

CHAPTER 5

ANALYSIS AND SIMULATION OF WEIGHTED AVERAGE SYNCHRONIZATION ALGORITHM

Chapter 5

Analysis and Simulation of Weighted Average Synchronization Algorithm

5.1. Simulation Model

Simulation is a process of creating a virtual model of a system using which various ideas or algorithms can be evaluated. In order to test and predict the behaviour of our algorithm we have created a simulation model. We have designed a probabilistic simulation environment to verify WASA in different conditions. The basic parameters of our simulation are number of clocks, resynchronization time, clock granularity and maximum allowable skew.

The clocks at the start of the simulation are specified by a probability distribution function. Another uncertainty is the drift rate clocks. The drift rate of clocks is always different from one another. The manufacturers of the clocks generally provide the drift rate of the clock. The manufacturers also specify the drift range of the clock. The drift range is usually given as $[-\rho, +\rho]$, where ρ is the maximum drift rate.

²The work presented in this chapter has been published in “**Analysis and Simulation of Weighted Average Synchronization Algorithm: A Novel Precise Clock Synchronization Algorithm**”. High Technology Letters. ISSN: 1006-6748. Volume 27, Issue 3, 2021. Scopus (UGC CARE LIST JOURNAL).

Clocks drifting outside this range are considered faulty. The maximum drift rate, resynchronization time and the maximum skew are related by the equation: $2\rho \times R = \delta$.

There are three major types of failure possible in a network namely (i) link failure (ii) node failure and (iii) Clock failure. Here we have considered only clock failure. In clock failure, we have design timing clock failure and malicious clock failure in the simulation model.

5.2. Input Data Analysis

For design of a system, we generally have to an assessment of what is expected from the proposed system. System parameters and system design is chosen so that the system start performing as expected. Similarly, in our simulation we have two sets of parameters: System parameters and design parameters. In the system parameters are we give our expectations in term of synchronization tightness and the network parameters. The parameters required to achieve this synchronization tightness in the given network are the design parameters. The system parameters are as given in the table 5.1. below:

Ser No	Parameters	Quantity
1.	Maximum Skew, δ <ul style="list-style-type: none"> ● The maximum skew of any two clocks should be 20 msec. ● At any point of time, any clocks should not deviate more than 10 msec from its value at the start of the resynchronization period. 	20 msec
2	Precision <ul style="list-style-type: none"> ● The clocks should be tightly synchronized ● This tightness will be measured at the end of the resynchronization process 	Should be must lesser than

	<ul style="list-style-type: none"> The precision offered should be much lesser the maximum skew 	Maximum Skew
3	Temporal Granularity <ul style="list-style-type: none"> The clocks should able to distinguish between two readings a second apart. 	1 msec
4	Number of nodes in the network, n <ul style="list-style-type: none"> The precision and maximum skew should be guaranteed in a network of at least 40 nodes 	40
5	Number of bad clocks <ul style="list-style-type: none"> The system should give the precision even in presence of 33% bad clock The Bad clocks may consist of timing failure and malicious type or entire of one type 	33% of n
6	Network topology <ul style="list-style-type: none"> The network topology is fully connected static network All links are bi-directional 	Fully connected static network
7	Fault tolerant <ul style="list-style-type: none"> The system should give the desired precision even in presence of faults 	

Table 5.1: System parameters for simulation

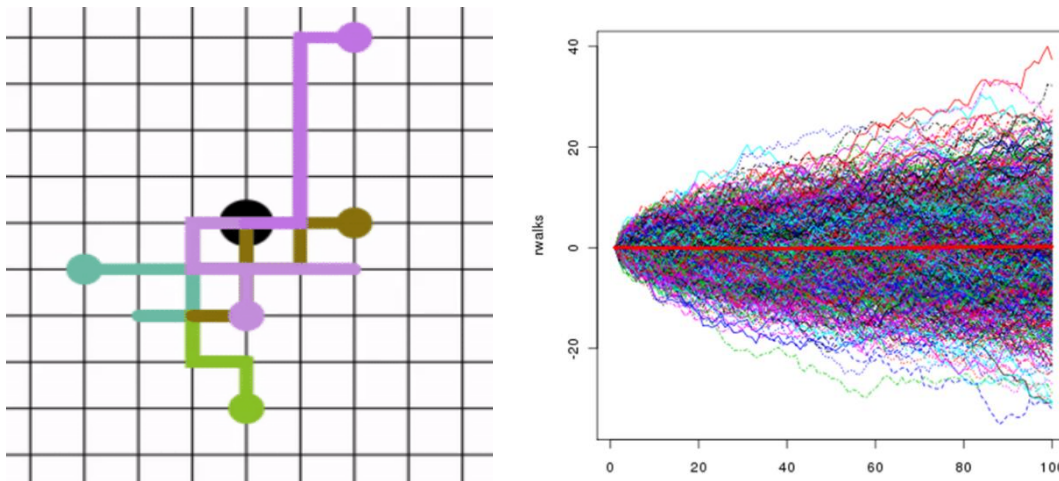
For design of the simulation model, we studied the system parameters and the requirement given and carefully decide the various design parameters so that the expectations from the system can be fulfil. The design parameters are listed in table 5.2.

Ser No	Parameters	Quantity	Remrks
--------	------------	----------	--------

1	<p>Initial drift</p> <ul style="list-style-type: none"> • Every clock has a drift rate • This drift rate is given by given by the manufacturer • The clock drift consist of two parameters namely step size, d_s and step frequency, d_f • Their values are given by the manufacture • Drift rate $\rho = d_s X d_f$ 	<111 microsec /sec	
2	<p>Distribution of drift rates</p> <ul style="list-style-type: none"> • The drift rates of all the clocks are distributed Normally 	Normal Distribut ion, $N(\mu, \sigma)$	
3	<p>Drift range</p> <ul style="list-style-type: none"> • The range within all the drift rates of the clocks resides • Given by manufacturer 	$[-\rho_{max}, \rho_{max}]$	$\rho_{max} = 111$ <i>microsec</i> /Sec
4	<p>Resynchronization time period, t</p> <ul style="list-style-type: none"> • The resynchronization process is done in rounds • Each round is of the duration t • t is obtained from the relation $2\rho_{max}t = \delta$ 	10 sec	

Table 5.2: Design parameters for simulation

We have tried to create a simulation model with an aim to make it realistic. We have made the drift rates of the different types of clocks different. The drift rate of good clocks are within the range $[-\rho_{max}, \rho_{max}]$ and within this range they are Gaussian distributed. The drift rate of the bad clocks are beyond the range $[-\rho_{max}, \rho_{max}]$ and the drift rate is determine using a random function. The drift rates of the malicious clocks are within the range $[-\rho_{max}, \rho_{max}]$ and have to two properties: (i) a malicious clock may have different clock value when different clock run the algorithm; (ii) a malicious clock may change it drift rate at every clock tick. This is achieved by using a random walk process where the next step of the malicious clock is decided by another random function. Pictorial description of the concept of Random Walk model is below.



h ave tested the simulation with 20, 30 and 40 nodes. In the simulation, for every set of parameters we have 10 runs for 20000 to 40000 seconds in real time. The detail discussion and analysis is given in the next page.

5.3. Results and Discussions

The simulation on Weighted Averaging Synchronization Algorithm is conducted as described above. The simulation results and analysis of these results are discussed in this chapter. The goal for this analysis is to examine the behaviour of WASA and to demonstrate

that this algorithm offers tighter precision while tolerating various faults. First, we give our analysis on the performance of the algorithm in different environments. Then the impacts of various failure on the clock synchronization is presented. Finally, we conclude by comparison of the simulation results with that of another algorithm.

We have conducted the simulation for wide varieties of parameters. In this series of simulations, we have taken a network for 20 and 40 nodes. We have conducted multiple runs for each set of parameters for 20000 to 40000 seconds. During each run, there are multiple resynchronization periods and we obtained a precision value for each resynchronization period. From these precision values, we have collected the worst precision and the same is used for plotting the graphs presented.

Each data point in the graphs represents the estimated value of the performance measure of interest i.e. Precision and it is the mean of multiple simulation experiments. The variance is also show in the plots. Suppose we choose n runs and want to obtain a 95% confidence interval for the expected precision, $E(x)$, then we can use the formula:

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

Where $\bar{I}(n)$ 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.

We have calculated the confidence interval for 95% confidence level for each data point obtained for the all simulation environments and the same is presented with the results at their respective sub-sections.

The following sub-section give the description of the simulation result for a set of parameters in a particular environment. All the parameters during the particular simulation is fixed except the percentage of fault.

5.4. Precision in low fault environment

In this section, we focus on the achievable synchronization tightness in low fault environment where few of the clocks are faulty. This environment is representative of many real applications. Here we analyse the relationship between precision and the amount of total fault. The result for 40 nodes network is presented at figure 5.1. in environment where few of the clocks are faulty. This environment is representative of many real applications. Here we analyse the relationship between precision and the amount of total fault. The result for 40 nodes network is presented at figure 5.1.

The graph shows a significant fall in the precision from 0% - 5% of combined fault percentage. The precision do not deteriorate much from 5% - 15%. Precision decrease only by 4% between 5% - 15% of combined fault percentage.

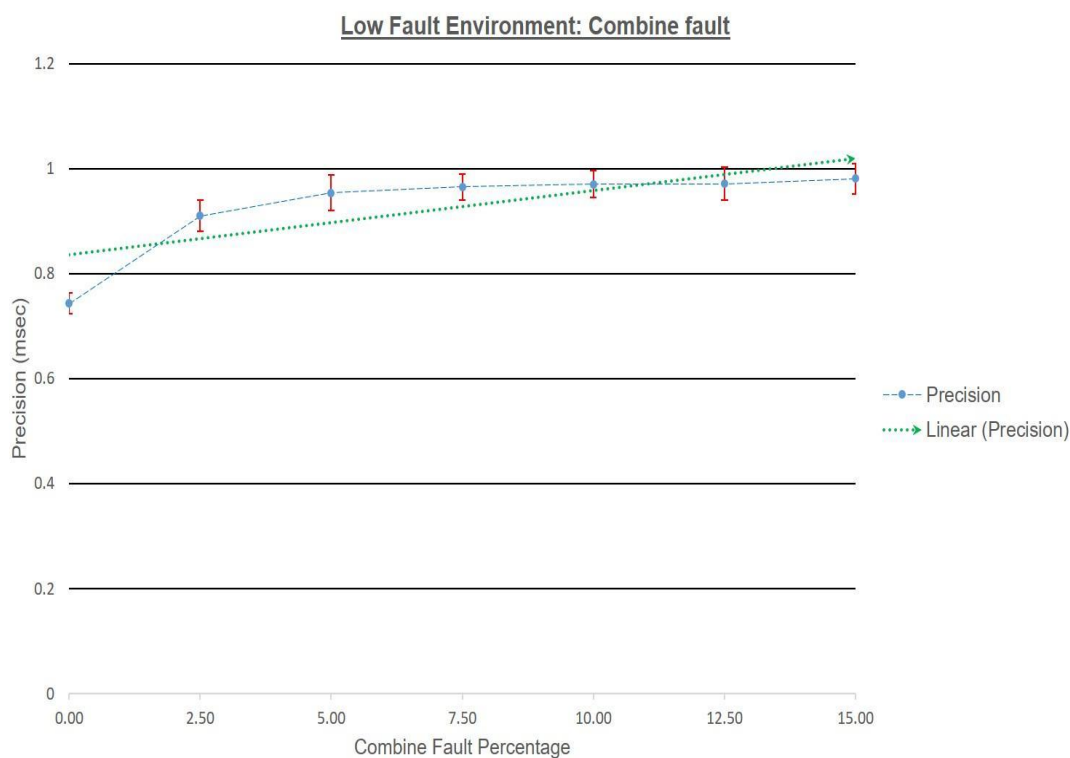


Figure 5.1: Performance of WASA in low fault environment

The precision given by WASA is below $1\mu\text{sec}$ for upto combined fault percentage of 15%. The result of the simulation indicate that the algorithm offer tight precision in presence of

low fault which is generally the case in a network. The average confidence interval of the samples is $[0.897\mu\text{sec} - 0.944\mu\text{sec}]$ with 95% confidence level.

5.5. Precision in High fault environment

Here we give the result and analysis of our algorithm in high fault prone environment. This type of environment may be rare especially the presence of large amount of malicious faults. The effect on precision offered by the algorithm with increase in the combined fault percentage is depicted in the figure 5.2 below.

The plot shows that the precision of the algorithm rapidly worsen at high combined fault percentage especially after 30% combined fault percentage. The average confidence interval of the samples is $[1.44\mu\text{sec} - 1.59\mu\text{sec}]$ with 95% confidence level.

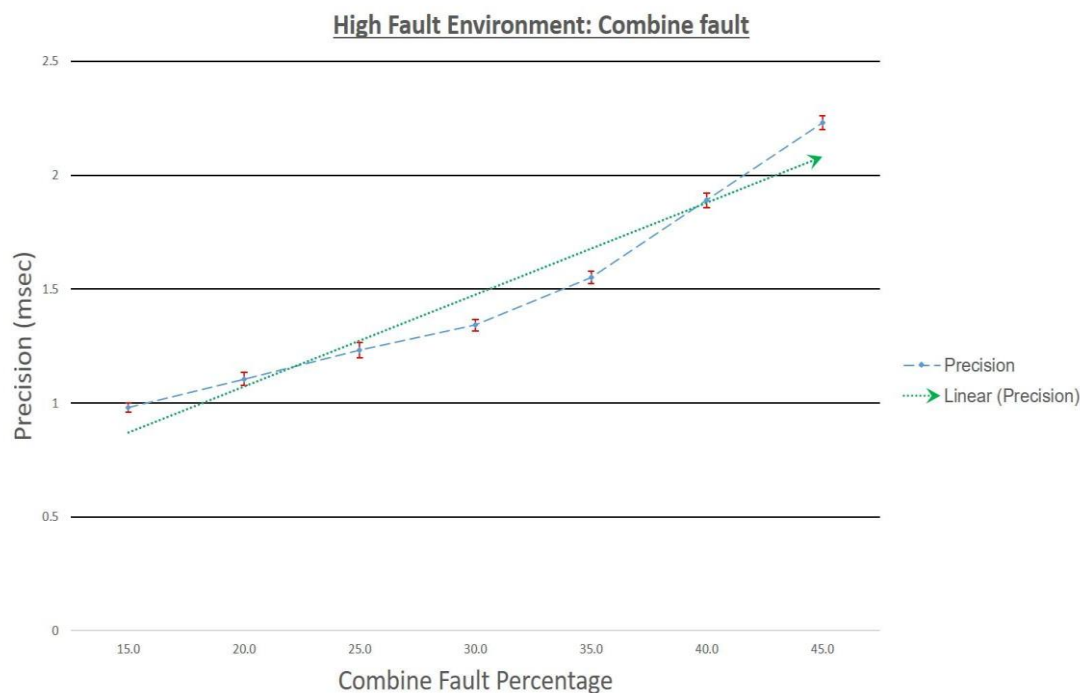


Figure 5.2: Performance of WASA in fault prone environment

5.6. Impact of timing Clock failure

The behaviour of the algorithm simulated when only timing clock failure type of fault is present is studied here. The plot of the same is given in figure 5.3. The precision offered by the algorithm deteriorate steadily as the amount of fault increase. There is significant fall in the precision as the fault percentage increase from 0% to around 5%. Thereafter the fall in precision is steady. The precision does not deteriorate much after 10%. The average confidence interval of the samples is $[0.949\mu\text{sec} - 0.995 \mu\text{sec}]$ with 95% confidence level.

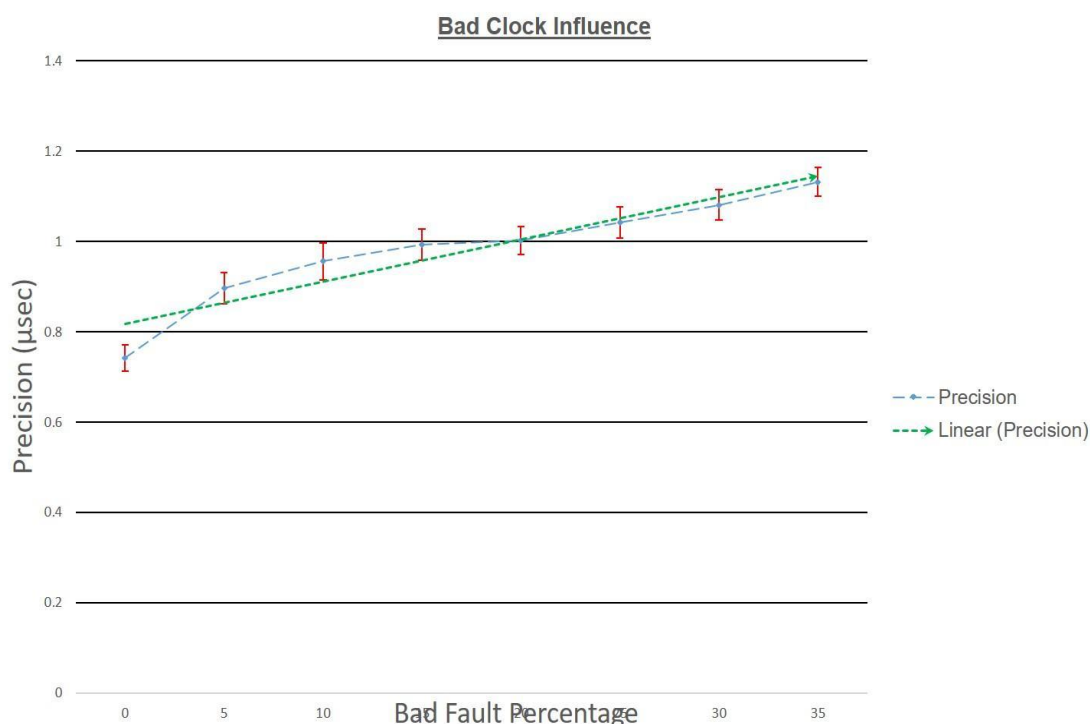


Figure 5.3: Relationship of Precision and bad clocks

5.7. Impact of malicious Clock failure

Malicious Clock are, as describe before, difficult to handle as they tend to destabilize the clock synchrony of the network. Thus, the effect of the malicious clocks on the precision provided by the algorithm is an important aspect of this simulation. The behaviour of WASA in presence of malicious clocks is plotted in figure 5.4. The trend of the graph shows that

the precision decrease by 11% between 10 % to 25% of malicious faults. Precision deteriorate by more than 37% between 30% - 40 % of malicious faults.

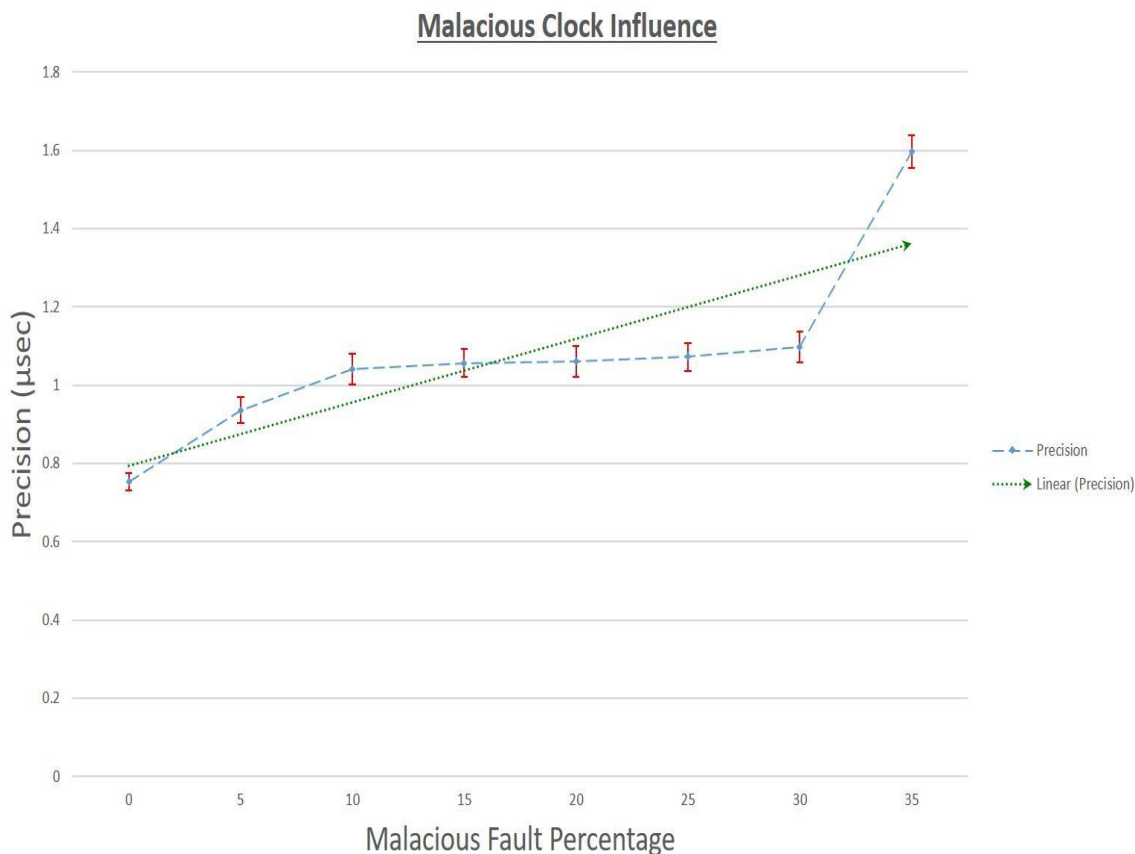


Figure 5.4: Relationship of Precision and Malicious clocks

The graph depict that the algorithm renders tight precision to the network for malicious fault up to 30%. The Precision get worse after 30%, which was expected. The average confidence interval of the samples is [1.07µsec – 1.11 µsec] with 95% confidence level. The average confidence interval of the samples is [1.07µsec – 1.11 µsec] with 95% confidence level.

5.8. Precision Comparison

We have simulated Sliding Window Algorithm proposed (SWA) in [Pfluegl and D. M. Blough , 1995]. The same simulation parameters used for simulation of WASA is taken for simulation of SWA. Figure 5.5 shows the plots of WASA and SWA in presence of malicious clocks. The trend of the plot reveals that WASA offer an improvement of around 30% in presence of 10% fault and also precision improve by more than 33 %

at 30% fault. The performance of WASA shows a significant improvement over at all range of malicious fault percentage. The performance of WASA, in terms of precision, is better than that of SWA by at least 19% at presence of 10% malicious faults. These enhancements in precision over SWA keep on strengthening as the percentage of fault increase. The precision given by WASA is around 60% better than SWA at 30% fault. The average confidence interval of the samples is [1.53 μ sec – 1.66 μ sec] for SWA with 95% confidence level.

Clock synchronization can be seen as a process of convergence of the clock values so as the clocks are brought within a bound. So it is importance to study how the clocks are converging during the synchronization process. We have done an analysis of the pattern of this convergence. The plot for the same is given in figure 5.6.

In the plot time variance show decaying pattern and finally get converged. The time variance at low fault environment converge the earliest and High fault environment took maximum time to converge. The plot shows that clocks at low fault environment synchronize the earliest as expected. Presence of high percentage of malicious faults disrupts time convergence.

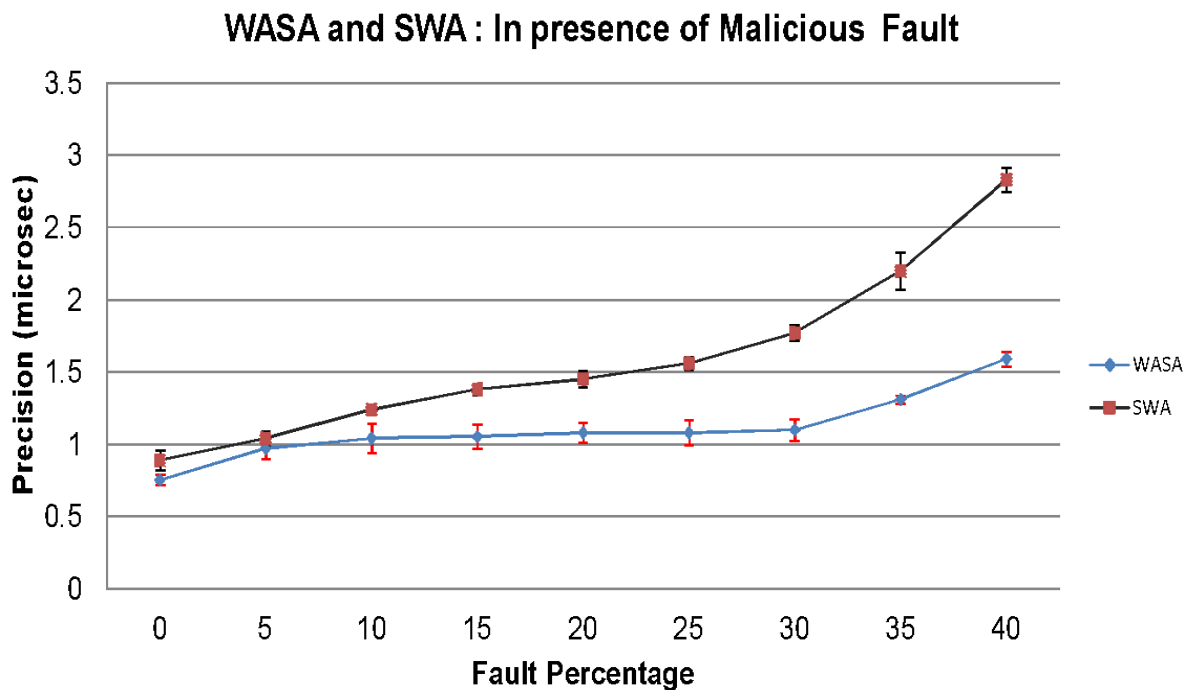


Figure 5.5: Performance of WASA and SWA in presence of malicious fault

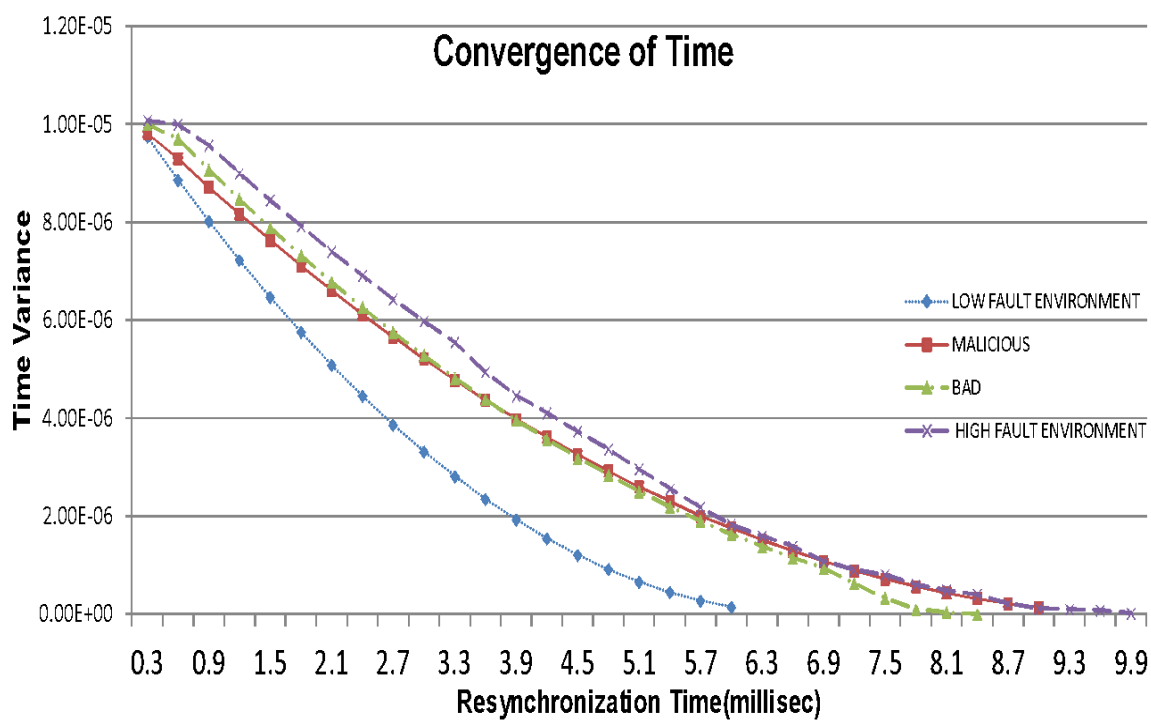


Figure 5.6 : Convergence of time at various environment

The study of the above plots of WASA shows that it gives tight precision. It can also be noted that the algorithm gives much better precision than SWA in presence of malicious

5.9. Conclusion

During the course of this study, we have done a thorough study of various methods of clock synchronization used in the distributed networks. One of the most challenging aspects of clock synchronization is making it fault tolerant even in the presence of malicious clocks.

The malicious clocks can disturb the synchrony of the clocks in the network if they are one third or more of the total clocks in number. Based on our study and analysis of the clock synchronization problem, we have developed a novel clock synchronization method. The algorithm, WASA, is a fault tolerant clock synchronization algorithm, which synchronizes clocks even in presence of faulty clocks including the malicious clocks. The algorithm gives improved precision as compared to some of the previously proposed methods. The WASA guarantees clock synchronization in presence of malicious clocks if the upper bound of malicious nodes is just under one-third of the network size. An improvement of 33% in precision is achieved in the worst-case scenario as compared to [Pfluegl and D. M. Blough, 1995]. Even though the worst-case scenario occurrence is very less but our study provides meaningful analytical insight. This analysis may have application in the high integrity systems especially in safety critical systems.

