www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

Design of Secure Elastic Timer Protocol in IoT-Comparative Analysis

Kapil Mehta[1]; Yogesh Kumar[2]
[1]Research Scholar, Department of Computer Science and Engineering, CT University, Ludhiana, Punjab, India.
Assistant Professor, Department of Computer Science and Engineering, Chandigarh Group of Colleges, Landran, Mohali, Punjab, India.
[2]Professor, Department of Computer Science and Engineering, CT University, Ludhiana, Punjab, India.

Abstract
IoT is a most integral part of every day's life where everyone has to access the real-time information that makes people lives much more convenient through a network of digitally connected devices. In the present age, IoT protocols are designed to operate in multiple layers that allows devices to communicate wherever security is of much concern. IoT is much vulnerable to attacks due to the decentralization. During Communication between devices over an IoT network, Synchronization and Security are the main concerns and failing so, devices may miss the messages, collide with other messages or waste the energy to get back in sync. Data Exchange through IoT requires synchronization of real time aspects with network model of Internet. For the execution of synchronized tasks, synchronization of clocks of IoT is of extreme importance to allow the sequential ordering of information. There are various research studies on synchronization approaches so far. Elastic Timer Protocol is proposed that is beneficial as compared to standard Time-Lay Clock synchronization technique in case of energy consumption and convergence time. This Paper introduces the Elastic Timer Protocol that uses RSA to ensure the security of a network that works in the better form in case of reduced Energy Consumption and convergence time. The Comparison of the proposed Elastic Timer Protocol in comparison with Time Lay clock synchronization is also presented in terms of various performance metrics like Energy Consumption, Delay, Throughput and Loss Rate of Packets. Moreover, the simulation results revealed that the proposed work in terms of performance metrics mentioned above, performs better with superior performance characteristics.

Key-words: - Internet of Things, Time-Synchronization, Elasticity, Security.

1. Introduction

IoT is the power of the Internet that can be extended beyond computers and smartphones to the connectivity of things, processes and environments. With the connectivity of the Internet, one can send and receive information which makes things smarter and smarter is always better. As, automation technology advances day-by-day, human life is getting more comfortable, more convenient with ease. Everyone has a choice to opt for automated systems as compared to manual systems. In the present age, manual systems are least preferred over Automatic systems. Recent trends in connectivity and intelligence, pervasive devices interact with one another for communication. This emerging communication is called Internet of Things (IoT) [1]. To support IoT, there are various interconnection possibilities that bring connectivity concerns between heterogenous objects of different nature [2]. As everything around, us is connected through the Internet then it is mandatory that every device should be synchronized with each other by clocks. Although clock synchronization is a most discussed issue for many years that introduce various challenges [3,4,5]. First objective is to reduce costs in IoT devices, that leads to use of lower quality hardware components. Secondly, IoT devices are operational in environments operated at different temperatures. At last, in standard clock synchronization protocols, devices have limited communication and computation capability which can push their ability for participation.

Wireless IoT networks can be created by grouping sensor nodes that share data over a network through wireless communication links. Clock Synchronization is a process that adjusts the clock signals for synchronization with other physical devices available in the network. IoT does not involve human intervention and allows the transmission from the physical world to Internet and vice-versa [6]. This is possible by the communication between various entities. During communication, applications require ordered information that leads to requirement of time that can be shared between objects for the processing [7,8].

The clock timings are different than the universal time (UTC) due to the various factors mentioned as below [9]. The effect of these errors is due to:-

1. Signal transmission of time report of a reference clock
2. Unpredicting and unbiasing in message delivery and time measurements
3. Clock-tick adjustments

1.1. Factors Affecting Time Synchronization

There are various deciding factors that hinder the accurate time synchronization includes count of nodes, topology mechanism and various other factors as mentioned below:

a. Temperature- It affects the clock accuracy by modifying the frequency and deviating from the reference clock.

b. Oscillator Aging-A minute change in the frequency of quartz crystal over a period of time.

c. Battery Charge- To power the clock device, a battery is required that helps to track the time. Disconnecting the power will lose tracking of time.

d. Pressure- It affects the clock rate, as pressure increases; tick of clock slows down and vice versa.

1.2 Clock Synchronization Errors

For the generation of clock signals, physical clocks forming the network use crystal oscillators whose frequency determines the clock. Sometimes, frequency of such clocks is unstable due to the manufacturing issues or imperfections, environmental changes like temperature, pressure etc. as discussed above, and physical clocks do not work with the same speed, so a time-to-time synchronization is required. For more illustrations, consider the Example-Suppose two nodes N1 and N2 of a network want to communicate. Node N1 wish to send a message to node N2 and then Node N1 has to synchronize with node N2's clock. There may be a change due to the delay between transmitting or receiving a message, before the destination Node N2 receives the clock value of Source Node N1. This delay can be occurred due to the various factors [10,11,12] as mentioned below:

Send Time- It is the time that is required to prepare and send the message by any operating system.

Access Time- Time required by a node to access the transmission channel of destination node.

Send/Receive Time- Transmission time stand for the transmission of the message from one node to another. On the other hand, reception time stands for the time utilized by a node to receive the message sent by other nodes.

Propagation Time- It is the time taken to travel from one node to another by the message.

Depending upon nature, the time associated can be deterministic or non-deterministic.

It is required to calculate two interdependent components during the synchronization of a local clock of a network [13,14]:

a. Clock Offset-It is the known or unknown delay of a given clock source. It is measured in terms of time units.

b. Clock Skew- The difference of expected time arrival and actual time arrival. It can be measured in terms of time units and could be positive or negative.

Both the components mentioned above are treated as instability parameters and make sure that the network system does not have time fluctuations and variations.

This paper is organized Section-wise as mentioned here: Section 1 is the Introduction section. Section 2 provides the brief of literature or related work. Section 3 introduces the Time Lay Clock Synchronization. Section 4 presents the proposed Elastic Time Clock Synchronization Protocol. Section 5 provides results of various performance metrics, and section VI culminates the conclusion and future work.

2. Related Work

There are numerous solutions to the problem that is required to be solved. The problem can be centralized that depends on centralized server connected to an accurate time source UTC described by Cristian [15] and Gusella and Zatti [16]. Every time to get the actual time, the request is made to the server and response is given by appending the current timings of a clock and send that back to client. On the other hand, the server polls the machines time by time periodically. When all the responses are received, server provides the time by using the round trip of messages.

Lamport [17] uses the concept of casuality and defines an ordering of events. For instance, if one event affects the outcome of other event then it is termed as partial ordering. There are various rules to be considered for partial ordering. First is the incrementing of local clock and second is the local timestamping.

Fidge [18] also uses the concept of casuality for the partial ordering of events. This includes the vector of values instead of using timestamping. Choosing vectors instead of timestamps maximize the performance of processors that will order intra events and inter events both in processes.

Tuncer presented the concept of Elastic Clocks [19] through which margins can be reduced without the sacrifice of robustness. Elastic Clocks allows non-static variability that can be adjusted by the clock at every cycle which adds gaps between timings.

T. Inzerilli et al. proposed a location-based approach which is optimized and can handle the synchronization with soft mobile-controlled vertical handover. The location information is utilized and accessed for attaining handover decisions [20].

A. Tekeoglu et al. introduced a run for security examination and privacy of IoT devices. The proposed system gives the better performance in security, authenticity and privacy. Along with that, vulnerability attacks, and issues related to insecure protocol versions was found [21].

I. Nasr et al. presented a noncoherent clock synchronization algorithm for timing detection. Detectors work with coherent timing that work on fading channel technique due to which the complexity of a network can be reduced. The technique in comparison with non-aided data coherent technique works well in terms of root mean square errors [22].

3. Time Synchronization Protocols

Synchronization between clocks is required for the efficient communication. There are variety of synchronization protocols available, the list is as below: -

A.) Network Time Protocol (NTP) – It is the protocol which allows NTP Computer sends a request of time signal with NTP Server Computer. NTP uses Coordinated Universal Time-UTC for the synchronization of clock timings with higher precision for accurate results [23].

B.) Simple Network Time protocol (SNTP) -It is a time synchronization protocol which uses lesser processing power during synchronization than NTP because it frees up the processor for several tasks. It works in the manner that every node is dependent upon the clock timings of another node.

C.) Precision Time Protocol (PTP) – It is a protocol that allows a grandmaster clock operates at the root node and distribute time information to other slave clocks. By using this mechanism, all clocks make a synchronization tree and works in better accuracy than NTP.

D.) Datacenter Time Protocol (DTP) – It use control packets or messages instead of standard packets in the physical layer for communication of several protocol messages. It is a highly scalable protocol with a bounded precision.

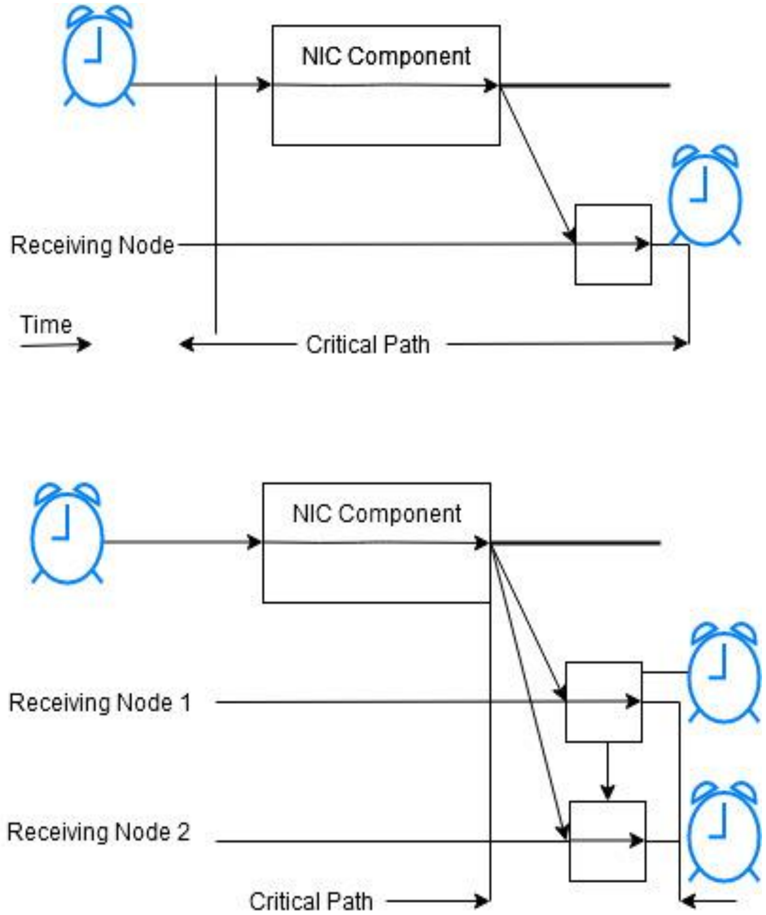
E.) Mobile Network Time Protocol (MNTP) -It is designed for lossy channel conditions to host mobile nodes. MNTP is defiant to clock drift conditions too. MNTP can be implemented on a wireless laptop and illustrate its capability in different set of operating conditions.

All these techniques work on the principle of Time-Lay Clock Synchronization which is a technique that sets the clock of the nodes of the network according to the timings of a third-party clock. For setting the clock of the nodes of a network, one node is to be a master clock. The master clock will broadcast a signal to the set of nodes present in the network. The first signal is to alert the receiver nodes to compare their clock timings with one another to calculate the relative offsets. The second signal is to send a syn-packet and allow the nodes to use the arrival time of a packet for synchronization as a reference time. The sync-packet will be broadcasted to all the receivers without any partiality. The nodes who receive the sync-packet will exchange the timing information with one another so that offset can be calculated to retain a local timescale. As compared to the traditional techniques that majorly focused on broadcasting a timestamp in a synchronization packet, time lay clock synchronization removes delay and uncertainty issues by automatically allowing the nodes to set their clocks after alerting the master clock [24].

Time-Lay synchronization can be scaled to 'n' receivers where 'n' is more than two. Once the sync-packet is received by the nodes they can note their time and can start exchanging timing information with their neighborhood [25].

Using Time-Lay Clock Synchronization is beneficial, as there is no scope of uncertainty from the sender side. By removing the Sender, the uncertainty relies on propagation and receiver time. In case of less size networks, it is assumed that sync-packet will arrive at the receiving nodes at an instant rate, so propagation time is completely negligible in such cases [26]. As shown in Figure 1, the traditional synchronization technique contains sender in the critical path which is supposed to synchronize with the receiving nodes. But in the Time-Lay clock synchronization, there is no sender in the critical path as this technique sets the clock timings with one another [27].

Figure 1 - Time-Lay Clock Synchronization



The cons related to Time Lay Clock Synchronization are as below:

1. One main challenge in Time-Lay Clock Synchronization is the computation of timing bottleneck which triggers the critical path adaption and restricts dynamic timing enhancement.

2. Reliable Synchronization is not assured in time-lay clock synchronization if faulty nodes are introduced that reproduces a trade-off between timing information and fault tolerance [28].

3. It doesn't work well for error detection and error handling conditions that do not relate with the algorithms for achieving high precision of clock values [29].

4. Proposed Elastic Timer Protocol

The Proposed Elastic Timer Protocol works on Elastic Clock Synchronization that assigns the resources according to the requirements of the network which is designed to support the reduced Energy consumption and less time to transmit and receive messages over the channel. The objective of Elastic Clock Synchronization Algorithm is to provide an issue free service to guarantee the efficient delivery of data. This technique is primarily based on the feature Elasticity that allows a system to adapt the workload changes by over or under-provisioning the available resources that meet the actual demand as possible. This is a novel concept that works on the basis of varying load by altering the use of computing resources in comparison to the previous approaches dealing with IoT.

The pros of Elastic Timer Protocol are as below: -

1. Elastic Timer Protocol is designed to be most scalable and flexible to maintain the tight synchronization between nodes.

2. Elastic Timer Protocols allow Elastic Clocks that reduce margins between the devices without sacrificing robustness.

3. No modification in the logical configuration of hardware. Elastic Clocks substitute global clock signal by the local clocks.

4. Elastic Timer Protocol handles dynamic variability with the clock adjustments at every cycle can be handled by adjusting the clock tick at every cycle that adds micro margins for the variability.

5. Normal Clocks (Rigid Clocks) work at single frequency, due to which energy and time is wasted in most of the cycles which is overcome by Elastic Clocks that follow the critical path variability with smaller margins as these do not work with fixed frequency.

6. Elastic Timer Protocol offers better power consumption by reducing the clock margins that make the circuit to run with a better performance.

The proposed Elastic Timer Protocol works with a global elasticity property that overcomes the wastage of resources raised due to underperforming or overperforming of varying load. Elastic Clock Synchronization adds a new property of handling the load variability to the nodes that can handle load spikes without restarting a system. Elasticity is the degree of freedom through which a system is able to handle changes in workload through independently and autonomously provision and deprovision the resources to assure that available resources should match the present demand as close as possible.

Elastic Clock Timer Protocol overcomes the weakness of standard clock synchronization schemes that waste resources because of the conditions having load variability. Using Elastic Clock Synchronization, source nodes sense the communication channel and send a signal to the destination node. Notifications are processing from the source node to the access point layer to dispatch to the destination node. Source and Destination nodes represent the entry and exit point that will handle the client connections and sends the received notifications to the targeted subscribers, it forwards subscriber subscriptions and publisher notifications to the *Access Point*.

Each layer of a network should have the property to scale in and out irrespective of each other. So, coordination within the layers is required for the maintenance of state consistency and it is performed by the cluster head who will create and remove nodes to fulfill the demand of interested layers. These Four layers are: -

Access Point (AP):- It acts as a portal for devices for the connection with LAN. An access point dispatches messages to the desired matcher and broadcasts packets to all other matchers.

Matchers (M): - Matchers receive the messages, match messages against receivers, and dispatch the packet containing information to the *Exit Point* layer.

Exit Point (EP): - It collects all partial results coming from the *Matcher* layer, merge them in the final set and then sends the notification to all *Connection Points* found in this set.

Connection Point (CP): - It represents the entry and exit point of a system. It also handles the connection with clients and sends the received notifications to the nodes as per their interest.

To perform scaling, the system should be capable to match the variable demand of workload that would also save the cost of deployed infrastructure.

The comparison of Proposed Elastic Timer Protocol with classical Time Lay Clock Synchronization is elaborated in Table-1.

Table 1 - Comparison of Proposed Elastic Timer Clock with Time Lay Clock Synchronization

Estimator	Time Lay Clock Synchronization Protocol	Proposed Elastic Timer Synchronization Protocol
Dynamic Timing Enhancement	Computation of timing bottleneck restricts dynamic timing enhancement and critical path adoption	No restrictions in dynamic timing enhancement
Reliable Synchronization	No Reliable Synchronization is assured if faulty nodes are introduced.	Fault tolerance capability to ensure reliable sync.
Error Detection	Higher precision of clock skew errors	Lower probability of Clock skew errors.
Scalability	Less Scalability/Loose Synchronization	High Scalability/Tight Synchronization
Power Consumption	Higher Power Consumption with high clock margins	Better power consumption by reducing the clock margins
Variability	Higher Variability of Critical Paths with Clock Margins	Lower variability of clock margins with elasticity

5. Results and Analysis of the Proposed Elastic Timer Protocol

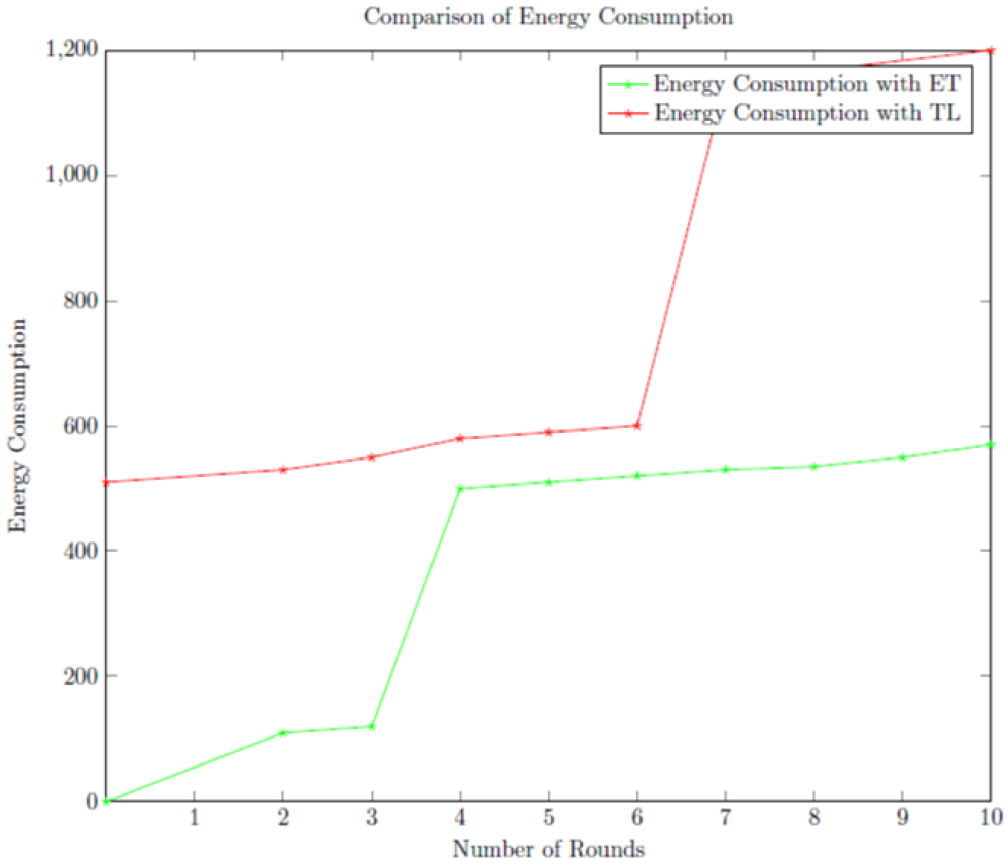
This area will present the performance evaluation of Elastic Time Synchronization over Time Lay Clock Synchronization for parameters like Consumption of Energy, Throughput, Loss Rate and Delay. Better results have been generated while using RSA with both the techniques. For the evaluation of the performance of simulation experiments, The MATLAB R2016a simulator was used. MATLAB helps to show the graphical representation of values of evaluated performance metrics. The MATLAB has various in-built functions, toolboxes, GUI based drag and drop dialog boxes, Artificial Neural Networks etc. MATLAB helps to implement both the algorithms, plotting graphs and designing graphical user interfaces. The network scenarios nodes were randomly placed. Nodes were selected manually or automatically to share the packets from one area to another. First of all, network of sensor nodes will be created of fixed Size. In our case, we have created a network of 50 nodes. Sensor nodes residing in a network grouped into clusters. Each group of nodes (Cluster) has a Cluster Head that will be chosen by any election algorithm. The node elected as a Cluster Head has the highest energy. Each Cluster Head transmits signal containing time to the sensor nodes of the region and synchronize with other cluster heads of different clusters for exchanging the time value. Cluster Head can calculate the average clock time. Cluster Heads can set the time of every sensor node. All the cluster head set their own time according to the final result. All the Cluster Heads work in the same way and calculate their average time. Base Station monitors the functioning of clusters and works as an intermediary during inter-cluster communication.

A. Energy Consumption- Energy Consumption is always a challenge as users have to pay huge energy bills for the operation of devices. The transmission of packets from one node to another is termed as Energy Consumption. The product of the number of transmitted packets with per unit energy is the energy consumption of a network.

Energy Consumption=Count of packets transmitted X Per unit Energy

Figure 2 shows the comparison of consumption of energy. As the number of rounds of simulation is increasing, Energy Consumption is lesser in Elastic Time Synchronization when compared to Time Lay Clock Synchronization.

Figure 2 - Comparison of Energy Consumption



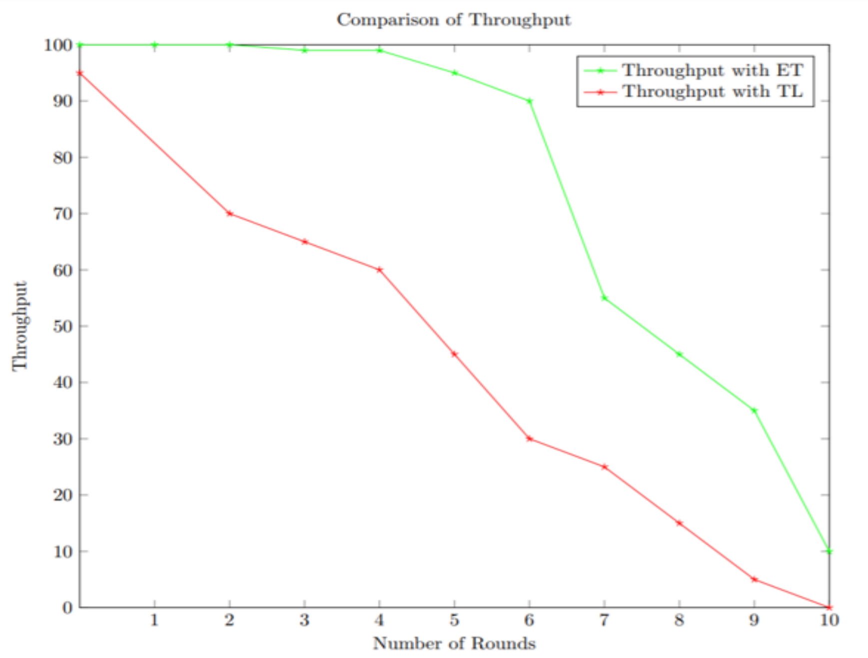
B. Throughput-It is measured by calculating the number of packets transferred per unit of time. The extent of data sent by a transmitter or received by a receiver can be find out by using throughput.

Throughput = ((Count of packets received)/ (Count of packets sent)) *time

Fig 3 shows the comparison of Throughput where Throughput is maximum in case of Elastic Time Synchronization when compared to Time Lay Synchronization.

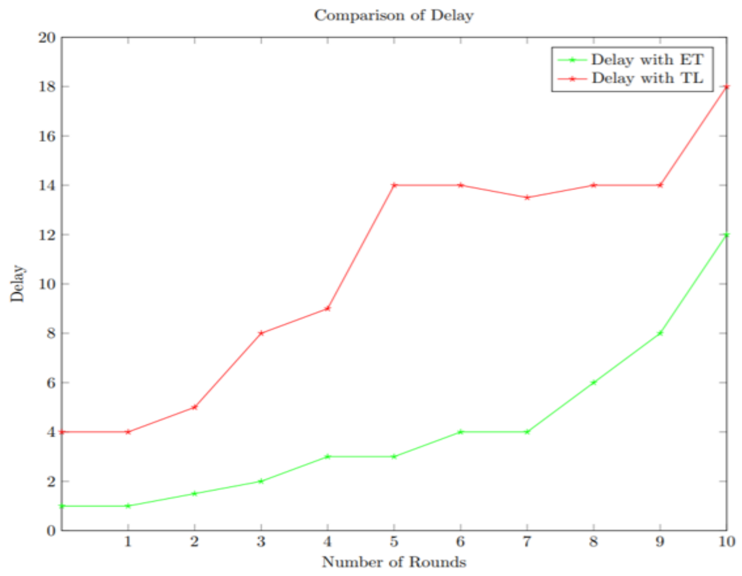
ISSN: 2237-0722
Vol. 11 No. 4 (2021)
Received: 24.05.2021 – Accepted: 21.06.2021

Figure 3 - Comparison of Throughput



C. Delay- The delay in a network can be calculated as a latency for one bit of data to travel across the network from one end point of communication to another. Delay is a performance characteristic that can be calculated in multiples or fractions of a second. As the number of rounds are increasing. Delay is minimum in case of Elastic Time Synchronization as compared to Time Lay Clock Synchronization. Fig 4 shows comparison of Delay in Elastic Time synchronization with Time-Lay Clock Synchronization.

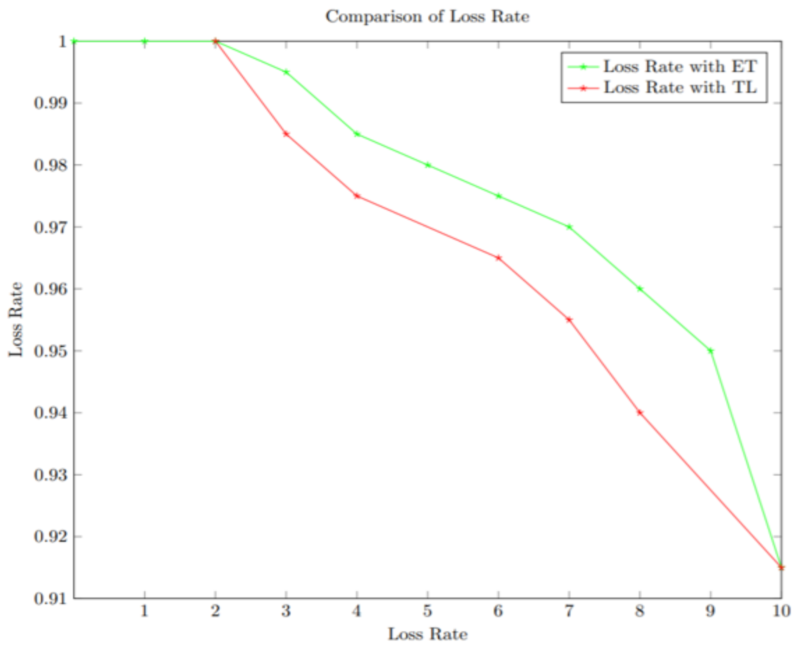
Figure 4 - Comparison Graph of Delay



ISSN: 2237-0722
Vol. 11 No. 4 (2021)
Received: 24.05.2021 – Accepted: 21.06.2021

D. Loss Rate- Loss Rate represents the ratio of count of lost packets to the total number of packets sent. Deadline is assigned to every packet before it starts executing, the objective is to minimize the count of lost packets due to expiry of the deadline. To evaluate the loss performance, a fraction of dropped packets should be analyzed. The fraction of loss packets is smaller in case of Elastic Time Synchronization as compared to Time Lay Clock Synchronization. Fig. 5 shows the comparison of Loss Rate in Elastic Time Synchronization and Time-Lay Clock Synchronization.

Figure 5 - Comparison Graph of Loss Rate



6. Conclusion and Future Scope

Elastic Timer Protocol was proposed in this paper that ensures efficient and perfect time synchronization strategy to function any IoT application. As compared to proposed Elastic Timer Protocol, there are various other protocols like NTP, PTP, SNTP, DTP and MNTP that uses GPS and works on the principle of Time Lay Clock Synchronization. The differentiation on the basis of various performance metrics like Energy Comparison, Throughput, Loss Rate, Delay was also shown for both the techniques. The proposed Elastic Timer Protocol ensures the efficient time synchronization for the operation of any wireless IoT application. The proposed Elastic Timer Protocol increases the elasticity of the protocol, with the help of which, optimality is guaranteed that enables the network configuration in terms of energy consumption and convergence time as compared to Time-Lay Clock Synchronization. It is analyzed and concluded that the proposed technique performs well in terms of calculated parameters and exhibit superior performance characteristics. In Future work, the proposed technique will be used and get evaluated for the testing of performance for more parameters or other synchronization protocols.

References

CASAGRAS. Project Final Report: RFID and the Inclusive Model for the Internet of Things; Technical report European Union Framework 7 Project 216803; European Commission: London, UK, 2009.

Moreno, M.F.; Cerqueira, R.; Colcher, S. Synchronization Abstractions and Separation of Concerns as Key Aspects to the Interoperability in IoT. *In Interoperability, Safety and Security in IoT; Springer: Cham,* Switzerland, 2016; pp. 26–32.

K. Marzullo and S. Owicki. 1983. Maintaining the Time in a Distributed System. *In ACM PODC.*

ACM IMC. DOI: Link 25. D.L. Mills. 1981. DCNET Internet Clock Service. Link. (April 1981).

D.L. Mills. 1985. Algorithms for Synchronizing Network Clocks. Link. (1985).

Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* 2013, 29, 1645–1660.

Wu, Y.C.; Chaudhari, Q.; Serpedin, E. Clock synchronization of wireless sensor networks. *IEEE Signal Process. Mag.* 2011, 28, 124–138. [CrossRef]

Savaglio, C.; Pace, P.; Aloi, G.; Liotta, A.; Fortino, G. Lightweight reinforcement learning for energy efficient communications in wireless sensor networks. *IEEE Access* 2019, 7, 29355–29364. [CrossRef]

Allan, D.W. Time and frequency(time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* 1987, 34, 647–654. [CrossRef] [PubMed]

Van Greunen, J, & Rabaey, J. (2003). Lightweight time synchronization for sensor networks. *Presented at the WSNA '03: Proceedings of the 2nd ACM international conference on wireless sensor networks and applications.* https://doi.org/10.1145/941350.941353.

Swain, A. R., & Hansdah, R. C. (2015). A model for the classification and survey of clock synchronization protocols in WSNs. *Ad Hoc Networks, (27),* 219–241. https://doi.org/10.1016/j.adhoc.2014.11.021.

Wu, Y.-C., Chaudhari, Q., & Serpedin, E. (2011). Clock Synchronization of Wireless Sensor Networks. *IEEE Signal Processing Magazine*, *28*(1), 124–138. https://doi.org/10.1109/msp.2010.938757.

Libri, A., Bartolini, A., Cesarini, D., & Benini, L. (2018). Evaluation of NTP/PTP fine-grain synchronization performance in HPC clusters. *Proceedings of the 2nd Workshop on AutotuniNg and ADaptivity AppRoaches for Energy Efficient HPC Systems - ANDARE '18*. https://doi.org/10.1145/3295816.3295819.

Lee, K. S., Wang, H., Shrivastav, V., & Weatherspoon, H. (2016). Globally Synchronized Time via Datacenter Networks. *Proceedings of the 2016 Conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*. https://doi.org/10.1145/2934872.2934885

F. Cristian, “Probabilistic clock synchronization,” *Distributed Computing,* vol. 3, no. 3, pp. 146–158, 1989.

R. Gusella and S. Zatti, *“The accuracy of the clock synchronization achieved by tempo in berkeley unix 4.3bsd,”* Tech. Rep. UCB/CSD-87-337, EECS Department, University of California, Berkeley, Calif, USA, 1987.

L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” *Communications of the ACM*, vol. 21, no.7, pp. 558–565, 1978.

C. Fidge, “Logical time in distributed computing systems,” *Computer*, vol. 24, no. 8, pp. 28–33, 1991.

E. Tuncer, J. Cortadella, and L. Lavagno, “Enabling adaptability through elastic clocks,” *In Proc. ACM/IEEE Design Automation Conference,* Jul. 2009, pp. 8–10.

T. Inzerilli, A. M. Vegni, A. Neri, and R. Cusani, “A Location-based Vertical Handover algorithm for limitation of the ping-pong effect”, *In Proc. of IEEE International Conference on Wireless & Mobile Computing, Networking & Communication,* vol. 14, no. 7, pp. 108-117, 2008.

A. Tekeoglu, A. Tosun, “A Testbed for Security and Privacy Analysis of IoT Devices”, *In Proc. of IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS),* vol. 13, pp. 343-348, 2016.

I. Nasr, L. Atallah, S. Cherif and B. Geller, “Time synchronization in IoT Networks: Case of a Wireless Body Area Network”, *In Proc. of IEEE Internet of Things Journal,* vol. 14, no. 5, pp. 864-949, 2016.

T. Jia et al., "An Adaptive Clock Management Scheme Exploiting Instruction-Based Dynamic Timing Slack for a General-Purpose Graphics Processor Unit with Deeo Pipeline and Out-of-Order Execution", *ISSCC,* pp. 318-319, Feb. 2019.

Herbst, N.R.; Kounev, S.; Reussner, R. Elasticity in Cloud Computing: What It Is, and What It Is Not. *In Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13),* San Jose, CA, USA, 26–28 June 2013; USENIX: San Jose, CA, USA, 2013; pp. 23–27. [Google Scholar]

Lehrig, S.; Eikerling, H.; Becker, S. Scalability, Elasticity, and Efficiency in Cloud Computing: A Systematic Literature Review of Definitions and Metrics. *In Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA ’15),* Montreal, QC, Canada, 4–8 May 2015; ACM: New York, NY, USA, 2015; pp. 83–92. [Google Scholar]

Mohindru, Vandana, Yashwant Singh, and Ravindara Bhatt. "Securing wireless sensor networks from node clone attack: a lightweight message authentication algorithm." *International Journal of Information and Computer Security* 12, no. 2-3 (2020): 217-233.

V. Rai, K. Mehta, J. Jatin, D. Tiwari and R. Chaurasia, “Automated Biometric Personal Identification-Techniques and Applications”, *4th International Conference on Intelligent Computing and Control Systems (ICICCS),* Madurai, India, 2020, pp. 1023-1030, doi: 10.1109/ICICCS48265.2020.9120896.

Mehta, K. et al. “Implementation of efficient clock synchronization using elastic timer technique in IoT.” *Advances in Mathematics: Scientific Journal* 9. 6(2020): 4025-4030.

D. Marwaha and A. Sharma, “A review on approximate computing and some of the associated techniques for energy reduction in IOT,” *in Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC* 2018, 2018, pp. 319–323.