CHAPTER 7

SIMULATION OF ACCURATE WEIGHTED AVERAGE SYNCHRONIZATION ALGORITHM

Chapter 7

Simulation of Accurate Weighted Average Synchronization Algorithm

7.1. Introduction

Any network consists of multiple elements and these elements need to function in certain fashion as designed. While all the network elements are required to function as designed, it may not be always possible to design a perfect system where every element functions perfectly. Hence, certain deviation within a limit may be acceptable. The size of this acceptable limit is an important parameter in deciding the robustness of the network and its elements. The elements of a network need to have some sorts of communication among themselves for the network to function. One of the most important information to communicate is how to share notion of time among these network elements. All the elements need to have their own notion of time and these notions of time (of the elements) need to be within certain acceptable window i.e. these elements need to have some common notion of time. Once these elements have some common notion of time, they need to maintain this common notion of time periodically. The process to ensure the network elements have common notion of time and the periodic maintenance of the same is nothing but network synchronization. Synchronization of communication network is one of the most important parameter for the network to function.

There are many network synchronization algorithm available like Egocentric averaging function [Pfluegl and D. M. Blough, 1985], Various Averaging methods [Jennifer Lundelius Welch and Nancy Lynch, 1988; Pfluegl and D. M. Blough , 1995], Midpoint function [Riccardo Gusella and Stefano Zatti, 1986], non-averaging convergence technique [Stephen R. Mahaney, Fred B. Schneider, 1985; Joseph Y. Halpern, et al, 1984; F. Cristian, H. Aghili, and R. Strong, 1986] and Sliding window method [Pfluegl and D. M. Blough, 1995] . Many modern synchronization algorithm uses GPS based solution [B. Sterzbach, 1997; J. J. Garnica, et al, 2010]. Network Time Protocol (NTP) [. D. L. Mills, 2006],

---

[4]The work presented in this chapter has been published in "**Accurate Weighted Average Synchronization Algorithm: Analysis and Simulation**". High Technology Letters. ISSN: 1006-6748. Volume 27, Issue 7, 2021. Scopus (UGC CARE LIST JOURNAL).

Precision Time Protocol v1 (PTPv1) [IEEE Standard 1588-2002], Precision Time Protocol v2 (PTPv2) [IEEE Standard 1588-2008] are also popular. Accurate Weighted Average Synchronization Algorithm (AWASA) is an unique algorithm designed to ensure accurate and precise synchronization which uses universal clock from sources like GPS/GLONASS/ IRNSS/Galileo clock as part of its algorithm. In this paper we intent to analyse the algorithm and conduction simulation of this algorithm.

7.2. Description And Analysis Of The Algorithm: AWASA

AWASA is a network clock synchronization algorithm which gives both an accurate and précised clock synchronization across a communication network. For description of the algorithm, let us first discuss the system model including network architecture, malfunction model and then the working of the proposed algorithm.

A. NETWORK ARCHITECTURE

The network used for description of the algorithm is a distributed hieratical network. The network can be fully or partially connected network. The network has three hierarchies namely Reference Layer (RL), Pseudo Reference Layer (PRL) and Non Reference Layer (NRL). In the RL there are few Master Nodes (MN), these MN have access to universal clock sources like GPS/GLONASS/ IRNSS/Galileo clock inputs for their clock synchronization. The MN are generally inter-connected and also share clocks input among themselves. Any MN is connected to two or more PMN in the PRL. In the PRL there are multiple PMN which are connected to two or more MN of RL and multiple Nodes in the NRL below. The PMN are also connected to other PMN in the PRL. Lastly in the NRL there are multiple Nodes which are connected their neighbouring Nodes and atleast two PMN in the PRL above. The network diagram is given below as fig. 1.

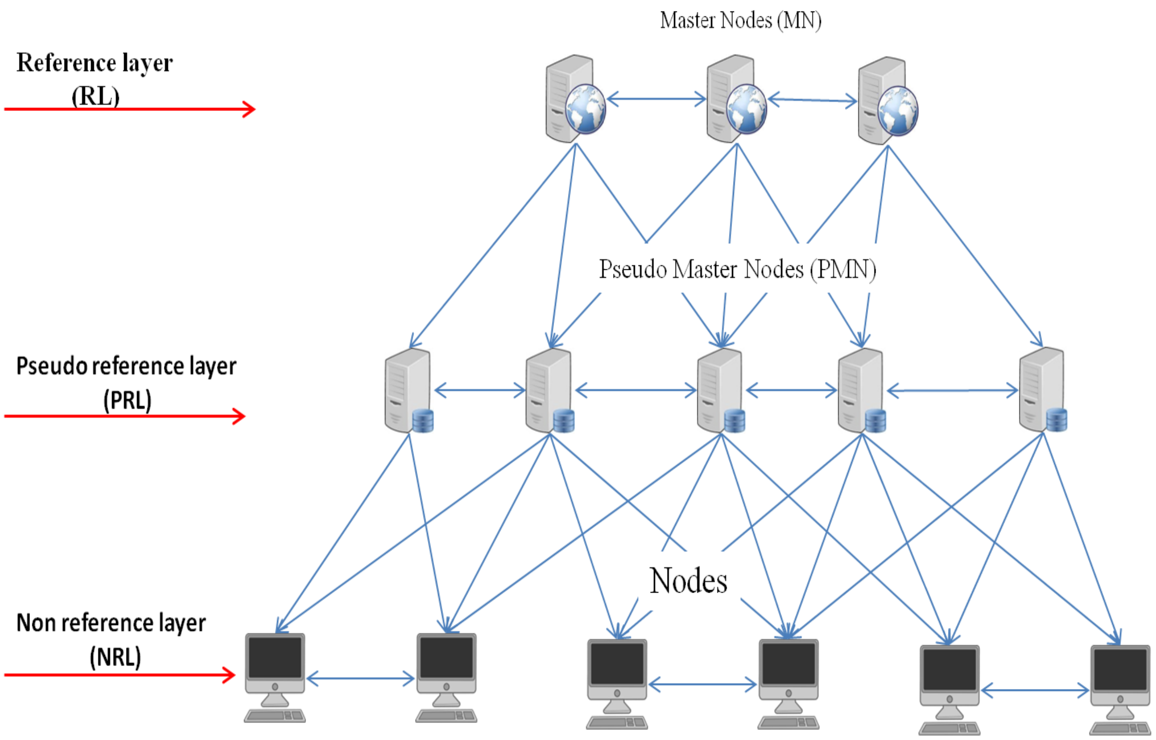

Fig. 7.1: Network Diagram of AWASA

B. MALFUNCTION MODEL

There are many type of faults encountered in a network. Some of the fault may be due to failure of clock in the MN/PMN/Nodes, failure of links between different types of nodes, failure of nodes as a whole or combination of two or more failures. In discussion of the algorithm, failure of clock of the nodes emphasized. Clock failure can due to simple mechanical failure of the physical clock of the node or some part of the same. In this scenario, the clock simple given incorrect or no clock input to other nodes. There is another type of clock failure in which a node gives conflicting or different clock values to different nodes at the same time. This malicious behaviour of the clock can be due to various factors. The effect of this malicious node/clock can have catastrophic clock failure in the network if such clock is not corrected or isolated in time. If presence of such malicious clock is beyond 33% then attaining clock synchronization may not be possible. This algorithm works in presence of such malicious clocks if the same is below 33%.

C. WORKING OF AWASA

The synchronization process starts when the Master node (MN) is at time t_s. The MN sends a synchronization request to all Pseudo Master Node (PMN). This communication is between RL and PRL. PMN will ack and reply if its clock state is between t_{rs} and t_{re} after due verification. This is the start of the resync process. Whenever the ACK is received by the MN it sends its clock value to all PMN to which it is connected. The MN is already in sync with trusted clock value as it has access to universal clock sources like GPS/GLONASS/ IRNSS/Galileo clock inputs for their clock synchronization. Now on receiving the clock value, the PMN correct their own clock value as to the new clock value and thus in sync with MN. The resynchronization process follows the IEEE 1588 standard. During the t_{pref} phase all the PMN required to be synchronized among them. The PMN runs WASA and arrive at a common clock value. Now in the t_{nref} this common clock value is shared with all connected Nodes using with IEEE 1588 standard. Now at the NRL, all nodes replaced their own clock value by the newly received clock value.

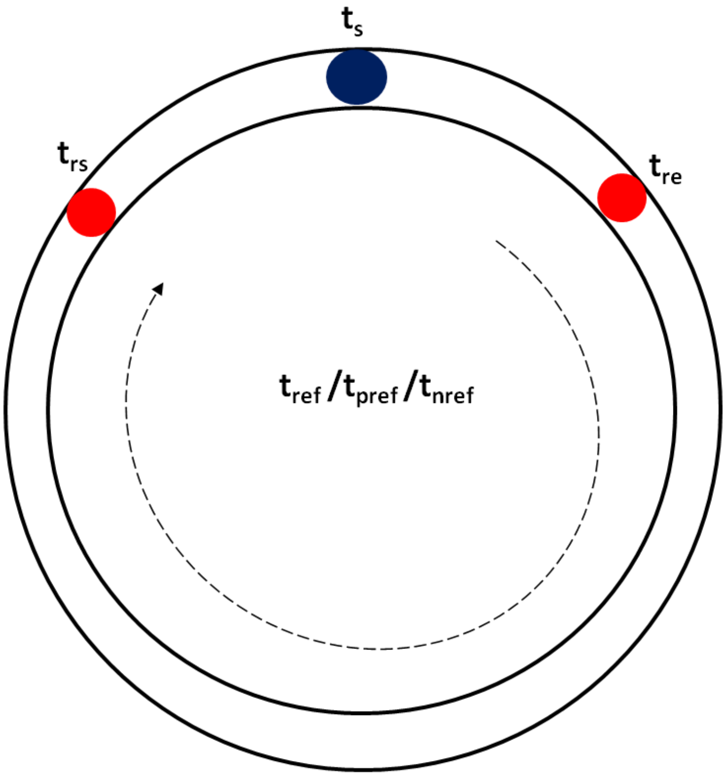

Fig. 7.2: Time line within a resynchronization period

7.3. SIMULATION

Simulation is process of creating a model of a proposed or existing solution/ algorithm/ plan and try, identify, understand and may be subsequently validate various factors involved to basically predict behavior of the proposed solution/ algorithm. Simulation can be done using various means and methods. Simulation of algorithm to test/ predict behavior of communication network can be carried out using software, hardware or combination of both.

For testing the behavior of AWASA we have created a simulation model using combination of software and hardware. A probabilistic simulation environment is created to validate AWASA in various conditions. In our simulation environment the basic parameters numbers of MN, PMN & Nodes and their clock value, resynchronization time, granularity of various clocks and maximum allowable clock skew/ drift.

The initial clocks values of MN at the start of the simulation are obtained from GPS clock input which are fed to the system. The clock values of PMN and nodes are assigned by a probability distribution function. One of the uncertainties of physical clocks is the drift rate. Clock drift rate is generally different from each other. Clock manufacturers usually provide clock drift rate. Manufacturers also generally specify clock drift range. The drift range is generally given as $[-\rho, +\rho]$, here ρ being maximum drift rate. If any clocks start drifting outside this window we considered the clock faulty. The maximum drift rate (ρ), resynchronization time (R) and the maximum skew (δ) are related by the equation: $2\rho \times R = \delta$.

A. INPUT DATA ANALYSIS

When designing a system/ algorithm, we need an assessment of what is expected from our proposed algorithm/ system. Hence, we generally chose system and design parameters according to our expected system environment. The outcome is than studied to check that the algorithm/ system is performing as predicted. Accordingly, in our simulation environment two sets of parameters are chosen namely System and Design parameters.

B. SYSTEM PARAMETERS

Here we defined various parameters, which are likely to be encountered in our communication network environment. Parameters like maximum skew, maximum drift, numbers of MN/ PMN/ Nodes, Temporal Granularity, numbers of bad nodes/clock, network topology etc. Details of the parameters taken for the simulation are given below in Table 7.1.

Ser no	Parameters	Quantity
1	**Maximum Skew, δ** (Maximum skew allowed is 0.1 μSec and the mean temporal distance between any two clocks should be maximum of 10 μSec)	**10 μS**
2	**Maximum Drift, ρ** (Maximum skew allowed is 0.1 μSec and the mean temporal distance between any clocks and reference clock should be maximum of 0.1 μSec)	**0.1 μS**
3	**Precision**	**< 10 μS**
4	**Accuracy**	**< 10 μS**
5	**Temporal Granularity**	**10 ns**
6	**Number of MN**	**> 2**
7	**Number of PMN**	**> 4**
8	**Number of Nodes**	**> 100**
9	**Number of Bad clock**	**Maximum of 40 %**
10	**Number of Malicious Clock**	**Maximum of 40 %**
11	**Network Topology**	**Partially connected**

Table 7.1. Simulation System Parameters

C. DESIGN PARAMETERS

Design parameters are those parameters, which are generally defined to set realistic limits or extend of various system parameters already defined. Design parameters try to ensure the simulation is functioning within certain pre-defined limits beyond which faithful deduction of the simulation result may not be possible. These definitions draw towards realistic network environment for the simulation. The design parameters defined are initials drift and skew, drift and skew rates, distribution of drift and skew, resynchronization time etc. Details of the design parameters are as given in table 2 below.

Ser No	Parameters	Quantity	Remarks
1	**Initial drift** (Drift consist of two parameters namely step size, d_s and step frequency, d_f ; Drift rate $\rho = d_s X\ d_f$)	**<10 ns**	
2	**Initial Skew** (Skew is related to precision of the network and initially all the nodes are in sync)	**0**	
3	**Distribution of drift and skew rates** (Drift & Skew rates of all the clocks are distributed Normally)	**Normal Distribution, *N*(μ,σ)**	
4	**Drift range** (It is the drift within which all the normal clocks are functioning)	$[-\rho_{max}, \rho_{max}]$	$\rho_{max} = 10\mu s$
5	**Skew range** **(It is the skew within which all the normal clocks are functioning)**	$[-\delta_{max}, \delta_{max}]$	$\delta_{max} = 10\mu s$
6	**Resynchronization time period, *t***	**100 μs**	

	(Resynchronization process is done in rounds; Each round is of the duration t; obtained from the relation $2\rho_{max}t = \delta$)		

Table 7.2. Simulation Design Parameters

A realistic simulation environment is created as far as possible by using various system and design parameters already defined above. The skew and drift rates of all clock varies from one another. The skew and drift rate of good clocks are within the range mentioned above in the table and they are Gaussian distributed. The skew and drift rate of the bad clocks are beyond the pre-defined, and the rates are determine using a random function. The rates of the malicious clocks are within the but they have to two properties: (i) malicious clock may also have different value while all normal clock value is as defined in the parameters; (ii) malicious clock may vary its skew and drift rate at every clock tick. This designed takes help of a random walk process. Here every next step of the malicious clock is defined by another random function.

The simulation carried by testing for multiple network sizes as defined in the parameters above. The sets of parameters are simulated multiple times in real time of varied duration ranging from 10000 seconds to 30000 seconds. The simulation is also done under various network conditions in terms of density of bad and malicious clocks. We now will discuss the results of the simulation.

7.4. Results and Discussions

Simulation of AWASA is carried out as described above using system and design parameters defined. Various aspects of the simulations including behaviours of the algorithm is being discussed in this sub section. The aim of this discussion/ analysis is to examine how AWASA behave under different network conditions and to see whether the algorithm actually offer tighter precision and better accuracy in term of clock synchronization. We will test the algorithm in various fault environments in presence of bad clock, malicious clock and combination of the both in varying percentage. We will also compare the algorithm with contemporarily clock synchronization algorithm.

The simulation has been carried out for various different set of system and design parameters under different network conditions. The numbers of nodes taken varies but satisfying the system parameters already defined. The simulation has been run multiple times for upto 30000 seconds, each simulation we assigned varied resynchronization periods. The results obtained are tabulated and graphs of the same are presented along with simulation result of WASA under similar simulation environment. Every data point that has being plotted in the graph gives the estimated value of the algorithms output of interest i.e. Accuracy and Precision. The data points are the means of multiple simulations output. The error margins are being plotted in the graph. When we run the simulation n runs and we want to ensure a 95% confidence interval, for the expected accuracy and precision, E(x), then we use the formula:

$$\bar{\mathrm{I}}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \frac{\sqrt{S}}{\sqrt{n}},$$

Here $\bar{\mathrm{I}}(n)$ is the sample mean for n runs, S is the variance of the samples and $t_{n-1,1-\frac{\alpha}{2}}$ is the t-distribution parameter for degree of freedom n-1. The confidence interval is calculated for 95% confidence level for every data point obtained for in all the various simulation environments. We have presented the same with the results.

7.4.1. Synchronization in Low Fault Environment

In this simulation setup, the network has less than 20% faulty clocks, combination of both bad and malicious clocks. The network created represents many real networks/ applications. We analyse the relationship between synchronization vis-a-vis the number of total combined faulty clocks. The result of the simulation in as given in figure 3 below.

As one as inferred from the graph AWASA offers a tight synchronization throughout the range of faulty clocks. These particularly true for faulty clocks percentage from 0% - 8%, where the synchronization time is less than 0.2 µs. As the percentage of faulty clocks raises there is a slightly increase in the synchronization time but this to be expected. However, the rise is linear and maximum synchronization time is less than 0.6 µs, which is much lesser than contemporary algorithm. We have also given out simulation result of WASA as plotted in the graph. The improvement of synchronization is quite significant as evident from the graphs. The result indicates that AWASA offers tight synchronization in presence of low combined fault. Average confidence interval obtained from the simulation is [0.013µs– 0.026 µs] with 95% confidence level. The trend lines indicate lesser deterioration in case of AWASA as compare to WASA.

Hence, the synchronization achieved is both accurate and précised in presence of low combined faults.

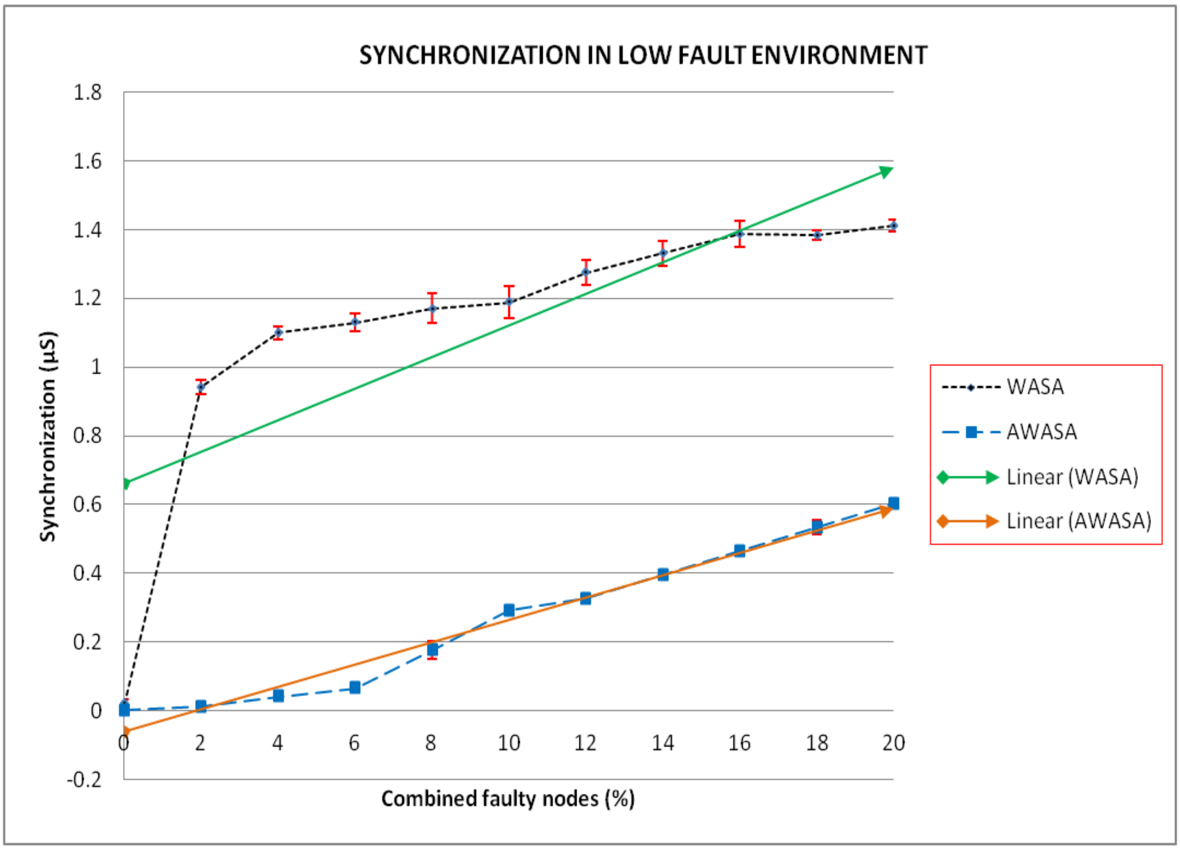

Fig. 7.3. Synchronization in Low Combined Fault Environment

7.4.2. Synchronization in High Fault Environment

Here in this simulation setup, the network has faulty clocks in the range of [22% to 40%], combination of both bad and malicious clocks. The network created represents many real networks/ applications. We analyse the relationship between synchronization vis-a-vis the number of total combined faulty clocks. The result of the simulation in as given in figure 4 below.

As one as inferred from the graph AWASA offers a tight synchronization upto 30 % of faulty clocks. However, the synchronization deteriorates significantly post presence 32% faulty clocks. In the faulty clocks percentage range from 22% - 30% where the synchronization time is less than 1.0 µs. As the percentage of faulty clocks raises there is a significant increase in the synchronization time but this to be expected, as the destabilizing behaviour of the malicious clocks come into play. We have also given out simulation result of WASA as plotted in the graph. The improvement of synchronization is quite significant as evident from the graphs as we compare the two algorithms. The result indicates that AWASA offers much better synchronization in presence of combined fault. Average confidence interval obtained from the simulation is [0.010µs– 0.105 µs] with 95% confidence level. The trend lines indicate lesser deterioration in case of AWASA as compare to WASA. Hence, the synchronization achieved is both accurate and précised in presence of high combined faults.

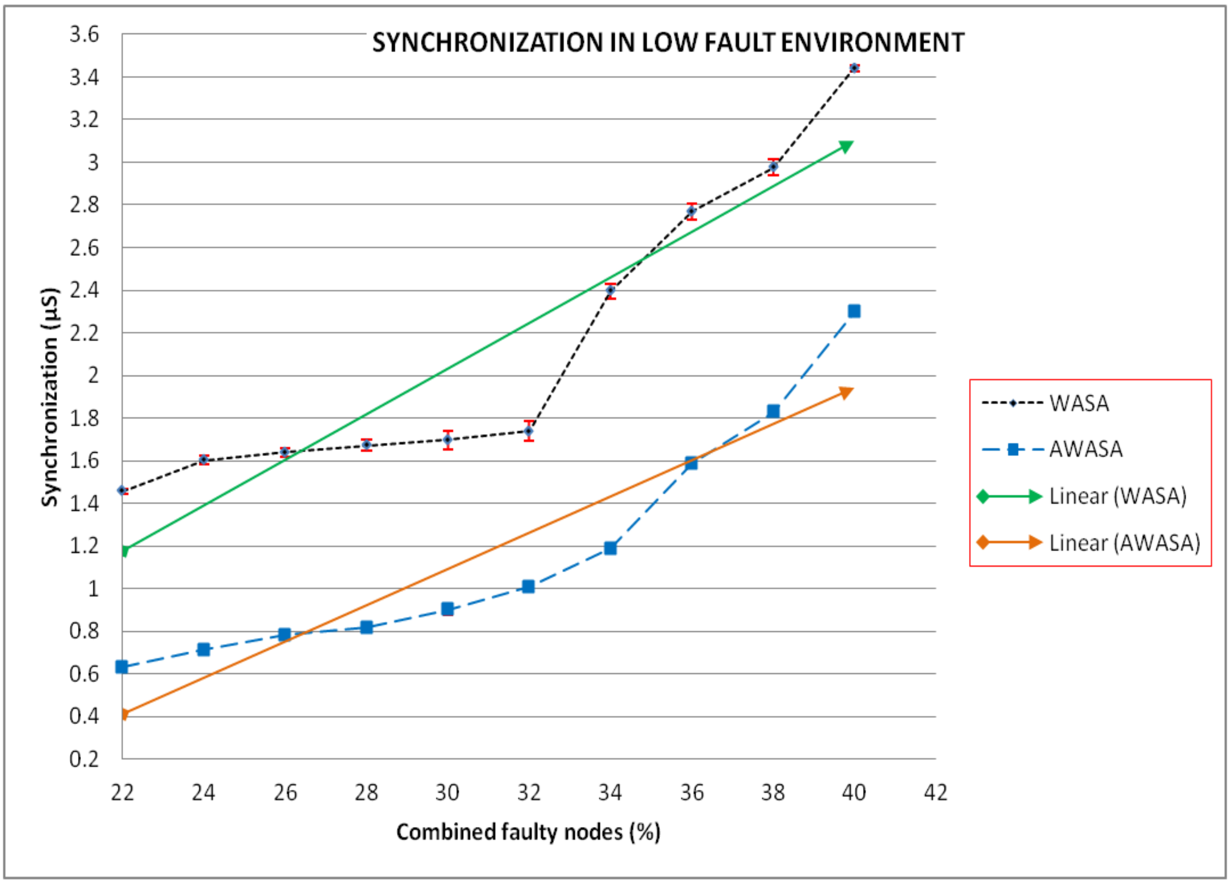

Fig. 7.4. Synchronization in High Combined Fault Environment

7.4.3. Synchronization in Presence Of Bad Clock

The network for this simulation has only bad clocks in the range of [0% to 40%]. We analyse the relationship between synchronization vis-a-vis the number of total bad clocks. The result of the simulation in as given in figure 5 below.

As we can see from the graph, that AWASA offers a tight synchronization upto 32 % of bad clocks. However, the synchronization deteriorates significantly post presence 32% faulty clocks. In the bad clocks percentage range from 0% - 32% the synchronization time is less than 0.29 µs. As the percentage of faulty clocks rises, there is a slight increase in the synchronization time but this to be expected. We have also given out simulation result of WASA as plotted in the graph. The improvement of synchronization is quite significant as evident from the graphs as we compare the two algorithms. The result indicates that AWASA offers much better synchronization in presence of combined fault. Average confidence interval obtained from the simulation is [0.012µs– 0.044 µs] with 95% confidence level. The trend lines indicate lesser deterioration in case of AWASA as compare to WASA. Hence, the synchronization achieved is both accurate and précised in presence of bad clocks.

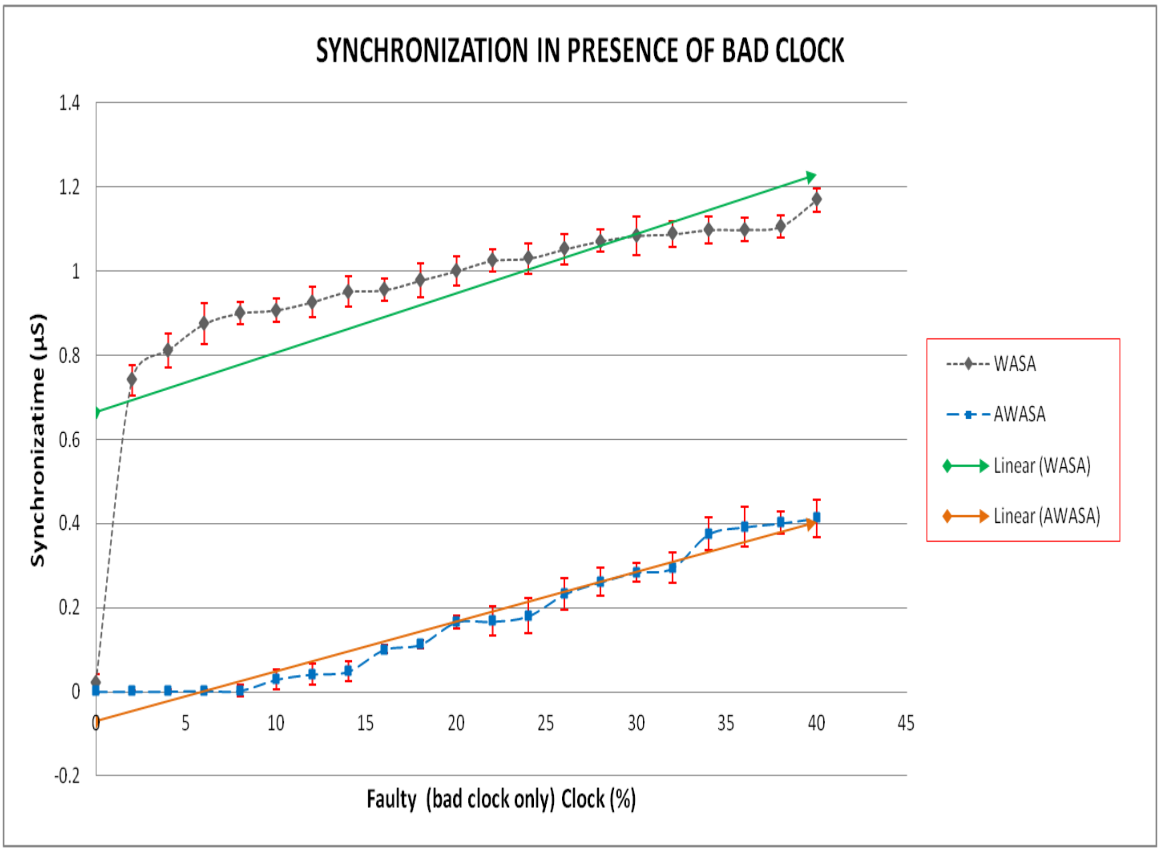

Fig. 7.5. Synchronization in presence of bad clocks

7.4.4. Synchronization in Presence Of Malicious Clock

We have discussed earlier that malicious clock is difficult to handle due to their destabilizing behaviour. Studying the effect of malicious is one of the major objective of this paper and hence the simulation. The simulation of a network has malicious clocks of varying proportion is carried out and the result of the same is plotted in figure 7. As we can make out from the graph that the algorithm offers tight synchronization upto range of 5% and then the synchrony deteriorates slightly upto 15 %. The deterioration increases more upto 30%, but beyond that the deterioration is quite significant as it rises exponentially. This is but expected knowing the destabilizing behaviour of the malicious clocks.

The improvement of synchronization is quite significant as evident from the graphs as we compare the two algorithms. The result indicates that AWASA offers much better synchronization in presence of combined fault. The trend lines indicate lesser deterioration in case of AWASA as compare to WASA. Hence, the synchronization achieved is both accurate and précised in presence of malicious clocks provided that range is within 30% to 33%.

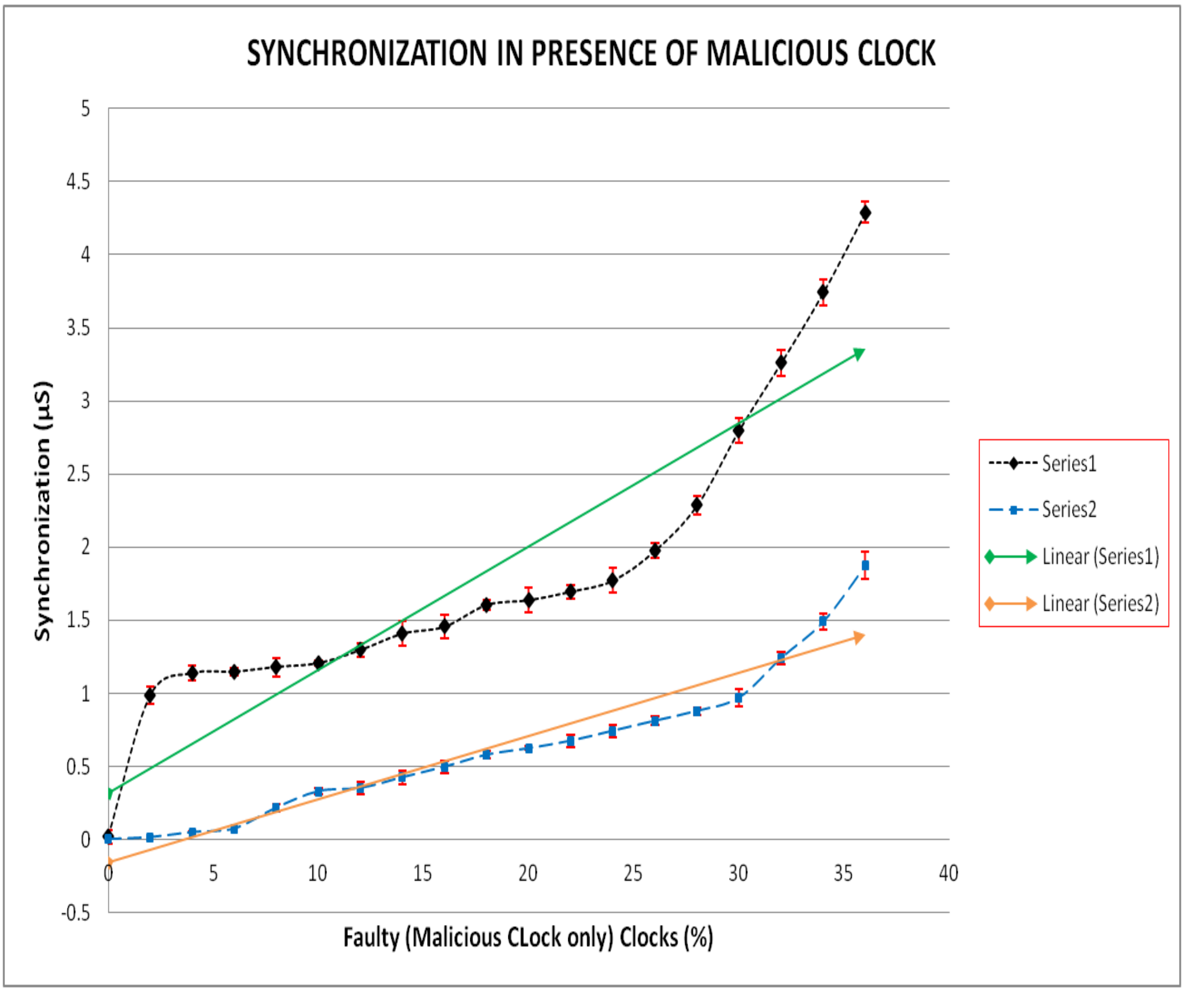

Fig 7.6. Synchronization in presence of malicious clocks

7.4.5. Conclusion

We have studied various algorithm and then concentrated on analysis and simulation of AWASA. We have also simulated WASA for comparison of the two algorithms. We have analysed the behaviour of AWASA in obtaining synchronization in various type of network environments satisfying the system and design parameters defined. The behaviour of AWASA and WASA are compared in the entire network environment created. From the various simulation results obtained it is very evident that AWASA gives better synchronization tightness in the entire network scenario created.

The behaviour of AWASA and WASA are compared in the entire network environment created. From the various simulation results obtained it is very evident that AWASA gives better synchronization tightness in the entire network scenario created. One of the difficult for the network to get synchrony. This observation is an important lesson, malicious clock should be identified, and controls, if possible, at least in critical network so as their destabilizing effect do not hamper the network operation.