Accepted Manuscript

Design and topological analysis of probabilistic distributed mutual exclusion algorithm with unbiased refined ordering

Susmit Bagchi

PII: S0167-739X(18)32328-8
DOI: https://doi.org/10.1016/j.future.2018.12.069
Reference: FUTURE 4690

To appear in: *Future Generation Computer Systems*

Received date : 29 September 2018
Revised date : 18 November 2018
Accepted date : 29 December 2018

Please cite this article as: S. Bagchi, Design and topological analysis of probabilistic distributed mutual exclusion algorithm with unbiased refined ordering, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2018.12.069

Design and Topological Analysis of Probabilistic Distributed Mutual Exclusion Algorithm with Unbiased Refined Ordering

Susmit Bagchi

Department of Aerospace and Software Engineering (Informatics),
Gyeongsang National University, Jinju, South Korea 660701

Author Correspondence Email: profsbagchi@gmail.com

Abstract

The applications of distributed computing systems are pervasive in nature involving multiple shared resources. The distributed mutual exclusion algorithms of various classes are employed to control concurrency of accessing shared resources maintaining data consistency. In general, the distributed mutual exclusion algorithms are designed based on fixed or dynamic graph structures formed by a set of processes, where the distributed mutual exclusion mechanisms are realized depending upon timestamp based ordering of events or by employing token circulation in the graph. On the contrary, in large scale heterogeneous distributed systems, an aggregate set of processes can be generated under special circumstances, where processes in a group are equally eligible to enter into critical section. In order to maintain safety and liveness properties of mutual exclusion in such cases, the probabilistic characterization as well as topological analysis of aggregate set in computing space is necessary. This paper proposes a probabilistic algorithm and its topological characterization for mutual exclusion in aggregate set of processes. The analysis of failure model of strictly ordered distributed inclusion-exclusion designs is constructed in the presence of aggregate set. The unbiased probabilistic algorithm is based on two-phased elastic randomization. The algorithm is evaluated through detailed simulation and, the related probabilistic characterization in topological subspace is evaluated. A detailed comparative analysis of the algorithm with respect to other distributed mutual exclusion algorithms is presented.

Keywords: Distributed computing, logical clock, mutual exclusion, probability, random variable, topological spaces.

1. Introduction

The present day distributed computing systems have two distinct characteristics namely, multi-level heterogeneities and large scale involving thousands of computing nodes. The multi-level heterogeneities include network level heterogeneity, hardware level heterogeneity and, system software level heterogeneity. Traditionally, the distributed computing systems are modeled as arbitrary graph structures, where nodes of a graph represent distributed processes and the edges of a graph represent network links. However, a distributed computing system can be modeled in view of topological spaces comprised of sets of distributed events generated by individual processes [23]. In any case, a distributed computing system maintains a set of shared resources concurrently accessed by a subset of distributed processes, which requires designing of mutual exclusion for Critical Sections (CS) [12]. The main aim of mutual exclusion is to maintain data consistency, liveness and fairness of computation involving shared resources [2, 6, 11]. The traditional distributed mutual exclusion (mutex) algorithms are designed employing two approaches namely, (1) logical clock based timestamps for ordering of requests in a group of processes, and (2) repeated circulation of a token between processes [7]. If a token is lost then the fault detection and regeneration of a new token may incorporate unpredictable delay in a system.

A wait-free (minimum delay) synchronization algorithm is proposed for concurrent distributed systems programming intended to systems having moderate scale [21].

However, the present day distributed systems such as, Mobile Cloud and Grid computing systems, are having geographic scale comprised of thousands of nodes and multi-level heterogeneities in architecture. These large scale systems are hybrid in nature, where traditional graph based or token based distributed mutual exclusion algorithms may not be adequate. The reasons are that, the distributed systems having geographic scale and highly heterogeneous networks are open to unpredictable network partitioning, message latency, duplicate transmissions, and random node failures. The fail-proof token circulation, token loss detection, regeneration, and realization of timestamp based ordering of events in such large-scale geographically distributed systems are extremely challenging. In such cases, the majority of traditional as well as improved distributed mutex algorithms are not completely suitable to guarantee safety, liveness and fairness [1, 10]. For example, the Ricart-Agrawala mutex algorithm requires modifications to adapt to large scale heterogeneous distributed systems [9]. However, the designing of probabilistic mutex algorithms and their characterizations in large scale distributed systems are not well studied. Moreover, the topological analysis of behaviour of a group of processes equally eligible for critical section in a distributed system is not explored.

1.1 Motivation

The large scale distributed computing systems contain multiple shared resources, and the distributed mutual exclusion algorithms are essentials for maintaining data consistency under concurrent execution involving those shared resources. However, the traditional distributed mutex algorithms are not completely suitable for present day large-scale geographically distributed mobile computing systems due to multi-level heterogeneities as well as scale of the systems [1]. The network level heterogeneity may results in random network partitioning and unpredictable delay in message transactions leading to the failures of traditional distributed mutex algorithms. Furthermore, the adaptation of distributed mutex algorithms to grid topology requires modifications at router layer. The router layer of grid topology is modified while employing Naimi–Trehel-Arnold algorithm to minimize network routing delays during distributed mutex [1]. However, this leads to a comparatively rigid design requirements involving network hardware layer of a large-scale geographically distributed systems. The token based adaptation of Naimi–Trehel-Arnold mutual exclusion algorithm is not suitable for fail-prone geo-distributed systems involving mobile computing nodes [4]. The reason is that, locating and recovering a lost token in a mobile distributed system enhances computational complexity to a high degree.

On the other hand, the generalized (*l*, *k*)-CS group mutual exclusion algorithms aim to incorporate flexibility in traditional mutual exclusion model in a group of processes [11, 20]. However, this algorithmic model would violate safety property due to reduction of tractability of shared data modifications if the group size increases. Moreover, the token based (*l*, *k*)-CS group mutual exclusion algorithms would fail in mobile distributed systems due to possibility of unpredictable network latency and network partitioning. On another extreme, the finite-population queuing model based distributed mutex algorithm assumes that, systems can be modeled using Petri Nets and, multicasting in network is reliable along with FIFO ordering, which are not suitable for geo-distributed mobile computing systems [6]. In general, the efficiency of distributed local mutex algorithm for mobile ad hoc networks (MANET) is highly dependent on the mobility of computing nodes [19]. The distributed local mutex algorithm is efficient if the computing nodes are effectively stationary residing in geographic proximity reducing network delay. The rooted graph based algorithms such as, Raymond and Ricart-Agrawala mutex algorithms, require adaptation for realizing mutual exclusion if the scale of distributed systems increases [5]. However, the Ricart-Agrawala algorithm would not be able to provide safe mutual exclusion in the presence of (l, k)-CS group (i.e. aggregate set), where $l > 1$ and $k > 1$ ($l = k$). It is interesting to note that, concurrency control can be modeled in topological spaces of distributed computation having complex structures [24]. Furthermore, the probabilistic estimation models offer acceptable solutions in the systems having inherent ambiguity as well as randomness in decision making [30, 31].

This paper proposes a probabilistic mutual exclusion model and corresponding algorithm for large-scale geo-distributed systems. This paper considers that aggregate set is a subspace in topological event space of distributed computation, and the algorithm can be viewed as a refinement of logical clock based mutex algorithms imposing total ordering on processes. The algorithm allows generation of aggregate set and resolves it using unbiased phased randomization. The probabilistic characterization of topological subspace of generated aggregate set is analyzed to gain insight. The main contributions of this paper are as follows.

- A unbiased probabilistic distributed mutex algorithmic model is proposed independent of network topology and delay distribution.
- The proposed algorithmic model considers generation of aggregate set and, resolves it probabilistically maintaining safety and fairness properties.
- The unbiased probabilistic model is based on two-phase elasticity of randomization, which reduces the requirement of repeated execution rounds for realizing mutex.
- Analysis of probabilistic characterization of topological normed spaces of arbitrary aggregate set is formulated.
- Computational estimation of variations of probability measures for different aggregate sets having varying densities of processes is presented.

Rest of the paper is organized as follows. Section 2 presents related work. Section 3 illustrates analysis of aggregate set generation, failure modes, and respective probabilistic characterization of aggregate set. The phased randomization model for generating order for mutex in aggregate set is presented in section 4. The corresponding mutex algorithms are presented in section 5. The analysis of algorithmic correctness and a set of analytical properties are presented in section 6. The implementation and evaluation of the algorithms are presented in section 7. Section 8 illustrates comparative analysis. Finally, section 9 concludes the paper.

2. Related Work

The mutual exclusion algorithms for distributed systems can be classified into two broad categories namely (a) function based algorithms and, (b) structure based algorithms. The structure based algorithms require a specific graph structure for the designing of distributed mutual exclusion algorithms. The function based categories can be further classified into two subcategories such as, permission based mutex algorithms and, token based mutex algorithms [7]. In case of permission based distributed mutex algorithms, the timestamps (generated by logical clocks) based ordering and consensus within a group of processes are employed, where the group is considered to be in closed category. The examples of permission based distributed mutex algorithms are Lamport algorithm [2], Ricart-Agrawala algorithm [12], Singhal algorithm [14] and, Maekawa algorithm [13]. However, the token based distributed mutual exclusion algorithms are designed considering rooted graph structures, where the repeated circulations of a token take place in the structure. The examples of token based mutex algorithms are Suzuki-Kasami algorithm [15] and, Naimi-Trehel-Arnold algorithm [16]. A distributed mutual exclusion in

MANET is proposed based on dynamic logical ring topology employing token circulations [8]. The periodic graph partitioning mechanism is employed forming cluster of computing nodes and arranging them into a logical ring topology for implementing Ricart-Agrawala mutex algorithm [9]. Another token based mutual exclusion algorithm in MANET is proposed to reduce hop frequency [10]. The main challenge with token based distributed mutual exclusion algorithms is to maintain fairness property. This is because a process holding a token may be in CS for arbitrary long time.

A fair and starvation-free distributed mutex algorithm is proposed, which is a refinement of Kanrar-Chaki algorithm [4, 32]. In this algorithm, heuristics are inserted in order to reduce the frequency of priority enhancement in a queue of requests received from processes [4]. A tree-based distributed mutex algorithm is proposed independent of the distributed shared memory [17]. The algorithm considers that a distributed system has a spanning tree topology. The hybrid version of distributed mutex algorithm is proposed based on logical array topological structure employing token circulations as well as wraparound in 2-D [18]. A priority based distributed mutual exclusion algorithm is proposed based on rooted tree structure, which is a logical tree structure [5]. The algorithm is dependent on the network layer supports from routers. A hybrid group based distributed mutual exclusion algorithm is proposed considering multiple groups of processes [3]. The algorithm aggregates a set of nodes into multiple groups, where mutual exclusion is realized by transacting inter-group or intra-group messages. This results in high message complexity due to several rounds of messaging within a group as well as inter-group. In cloud computing paradigm, analysis and similarity assessment of events generated by distributed processes can be performed based on shared data requiring distributed mutex in closed group of processes [28]. In case of very large scale mobile distributed systems, formation of process groups can be analyzed in view of topology, which is inline with the topological data grouping in massive data sets [26, 27, 29].

3. Probabilistic Characterization of Aggregate Set

The standard mutual exclusion algorithms for distributed systems are designed based on partial ordering of events. The partial ordering is performed based on local logical clocks associated to each distributed processes. The CS requests are time stamped by using local logical clock values of individual processes and the request messages are broadcasted in a closed group of processes. As a result, a single process within the closed group can enter into CS depending upon lowest value of logical clocks. In general, this mechanism is efficient and easy to realize. However, in special cases the algorithm may lead to generate an aggregate set of processes and would not be able to provide strict mutex in the presence of (*l*, *k*)-CS set. Moreover, the timestamp based mechanism may not work in open group of processes. The detailed analysis of generation of aggregate set (i.e. process subgroup) and failure model is given in next section.

3.1 Generation of Aggregate Set and Failure Model

Let $P=\{p_i : 1 \le i \le N, i \in Z^+\}$ be a closed group of processes in a distributed system having arbitrary topology. Let E_i be a set of events generated by respective process $p_i \in P$ and, $\forall p_i \in P, C : E_i \rightarrow Z^+ \cup \{0\}$ be the clock function associated to individual processes. According to model of distributed computation, the following biconditional is maintained over an anti-symmetric relation $\forall p_i \in P, < \subset E_i^2$ in any system,

$$\forall p_i \in P, \forall e_i, e_k \in E_i, [(e_i, e_k) \in <] \Leftrightarrow [C(e_i) < C(e_k)] \qquad (1)$$

Let a predicate be defined as, $\exists x, y \in Z^{+} \cup \{0\}, \Psi(x, y) \in \{0,1\}$ where,

$$\begin{aligned} & x = C(e_a), y = C(e_b), \\ & [\Psi(x, y) = 1] := \left([x < y] \Rightarrow [(e_a, e_b) \in <]\right) \end{aligned} \qquad (2)$$

However, as the clock function is not a global bijection in a system, hence a distributed system maintains following condition for consistency of computation,

$$\begin{aligned} & \exists \{e_a, e_b\} \subset \bigcup_{i=1}^{N} E_i, \\ & \Psi(x, y) = 0, \\ & [(e_a, e_b) \in <] \Rightarrow [(e_b, e_a) \notin <] \wedge [x < y] \end{aligned} \qquad (3)$$

Assume that, initially $\forall p_i \in P, \exists e_a \in E_i, C(e_a) = 0$ and, $CS(P)|_{\Delta t} \subset P$ represents a set of processes willing to enter CS within Δt time-window, where $\Delta t \to 0$. As the function $C(.)$ is not invertible, so $\exists A \subset \bigcup_{i=1}^{N} E_i$ such that following axioms are satisfied,

$$\begin{aligned} & \forall e_a, e_b \in A, \\ & e_a \in E_a, e_b \in E_b, a \neq b, \\ & C(e_a) = C(e_b) \end{aligned} \qquad (4)$$

This may lead the distributed system under consideration to violate safety and liveness properties of mutual exclusion of logical clock based algorithms if the following axioms are satisfied,

$$\begin{aligned} & |CS(P)|_{\Delta t}| \geq 1, |A| > 1, \\ & \Delta t \to 0, \\ & [e_x \in A] \Rightarrow [p_x \in CS(P)|_{\Delta t}] \end{aligned} \qquad (5)$$

Thus, the logical clock based traditional mutex algorithms may not be able to select a distinct process under the existence of conditions satisfying axioms mentioned earlier. Hence, A is a generated CS-aggregate set and the corresponding execution sequence is presented in Fig. 1 as schematic representation.

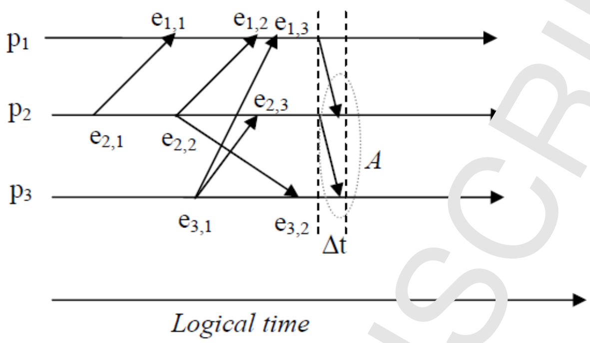

Fig. 1. Schematic representation of generation of aggregate set

It is to note that, complete restriction on set A is not possible in a large-scale distributed system, which is inline to the concept of generalized (*l*, *k*)-CS set. Hence, a mechanism is necessary to resolve A while maintaining safety and liveness properties of CS. In the next section, a detailed probabilistic characterization of generation of aggregate set A is presented in view of topological spaces.

3.2 Probabilistic Characterization of Set A

Let the entire set of events generated in a distributed system be represented by, $E_D = \bigcup_{i=1}^{N} E_i$. However, the CS requests can be generated randomly by distributed processes and, the corresponding real (R) valued random variable in 1-D is given by,

$$X_D : E_D \to R \qquad (6)$$

In a group of processes in a distributed system, the inter-dependency between processes is important factor to be considered. Let the probabilistic characterization of A be represented by, $A_i \subseteq X_D(A \subset E_D)$, where $i \in I$ (index set). In the correlated event space, a probability distribution function is defined as,

$$pr : E_D^2 \to [0,1] \qquad (7)$$

Let τ_g be the product topology in 2-D space of randomized events of processes and, the corresponding product topology generating function on randomized real space be given as, $g_\tau : R^2 \to \{R^2\}$ such that, the following conditions are satisfied, where $\Omega(.)$ generates power set,

$$
\begin{aligned}
& g_\tau(R^2) \subseteq \{X_D^2\}, \\
& \tau_g \subset \Omega(X_D^2)
\end{aligned}
\qquad (8)
$$

However, the generation of topology τ_g is not arbitrary and, it follows a restriction as given below considering $\overline{K_i}$ is a closure of open set K_i,

$$
\begin{aligned}
& e_{ab} = (e_a, e_b) \in E_D^2, \\
& \alpha_{ab} = (X_D(e_a) \in R, X_D(e_b) \in R), \\
& \forall K_i \in \tau_g, \ \sum_{\forall \alpha_{ab} \in \overline{K_i}} pr(e_{ab} \mid K_i \in \tau_g) = 1
\end{aligned}
\qquad (9)
$$

The degree of correlation of events in space is computed as a norm affecting the shape of probability distribution within topological space generated by $g_\tau(.)$. The computable q-norm in correlated event space and, the respective locally uniform as well as complete (LUC) probability distribution in topological event space are defined as,

$$
\begin{aligned}
& \forall e_{ab} \in B \subset E_D^2, q \in R^+, \\
& \| e_{ab} \|_q = \left(X_D(e_a)^q + X_D(e_b)^q \right)^{1/q}, \\
& pr(e_{ab} \mid K_i \in \tau_g) = \frac{\| e_{ab} \|_q}{\sum_{\forall \alpha_{ab} \in \overline{K_i}} \| e_{ab} \|_q}
\end{aligned}
\qquad (10)
$$

Furthermore, if $G, H \subset E_D^2$ be such that, $\exists V \in \tau_g$ and $G \cap H \neq \phi$ then, definition of $pr(.)$ is refined as,

$$
\begin{aligned}
& |G| > 1, |H| > 1, \\
& M = G \cup H, V \in g_\tau(X_D(M) \times X_D(M)), \\
& \alpha_{ab} \in V : pr(e_{ab} \mid V \in \tau_g) = \frac{\| e_{ab} \|_q}{\sum_{\forall \alpha_{ab} \in \overline{V}} \| e_{ab} \|_q}
\end{aligned}
\qquad (11)
$$

Evidently, the probabilistic characterization is unbiased in nature having uniform as well as normed distribution. In the next section, a probabilistic model is constructed to resolve set A through the noncommutative composition of functions maintaining SL (Safety-Liveness) properties of mutex.

4. Resolving aggregate set A

It is considered in this paper that, $A \neq \phi$ and, the SL properties are maintained while resolving set A. If one considers that, unbiased randomization preserves fairness, then the proposed model in this section also preserves fairness of mutex.

Let two monotone sequences be defined as,

$$
\begin{aligned}
& n = |A|, B_n \subset Z^+, C_n \subset Z^+, \\
& B_n = (b_m)_{m=1}^{m=n}, b_m \in (0,+\infty), b_{m+1} = b_m + 1, \\
& C_n = (c_m)_{m=1}^{m=n}, c_m \in (0,+\infty), c_{m+1} > c_m
\end{aligned} \tag{12}
$$

If $f_S : Z^+ \to Z^+$ is a surjection function and, $\exists D \subset Z^+$, then $f_S(.)$ can be defined as,

$$
\begin{aligned}
& \forall x_m \in D, nx_m > 0, \\
& [\sin(nx_m) > 0] \Rightarrow \left[f_S(x_m) = \lfloor \sin(nx_m)^{-1} \rfloor \right], \\
& [\sin(nx_m) \le 0] \Rightarrow \left[f_S(x_m) = | n - x_m b_m | \right]
\end{aligned} \tag{13}
$$

The surjection is hybrid in nature having partly periodic and partly linear combination depending on interval conditions. The hybrid function ensures unbiased distribution in two disjoint domains having disparate distribution profiles of points in a set.

An integer valued translation function $g(.)$ is defined as follows,

$$
\begin{aligned}
& E = f_S(D), 1 \le k \le n, \\
& \forall y_k \in E, g(y_k) = y_k + c_k
\end{aligned} \tag{14}
$$

Furthermore, a random variable $X(.)$ is defined as,

$$
\begin{aligned}
& 1 \le j \le n, c_1 = b_n, \\
& H = \{ w_j : w_j = g(y_j) \wedge c_j = 2c_{j-1} \}, \\
& X : H \to Z^+ \cup \{0\}
\end{aligned} \tag{15}
$$

A noncommutative composite function is $(X_\circ g)(.)$ which prepares set A by incorporating elasticity along with unbiased randomization. The elastic diameters and ratio of diameters r are computed as,

$$
\begin{aligned}
& W \subset R^+ \cup \{0\}, \\
& diam(W) = | \max(W) - \min(W) |, \\
& G = \{ x : x \in (X_\circ g)(.) \}, \\
& G_f = \{ y : y \in f_S(.) \}, \\
& r = \frac{diam(G_f)}{diam(G)}
\end{aligned} \tag{16}
$$

The ratio of diameters may not be constant and it determines the varying elasticity of set during randomization at different instances. The elasticity of set affects the chances of collision during composite mapping. If the frequency of collision is low, then the computational complexity is high due to requirement of generating a complete injection map covering set in multiple rounds.

5. The Algorithms

This section presents a set of algorithms in pseudo-code form for probabilistic mutual exclusion, which is a refinement of logical clock based algorithm following partial ordering. First, the randomization algorithm is presented. The resolution algorithm of aggregate set is presented next. Finally, the refined algorithm for mutual exclusion is presented. The randomization algorithm is presented in Fig. 2.

```
//Algorithm : randomize (A)

∀p_j ∈ P:
Integer   m = 0, n = |A|, k, y;
Integer array  a[n];
Set   B = {p_k : e_k ∈ A};

while (m != n && B ≠ φ){
        k = pid (p_k ∈ B);
        y = f_S (k);
        a[m] = (X ∘ σ)(y);
        B = B\{p_k};
        m = m + 1;
}
```

Fig. 2. Pseudo-code representation of randomization algorithm

The randomization algorithm follows the randomization model to generate a set of processes having distinct infimum. The computation considers individual process identifier(s) (pid) of CS-requesting processes during randomization. If the distinct infimum cannot be generated in a single round, then multiple rounds would be required executing same algorithm realizing repeated randomization. The pseudo-code representation of resolving aggregate set in multiple rounds is given in Fig. 3.

```
//Algorithm : resolve (A)

∀pj ∈ P:
Integer n, y, z;
Set Bk , B;
Structure message  m[ ] = +∞; //initialized to invalid

Label:  n = |A|;
        Bk = {pk : ek ∈ A};
        if (n = 1 && pj ∈ Bk) enter_CS();
        else {  y = randomize (A);
                m [j] = < pj , y>;
                send (m[j], Bk \{pj});
                z = n – 1;
                while (z != 0) { receive (m[k], pk ∈ Bk);
                                 z = z – 1; }
        A = φ;
        B = find_min_equal (m[n]);
        A = A ∪ {ek: pk ∈ B};
        goto Label;
}
```

Fig. 3. Pseudo-code representation of aggregate set resolution algorithm

The `find_min_equal()` function represents searching the process in randomized set having infimum randomized value. The processes in aggregate set execute the resolution algorithm and mutually exchange the randomized local values. The reason for exchanging values is to ensure to global consistency of values within a subset of processes. If the distinct infimum of values is found, then the process generating the infimum proceeds into CS. Otherwise, if no single process is found having a distinct infimum, then repeated execution of the algorithm is performed. On each round of execution, the reduced infimum valued set is considered and other elements are discarded.

The refined probabilistic mutual exclusion algorithm is presented in pseudo-code format in Fig. 4. According to algorithm, if the first round of execution of standard logical clock based algorithm successfully generates a distinct infimum, then further randomization is not required and the process generating infimum proceeds to CS. In this case, the aggregate set is not formed by the execution sequence.

```
// Algorithm : pmutex(P)

∀pj ∈ P:
Boolean want_CS;
Integer  my_clock, agreed = 0;
Structure message m;
Set Bk = φ , A = φ ;
Macro LEAVING_CS = 0; //set to true at end of CS within CS procedure

if (want_CS = = 1) {
        Bk = Bk ∪ {pj };
        m.clock = my_clock;
        m.data = CS_request;
        send (m, P\{pj});
        }
if (receive (m)){
        if (want_CS = = 1){
                if (m.data = = CS_request && m.clock < my_clock) send (ok, m.pk);
                else if (m.data = = CS_request && m.clock = = my_clock) {
                                Bk = Bk ∪ {m.pk }; A = A ∪ {m.ek};}
                }
        else send (ok, m.pk);
        if (m.data = = ok) agreed = agreed + 1;
        else if (want_CS = = 1 && agreed = = |P| – 1) {
                if (|Bk | = = 1) enter_CS();
                else resolve (A);
                }
}
if (LEAVING_CS) send (ok, P\{pj});

//end
```

Fig. 4. Pseudo-code representation of refined probabilistic mutex algorithm

However, if the aggregate set is generated due to combinatorial execution sequence, then the algorithm fails to determine distinct infimum and proceeds to resolving the aggregate set. The algorithm calls for set resolution, which in turn calls for randomization in repeated rounds, if required. In any case, the algorithm successfully generates a distinct process generating infimum and selects the process to enter into CS. Note that, once full monotonic sequencing is performed on an aggregate set, each process in the set can enter CS one by one following that monotone. It indicates that, after generating a randomized monotone sequence, repeated executions to generate different sequences are not required. This reduces the overall computational complexity of the algorithm.

6. Analysis of Correctness and Topological Properties

This section presents analysis of algorithmic correctness and a set of associated properties. It is assumed that, the points in a set can be associated to a distribution function, where the underlying metric space of distributed systems is probabilistic in nature (i.e. events generated by processes are random in nature). Thus, the standard probably space of events generated by processes having a topological structure would result in formation of probabilistic topological space [25]. The estimation of interrelationship between event samples in a computing space is important for formation of aggregate set of processes, data and events [29]. The aggregate set A generated by processes is considered as a subspace in view of analysis, where $\overline{A}$ represents corresponding closed set.

6.1 Algorithmic Correctness Analysis

Let be $\exists n \exists m \exists s \in Z^+$ such that, $n = |A|$ and $m = |(X_{\circ} g)(U)|$ in a system, where initially $s = 1$ and $U \subset R$ is a finite set. In a distributed system, $\forall x \forall y \in A$, let $ip_x \in Z^+$ and $ip_y \in Z^+$ be representing respective process IDs (pid). According to the algorithm, if $n = m$ for $s = 1$ then $\forall x \forall y \in A, (X_{\circ} g)(ip_x)\big|_{s=1} \neq (X_{\circ} g)(ip_y)\big|_{s=1}$. Thus, the algorithm terminates with $s = 1$ by generating a monotone sequence, $J = (\beta_i \in R)_{i=1}^{|A|}$. However, if $n > m$ then $\exists s > 1$ such that, $\forall x \forall y \in A, (X_{\circ} g)(ip_x)\big|_{s>1} \neq (X_{\circ} g)(ip_y)\big|_{s>1}$ and, the algorithm terminates by generating $J = (\beta_i \in R)_{i=1}^{|A|}$ at $s > 1$. It is not possible to attain $n < m$ in a system in any case, and the algorithm executes in rounds to attain the condition, $|J| = |A|$. Furthermore, as $|P|$ is finite, thus monotone sequences B_n, C_n are finite as well as bounded by corresponding definitions. Hence, the algorithm successfully resolves aggregate set and sequences the processes in a convergent form.

6.2 Topological Analysis of Aggregate Set

In this section, the analysis of characteristics of aggregate set is presented considering underlying distributed computing space having topological nature. The aim of performing rigorous analysis is to gain a better insight to the dynamics of the system. The analytical results are presented as a set of theorems considering any probabilistically characterized set A_i in the topological space for any aggregate set A in the corresponding event space of processes in a system. If the structures of aggregate sets are different, then the corresponding probabilistically characterized sets are also represented using different indexes. In this section, N^+ represents a set of natural numbers.

6.2.1 Theorem 1: If $A_I \subset \tau_g$ such that, $\forall A_i, A_m \in A_I, A_i \cap A_m = \phi$ and $i, m \in N^+, i \neq m$, then $\sum_{\forall \alpha_{im} \in \overline{K}} pr(e_{im} \mid K \in \tau_g) \leq \sum_{\forall A_i \in A_I} \sum_{\forall \alpha_{im} \in \overline{A_i}} pr(e_{im} \mid A_i \in \tau_g)$ where, $K = \bigcup_{i=1}^{n} A_i$.

Proof: Let $A_I \subset \tau_g$ such that, $\forall A_i, A_m \in A_I, [i \neq m] \Rightarrow [A_i \cap A_m = \phi]$. If $|\tau_g| < +\infty$, then $\exists n \in N^+$ such that, $|A_I| = n$. However, if $pr(.)$ is locally uniform and complete, then within the respective finite topological space,

$$\forall A_i \in \tau_g, \sum_{\forall \alpha_{im} \in \overline{A_i}} pr(e_{im} \mid A_i \in \tau_g) = 1 \qquad (17)$$

Thus, if $K=\bigcup_{i=1}^{n}A_i$, then due to local uniformity and completeness in topological space, $\sum_{\forall\alpha_{im}\in\overline{K}} pr(e_{im}\,|\,K\in\tau_g)=1$. Hence, the topological space maintains the property that, $\sum_{\forall\alpha_{im}\in\overline{K}} pr(e_{im}\,|\,K\in\tau_g) < \sum_{\forall A_i\in A_I}\sum_{\forall\alpha_{im}\in\overline{A_i}} pr(e_{im}\,|\,A_i\in\tau_g)$.

6.2.2 Theorem 2: If $pr(.)$ is LUC everywhere in τ_g and, $\exists\{A_i,A_k\}\subset\tau_g$ where $A_i\cap A_k=C\neq\phi$ such that, $\sum_{\forall\alpha_{im}\in\overline{A_i\backslash C}} pr(e_{im}\,|\,A_i\in\tau_g)=\sum_{\forall\alpha_{im}\in\overline{A_k\backslash C}} pr(e_{im}\,|\,A_k\in\tau_g)$, then the distribution will maintain $\sum_{\forall\alpha_{im}\in\overline{A_i\backslash C}} pr(e_{im}\,|\,A_i\in\tau_g)=\frac{1}{2}\left(1-\sum_{\forall\alpha_{im}\in\overline{A_i\cap A_k}} pr(e_{im}\,|\,A_i\cap A_k\in\tau_g)\right)$.

Proof: Let be $\exists\{A_i,A_k\}\subset\tau_g$ where, $A_i\cap A_k=C\neq\phi$. By following the topological properties, $[\{A_i,A_k\}\subset\tau_g]\Rightarrow[A_i\cup A_k\in\tau_g]$. If $K=A_i\cup A_k$ and, $pr(.)$ is LUC everywhere in τ_g , then $\sum_{\forall\alpha_{im}\in\overline{K}} pr(e_{im}\,|\,K\in\tau_g)=1$. However, the rearrangement in topological space can be done as, $K=(A_i\backslash C)\cup(A_k\backslash C)\cup C$, where C is an open set maintaining condition that,

$$\sum_{\forall\alpha_{im}\in\overline{A_i\backslash C}} pr(e_{im}\,|\,A_i\in\tau_g)=\sum_{\forall\alpha_{im}\in\overline{A_k\backslash C}} pr(e_{im}\,|\,A_k\in\tau_g) \qquad (18)$$

Hence, the local distributions within the sub-spaces maintain, $\sum_{\forall\alpha_{im}\in\overline{A_i\backslash C}} pr(e_{im}\,|\,A_i\in\tau_g)=\frac{1}{2}\left(1-\sum_{\forall\alpha_{im}\in\overline{A_i\cap A_k}} pr(e_{im}\,|\,A_i\cap A_k\in\tau_g)\right)$.

Lemma : If $pr(.)$ is LUC everywhere in τ_g , then $\sum_{\forall\alpha_{im}\in\overline{K\backslash C}} pr(e_{im}\,|\,K\in\tau_g) < \sum_{\forall\alpha_{im}\in\overline{C}} pr(e_{im}\,|\,C\in\tau_g)$.

Proof: Let be $\exists\{A_i,A_k\}\subset\tau_g$ and, $K\in\tau_g : K=A_i\cup A_k$. If $C=A_i\cap A_k$, then according to topological property, $C\in\tau_g$. If $C\neq\phi$ and, $pr(.)$ is LUC everywhere in τ_g , then within the sub-spaces,

$$\left[\sum_{\forall\alpha_{im}\in\overline{K}} pr(e_{im}\,|\,K\in\tau_g)=1\right]\Rightarrow\left[\sum_{\forall\alpha_{im}\in\overline{K\backslash C}} pr(e_{im}\,|\,K\in\tau_g)<1\right] \qquad (19)$$

However, due to LUC property, $\sum_{\forall\alpha_{im}\in\overline{C}} pr(e_{im}\,|\,C\in\tau_g)=1$.

Hence, $\sum_{\forall\alpha_{im}\in\overline{K\backslash C}} pr(e_{im}\,|\,K\in\tau_g) < \sum_{\forall\alpha_{im}\in\overline{C}} pr(e_{im}\,|\,C\in\tau_g)$.

This indicates that, locality of probability measure within topological space has an effect on respective distribution profiles.

6.2.3 Theorem 3: Probabilistic estimation is topologically commutative, $\forall A_i \in \tau_g$ if $\exists A_k = \{(x_i, x_m) : (x_m, x_i) \in A_i\}$, such that $A_k \in \tau_{g\otimes}$ then $\forall \alpha_{im} \in A_i, \forall \beta_{mi} \in A_k$, $\sum_{\forall \alpha_{im} \in \bar{A}_i} pr(e_{im} \mid A_i \in \tau_g) = \sum_{\forall \beta_{mi} \in \bar{A}_k} pr(e_{mi} \mid A_k \in \tau_{g\otimes})$.

Proof: Let the two topological spaces τ_g and, $\tau_{g\otimes}$ be defined over X_D^2 such that,

$$\begin{aligned} &\forall A_i \in \tau_g, \exists A_k = \{(x_i, x_m) : (x_m, x_i) \in A_i\}, \\ &[A_i \neq \phi, A_i \in \tau_g] \Rightarrow [A_k \neq \phi, A_k \in \tau_{g\otimes}] \end{aligned} \quad (20)$$

However, due to symmetry, $\| e_{im} \|_q = \| e_{mi} \|_q$. Moreover, due to LUC property, $\forall \alpha_{im} \in A_i, \forall \beta_{mi} \in A_k$, $\sum_{\forall \alpha_{im} \in \bar{A}_i} pr(e_{im} \mid A_i \in \tau_g) = \sum_{\forall \beta_{mi} \in \bar{A}_k} pr(e_{mi} \mid A_k \in \tau_{g\otimes}) = 1$.

6.2.4 Theorem 4: If $\{e_i, e_m\} \subset E_D, \| e_{im} \|_q > 0$ and $\Delta(\alpha_{im})$ is a neighbourhood base of $\alpha_{im} \in D \in \tau_g$ such that, τ_g is first countable with $\{H \subset D, G \subset H\} \subset \Delta(\alpha_{im})$, then $\sum_{\forall \alpha_{im} \in \bar{G}} pr(e_{im} \mid D \in \tau_g) < \sum_{\forall \alpha_{im} \in \bar{H}} pr(e_{im} \mid D \in \tau_g)$.

Proof: Let $\{e_i, e_m\} \subset E_D, k \in Z^+$ and, $D \in \tau_g$ be such that, τ_g is first countable having $\Delta(\alpha_{im}) = \{A_k \subset D : \alpha_{im} \in A_k \wedge A_k \supset A_{k+1}\}$. Thus, in finite topological space $|\Delta(\alpha_{im})| < +\infty$ and, $\bigcup_{k=1}^{k=n} A_k \subset D$. Now, if $\| e_{im} \|_q > 0$ and, $\sum_{\forall \alpha_{im} \in \bar{D}} pr(e_{im} \mid D \in \tau_g) = 1$, then $\sum_{\forall \alpha_{im} \in A_k \subset D} pr(e_{im} \mid D \in \tau_g) < 1$. Hence, if $H = A_k, G = A_{k+1}$ having positive norm everywhere in space, then $\sum_{\forall \alpha_{im} \in \bar{G}} pr(e_{im} \mid D \in \tau_g) < \sum_{\forall \alpha_{im} \in \bar{H}} pr(e_{im} \mid D \in \tau_g)$.

6.2.5 Theorem 5: If $H \subset \tau_g$ and $f : H \to H$ where, $\forall A_i, A_m \in H, A_i \cap A_m = \phi$ and $f(A_i) \subset A_m$, then $\exists n \in Z^+, n > 0$ such that, $f^n(.)$ is topologically convergent.

Proof: Let be $H \subset \tau_g$ for the topological space τ_g such that, $|\tau_g| \in Z^+$ and, $f : H \to H$. If $H = \{A_i : A_i \in \tau_g \wedge i \in Z^+\}$ such that, $\forall A_i, A_m \in H, A_i \cap A_m = \phi$, then $\forall a, b \in Z^+, [a < b] \Rightarrow [f^a(A_i) \supset f^b(A_i)]$ if $f(A_i) \subset A_m$. Thus, $f^a(A_i) \cap f^b(A_i) = \phi$ and, $|\lim_{a \to +\infty} f^a(A_i)| < +\infty$. Hence, $f^n(.)$ is topologically convergent, where $\exists n \in Z^+, n > 0$.

7. Experimental Evaluation

7.1 Implementation Framework

The experimental evaluation of the algorithm is performed by implementing it in C language on Linux Fedora 2.6 platform. The implantation simulates a set of distributed processes by using multithreaded programming environment. The schematic representation of implementation architecture is presented in Fig. 5.

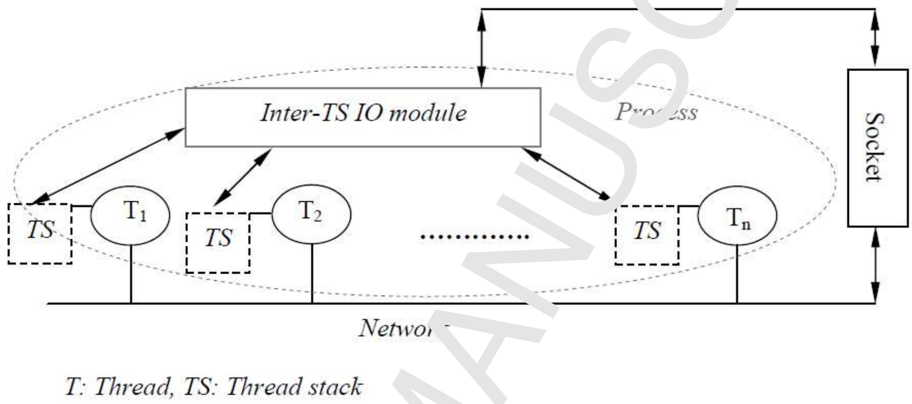

Fig. 5. Schematic representation of implementation model

The threads simulate a set of distributed processes connected by network. Each thread is associated with isolated thread stack frames, where the data IO between threads are implemented by inter-TS IO module over the Socket interface. The set of threads reside within an address space of a process. The threads execute the algorithms locally in respective nodes and exchanges computed values over network. The random variable *X*(.) is realized by seeded `srand(seed)` library function, where `seed` is computed by following randomization model presented earlier. The threads proceed to create a total order based on exchanged randomized key values. If the threads detect generation of same key values by a subset of threads, then that subset of threads proceed to second round of execution of same algorithm to establish a total order. The evaluation of system performance is conducted by measuring the following parameters: (a) variations of diameter of set after hybrid surjective mapping and composite mapping, (b) frequency of surjective collision, (c) variations of diameter ratio (r) with respect variations in number of nodes, (d) requirement of computation rounds for monotone sequencing of processes and, (e) surface map of interdependency between collision, diameter ratio and number of nodes in a system. In the experimentation, the initial base value is set to integer value 2 ($b_1 = 2$).

7.2 Evaluation of Algorithmic Performance

The variation of diameter of surjective map of a set with respect to number of nodes is presented in Fig. 6. The experimental result illustrates that, initially the diameters are not heavily inflated with respect to varying number of nodes. However, the diameter starts to get inflated strongly after node count exceeds a threshold (in this case 20). The diameter inflation peaks at node count equals to 25 and sets down to a lower value gradually if the node count is further increased. This indicates that, the surjection provides an expansion zone and a contraction zone depending on the node counts. This behaviour is due to the influence of periodic trigonometric function controlled by shifting base value in iterations.

The corresponding variation of diameter of set under randomized composite mapping is illustrated in Fig. 7. The composite map expands the diameter 100 folds approximately, on the average covering the entire range of node counts. However, the inflation dynamics is relatively monotonic and exhibits saturation effects for a given base value. Moreover, the monotonic expansion of diameter is nonlinear in nature.

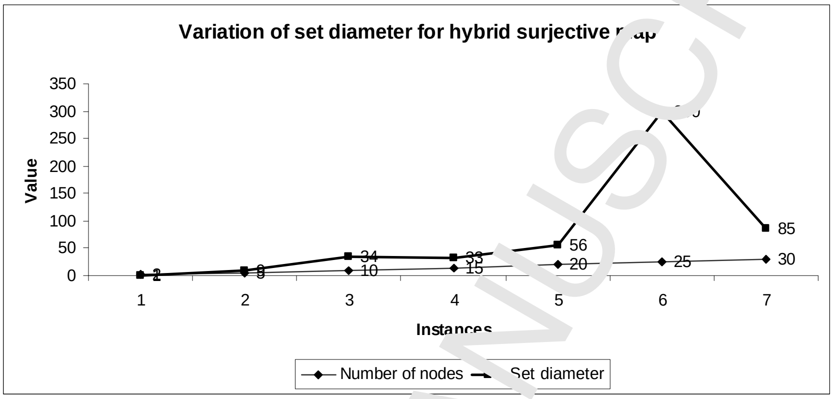

Fig. 6. Variation of diameter of surjective set mapping

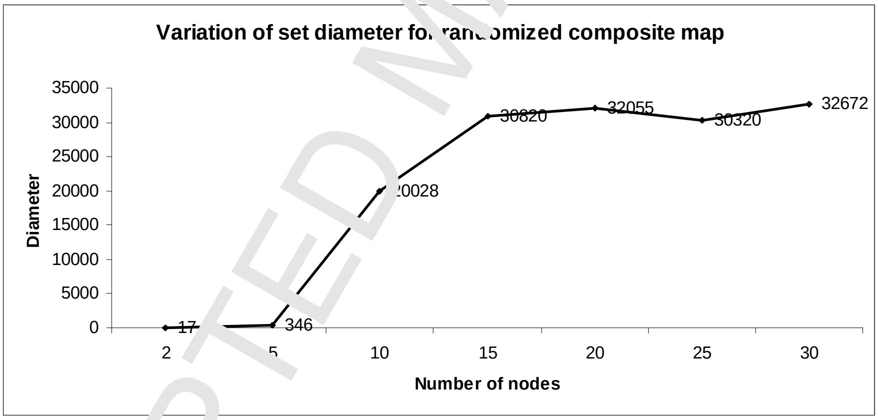

Fig. 7. Variation of diameter of randomized composite mapping

The variation of collision frequency under hybrid surjective map is illustrated in Fig. 8. The curve illustrates that collision frequency is varying with indeterminism and has nonlinear profile. However, the variation is having computable as well as distinct supremum and infimum values. It indicates that collision frequency is band-limited in nature. The value of diameter ratio of aggregate set for full execution of algorithm is illustrated in Fig. 9. The profile of diameter ratio variation illustrates that, with lower number of nodes, the ratio is much larger (i.e. composite map is highly elastic). However, the elasticity of randomization decreases monotonically as the number of nodes is increased. However, monotonic increase in umber of nodes results in nonlinear variation in elasticity in randomization.

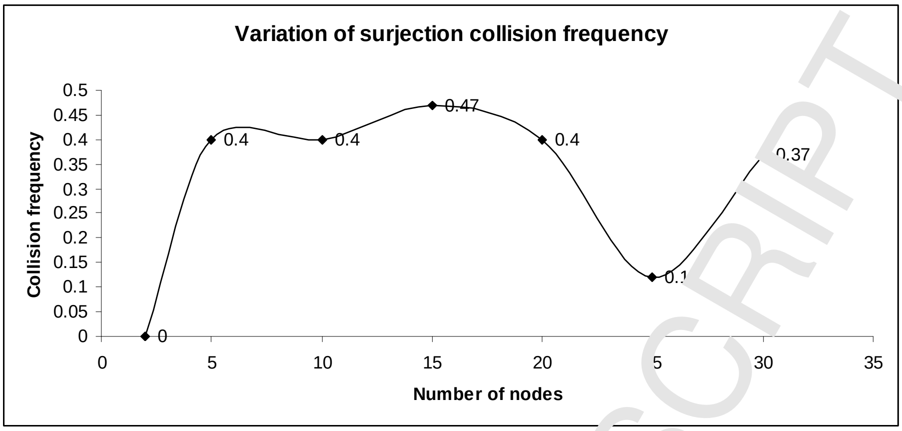

Fig. 8. Variation of collision frequency under hybrid surjection

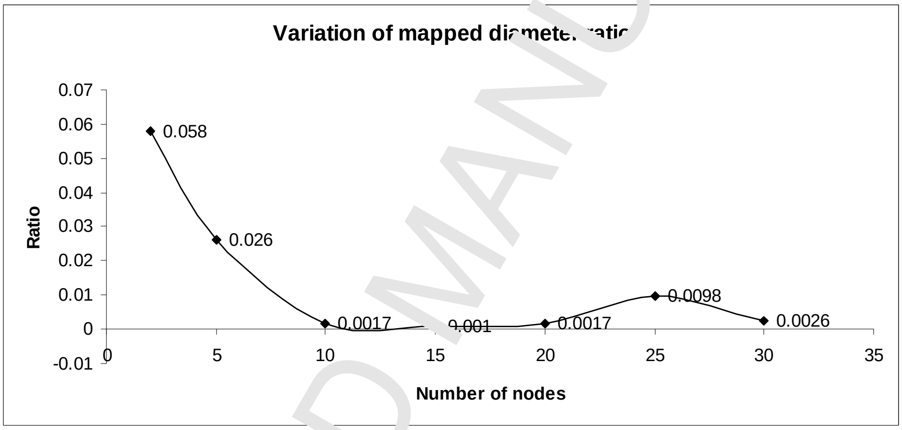

Fig. 9. Variation of diameter ratio under two maps

The reduction in elasticity enhances the requirement of multiple rounds of computation for ordering of processes to enter in CS. This effect is visible in Fig. 10. If the base value of primary sequence is lower, then the requirement of multiple rounds of computation starts at node count of 35. However, a single round of execution is enough for relatively lower node count.

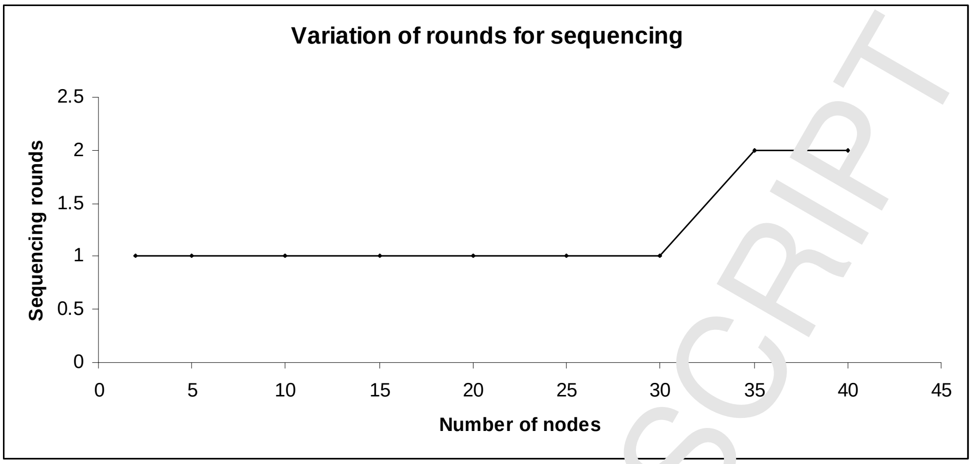

Fig. 10. Shifts in sequencing rounds

The dynamics of collision, node count and diameter ratio are illustrated in Fig. 11. The surface map is nonlinear and not strictly monotonic in nature. The surface map illustrates that, there exist multiple zones of elevations and, the nonlinear randomized variation within a zone is band-limited (i.e. computable in nature).

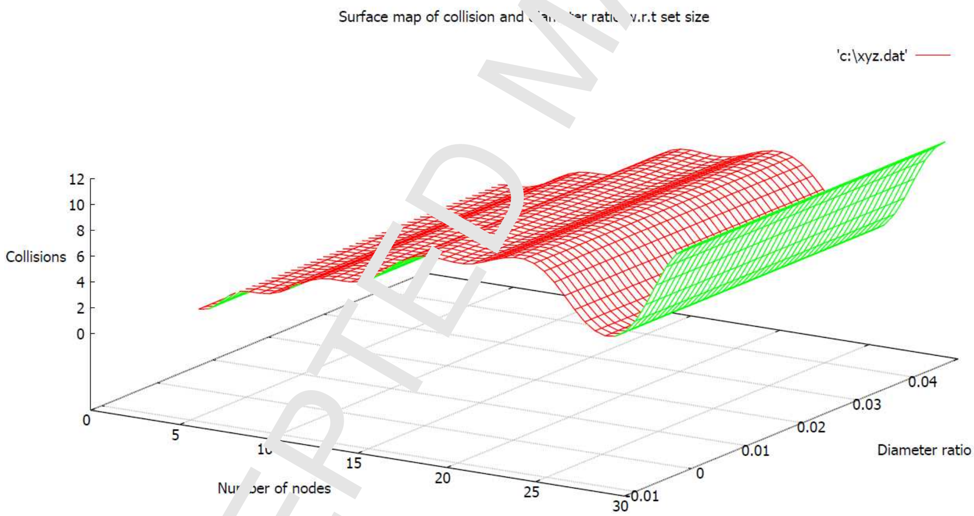

Fig. 11. Dynamics of collision, node count and diameter ratio

Furthermore, in all cases the aggregate set is successfully resolved and mutual exclusion is attained without failures.

7.3 Evaluations of Probabilistic Characterization

Experimental evaluations of probabilistic characterization model are carried out by using numerical simulation of different clusters of events with varying densities in event subspaces. The randomization of sampling of events to generate aggregate set A is simulated by using seeded randomized function. The distribution profiles of events clusters are presented in Fig. 12. The nature of density distribution is divergent between two profiles as presented in Fig. 12. One of the cluster density distribution profiles is approximately converged within a limiting value with respect to range of distribution representing monotonically increasing high density clusters of events. The second profile represents dilution effect (i. e. reduction in events cluster density) with the increasing norms with respect to increasing cluster size distribution. As a result, the second profile has a divergent distribution profile. The probabilistic estimations are performed for both low density and high density events clusters with varying norms in order to determine the resultant effect of norm on the respective estimations.

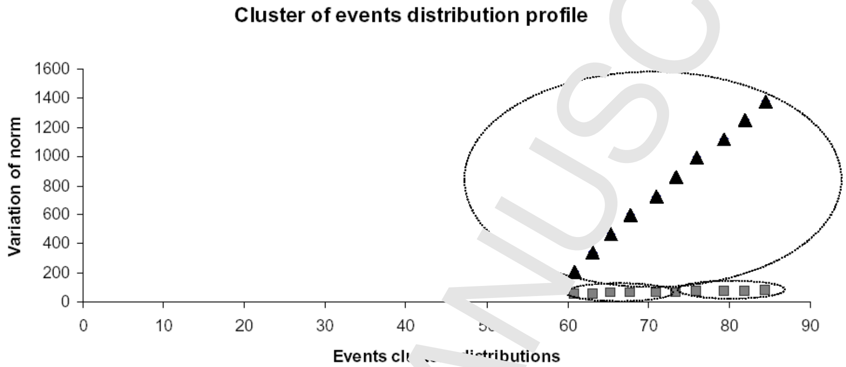

Fig. 12. Distributions profiles of events clusters with varying densities

7.3.1 Characterization I: High density of A

In this experiment, the density of clusters of events is considered to be high indicating a large cardinality of aggregate set A, which results in reduction in distances (d values) between samples within topological spaces of computation. The snapshot of variations of 2-norm values of distances (d values) between events is presented in Fig. 13. The correlation between d values and 2-norm values illustrates that both are monotonically increasing in nature with relative uniformity of distances with increasing number of events collected in a cluster from the underlying topological subspaces.

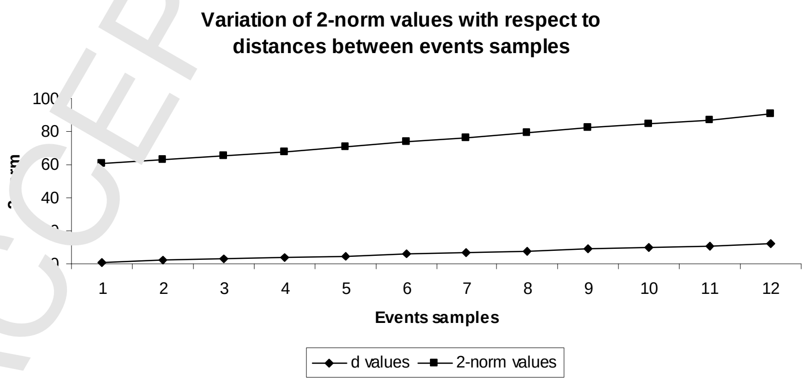

Fig. 13. Correlation of distances and 2-norm values of events samples

The corresponding distribution surface of probability estimation in 2-normed space is presented in Fig. 14. The distribution surface illustrates that, in high density clusters of events, the variations of probability estimations are highly non-uniform in nature. The non-uniformity of probability estimation is highly dependent on the cluster size affecting the relative distances between events. In other words, the structure of underlying topological subspace has influence on probability estimations in high density events clusters in 2-normed space.

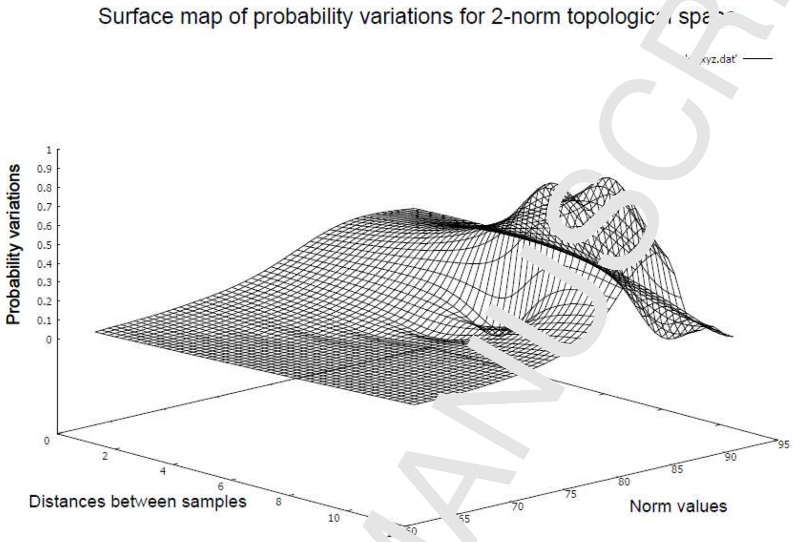

Fig. 14. Distribution surface of probability estimation in 2-normed high density events space

Next, the norm dimension is increased to 3 while keeping the cluster densities of events unchanged in order to detect the influence of norm dimension on estimations. The resulting variations of profiles of d values and 3-norm values are presented in Fig. 15. The profiles illustrate that, d values and 3-norm values tend to diverge if the number of samples are monotonically increased. This indicates that, correlation of d values and norm dimensions are mutually repulsive in nature.

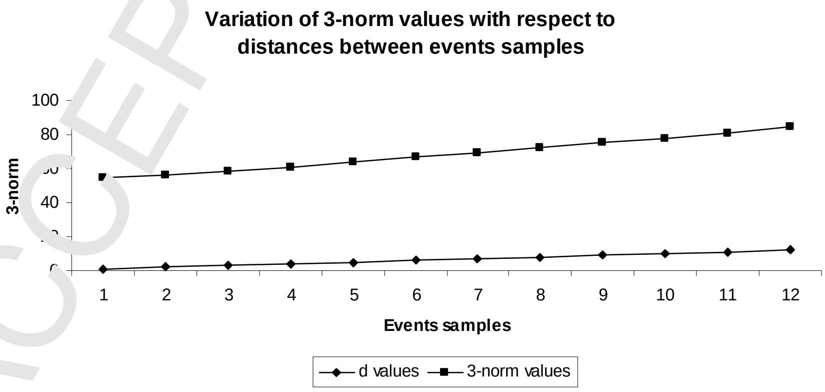

Fig. 15. Correlation of distances and 3-norm values of events samples

As a result, the probability estimation surface appears to be smoother in this case as illustrated in Fig. 16.

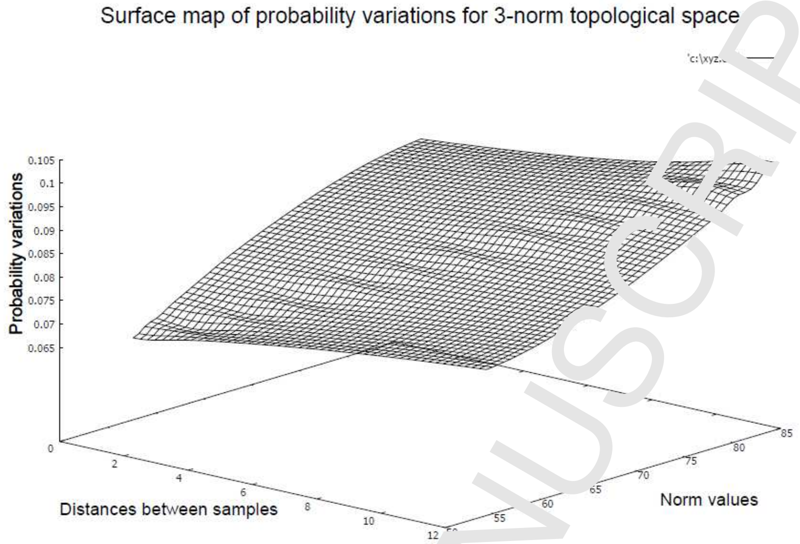

Fig. 16. Distribution surface of probability estimation in 3-normed high density space

The estimation surface illustrates that, the variations of d values, 3-norms and probability variations are mutually adjusting keeping the surface relatively smooth. However, there are appearances of occasional low intensity periodicity on probability estimation surface limited within a band.

7.3.2 Characterization II: Low density on A

In this experiment, the density of clusters of events is considered to be low, which results in increase in distances (d values) between samples within topological subspaces. The variations of 2-norm values and d values with respect to sample size are illustrated in Fig. 17. It is evident from figure that, the variations are mutually divergent in nature if the sample size is increased monotonically.

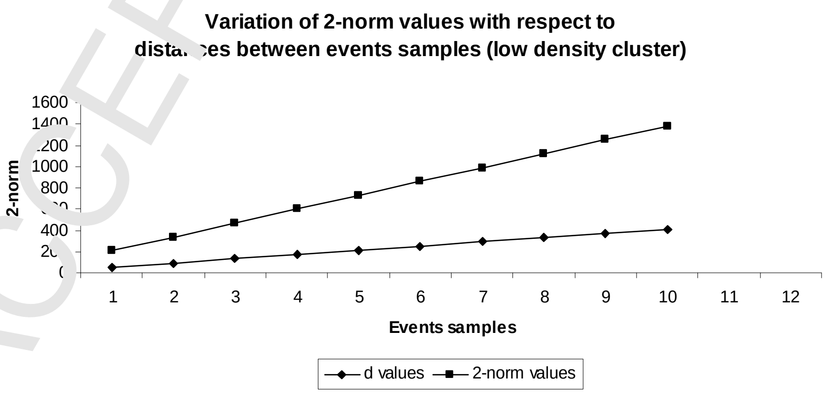

Fig. 17. Variations of d values and 2-norm values of events samples in low density cluster

The resulting surface of probability estimation in low density cluster of events is presented in Fig. 18. In the low density subspaces within clusters of events, the 2-normed estimation appears to be relatively stable all most everywhere. However, there are localized band limited periodicities in estimations on the surface of probability variations. This indicates that in low density clusters of events, the 2-norm based estimation may not distinguish probabilities of appearances of samples with sharpness.

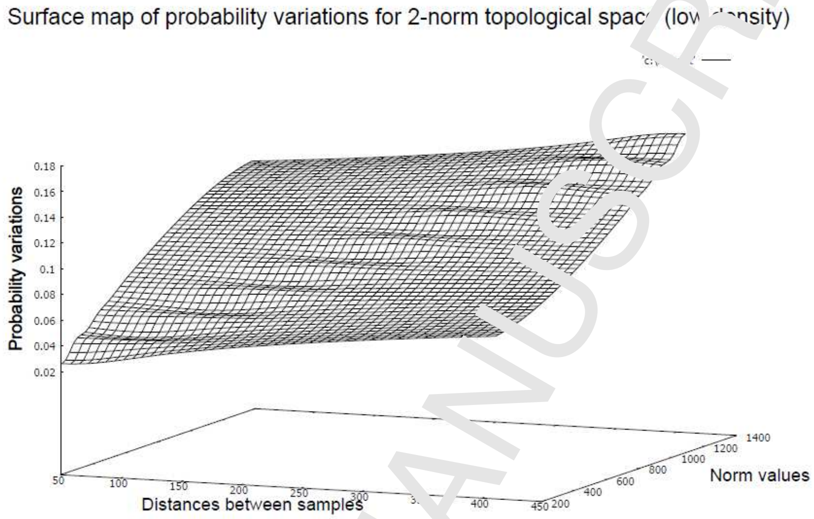

Fig. 18. Distribution surface of probability estimation in 2-normed low density space

The corresponding variations of d values and 3-norm in low density clusters of events with respect to monotonically increasing sample size are presented in Fig. 19. As expected, the correlation between d values and dimensions of norms is invariant in low density clusters of events. The main reason is that, the reduction in density results in reduction of mutual interference between events within a computing subspace.

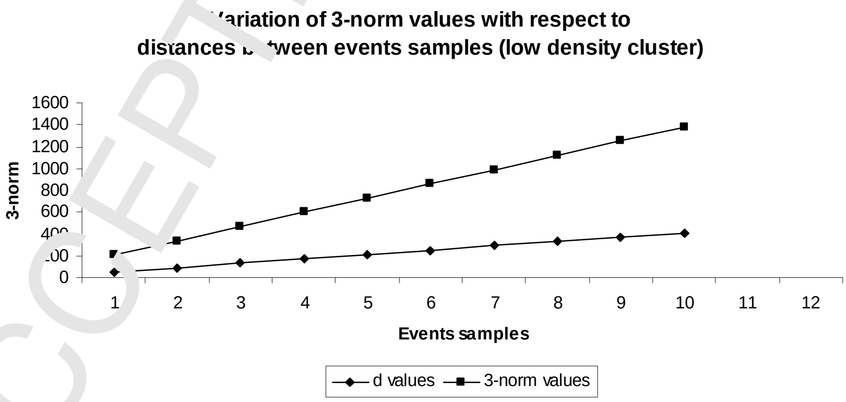

Fig. 19. Variations of d values and 3-norm values of events samples in low density cluster

The resulting surface of probability variations is presented in Fig. 20. Evidently, the identical invariance is observable in probability estimation surface, which is insensitive to the dimension of computation of norm.

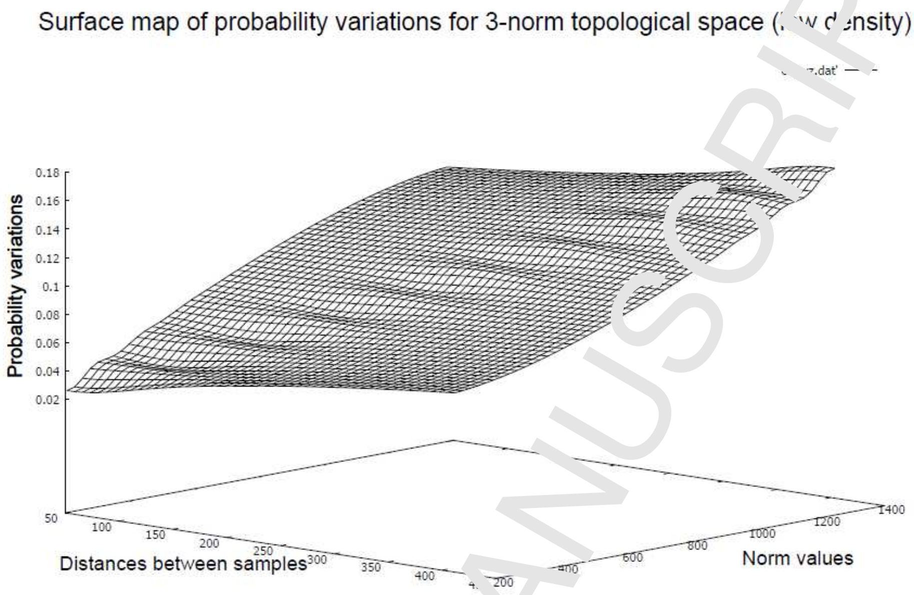

Fig. 20. Distribution surface of probability estimation in 3-normed low density space

In other words, the enhanced distances between samples in subspace effectively reduce the normed projection values making it relatively uniform in nature.

8. Comparative Analysis

This section presents comparative analysis of the proposed probabilistic mutex algorithm (called as PMA) with respect to a diverse set of algorithms in the domain. The number of nodes considered for PMA in comparative analysis represents the size of aggregate set. The set of algorithms considered for comparative analysis are: Token based distributed group mutex algorithm (TGM) [11], Optimal algorithm for mutex (OAM) [12], pN algorithm for mutex (PNA) [13], Tree based mutex algorithm (TMA) [17], Kanrar-Chaki algorithm (KCA) [4], Chang mutex algorithm (CMA) [3], Common optimal fair starvation-free algorithm (COA) [4], Hybrid token based mutex algorithm (HTA) [18] and, Local mutex algorithm (LMA) in static network [19]. The comparative analysis of algorithms considers three parameters such as, (1) message complexity, (2) response time and, (3) failure/collision count. The message complexities of algorithms are compared in two classes. In first category, algorithms are grouped depending on their computable deterministic message complexities. In the second category, the comparison is performed considering algorithms having varying complexities with a range (bounded).

The comparison of deterministic message complexities is presented in Fig. 21. The message complexity of OAM algorithm is a monotonically increasing function having relatively steep slope due to rounds of multicasting. The message complexities of TMA and PNA are nearly comparable. The message complexities of these two algorithms are having extremely low growth factor with respect to number of nodes because, communication does not need to cover the whole group. On the other hand, message complexities of PMA and TGM are comparable within a range for relatively lower number of nodes. However, the message complexities tend to diverge for PMA and TMA as the number of nodes increases. The PMA has relatively lower message complexity as compared to TGM, because PMA requires subset of processes, in general.

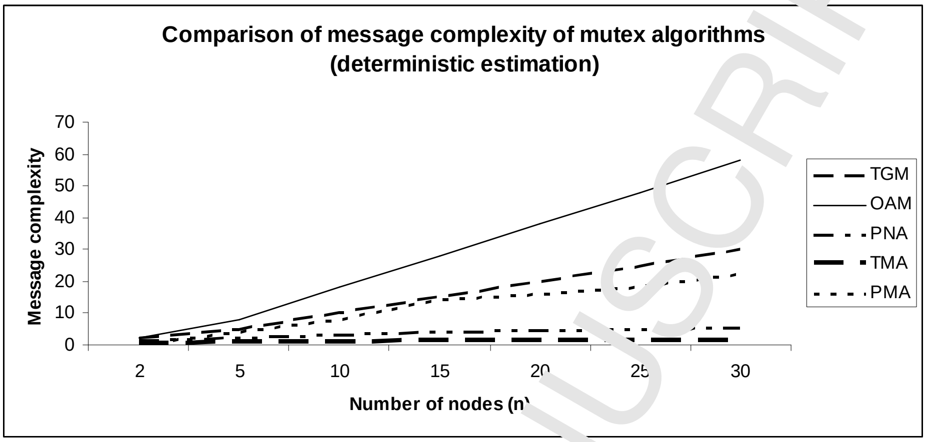

Fig. 21. Comparison of deterministic message complexities

The comparison of varying message complexities (bounded within an interval) is depicted in Fig. 22. It is observable that, message complexity of HTA is bounded within a bounded region (between HTA-I and HTA-II). However, the region of variation is not highly divergent in nature. On the other hand, PMA exhibits a uniform variation of message complexity having deterministic characteristics in all cases. If the number of CS-requesting nodes is lower, then the overall message complexity of PMA is lower than HTA. However, overall message complexity of PMA tends to monotonically increase if aggregate set size is monotonically increased.

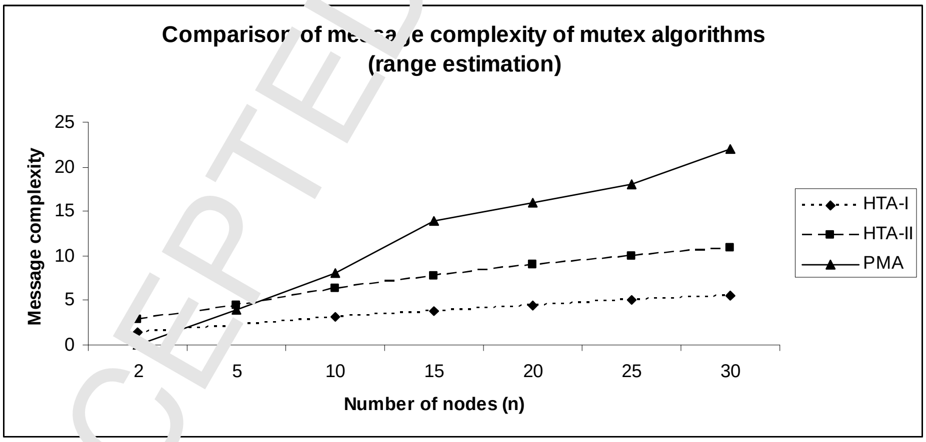

Fig. 22. Comparison of varying message complexities

The response time of algorithms are measured by using system clock having millisecond resolution in order to determine computational complexities of algorithms. The comparison of response time of mutex algorithms is presented in Fig. 23. The algorithmic response time represents the averaged execution time of algorithm while generating the mutex decision given a set of CS-requesting processes.

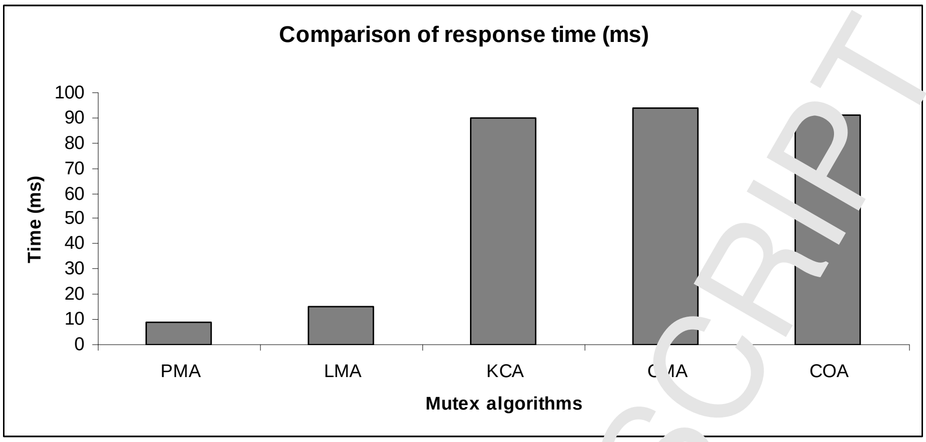

Fig. 23. Comparison of algorithmic response time

It is observable that, response time of KCA, CMA and COA are comparatively higher (nearly 3 folds) than the PMA and LMA models. However, algorithmic response times of PMA and LMA are comparable, where PMA offers relatively lower response time in a static network. The comparative analysis of varying failure/collision locality set size is presented in Fig. 24. In this case, the LMA is evaluated in static network following Linial model [22].

The variation of failure/collision between Linial LMA and PMA is diverging in nature with monotonically increasing number of nodes. The PMA model successfully reduces failure/collision rate by incorporating two-phased elastic randomization. However, in any case, the failure/collision count is monotonically increasing with respect to monotonic increment of number of CS-requesting processes.

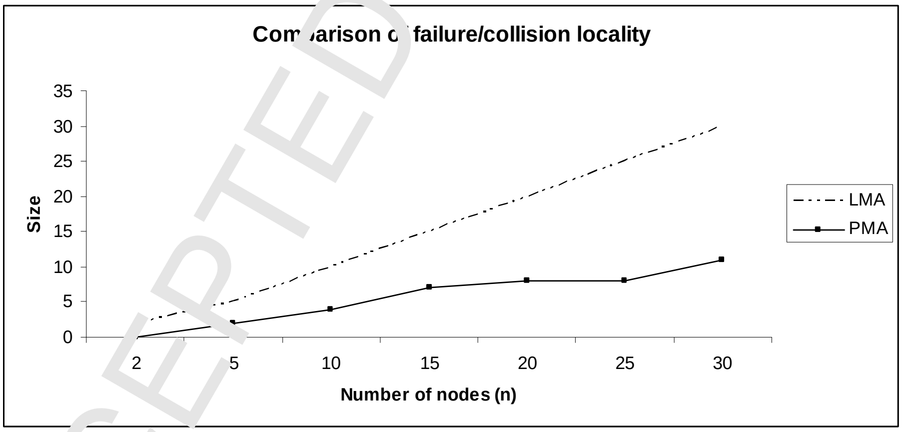

Fig. 24. Comparison of varying failure/collision locality set size

Although the failure/collision count is enhanced with respect to node count, however PMA model does not incur repeated rounds of execution of algorithm for generating the total order among the set of CS-requesting processes. This reduces the computational complexity to some extent reducing the overall response time.

9. Conclusion

The traditional distributed mutual inclusion and exclusion algorithms intended for shared resources are not completely suitable for direct applications in heterogeneous large scale mobile distributed systems. The generalized distributed mutual inclusion and exclusion algorithms execute in localized computing systems and, these algorithms tend to generate subset of processes in a queue waiting for critical section execution. These processes are equally eligible to enter into critical section. In specific cases, this violates safety property of distributed critical section if these processes are allowed to enter in critical section concurrently. The proposed failure analysis model identifies such conditions in analytical forms. The resolution of such group of processes is performed by employing a probabilistic algorithmic model. The computational evaluation of the algorithm illustrates that, it is suitable for maintaining safety property by incorporating concept of multi-phased elastic randomization. The detailed analysis of algorithmic correctness, probabilistic characterizations, and underlying topological structures are presented in this paper. The profiles of topological as well as probabilistic characterizations are evaluated through numerical simulations. The proposed distributed mutual exclusion algorithm is computationally inexpensive in nature.

References

[1] M. Bertier, L. Arantes, P. Sens, Hierarchical token based mutual exclusion algorithms, Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid, IEEE, USA , 2004, p.539-546.

[2] Leslie Lamport, Time, clocks, and the ordering of events in a distributed system, Communications of the ACM, Vol.21, No. 7, p.558-565, July 1978, DOI: 10.1145/359545.359563.

[3] I. Chang, M. Singhal, M.T. Liu, A hybrid approach to mutual exclusion for distributed system, in: Proceedings of the 14th IEEE Annual International Computer Software and Applications Conference, 1990, pp. 289-294.

[4] J. Lejeune, L. Arantes, J. Sopena, P. Sens, A fair starvation-free prioritized mutual exclusion algorithm for distributed systems, Journal of Parallel and Distributed Computing, Vol. 83, pp. 13-29, 2015.

[5] Ahmed Housni, Michel Trehel, Distributed Mutual Exclusion Token-Permission Based by Prioritized Groups, AICCSA '01 Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications, IEEE, USA, 2001, DOI: 10.1109/AICCSA.2001.933984.

[6] Alexander Kostin, Ljudmila Ilushechkina, Erhan Basri, A Comparative Performance Study of Distributed Mutual Exclusion Algorithms with a Class of Extended Petri Nets, ICA3PP 2005: Distributed and Parallel Computing, Lecture Notes in Computer Science, Springer, Volume 3719, 2005, pp. 11-22.

[7] Michel Raynal, A simple taxonomy for distributed mutual exclusion algorithms, ACM SIGOPS Operating Systems Review, Volume 25, Issue 2, April 1991, pp. 47-50.

[8] R. Baldoni, A. Virgillito, R. Petrassi, A distributed mutual exclusion algorithm for mobile ad-hoc networks, Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications, Italy, IEEE, 2002, DOI: 10.1109/ISCC.2002.1021727.

[9] Kayhan Erciyes, Cluster Based Distributed Mutual Exclusion Algorithms for Mobile Networks, Euro-Par 2004 Parallel Processing, Lecture Notes in Computer Science, Springer, Volume 3149, 2004, pp 933-940.

[10] A. Derhab, N. Badache, A distributed mutual exclusion algorithm over multi-routing protocol for mobile ad hoc networks, International Journal of Parallel, Emergent and Distributed Systems, Volume 23, Issue 3, Taylor & Francis, 2008, pp. 197-218.

[11] T. Masuzawa, H. Kakugawa and S. Kamei, A Token-Based Distributed Group Mutual Exclusion Algorithm with Quorums, IEEE Transactions on Parallel & Distributed Systems, Vol. 19, No. 09, pp. 1153-1166, 2008.

[12] G. Ricart, A. Agrawala, An optimal algorithm for mutual exclusion in computer networks, CACM: Communications of the ACM, 24, 1981.

[13] M. Maekawa, A pN algorithm for mutual exclusion in decentralized systems, ACM Transactions on Computer Systems, 3(2), pp.145-159, May 1985.

[14] M. Singhal, A dynamic information structure for mutual exclusion algorithm for distributed systems, IEEE Transactions on Parallel Distributed Systems, 3(1), pp. 121-135, 1992.

[15] I. Suzuki, T. Kasami, A distributed mutual exclusion algorithm, ACM Transactions on Computer Systems (TOCS), 3(4), pp. 344-349, 1985.

[16] M. Naimi, M. Trehel, and A. Arnold, A log (N) distributed mutual exclusion algorithm based on path reversal, Journal of Parallel and Distributed Computing, 34(1), pp. 1-13, 1996.

[17] K. Raymond, A tree-based algorithm for distributed mutual exclusion, ACM Transactions on Computer Systems, Vol. 7, No. 1, pp. 61-77, 1989.

[18] H. Taheri, P. Neamatollahi, M. Naghibzadeh, A hybrid token-based distributed mutual exclusion algorithm using wraparound two-dimensional array logical topology, Information Processing Letters, Vol. 111, No. 17, pp. 841-847, 2011.

[19] H. Attiya, A. Kogan, J. L. Welch, Efficient and robust local mutual exclusion in mobile ad hoc networks, IEEE Transactions on Mobile Computing, Vol. 9, Issue 3, 2010, pp. 361-375.

[20] S. Kamei, H. Kakugawa, An asynchronous message-passing distributed algorithm for the generalized local critical section problem, Algorithms journal, MDPI, Vol. 10, Issue 2, 2017.

[21] J. S. Barrios, M. E. Bermudez, Using wait-free synchronization in the design of distributed applications, Future Generation Computer Systems, Vol. 22, Issue 1-2, Elsevier, 2006, pp. 46-56.

[22] N. Linial, Locality in distributed graph algorithms, SIAM Journal of Computing, Vol. 21, Issue 1, 1992, pp. 193-201.

[23] R. Conde, S. Rajsbaum, An introduction to topological theory of distributed computing with safe-consensus, Electronic Notes in Theoretical Computer Science, Elsevier, Vol. 283, 2012, pp. 29-51.

[24] L. Fajstrup, M. Raussen, E. Goubault, Algebraic topology and concurrency, Theoretical Computer Science, Vol. 357, Issue 1-3, Elsevier, 2006, pp. 241-278.

[25] M. J. Frank, Probabilistic topological spaces, Journal of Mathematical Analysis and Applications, Vol. 34, No. 1, Elsevier, 1971.

[26] A. Ranganathan, E. Menegatti, F. Dellaert, Bayesian inference in the space of topological maps, IEEE Transactions on Robotics, Vol. 22, No. 1, IEEE, 2006.

[27] V. Snasel, J. Nowakova, F. Xhafa, L. Barolli, Geometrical and topological approaches to big data, Future Generation Computer Systems, Vol. 67, Elsevier, 2017.

[28] J. Li, X. Chen, Q. Huang, D. S. Wong, Digital provenance: Enabling secure data forensics in cloud computing, Future Generation Computer Systems, Vol 37, Elsevier, 2014.

[29] D. N. Reshef et al., Detecting novel associations in large data sets, Science, Vol. 334, Issue 6062, 2011.

[30] J. Wang, K. Qiao, Z. Zhang, An improvement for combination rule in evidence theory, Future Generation Computer Systems, Vol. 91, Elsevier, February 2019, pp. 1-9.

[31] C. H. Tai, C. T. Chang, Y. S. Chang, Hybrid knowledge fusion and inference on cloud environment, Future Generation Computer Systems, Vol. 87, Elsevier, October 2018, pp. 568-579.

[32] S. Kanrar, N. Chaki, FAPP: A new-fairness algorithm for priority process mutual exclusion in distributed systems, Journal of Networks, Vol. 5, Issue 1, 2010, pp. 11-18.

Susmit Bagchi has received B.Sc. (Honours) from Calcutta University in 1993, B. E. in Electronics Engineering in 1997 from Nagpur University, M. E. in Electronics and Telecommunication Engineering in 1999 from Bengal Engineering and Science University (presently IIEST). He has obtained Ph.D. (Engineering) in Information Technology in 2008 from Bengal Engineering and Science University. Currently, he is Associate Professor in Department of Aerospace and Software Engineering (Informatics), Gyeongsang National University, Jinju, South Korea. His research interests are in Distributed Computing Systems and Analysis.

Highlight:

- Distributed mutual inclusion-exclusion failure analysis and algorithm
- Probabilistic algorithm for refined distributed mutual exclusion
- Refined ordered probabilistic distributed mutual exclusion algorithm