A Theoretical Evaluation of Peer-to-Peer Internal Clock Synchronization*

Sirio Scipioni, Leonardo Querzoni, Sara Tucci Piergiovanni, Roberto Baldoni,
Sapienza University of Rome
Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
Rome, Italy
{scipioni | querzoni | tucci | baldoni}@dis.uniroma1.it

ABSTRACT

Synchronized clocks are usually considered as a prerequisite for many distributed applications. Existing solutions mainly deal with this problem in static environments with well defined characteristics and limits. The needs of an emergent class of large-scale peer-to-peer applications that have to operate without any assumptions on the surrounding environment have recently revitalized this research area with the proposals of new solutions characterized by self-organization capabilities and strong adaptability to dynamic settings.

This paper reports about the properties of a clock synchronization algorithm for large scale applications. The algorithm implements an internal clock synchronization mechanism which combines the gossip-based paradigm with a nature-inspired approach coming from the *coupled oscillators* phenomenon. Using a theoretical approach, the paper focuses on the convergence properties of the algorithm, characterizing its synchronization speed (decay factor) the final synchronization point and error.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications; G.3 [**Probability and Statistics**]: Probabilistic Algorithms; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms

Theory, Algorithms, Performance

Keywords

Peer-to-Peer Systems, Internal Clock Synchronization, Theoretical Analysis

*This work was partially supported by the European Network of Excellence ReSIST and by a grant from an agreement CINI-Finmeccanica.

Autonomics 2008, September 23 - 25, 2008, Turin, Italy

1. INTRODUCTION

Clock synchronization is a fundamental building block for many distributed applications. As such, the topic has been widely studied for many years, and several algorithms exist which address different scales, ranging from local area networks (LAN), to wide area networks (WAN). However there exists an emergent class of applications and services, operating in very challenging settings, for which the problem of synchronizing clocks has been attacked only recently [2, 12]. These applications have to operate without any assumption on deployed functionalities, pre-existing infrastructure, or centralized control, while being able to tolerate network dynamism, due to crashes or to node joining or leaving the system, and scaling from few hundred to tens of thousands of nodes. These new algorithms are built with self-organization capabilities and strong adaptability in order to support the adverse settings they are supposed to work on.

A common way to resolve the clock synchronization problem in absence of deployed functionalities such as external time sources is recurring to *convergence function-based* techniques. These techniques are based on two steps in which a node 1) estimates through a message exchange the clock value of other nodes obtaining a so-called *clock estimate* of other nodes; 2) uses a *convergence function*, which takes as argument a set of clock estimates and returns a single clock value, to adjust its local clock. A convergence function compute some kind of averaging on clock estimates and usually some of them are designed to tolerate erroneous clock estimates or faulty values. Consequently computing a mean is often a basing building block for most internal clock synchronization algorithms.

A promising approach to tackle this kind of problems is to embrace a fully decentralized paradigm in which peers implement all the required functionalities, by running so called $gossip-based$ algorithms. In this approach, due to the large scale and geography of the system, each peer is provided with a neighborhood representing the part of the system it can directly interact with. The algorithm running at each peer computes local results by collecting information from this neighborhood. These results are computed periodically leading the system to gradually compute the expected global result.

In this paper we present a theoretical analysis of the synchronization properties of a mean-based convergence function in a peer-to-peer system. While previous studies [2, 12] provide only simulation-based experimental evaluations of the protocol properties, we started from this basing building block of clock synchronization (i.e. a mean) in order to show

that is possible to provide theoretical properties in terms of mean and variance of the distribution of clocks. In particular, using a simple mean of differences between the clock value of a node and its neighbours as convergence function, we show two important properties: 1) convergence speed and synchronization error, in presence of errors induced by network perturbation, depend only from the size of the local view of nodes and from the distribution of network errors; 2) the synchronization error, in absence of network perturbation, has an lower bound that depends on the distribution of drift of hardware clocks, on local view size and on time interval elapsing between two synchronization round. The results stemmed from the theoretical evaluation evaluation have been validated through simulation-based experiments.

The rest of the paper is organized as follows: Section 2 presents the system model along with the algorithm, while in section 2.1 is presented the algorithm used in our evaluation and the convergence function. The theoretical evaluation is presented in Section 3. Section 5 discusses related works, while Section 6 concludes the paper.

2. SYSTEM MODEL

We consider a system constituted by a finite but unknown set of uniquely identified nodes. Nodes can crash at any time during their computation. A node that does not crash during the entire system lifetime is considered correct.

Each node n_i has a finite set of neighbors. In the following we refer to this set as the *local view* (lv_i) of node n_i. Neighbors nodes communicate by exchanging messages through point-to-point communication. Communication between correct nodes is reliable, but message transmission delays can be unpredictable, but finite.

The local view of a node n_i is based on a Peer Sampling Service[13]. We assume the nodes belonging to the n_i's view provided by the Peer Sampling Service, to be a uniform random sample of the whole system population. If a node n_i crashes, the Peer Sampling Service will not include anymore n_i in any local view.

We also assume that every node is equipped with a hardware clock. Depending on its quality and the operating environment, its frequency may drift. Manufacturers typically provide a characterization for ρ – the maximum absolute value for clock drift. Ignoring, for the time being, the resolution due to limited pulsing frequency of the clock, the hardware clock can be described by:

$$C(t) = ft + C_0;$$

where: $(1-\rho) \leq f \leq (1+\rho)$.

2.1 The Mean-Based Algorithm

In the follows we present a mean-based algorithm that uses a mean of the clock difference between a node n_i and its neighbours in order to reach a synchronization point. Furthermore the clock difference will be estimated through a Remote Clock Reading Procedure with an error ϵ which depends on the mechanism used to perform the estimation. In this paper we assume the Remote Clock Reading Procedure is the same used in NTP [19, 20]. Under this assumption, the real offset is such as the error is bounded by $\pm RTT/2$, where RTT is the round trip time but, as it is showed in [2], the error strictly depends from channel delay. We say $E_{T_{i,j}}(n)$ the error induced by channel between n_i and n_j belonging to S_t at round n. At last we should note that the value C_i is computed periodically, every ΔT.

As a result, the clock synchronization algorithm proceeds in synchronization rounds, where a node n_i performs at each round the following steps:

1. Ask to Peer Sampling Service a random list of neighbours.
2. Evaluate the difference with every neighboring clock, using the Remote Clock Reading Procedure.
3. Compute new clock by mean of the equation

$$C_i(n+1) = C_i(n) + f_i \Delta T + \frac{1}{N_i} \sum_{j=1}^{N_i} [(C_j(n) - C_i(n)) + E_{T_{i,j}}(n)] \quad (1)$$

4. Update the value of C_i.

Where ΔT is the time interval between two consecutive synchronization rounds and N_i is the size of local view lv_i of n_i.

3. EVALUATION

The aim of this section is to show the behaviour of the proposed coupling algorithm when the different clocks are connected by a random communication graph provided by the Peer Sampling Service. In this section first we defining three metrics and we will show that, basing on the Peer Sampling Service that we assume is able to return a number n of random nodes in system, our algorithm show some interesting properties that can be described using some statistical well-know results.

3.1 Evaluation Metrics

The metrics used to evaluate the proposed algorithm are its synchronization error, synchronization point and decay factor. A precise definition, for each metric, is provided below.

Synchronization Error.

The synchronization error (SE) at time t is the standard deviation of the various processes' clock values at same time t. In an ideal setting this value should converge to zero, i.e. all clocks are perfectly synchronized on a same value.

Synchronization Point.

The synchronization point (SP) at time t is the mean of the processes' clocks values at time t.

Decay Factor.

The Decay Factor is the factor by which is reduced the synchronization error in each round, i.e. this metric measure the convergence speed of the algorithm.

3.2 Statistical Analysis

The aim of this section is to show the statistical properties coming from the use of views representing a random sample of the entire population of nodes. Note that Formula 1 represents the facts that at each algorithm step a node performs a mean of N_i samples chosen uniformly at random from the entire population. For sake of simplicity we also assume that $N_i = n, \forall i = 1...N$ and N is fixed.

In the following we will prove three theorems. The first theorem formally shows that the synchronization error of the system decays as $1/\sqrt{n}$ at each round in absence of errors introduced by clock drifts and communication channel delays. The second theorem discuss the contribution of clock drift to the synchronization error with perfect clock estimates. Finally, the third theorem formally shows that the system will eventually show a synchronization error only introduced by clock drifts and communication channel delays where the error introduced by clock drifts is negligible. Moreover we prove three lemmas that describe how varying the synchronization point of the system during time in the three scenarios previously described(i.e. in absence of clock drift and network perturbation, with only clock drift, in presence of clock drift and imperfect clock estimates).

3.2.1 Analysis with no error

Let consider the behaviour of our algorithm without the errors introduced by clock drifts and communication channels, i.e. with perfect offsets estimates. We can consider to have all N nodes at the initial time with clock values following an arbitrary distribution. Clock values can be then represented by a random variable X with an associated probability density function $p(X)$ with unknown mean μ and an unknown variance $\sigma^2 > 0$. Now, considering the possibility for each node to take a random sample of n nodes $X_1, X_2, ...X_n$, each node can calculate the mean of the sample m. From the well-known Central Limit Theorem (CLT) we have that m is approximately equal to μ, while the variance of the sample, denoted as s, is such that $\frac{\sigma^2}{n} = s^2$. So as the sample size increases the distribution of the sample means becomes more concentrated about the mean value μ. Thanks to the iterative nature of the algorithm, as the number of rounds increases, also the number of sample increases (n samples are taken at each round). This implies that at each round the spread of computed sample means decreases, leading to calculate at each node the value μ when the number of synchronization rounds tends to infinity. More formally, we can prove the following theorem:

THEOREM 1. *Let $p(X^0)$ be the initial distribution of clock values with finite variance $\sigma^2_{X^0}$. Let us assume no clock drifts and perfect offset estimates. Under these hypothesis, the mean-based algorithm is able to reduce the synchronization error SE of a factor $\frac{1}{\sqrt{n}}$ in each synchronization round and converges to $SE = 0$.*

PROOF. By induction on the number of synchronization rounds.

round 1: each node extracts n samples $X^0_1, X^0_2, ...X^0_n$ from the clock values of nodes belonging to distributed system. Each node i computes a sample mean m^0_i on the extracted values and updates its clock to that sample mean m^0_i. From CLT, the whole set of computed sample means can be represented by a new random variable X^1 with distribution $p(X^1)$ with variance:

$$\sigma^2_{X^1} = \frac{\sigma^2_{X^0}}{n}, \tag{2}$$

round 2: each node i extracts again n samples $X^1_1, X^1_2, ...X^1_n$ from the new distribution X^1 shown at the end of the first round and computes the sample mean m^1_i. Applying also at this round CLT, we obtain the distribution at the end of the second round $p(X^2)$ with variance:

$$\sigma^2_{X^2} = \frac{\sigma^2_{X^1}}{n} \tag{3}$$

Equation 4 becomes, substituting $\sigma^2_{X^1}$,

$$\sigma^2_{X^2} = \frac{\sigma^2_{X_0}}{n^2} \tag{4}$$

round i: each node still computes a sample mean of the clock values of its neighbours, and consequently after round i, the variance is:

$$\sigma^2_{X^i} = \frac{\sigma^2_{X^{i-1}}}{n} \tag{5}$$

Consequently the variance of our system at round i becomes:

$$\sigma^2_{X^i} = \frac{\sigma^2_{X^0}}{n^i} \tag{6}$$

At each round then, the variance of the initial distribution $p(X^0)$ decreases of a factor $\frac{1}{n}$ and consequently the standard deviation SE of a factor $\frac{1}{\sqrt{n}}$. For a number of synchronization rounds that tends to infinity $SE = 0$ and the theorem follows.

□

Moreover we can prove a lemma in order to describe the behaviour of system synchronization point around which clock value are distributed. In this case, the synchronization point moves on a line having an unitary slope and μ_X as y-intercept.

Lemma 1. Let $p(X^0)$ be the initial distribution of clock values with finite variance $\sigma^2_{X^0}$, mean μ_{X^0} and let $\rho = 0$. Under these hypothesis, the mean-based algorithm with perfect offsets estimates converges in a round i to a synchronization point $SP(i) = \mu_{X^0} + i * \Delta T$ with a $SE = 0$ when $i \to \infty$.

PROOF. The proof follows directly from the previous theorem and from the application of CLT. In fact at end of round i the mean value of the sample mean computed by each node is described by two terms: first derives from the CLT, in fact for CLT the mean value of a sample mean of a population is exactly the mean of the population, and the second from the $f * \Delta T$ in equation 1. From hypotesis $\rho = 0$ so $f = 1$ and from our assumption the $E[\Delta T] = \Delta T$ because each clock executes next round after the same time interval ΔT. More formally

$$\mu_{X^i} = \mu_{X^{i-1}} + \Delta T \tag{7}$$

Consequently the SP at a round i is determined by the following equation:

$$SP(i) = \mu_{X^i} = \mu_{X^0} + i * \Delta T \tag{8}$$

from theorem 1 follows that $SE = 0$ when $i \to \infty$. □

3.2.2 Analysis with clock drifts

However, the contribution of clock drifts has to be included. This contribution to the standard deviation of the system does not decrease and eventually remains the only significant contribution to the standard deviation of the system. This can be represented by a random variable and an associated probability density function. In the following we will denote as $p(R)$, σ_R^2 and μ_R the probability distribution, the variance and the mean of clock frequencies. Using this notation we can prove the following theorem.

THEOREM 2. *Let $p(X^0)$ be the initial distribution of clock values with finite variance $\sigma_{X^0}^2$. Let $p(R)$ be the distribution of clock drifts with variance σ_R. Under these hypothesis, the mean-based algorithm with perfect offsets estimates is able to converge to $SE = \sigma_R \Delta T * \sqrt{\frac{n}{n-1}}$.*

PROOF. By induction on the number of synchronization rounds.

round 1: as shown in the proof of the previous theorem, from CLT, the whole set of sample means computed by each node i can be represented by a new random variable X^1 with distribution $p(X^1)$. In this case, the variance of this distribution is constituted by two terms, the first term follows from the application of CLT, as the previous theorem, and the second term includes the contribution of clock drifts. In particular the first term is equal to $\frac{\sigma_{X^0}^2}{n}$. As for the second term, let us note that we have to include the value $f * \Delta T$, where the term f depends on the drift ρ. This relation makes f a random variable described in the whole population by the distribution $p(R)$. Consequently, after this first round the distribution $p(X^1)$ of clock values has a variance :

$$\sigma_{X^1}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^0}^2}{n}, \tag{9}$$

Note that the first term is constituted by the frequency variance multiplying the factor ΔT, in fact as R represents clock frequencies possibly used by different nodes, the total variance depends also on the duration of the round.

round 2: the sample mean computed at round 2 applying also at this round CLT and taking into consideration the distribution on drifts $p(R)$, has distribution at the end of the second round $p(X^2)$ with variance:

$$\sigma_{X^2}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^1}^2}{n} \tag{10}$$

Equation 4 becomes, substituting $\sigma_{X^1}^2$,

$$\sigma_{X^2}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_R^2 \Delta T^2}{n} + \frac{\sigma_{X^0}^2}{n^2} \tag{11}$$

round i: as previous round and previous proof the distribution $p(X^i)$ has variance:

$$\sigma_{X^i}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^{i-1}}^2}{n} \tag{12}$$

Consequently the variance of our system at round i becomes:

$$\sigma_{X^i}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_R^2 \Delta T^2}{n} + \cdots + \frac{\sigma_R^2 \Delta T^2}{n^{i-1}} + \frac{\sigma_{X^0}^2}{n^i} \tag{13}$$

Where the first n terms describes a geometric series with a common ratio $r = \frac{1}{n} < 1$.

$$\sigma_{X_i}^2 = \sum_{j=0}^{i} \frac{\sigma_R^2 \Delta T^2}{n^j} + \frac{\sigma_{X^0}^2}{n^i} \tag{14}$$

Consequently the variance of the whole system converges to a value that depends only from σ_R^2 as the synchronization rounds go to infinity, in fact after a transitory the terms $\frac{\sigma_{X^0}^2}{n^i}$ becomes negligible and the geometric series converges to $\sigma_R^2 \Delta T^2 * \frac{n}{n-1}$. The synchronization error SE consequently becomes:

$$SE = \sigma_R \Delta T * \sqrt{\frac{n}{n-1}} \tag{15}$$

and the theorem follows.

□

In the following we discuss a lemma similar to the previous one, where we can analytically describe the behaviour of the system synchronization point in presence of clock drifts. In presence of clock drift, the synchronization point moves on a line having a slope equals to μ_R and μ_X as y-intercept.

Lemma 2. Let $p(X^0)$ be the initial distribution of clock values with finite variance $\sigma_{X^0}^2$ and mean μ_{X^0}. Let $p(R)$ be the distribution of clock drifts with variance σ_R. Under these hypothesis, the mean-based algorithm with perfect offsets estimates is able to converge at round i to a synchronization point $SP(i) = \mu_{X^0} + \mu_R * i * \Delta T$ with a $SE = \sigma_R \Delta T * \sqrt{\frac{n}{n-1}}$ when $i \to \infty$.

PROOF. The proof derives from the previous theorem and from the lemma 1. In this case considering f distributed with a mean μ_R and variance σ_R

$$E[f * \Delta T] = E[f] * \Delta T = \mu_R * \Delta T \tag{16}$$

Consequently the mean at round i has two terms: the first term follows from the application of CLT, as the previous theorem, and the second term includes the contribution of clock drifts.

$$\mu_{X^i} = \mu_{X^{i-1}} + \mu_R * \Delta T \tag{17}$$

Finally substituting $\mu_{X^{i-1}}$ we obtain

$$SP(i) = \mu_{X^i} = \mu_{X^0} + \mu_R * i * \Delta T \tag{18}$$

and from theorem 2 follows that $SE = \sigma_R \Delta T * \sqrt{\frac{n}{n-1}}$ when $i \to \infty$. □

3.2.3 Analysis with Clock Drift and Network Errors

Finally, let us introduce errors induced by imperfect offsets estimates, i.e. errors in remote clock reading procedure due to unknown channel delays. Also this type of error is a random variable with an associated probability density function. We denote as $p(E)$, σ_E^2 and μ_E the probability distribution, the variance and the mean of the errors in remote clock readings. Note that $p(E)$ is strictly related to the asymmetry of channels and it is not a normal distribution[2]. This is not a problem for our analysis because we do not manage directly errors but only sample means of errors and for CLT they converge to a normal distribution despite the shape of original distribution. Thus, we can prove the following theorem:

THEOREM 3. *Let $p(R)$ and $p(X^0)$ the distribution of clock drifts and the initial distribution of clocks, with respectively variance σ_R^2 and finite variance $\sigma_{X^0}^2$. Let $p(E)$ the distribution of errors in remote clock reading with finite variance σ_E^2. Under these hypothesis the synchronization error SE converges to $SE = \frac{\sigma_E}{\sqrt{n-1}}$.*

PROOF. By induction on the number of rounds:

round 1. Each node computes a sample mean, but each sample is now a sum of two random variables, namely X^i and E where X^i represents the distribution of correct clock values at the beginning of round i and E the error induced in the Remote Clock Reading Procedure by the channel asymmetry. Consequently we can apply separately the CLT to X^i and E. In this manner, the variance of distribution of clock value is constituted by three terms: the first and second term are respectively equal to $\frac{\sigma_{X^0}^2}{n}$ and $\sigma_R^2 \Delta T$, as we showed in previous theorem, and the last one follows directly to the application of CLT to E so it is equal to $\frac{\sigma_E^2}{n}$. At the end of the first round we obtain:

$$\sigma_{X^1}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^0}^2}{n} + \frac{\sigma_E^2}{n} \tag{19}$$

round 2. Applying also at this round CLT and taking into consideration the clock drifts, the distribution at the end of the second round $p(X^2)$ has variance:

$$\sigma_{X^2}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^1}^2}{n} + \frac{\sigma_E^2}{n} \tag{20}$$

Equation 20 becomes, substituting $\sigma_{X^1}^2$,

$$\sigma_{X^2}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_R^2 \Delta T^2}{n} + \frac{\sigma_{X_0}^2}{n^2} + \frac{\sigma_E^2}{n} + \frac{\sigma_E^2}{n^2} \tag{21}$$

round i. At a generic step i, as previously described, the variance of the distribution $p(X^i)$ is:

$$\sigma_{X^i}^2 = \sigma_R^2 \Delta T^2 + \frac{\sigma_{X^{i-1}}^2}{n} + \frac{\sigma_E^2}{n} \tag{22}$$

We have to note that the term $\frac{\sigma_E^2}{n}$ remains the same in each round. Consequently substituting $\frac{\sigma_{X^{i-1}}^2}{n}$ we can expand Equation 22 and writing in terms of series we obtain:

$$\sigma_{X^i}^2 = \sum_{j=0}^{i} \frac{\sigma_R^2 \Delta T^2}{n^j} + \frac{\sigma_{X^0}^2}{n^i} + \sum_{j=1}^{i} \frac{\sigma_E^2}{n^j} \tag{23}$$

The last term is a geometric series with a common ration $r = \frac{1}{n} < 1$, so the series, starting from $j = 1$ converges to $\frac{\sigma_E^2}{n-1}$. Moreover $\frac{\sigma_{X^0}^2}{n^i}$ becomes rapidly small and after a few round $\frac{\sigma_{X^0}^2}{n^i} << \frac{\sigma_E^2}{n-1}$. At last, usually σ_R^2 is smaller than σ_E^2 of several orders of magnitude (i.e. considering slow channels presented in [3] the difference is about ten orders of magnitude), under this assumption $\sigma_R^2 \Delta T^2 * \frac{n}{n-1} << \frac{\sigma_E^2}{n-1}$ also for larger value of ΔT. Thus, Equation 23 for a number of synchronization rounds that tends to infinity, the variance of the system becomes:

$$\frac{\sigma_E^2}{n-1} \tag{24}$$

and then, standard deviation is:

$$SE = \frac{\sigma_e}{\sqrt{n-1}} \tag{25}$$

and the theorem follows.

□

Finally we can analytically discuss the behaviour of synchronization point in presence of both errors, i.e. clock drifts andimperfect estimates. Note that in presence of both errors the synchronization point is described by a line with $\mu_X + \mu_E$ as slope and μ_R as y-intercept.

Lemma 3. Let $p(R)$ and $p(X^0)$ the distribution of clock drifts and the initial distribution of clocks, with respectively variance and mean σ_R^2, μ_R and $\sigma_{X^0}^2$, μ_{X^0}. Let $p(E)$ the distribution of errors in remote clock reading with mean μ_E and finite variance σ_E^2. Under these hypothesis the synchronization error SE converges to $SE = \frac{\sigma_E}{\sqrt{n-1}}$, when $i \to \infty$, and the system to a synchronization point at round i described by $SP(i) = \mu_{X^0} + \mu_R * i * \Delta T + \mu_E$.

PROOF. The proof follows from the proof of previous theorem and from the CLT. We have that at round i the mean of sample mean is composed by three terms, similarly to the previous proof: the first term follows from the application of CLT, the second term includes the contribution of clock drifts and the third one includes the network errors introduced by the remote clock reading procedure. In Lemma 2 we showed the contribution of the first two terms to the mean at a round i. At last adding the contribution of $p(E)$

$$\mu_{X^i} = \mu_{X^{i-1}} + \mu_R * \Delta T + \mu_E \tag{26}$$

Where μ_E is the term introduced by $p(E)$ from CLT, similarly we showed in the proof of Lemma 1

Consequently as we have showed in previous proof, substituting $\mu_{X^{i-1}}$

$$SP(i) = \mu_{X^i} = \mu_{X^0} + \mu_R * i * \Delta T + \mu_E \tag{27}$$

and from theorem 3 follows that $SE = \frac{\sigma_E}{\sqrt{n-1}}$ when $i \to \infty$. □

4. EXPERIMENTAL VALIDATION

The aim of this section is an experimental validation of theorems showed in previous section. We run several simulation using Peersim in order to verify the behaviour of the mean-based algorithm in a peer-to-peer environment and to compare the obtained experimental results with the analytical ones. We define three scenarios in order to validate theorems presented in previous section: 1) a scenario with perfect clock estimates and without clock drifts; 2) a scenario with clock drift and perfect clock estimates; 3) a scenario with clock drifts and errors in remote clock reading. Every scenario is composed by 64K nodes and no nodes are added/removed during the simulation.

In the first scenario we evaluate the convergence speed in terms of number of synchronization rounds required to reach a predefined $SE = 10\mu s$. In Figure 1 we compare the behaviour of our simulation results with the analytical ones starting with different variance of initial distribution of clocks $p(X^0)$. In this case $p(X^0)$ is assumed to be a rectangular distribution. The difference between theoretical results and experimental ones are noticeable only with small view size. This is due to the CLT that show better results when the number of sample (i.e. the number of elements in a local view) is large. The term "large" is relative: the rule of thumb is that a sample size n of at least 30 will suffice; although for many distributions smaller n can be sufficient (e.g. normal, rectangular, binomial, etc. . .).

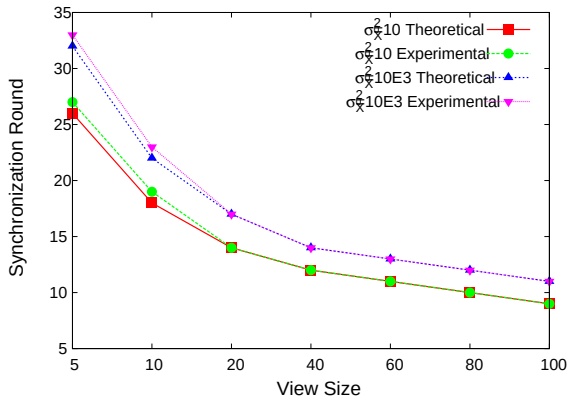

Figure 1: Convergence dependency on Variance of Initial Distribution and View Size.

In the second scenario we introduce clock drifts. We model $p(R)$ as a normal distribution and assume $\Delta T = 30s$. Figure 2 shows that practically there are not difference between theoretical results obtained applying the theorem 2 and the experimental ones obtained executing simulations. In particular it shows that for common value of standard deviation of frequency, in the order of $10^{-6} - 10^{-7}$ (e.g. in [18] it is presented a comparison between clock frequencies in common CPU) the impact of clock drifts on the accuracy of clock synchronization is small also for large ΔT.

In the third scenario we introduce network errors. In this setting the remote clock reading procedure does not produce anymore perfect estimates. We describe the network errors,

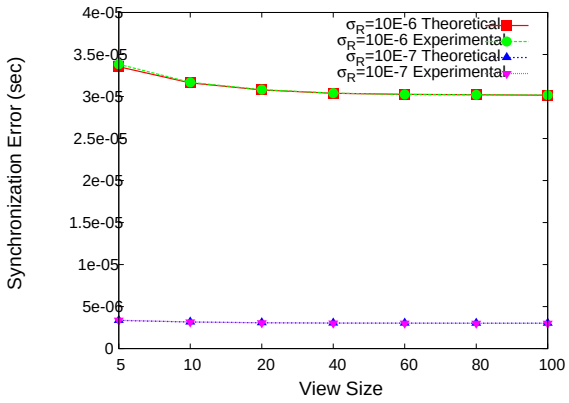

Figure 2: Synchronization error varying Clock Drift Variance and View Size

introduced in our computation, through two normal distribution to model respectively the RTT of message exchange and the asymmetry of channels. The distribution of RTT has mean and standard deviation derived by fitting several round-trip data set measured over the Internet [3]. In particular in this scenario we evaluate a "slow channel", as it is described in [3],while we let the variance of distribution of channel asymmetry can assume different values. Because we assumed that we evaluate the clock differences as NTP does, the error distribution $p(E)$ is the product of these two distributions, as it is shown in [2]. In order to validate the Theorem 3, in Figure 3 we show the results obtained comparing two different variance of channel asymmetry and consequently of $p(E)$. In this settings the differences are very little noticeable also for the smallest view size because the distribution of $p(E)$ produced by the product of two normal are more "regular" than the rectangular distribution used in first scenario. Another important point is the order of magnitude of SE introduced by imperfect estimates. As we said in the proof of Theorem 3 usually σ_R^2 is smaller than σ_E^2 of several orders of magnitude and Figure 3 confirms that the impact of clock drift with respect to the network channel errors is negligible when we consider channels like the ones described in [3]. Error due to clock drifts and imperfect clock estimates can become comparable only in dedicated LAN, where RTT is very small and also channel asymmetry can be unnoticeable.

5. RELATED WORK

We can divide clock synchronization algorithms in two main classes: deterministic and probabilistic. Deterministic clock synchronization algorithms [9, 14, 15, 16, 17, 6, 5, 10] formally guarantee strict properties on the accuracy of the synchronization but assumes that a known bound on message transfer delays exists. In particular by mean of existence of this bound on message delay they can guarantee an upper bound on the difference between any two clock values. Lamport in [14] defines a distributed algorithm for synchronizing a system of logical clocks which can be used to totally order events, specializes this algorithm to synchronize physical clocks, and derives a bound on how far out of synchrony the clocks can become. Several works of Dolev et al. [7, 8, 9, 11] propose and analyze several decentralized synchronization protocols applicable for WAN but

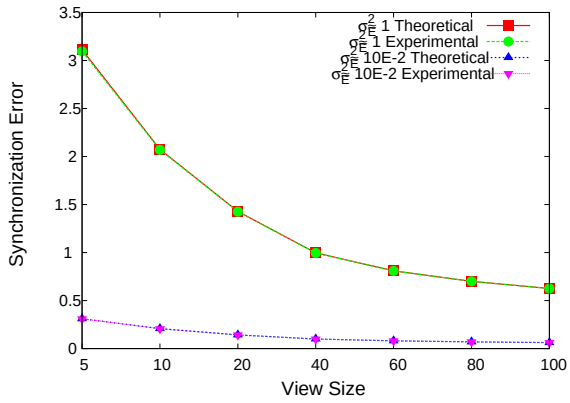

Figure 3: Synchronization error varying Network Error Variance and View Size

that require a clique-based interconnecting topology, which is hardly scalable with a large number of nodes.

However the deterministic approach, normally tuned to cope with the worst case scenario, assures a bounded accuracy in LAN environments but loses its significance in WAN environments where messages can suffer high and unpredictable variations in transmission delays. Clock synchronization algorithms based on a probabilistic approach were proposed in [4, 1] in order to try to overcome this problem. The basic idea is to synchronize clocks in the presence of unbounded communication delays by using a probabilistic remote clock reading procedure. Each node makes several attempts to read a remote clock and, after each attempt, calculates analytically the maximum error. By retrying often enough, a node can read the other clock to any required precision with a probability as close to 1 as desired. This implies that the overhead imposed by the synchronization algorithm and the probability of loss of synchronization increases when the required synchronization error is reduced. A formal evaluation of relationship between required error and probability of obtaining synchronization is proposed. The master-slave approach and the execution of several attempts are basic building blocks of the most popular clock synchronization protocol for WAN settings: NTP [19, 20]. NTP works in a static and manually-configured hierarchical topology. Moreover it requires the presence of some nodes directly connected with a external time reference in order to obtain external time synchronization.

The peer-to-peer approach for synchronizing very large systems is a novel solution and is interesting for the inherent scalability of this approach. We know only two previous studies [2, 12] where are proposed only simulation-based experimental evaluations of the protocol properties. In [12] the presence of a source node perfectly synchronized with real-time clock is assumed. Each node uses a peer sampling service to select another node in the network and to exchange timing information with. If the time read from the contacted node is of higher quality than its own time (e.g. the contacted node is the source), then the reading node will adopt the clock setting of the other one. In [2] it is presented convergence function-based approach to internal clock synchronization, where each node read by mean of a remote clock reading procedure neighbours' clock values and computes its new clock basing on read values.

6. CONCLUSIONS

In this paper we presented a theoretical analysis of the synchronization properties of a mean-based convergence function in a peer-to-peer system. The analysis focused on three main aspects of the algorithm behaviour: synchronization error, final synchronization point and decay factor that affect synchronization delay.

More specifically the analysis outlined two important properties: 1) convergence speed and synchronization error of the mean based protocol, in presence of errors induced by network perturbation, depend only from the size of the local view of nodes and from the distribution of network errors; 2) the synchronization error, in absence of network perturbation, has an lower bound that depends on the distribution of drift of hardware clocks, on local view size and on time interval elapsing between two synchronization round.

7. REFERENCES

[1] K. Arvind. *Probabilistic Clock Synchronization in Distributed Systems*, volume 5 of *IEEE Trans. on Parallel and Distrib. Systems*. 1994.

[2] R. Baldoni, A. Corsaro, L. Querzoni, S. Scipioni, and S. Tucci-Piergiovanni. An adaptive coupling-based algorithm for internal clock synchronization of large scale dynamic systems. In *OTM Conferences*, pages 701–716, 2007.

[3] R. Baldoni, C. Marchetti, and A. Virgillito. Impact of wan channel behavior on end-to-end latency of replication protocols. In *Proceedings of European Dependable Computing Conference*, 2006.

[4] F. Cristian. *A probabilistic approach to distributed clock synchronization*, chapter 3, pages 146–158. Distributed Computing. 1989.

[5] F. Cristian, H. Aghili, and R. Strong. Clock synchronization in the presence of omission and performance faults, and processor joins. In *Proceedings of the Int. Conf. Fault-Tolerant Computing*, pages 218–223, 1986.

[6] F. Cristian and C. Fetzer. Lower bounds for convergence function based clock synchronization. In *Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, pages 137–143, 1995.

[7] A. Daliot, D. Dolev, and H. Parnas. Linear time byzantine self-stabilizing clock synchronization. Technical Report TR2003-89, The Hebrew University of Jerusalem, 2003.

[8] A. Daliot, D. Dolev, and H. Parnas. Self-stabilizing pulse synchronization inspired by biological pacemaker networks. In *Proceedings of the Sixth Symposium on Self-Stabilizing Systems*, pages 32–48, 2003.

[9] S. Dolev. *Possible and Impossible Self-Stabilizing Digital Clock Synchronization in General Graph*, pages 95–107. Number 12(1) in Journal of Real-Time Systems. 1997.

[10] J. Halpern, B. Simons, and R. Strong. Fault-tolerant clock synchronization. In *Proceedings of the 3rd Ann. ACM Symposium on Principles of Distrib. Computing*, pages 89–102, 1984.

[11] T. Herman and S. Ghosh. *Stabilizing Phase-Clock.*, volume 5 of *Information Processing Letters*, pages 585–598. 1994.

[12] K. Iwanicki, M. van Steen, and S. Voulgaris. Gossip-based synchronization for large scale decentralized systems. In *Proceedings of the Second IEEE International Workshop on Self-Managed Networks, Systems and Services*, 2006.

[13] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service: experimental evaluation of unstructured gossip-based implementations. In *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, 2004.

[14] L. Lamport. *Time, clocks and ordering of events in a distributed system*, volume 21, pages 558–565. Commun ACM, 1978.

[15] L. Lamport and P. M. Melliar-Smith. Byzantine clock synchronization. In *Proceedings of 3rd Ann. ACM Symposium on Principles of Distributed Computing*, pages 68–74, 1984.

[16] L. Lamport and P. M. Melliar-Smith. Synchronizing clocks in the presence of faults. In *Journal of the ACM*, volume 32(1), 1985.

[17] J. Lundelius-Welch and N. Lynch. A new fault-tolerant algorithm for clock synchronization. In *Proceedings of the 3rd Ann. ACM Symposium on Principles of Distrib. Computing*, pages 75–88, 1984.

[18] H. Marouani and M. Dagenais. *Internal Clock Drift Estimation in Computer Cluster*, volume 2008 of *Journal of Computer Systems, Networks, and Communications*. 2008.

[19] D. Mills. Network time protocol (version 1) specification and implementation. Network Working Group Report RFC-1059, July 1986.

[20] D. Mills. Network time protocol version 4 reference and implementation guide. Electrical and Computer Engineering Technical Report 06-06-1, June 2006.