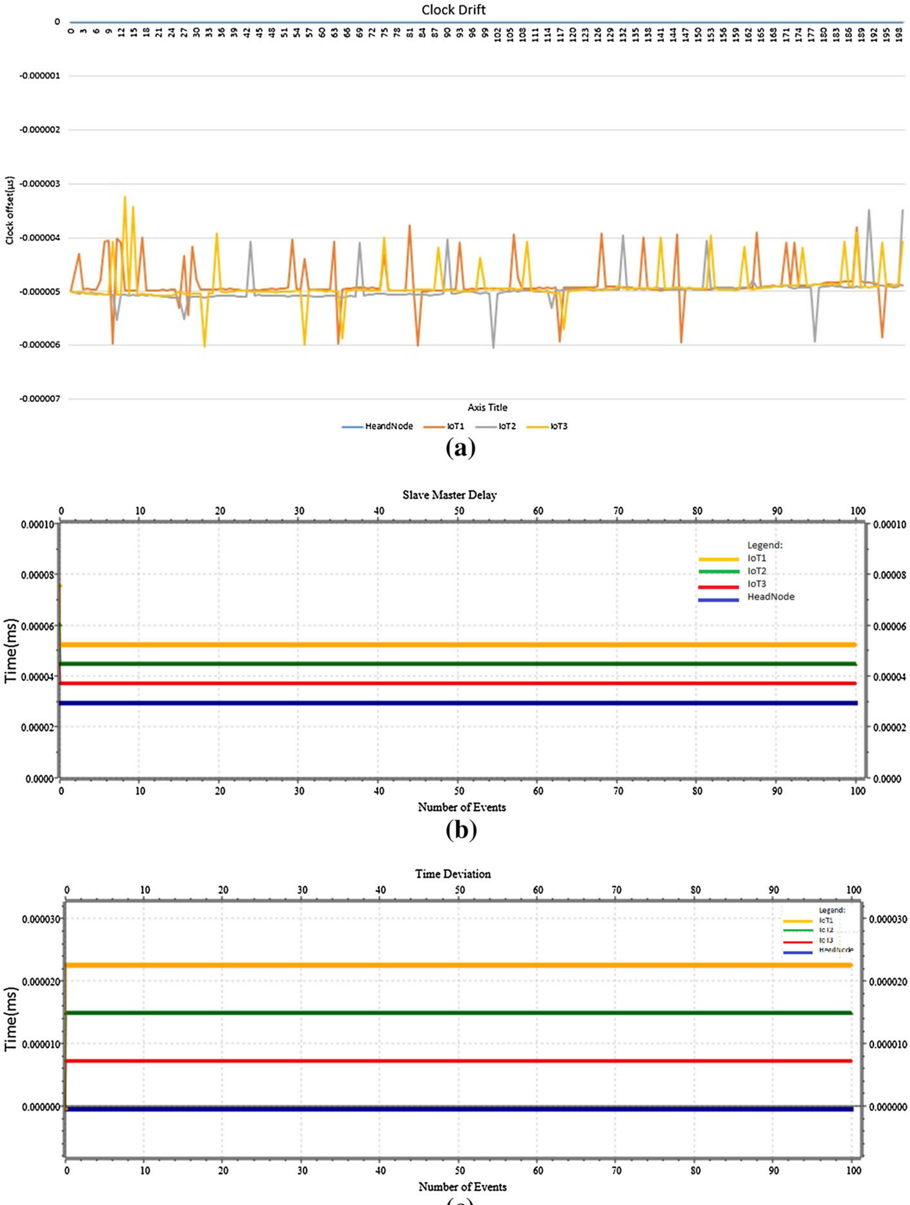


**Fig. 8** **a** Software clock drift. **b** Soft clock slave master delay before algorithm. **c** Soft clock time deviation before algorithm

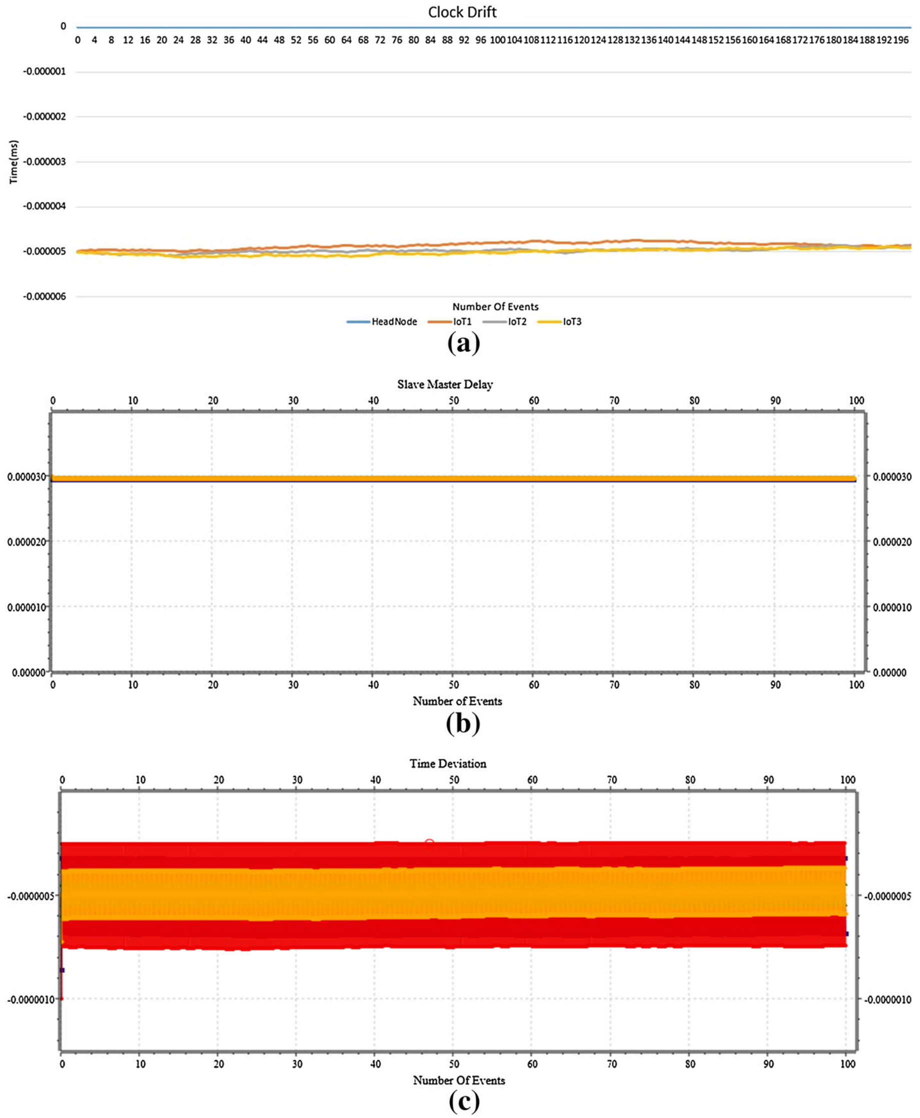


**Fig. 9** **a** Software clock drift after algorithm. **b** Soft clock slave master delay after algorithm. **c** Soft clock time deviation after algorithm

As previously mention that each node contains hardware and soft clock and software clock will adjust their clock as per requirement. With All part of proposed algorithm merged and result in Network of IoTs are as below.

Above Figs. 8a and 9a show that with proposed solution clock drift between is minimum as compare to generalize network drift. Figures 8b and 9b shows that master–slave delay is closed to nanosecond and Figs. 8c and 9c software clock time deviation is stabilized with master and slave IoTs which increase the data consistency and contention between the node is high which gives the surety of packet drops are less. Simulation results show that proposed algorithm works in expected manner.

## 7 Conclusion

Clock Synchronization with give effective and overcome the delay issue in data synchronization. The objective is to minimize the error, improve the sync rate and reduce the time delay in transmission of data using GALS algorithm to improve workability of IoT devices. Dressler and Nordin [3] shows that using GALS algorithm data transmission with minimum sync rate and error reduction can be minimized. Here proposed algorithm show that IoT synchronization is done in three step with global and local synchronization with proposed algorithm which is works separately at different scenario.

## References

1. Bojic, I., Lipic, T., Kusek, M. (2014). Scalability issues of firefly-based self-synchronization in collective adaptive systems. In *IEEE eighth international conference on self-adaptive and self-organizing systems workshops (SASOW), September 8, 2014* (pp. 68–73). IEEE.
2. Verma, P. K., Tripathi, R., Naik, K. (2014). A robust hybrid-MAC protocol for M2M communications. In *International conference on computer and communication technology (ICCCT), September 26, 2014* (pp. 267–271). IEEE.
3. Dressler, F., Nordin, N. (2015). On the synchronization of co-located IEEE 802.15. 4 networks for IoT applications. In *IEEE 1st international forum on research and technologies for society and industry leveraging a better tomorrow (RTSI), September 16, 2015* (pp. 29–33). IEEE.
4. Kubler, S., Nargund, I. D., Framling, K., Derigent, W. (2014). Peer-to-peer data synchronization agents. In *IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), August 11, 2014* (Vol. 3, pp. 32–39). IEEE.
5. Lévesque, M., Tipper, D. (2015). Improving the PTP synchronization accuracy under asymmetric delay conditions. In *IEEE international symposium on precision clock synchronization for measurement, control, and communication (ISPCS), October 11, 2015* (pp. 88–93). IEEE.
6. Schenato, L., Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *46th IEEE conference on decision and control, December 12, 2007* (pp. 2289–2294). IEEE.
7. Kopetz, H., & Ochsenreiter, W. (1987). Clock synchronization in distributed real-time systems. *IEEE Transactions on Computers, 100*(8), 933–940.
8. Solis, R., Borkar, V. S., Kumar, P. R. (2006). A new distributed time synchronization protocol for multihop wireless networks. In *45th IEEE conference on decision and control, December 13, 2006* (pp. 2734–2739). IEEE.
9. Gong, L. (1992). A security risk of depending on synchronized clocks. *ACM SIGOPS Operating Systems Review, 26*(1), 49–53.
10. Li, Q., & Rus, D. (2006). Global clock synchronization in sensor networks. *IEEE Transactions on Computers, 55*(2), 214–226.
11. Sun, K., Ning, P., & Wang, C. (2006). Secure and resilient clock synchronization in wireless sensor networks. *IEEE Journal on Selected Areas in Communications, 24*(2), 395–408.
12. Lee, S. H., Park, S. H., Choi, S. S. (2006). Clock time synchronization for wireless 1394 heterogeneous networks. In *International conference on consumer electronics, 2006. ICCE'06. 2006 digest of technical papers. January 7, 2006* (pp. 317–318). IEEE.
13. Gurewitz, O., Cidon, I., & Sidi, M. (2006). Network classless time protocol based on clock offset optimization. *IEEE/ACM Transactions on Networking (TON), 14*(4), 876–888.
14. Giridhar, A., Kumar, P. R. (2006). Distributed clock synchronization over wireless networks: Algorithms and analysis. In *45th IEEE conference on decision and control, December 13, 2006* (pp. 4915–4920). IEEE.
15. Rentel, C. H., & Kunz, T. (2008). A mutual network synchronization method for wireless ad hoc and sensor networks. *IEEE Transactions on Mobile Computing, 7*(5), 633–646.
16. Rentel, C. H., & Kunz, T. (2008). A mutual network synchronization method for wireless ad hoc and sensor networks. *IEEE Transactions on Mobile Computing, 7*(5), 633–646.
17. Frank, H., Mittnacht, A., Rappold, D., Kupzog, F., Mahlknecht, S. (2008). Clock synchronization of distributed computers in building automation networks using an AC power supply. In *34th annual*

*conference of IEEE on industrial electronics, 2008. IECON 2008. November 10, 2008* (pp. 2730–2735). IEEE.

18. Chaudhari, Q. M., Serpedin, E., & Qaraqe, K. (2010). Some improved and generalized estimation schemes for clock synchronization of listening nodes in wireless sensor networks. *IEEE Transactions on Communications, 58*(1), 63–67.
19. Choi, B. J., Liang, H., Shen, X., & Zhuang, W. (2012). DCS: distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems, 23*(3), 491–504.
20. Choi, B. J., Shen, X. (2010). Distributed clock synchronization in delay tolerant networks. In *IEEE international conference on communications (ICC), May 23, 2010* (pp. 1–6). IEEE.
21. Leng, M., & Wu, Y. C. (2011). Distributed clock synchronization for wireless sensor networks using belief propagation. *IEEE Transactions on Signal Processing, 59*(11), 5404–5414.
22. Leng, M., & Wu, Y. C. (2011). Distributed clock synchronization for wireless sensor networks using belief propagation. *IEEE Transactions on Signal Processing, 59*(11), 5404–5414.
23. Zennaro, D., Ahmad, A., Vangelista, L., Serpedin, E., Nounou, H., & Nounou, M. (2013). Network-wide clock synchronization via message passing with exponentially distributed link delays. *IEEE Transactions on Communications, 61*(5), 2012–2024.
24. Lévesque, M., Tipper, D. (2015). ptp++: A precision time protocol simulation model for OMNeT++/INET. arXiv preprint arXiv:1509.03169. September 10, 2015.

**Mr. Ravi Shankar Jha** M.Tech. researcher student in Jaypee University of Information and Technology, Wakanaghat, Solan. He received M.Tech. degree in Computer Science and Engineering from Jaypee University of Information and Technology, Wakanaghat in 2016 on “Data Synchronization in Machine to Machine Communication Through the Cloud”. He received B.Tech. degree in Computer Science and Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Madhya Pradesh in 2010.

**Dr. Punit Gupta** is Assistant Professor (Senior Grade) in the Department of Information Technology, Jaypee University of Information Technology (JUIT), Waknaghat, Solan-173215, Himachal Pradesh, India. He received B.Tech. degree in Computer Science and Engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Madhya Pradesh in 2010. He received M.Tech. degree in Computer Science and Engineering from Jaypee Institute of Information Technology (deemed university) in 2012 on “Trust Management in Cloud Computing”. He is a Gold Medalist in M.Tech. He joined Jaypee University of Information Technology Waknaghat on February 2013.