On Dependability Evaluation of Mesh Connected Multiprocessors*

Prasant Mohapatra

Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa 50011

Chita R. Das

Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract

Analytical techniques for reliability and availability prediction of mesh connected systems are proposed in this paper. The models are based on the submesh requirements. First, a reliability model is proposed assuming that a submesh can be always recognized if it exits. Analysis of the linear consecutive n-out-of-N system is extended using an *expanding row/column* technique to evaluate the submesh reliability. An alternative approach called *row folding* is also discussed. Due to the high complexity involved in computing the exact reliability, both of these techniques use approximation to estimate lower bounds. Next, the submesh reliability is computed based on two different allocation policies, known as the two-dimensional buddy system (TDBS), and the frame sliding (FS). The model with the TDBS is further extended to estimate the reliability of multiple working submeshes which is useful in a multiuser environment. Availability analysis for a submesh of the required size is conducted using a Markov chain (MC). State truncation is used to reduce the computation time, and the MC is solved using HARP. Validation of the analytical models is done through extensive simulation. Issues, such as, reliability comparison based on allocation policies, and methods for improving system reliability are addressed using the analytical models.

Index Terms: Allocation-Based Reliability, Availability Model, Consecutive n-out-of-N System, Expanding Row/Column Technique, Markov Chain, Mesh Connected Systems, Submesh Dependability.

* This research was supported in part by the National Science Foundation under grant number MIP-9104485.

I. INTRODUCTION

The mesh topology is becoming a popular architecture for parallel machines. This is mainly due to its structural regularity for easy VLSI implementation, low degree, low bisection width†, and suitability for a class of applications known as *near neighbor problems*. Several prototypes and commercial machines built on the basis of mesh topology are ILLIAC IV, Goodyear Aerospace MPP, Tera Computer System, Touchstone Delta, Wavetracer's DTC, MasPar, T3D, and Paragon [1]. Some of these machines are targeted for SIMD applications, while the others are suitable for MIMD environment.

Along with performance, dependability‡ characterization of multiprocessors is essential to evaluate their effectiveness for critical and real time applications. While dependability analyses of other parallel architectures such as hypercube and MIN have been reported in the literature [3], there has been almost no work reported on mesh dependability. In this paper, we propose analytical techniques for reliability and availability prediction of two-dimensional mesh connected systems that provide near neighbor interconnections.

The dependability models analyzed here are derived from the task requirement, and are termed *submesh dependability*. The probability of a single working submesh or multiple working submeshes is obtained using the models. A near neighbor problem needs a mesh of a specific dimension for its execution. Commercial systems like Intel Paragon and Cray T3D have the capability to allocate variable size submeshes to incoming jobs. A single submesh dependability is an appropriate measure for both SIMD and MIMD machines where the system is dedicated to a single user. An MIMD mesh can be partitioned into independent submeshes and can run multiple jobs simultaneously [4,5]. Multiple submesh dependability measure is hence suitable for multitasked MIMD systems.

† Bisection width of a network is the minimum number of wires that have to be removed in order to partition the network into two halves with identical (within ± 1) number of processors.

‡ Dependability is a generic term used to address reliability, availability, maintainability, and the related issues [2].

Fault-tolerant study of mesh connected computers has been primarily focussed on the design and analysis of reconfigurable systems [6-8]. Earlier work on reliability of two-dimensional mesh structures (processor arrays) approaches the problem from a different perspective [9,10]. It uses a successive row and column elimination (SRCE), or an alternate row and column elimination (ARCE) technique for system reconfiguration in the presence of faults. The minimum working configuration consists of a single functional row of nodes. This is an appropriate approach for systolic environment. Recent mesh architectures, like Touchstone Delta and Paragon, use complex and powerful processors. Eliminating a complete row or column because of one node failure is neither economical nor an efficient approach. Gradual reduction of mesh size may not be acceptable for real time applications. Moreover, an application may define its requirement in terms of a specific size submesh. Submesh reliability computation is essential for these environments.

To our knowledge, there is no reported work on modeling of mesh dependability based on the submesh requirement. Developing an exact model to find a functional submesh in a mesh with faulty nodes is very difficult. This difficulty is attributed mainly to the mesh topology. Mesh does not have a recursive or hierarchical structure. Therefore, previous recursive techniques used for hypercube analyses [11], or hierarchical decomposition techniques used for MIN analyses [12, 13] cannot be used here. The conventional method of constructing a Markov chain (MC) for reliability computation is also not feasible for analyzing mesh reliability. It is probably impossible to construct an exact MC to find the submesh reliability. The state space explosion for even small meshes might be another problem with the Markov model. Thus, the existing packages like HARP [14], SHARPE [15], and other software tools cannot be used. We therefore propose various techniques to compute the submesh reliability either via approximation or through the simplified allocation policies used for locating a submesh.

Three different models are developed for computing the submesh reliability. The first one is based on the *perfect submesh recognition* ability. An allocation scheme with this capability is reported by Zhu [16] where a submesh can be always recognized if it exists in the system. The technique used for this reliability model is derived from the analysis of *consecutive n-out-of-N* systems. A consecutive n-out-of-N system consists of

N components aligned in a line and is considered operational if at least n consecutive components are working. The methodology is extended for a functional two-dimensional submesh. The difficulty in this generalization lies in modelling the alignment of the working nodes in different rows and columns. Finding the alignment probability is very challenging and has been left as an open problem [17,18]. We approximate the alignment by using an *expanding row/column* technique to get reasonable estimate of the reliability for up to 50% degradation. The basis of the technique is to conceptually increase the number of required nodes in each row or column to ensure the alignment irrespective of the locations of the working nodes. An alternate method based on *row folding* is discussed which gives a lower bound for higher degradation.

The other two reliability models capture the submesh allocation policies proposed in the literature. These are the two dimensional buddy system (TDBS) [4], and the frame sliding (FS) method [5]. Allocation-based reliability is useful since the operating system assigns a submesh using the underlying allocation algorithm. We derive exact expressions for mesh reliability with these two allocation policies. The analysis is further extended to find reliability of multiple submeshes which is useful for a multiuser environment.

The next part of our study focuses on analysis of repairable systems. System availability is used as a dependability measure for repairable systems. Traditionally, a detailed MC is used for analyzing system availability. However, it is extremely difficult (may be impossible) to construct a MC to find the existence of an arbitrary submesh in a larger mesh with a single repair facility. The availability model is therefore developed for finding a square mesh with the TDBS as the underlying allocation strategy. The working states of the MC represent the number of working elements along with the configuration that satisfies the task requirements. In order to keep the model tractable, we use state truncation. A software package called HARP [14] is used to solve the MC. Extensive simulation is conducted to validate the reliability and availability models.

A comparative study is conducted to analyze the effect of allocation policies on system reliability. It is observed that an allocation policy with better recognition ability not only improves performance but also provides better reliability. The two approaches for computing submesh reliability with *perfect recognition* are compared to show their effectiveness for various degradation. Next, the system reliability of a torus is studied and compared with a mesh of the same size. The results indicate that the reliability is considerably improved by converting a mesh into a torus (mesh with wrap-around connections).

The remainder of this paper is organized as follows. Preliminaries of the mesh structure and model assumptions are given in Section II. Analysis of submesh reliability with perfect recognition ability are presented in Section III. In Section IV, the allocation-based reliability models are derived for the two allocation schemes. Multitask reliability analysis is also described in this section. The availability model is developed in Section V. Model validation and various reliability trade-offs are discussed in Section VI, followed by the concluding remarks in Section VII.

II. PRELIMINARIES

Figure 1 shows a mesh connected system with NM nodes arranged in N columns and M rows. It is assumed that the submesh size requirement of an incoming task is pre-defined. A task requirement of an $(n \times m)$ submesh can be satisfied by a group of nm working nodes arranged in n columns and m rows. A system is considered operational as long as a fault-free submesh of the required size is available for task execution.

The dependability models are based on the assumption that all the nodes have identical exponential failure time distribution. It is assumed that the failure rate of a node includes the failure rates of the node processor and the links. Let the failure rate of a node be denoted as λ. The reliability of a node at time t, $R(t)$, is then equal to $e^{-\lambda t}$. In case of a node failure, the detection, isolation, and reconfiguration process may not be always perfect. This concept, termed as *imperfect coverage* and denoted as C, is defined as the probability that a system reconfigures successfully, given that a fault has occurred. For repairable systems, μ represents the repair rate with exponential distribution of repair time. After a repair, the system may or may not be able to reconfigure to a working state. The probability that a system is reconfigured to a working state after a node is repaired is termed as the *imprecise repair factor,* and is denoted as r.

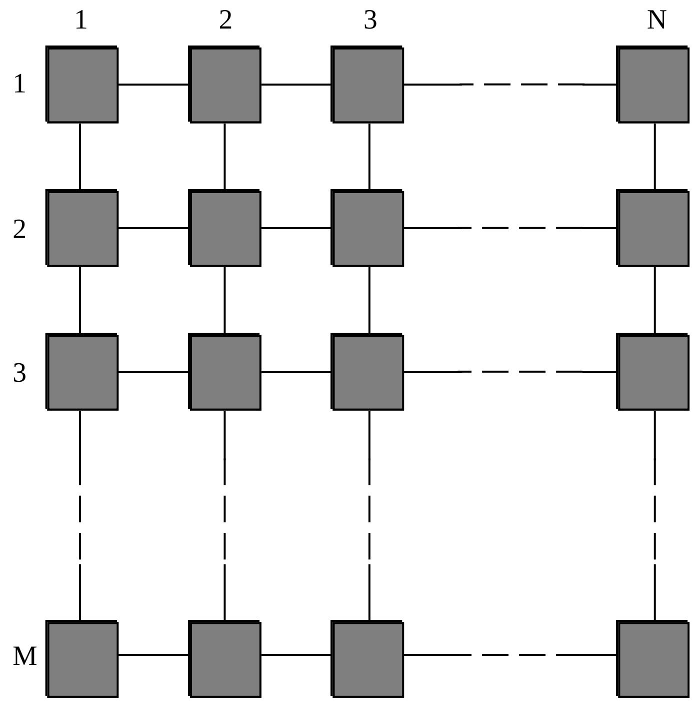

Fig. 1. An (NxM) Mesh Connected System

III. RELIABILITY WITH PERFECT SUBMESH RECOGNITION

The allocation scheme proposed by Zhu [16] has the capability of perfect submesh recognition which can always recognize a submesh in the system if it exists. The allocation algorithm is described in detail in [16]. For a mesh size of $(N \times M)$, if the task requires a submesh of $(n \times m)$, where $n \leq N$ and $m \leq M$, the time complexity of the perfect recognition scheme is $\Theta(MN)$ [16]. The space complexity of all the scheme is equal to $O(NM)$. In this section we will derive the system reliability with perfect submesh recognition ability.

The main idea of the approach presented here is derived from the *consecutive n-out-of-N* systems. The reliability of a *consecutive n-out-of-N* system is defined as the probability that at least n consecutive components work out of the N components aligned in a line. The exact reliability analysis of consecutive n-out-of-N components is presented next.

A. Consecutive n-out-of-N Reliability

Consecutive n-out-of-N systems have been studied extensively in the literature [17-19]. This reliability is derived with the aid of the following lemma.

Lemma 1: Let $Z(\alpha,\ \beta,\ \gamma)$ denote the number of ways in which α identical balls can be placed in β urns such that each urn has at most γ balls. Then,

$$Z(\alpha,\ \beta,\ 1) = \begin{cases} \binom{\beta}{\alpha}, & \text{for } 0 \le \alpha \le \beta; \\ 0, & \text{for } \alpha > \beta. \end{cases} \tag{1}$$

$$Z(\alpha,\ \beta,\ \gamma) = \sum_{i=0}^{\beta} \binom{\beta}{i} Z(\alpha - \gamma i,\ \beta - i,\ \gamma - 1), \qquad \text{for } \gamma \ge 2. \tag{2}$$

Proof: For $\gamma = 1$, and $0 \le \alpha \le \beta$, Z is just the number of ways in which α urns can be selected out of β urns. If $\alpha > \beta$, then at least one urn will have more than one ball. Hence, $Z = 0$, for $\alpha > \beta$. Consider the case when $\gamma = 2$. Let there be i urns which contain exactly two balls. The remaining $(\alpha - 2i)$ balls are distributed in $(\beta - i)$ urns such that each urn has at most one ball. This case is possible in $Z(\alpha - 2i,\ \beta - i,\ 1)$ ways. The i urns can be selected in $\binom{\beta}{i}$ ways. The general case can be derived recursively.

Q.E.D.

Consider a system with N components aligned in a line. Suppose i components are working and the remaining $(N-i)$ have failed. The system failure depends on the locations of the i components. It can be perceived that the i working components can be placed amongst $(N - i + 1)$ urns. The working components positioned before the first failed component are placed in the first urn. The working components between the first and the second failed components are placed in the second urn, and so on. The system fails if each of the urns have fewer than n components. There are $Z(i,\ N-i+1,\ n-1)$ arrangements that satisfy the condition. Let p be the probability that a component is working, and $q(= 1 - p)$ be the probability that a component has failed. The probability of system failure, P_f, is given as

$$P_f = \sum_{i=0}^{N} Z(i,\ N - i + 1,\ n - 1) p^i q^{N-i}. \tag{3}$$

The reliability of the *consecutive n-out-of-N* system, denoted as $R_{n|N}$, is then

$$R_{n|N} = 1 - P_f. \tag{4}$$

B. Model for Submesh Reliability

Reliability of an $(n \times m)$ submesh in an $(N \times M)$ system is computed on the basis of perfect submesh recognition ability. Perfect recognition ability means that a submesh is always recognized if it exists.

Reliability analysis of a submesh is done in three steps. First, we consider a single row (column) of nodes, and derive the exact reliability expression for at least n (m) consecutive working nodes. Second, the reliability of at least m (n) consecutive components is computed where each component represents a functional row (column). A row (column) is considered functional if at least n (m) consecutive nodes are working. An exact expression for this case can be also derived. Now, we have at least m (n) consecutive rows (columns), each of which has at least n (m) consecutive working nodes. These working nodes, however, may not be aligned to form an $(n \times m)$ submesh. The third step is to ensure the alignment of the working nodes to form an $(n \times m)$ submesh. All these steps are discussed in detail in this section.

Consider a single row of N nodes. The probability that there are at least n consecutive working nodes can be derived from equations (3) and (4) as

$$R_{n|N}(t) = 1 - \sum_{i=0}^{N} Z(i, N-i+1, n-1) R^i(t)[(1-R(t))C]^{N-i}. \quad (5)$$

The parameter C in the above expression represents the fault coverage factor of a node. The $(N \times M)$ mesh, shown in Figure 1, can be perceived as M components aligned in a vertical line, where each component represents a row of N nodes. For the task requirement under consideration, the reliability of a row, $R_{n|N}(t)$, becomes the reliability of each of the M component in the vertical line. Using the same methodology, the reliability of consecutive m-out-of-M components is given as

$$R_{m|M}(t) = 1 - \sum_{j=0}^{M} Z(j, M-j+1, m-1) R^j_{n|N}(t)[(1-R_{n|N}(t))]^{M-j}. \quad (6)$$

$R_{m|M}(t)$ denotes the probability that there are at least n consecutive working components in at least m consecutive rows. The analysis can be also done by first computing the reliability of consecutive m-out-of-M components. Equation (6) is then evaluated substituting m and M by n and N, respectively. The results obtained using both the approaches are identical when the alignment of working nodes is not taken into account. Selection of the appropriate approach while considering the alignment is discussed later.

The working components must be aligned so as to form an $(n \times m)$ submesh. The probability of such an alignment is extremely difficult to compute [18]. A rough estimation of the computation complexity can be derived as follows. Consider the alignment of only two rows. The probability that at least n of the components are aligned can be computed using four probabilistic terms as

$$\sum_{k=n}^{N} \text{Prob}\{k \text{ consecutive working nodes}\} \sum_{l=n}^{N} \text{Prob}\{l \text{ consecutive working nodes}\}$$

$$\sum_{i=1}^{N-n+1} \text{Prob}\{\text{alignment starts at } i\text{th column}\} \sum_{j=n}^{N} \text{Prob}\{j \text{ are aligned}\}.$$

The first two terms refer to the probabilities of at least n consecutive working components in the two rows. The third and fourth terms are the probabilities that the alignment starts at a certain column and extends to a specific length. The complexity is $O(N^2 N^2)$. For M rows, the complexity will be $O(N^2 N^M)$. A closer look will reveal even more terms that are not captured in the above expression. All these details are extremely difficult to model. We therefore use an approximation to analyze the alignment problem. The approximation is based on an *expanding row/column* technique and is derived from the following theorem.

Theorem 1: In an $(N \times M)$ system, if there are at least m consecutive rows each having $\lceil \frac{N+n}{2} \rceil$ consecutive working components, then a working $(n \times m)$ submesh can be obtained irrespective of the alignment of the components, where $n \leq N$, and $m \leq M$.

Proof: Consider two consecutive rows each of which has at least $(\lceil \frac{N+n}{2} \rceil)$ consecutive working components. The worst case misalignment happens when the consecutive patterns start from the opposite ends. In each row the pattern will exceed the mid point by at least $\frac{n}{2}$ components. There will be an overlap of at least n components among the two rows.

The argument can be similarly extended for m rows. Thus, if we consider at least m consecutive components where each component corresponds to a consecutive $(\lceil\frac{N+n}{2}\rceil)$-of-$N$ system, then a working $(n \times m)$ submesh is obtained irrespective of the alignments.

Q.E.D.

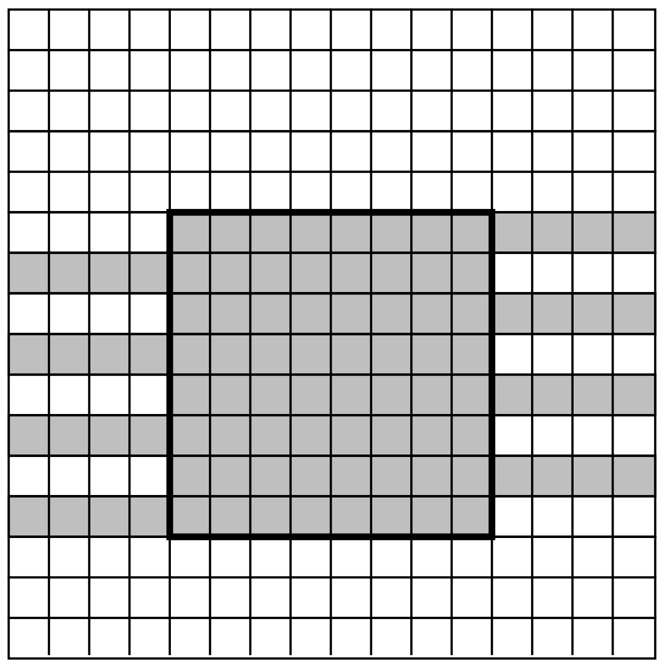

Fig. 2(a). Alignment of Consecutive Working Nodes

Figure 2(a) illustrates the alignment of eight consecutive components in eight rows. Here, $N = 16$, and $n = 8$. It can be seen that by considering $(\lceil\frac{N+n}{2}\rceil) = 12$ working components in each row, an (8×8) submesh is obtained irrespective of the location of the working components. The worst case scenario is depicted in Figure 2(a).

Corollary: In an $(N \times M)$ system, if there are at least n consecutive columns each having $\lceil\frac{M+m}{2}\rceil$ consecutive working components, then a working $(n \times m)$ submesh can be obtained irrespective of the alignment of the components, where $n \leq N$, and $m \leq M$.

Proof: It follows from the proof of Theorem 1.

Q.E.D.

Theorem 1 and its corollary are used for approximating the alignment of working nodes. The alignment is guaranteed by increasing the required number of nodes. There are two alternatives for the approximation. Using Theorem 1, at least $(\lceil \frac{N+n}{2} \rceil \cdot m)$ nodes are required for an $(n \times m)$ submesh, and a minimum of $(\lceil \frac{M+m}{2} \rceil \cdot n)$ nodes are required using the corollary. A tight bound is obtained by selecting the minimum of the above two requirements. It can be shown trivially that Theorem 1 should be applied when $Nm \leq Mn$, otherwise the corollary should be used. The explanation is supplemented by an example. Consider a (100×80) mesh, where the task requirement is a (60×70) submesh. Here, $Nm = (100)(70) = 7000$, and $Mn = (80)(60) = 4800$. As $Mn < Nm$, the corollary is used and a task requirement of $(n \times \lceil \frac{M+m}{2} \rceil)$, i.e. (60×75) is considered.

Equations (5) and (6) are used with the expanded submesh size to compute the system reliability. If $Nm \leq Mn$, equations (5) and (6) are evaluated by replacing n as $\lceil \frac{N+n}{2} \rceil$. Otherwise $(Mn < Nm)$, equation (5) is computed first replacing n and N by $\lceil \frac{M+m}{2} \rceil$ and M, respectively. The accuracy of the analytical results depends upon the additional nodes required for row or column expansion. The number of additional nodes required for a row or column expansion is less for low degradation. It is shown in Section VI that the model gives a very close estimation of system reliability in these cases. For higher degradation, the proposed approximation needs more working nodes to guarantee alignment, and thus should give a conservative reliability estimate.

C. Row Folding: An Alternative Approach

The technique proposed in [17] can be modified and extended to compute lower bound of the submesh reliability. The authors have proposed a model for an $n^2/N^2 : F$ system which is a square grid of size N. The system fails if and only if there is at least a square grid of side n that contains all failed components. We modify the technique to model an $(N \times M)$ mesh which is considered operational if there is at least one $(n \times m)$ submesh that contains all working components.

We represent the mesh as a matrix whose elements indicate the status of the nodes; 1 for working, and 0 for failed condition. Denoting the matrix in column major form as $X = (X_{i,j})$, an α-fold mesh ξ_j of rows $j,\ j+1,\ ...,\ j+\alpha-1$ is represented as a vector

$$(X_{1,j}, X_{1,j+1}, ..., X_{1,j+\alpha-1}, X_{2,j}, X_{2,j+1}, ..., X_{2,j+\alpha-1}, ..., X_{N,j}, X_{N,j+1}, ..., X_{N,j+\alpha-1}).$$

For example, if

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

the 2-fold mesh of rows 2 and 3 is (assuming that the bottom most row is numbered one)

$$\xi_2 = (0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 0),$$

and the 2-fold mesh of rows 3 and 4 is

$$\xi_3 = (1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1).$$

An α-fold mesh consists of $N\alpha$ elements. The authors in [17] state that if there are at least α^2 consecutive 1's in an α-fold mesh, an $(\alpha \times \alpha)$ submesh can be found. This is not always true. It depends on the location of the consecutive 1's. Although, the 2-fold mesh, ξ_2, in the example has four consecutive 1's, a (2×2) square grid with all 1's is not available.

It is imperative that if there are $(n+1) \cdot m$ consecutive 1's in an m-fold mesh, then the existence of a working $(n \times m)$ submesh is guaranteed. Let E_i be the event that an m-fold mesh ξ_i includes at least $(n+1) \cdot m$ consecutive 1's. The system works if at least one E_i occurs. The probability of E_i can be computed by analyzing a consecutive $(n+1)m$-out-of-Nm system. To make the E_i's independent, only the vectors ξ_1, ξ_{m+1}, ξ_{2m+1}, ..., ξ_{dm+1}, where $d = \lfloor \frac{M}{m} \rfloor$, are considered. These vectors have no common elements. The lower bound of the reliability using this row folding method is given as

$$R_{sys}(t) = 1 - (1 - R[E_i])^d, \tag{7}$$

where $R[E_i]$ is the reliability of a consecutive $(n+1)m$-out-of-Nm system.

The row folding method does not considers all possible locations of an $(n \times m)$ submesh and thus provides a pessimistic lower bound. This lower bound and the lower bound described in Section III.B are compared with the simulation results in Section VI.C.

IV. ALLOCATION-BASED RELIABILITY

The previous reliability model is based on perfect submesh recognition ability. The first-fit and the best-fit allocation algorithms proposed in [16] belongs to this catagory which cam recognize a submesh if it exits. Neither of the other two allocation policies (TDBS [4] and FS [5]) reported in the literature have perfect submesh recognition ability. The FS policy can be modified to recognize all possible submeshes. However, this will make the allocation policy overly complex. In this section, we develop reliability models capturing the effect of TDBS and FS allocation policies.

For a mesh size of $(N \times M)$, if the task requires a submesh of $(n \times m)$, where $n \leq N$ and $m \leq M$, the time complexities of the allocation policies are derived in [16]. If there are k allocated tasks in the system, the time complexities for the TDBS and FS policies are $O(MN)$ and $O((MNk)/(mn))$, respectively. The space complexities of both the schemes are the same and is equal to $O(NM)$.

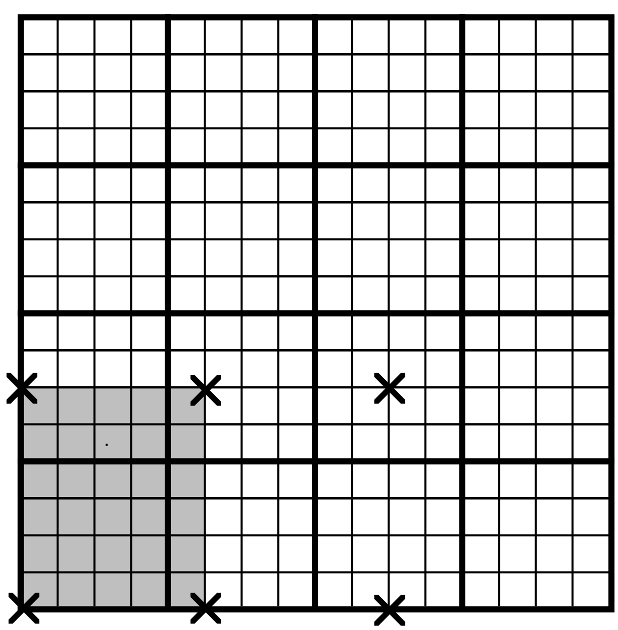

Fig. 2(b). Illustration of the TDBS and FS schemes on a (16x16) mesh

A. Reliability with the Two Dimensional Buddy Scheme

The two dimensional buddy system (TDBS) proposed in [4] is an extension of the one dimensional buddy policy used for memory management. The system size as well as the job (submesh) size are restricted to squares and the side lengths can only be powers of 2. Let (NxN) be the system size. If a job requires a mesh of size $(m \times m)$, then the submesh size is rounded up to the nearest power of 2. The required submesh size, denoted as $(n \times n)$, can be determined as $n = 2^{\lceil log_2 m \rceil}$. If an $(n \times n)$ submesh is not available, the algorithm searches for a larger submesh and splits it into smaller submeshes. An $(n \times n)$ submesh is then allocated to the job. A job is queued if a submesh of the required size is not available. After execution, a submesh is reclaimed and may be combined with other free submeshes to form a larger submesh.

An $(n \times n)$ submesh can be located in a fault-free $(N \times N)$ mesh at $(\frac{N}{n})^2$ different places with the TDBS. These locations are distinct and the status of nodes in one location is independent of the other locations. As an example, the sixteen (4×4) submeshes in a (16×16) mesh with the TDBS are shown enclosed in bold lines in Figure 2(b). The system reliability, $R_{sys}(t)$, is given as

$$R_{sys}(t) = \sum_{i=1}^{\rho} \binom{\rho}{i} [R_n(t)]^i \left(\sum_{j=1}^{n^2} \binom{n^2}{j} [(1 - R(t))C]^j [R(t)]^{n^2 - j} \right)^{\rho - i}, \qquad (8)$$

where $R_n(t)$ is the reliability of a submesh of size $(n \times n)$, and $\rho = (\frac{N}{n})^2$. $R_n(t)$ can be obtained as

$$R_n(t) = (R(t))^{n^2}. \qquad (9)$$

In equation (8), the probability of at least one working $(n \times n)$ submesh is multiplied with the probability of $(\rho - i)$ failed submeshes. Note that a submesh fails if any one of the n^2 nodes is faulty. The second part of equation (8) includes the effect of coverage factor, C, of the failed nodes.

B. Reliability with the Frame Sliding Scheme

The frame sliding (FS) strategy is proposed in [5]. Unlike the TDBS, it is applicable to any rectangular mesh, and a job is allocated exactly the same size submesh that it needs. The size of a "frame" is identical to that of a requested submesh. The shaded portion in Figure 2(b) represents a frame for a (6×5) submesh. Frames are examined one at a time starting from the lower leftmost available node (coordinate). When nodes in the currently examined frame are not all available (or working), the frame is slid over the "plane" of the mesh for searching next candidates, with the horizontal and vertical strides equivalent respectively to the width and height of the requested submesh. The "X"s represent the bottom leftmost coordinates of the (6×5) submeshes with the FS scheme. The search is continued until a frame involving only free (and working) nodes is found or all candidate frames are exhausted. In the later case, the job is queued until a submesh of the required size is available.

Let (NxM) be the system size, and $(n \times m)$ be the required submesh size. Using the FS method, there are $(\lfloor \frac{N}{n} \rfloor \times \lfloor \frac{M}{m} \rfloor)$ places where the submesh can be allocated. In Figure 2(b), using the FS method, the (6×5) submesh can be allocated at $(\lfloor \frac{16}{6} \rfloor \times \lfloor \frac{16}{5} \rfloor) = 6$ locations. There is no overlapping between these locations. Therefore the system reliability can be written as

$$R_{sys}(t) = \sum_{i=1}^{\theta} \binom{\theta}{i} [R_{nm}(t)]^i \left(\sum_{j=1}^{nm} \binom{nm}{j} [(1 - R(t))C]^j [R(t)]^{nm-j} \right)^{\theta - i}, \qquad (10)$$

where

$$\theta = \lfloor \frac{N}{n} \rfloor \times \lfloor \frac{M}{m} \rfloor, \quad \text{and} \quad R_{nm}(t) = [R(t)]^{nm}.$$

C. Multitask Reliability

Multitask reliability prediction is essential in a multiuser environment where each user needs a specific size submesh for execution of its task. The multitask reliability model developed here uses the two-dimensional buddy scheme for task allocation.

Since we are using the TDBS for allocation, the system size as well as all the submeshes need to be squares with side lengths of powers of 2. Let $J_1, J_2, \ldots, J_k$ denote the submesh sizes of k tasks. The multitask reliability for the specified requirements is defined as the probability that there are k working disjoint submeshes of dimensions $\{J_1,\ J_2,\ \ldots, J_k\}$ at time t. It is extremely difficult to obtain a closed form expression for computing multitask reliability. Let J denote the smallest square (with side lengths of powers of 2) such that all the required submeshes can be contained in it without overlapping. A conservative bound of the system reliability can be obtained by computing the probability of a single working submesh of size $(J \times J)$.

A tight bound of the multitask reliability can be obtained by using a best fit approach. Let the k jobs be arranged in decreasing order of the submesh size. Let the ordered list be denoted as $\{j_1,\ j_2,\ \ldots, j_k\}$. Note that j_i is a power of 2, for $1 \le i \le k$, and represents a submesh of size $(j_i \times j_i)$. The jobs are allocated starting with the largest submesh. We describe the methodology with an example. Consider a (256×256) mesh. Let there be three jobs which need submeshes of size (64×64), (32×32), and (16×16), respectively. The reliability of a (64×64) submesh in a (256×256) mesh, denoted as $R_{64}(t)$, can be computed using equation (8). The (64×64) submesh can be located at $(\frac{256}{64})^2 = 16$ different places with the TDBS. After the allocation of a (64×64) submesh, there are $(16 - 1)$ places of size (64×64), and thus a (32×32) submesh can be located at $(\frac{64}{32})^2(16 - 1) = 60$ possible places. $R_{32}(t)$, the probability of finding a (32×32) submesh after the allocation of a (64×64) submesh, can be computed using equation (8). Similarly, $R_{16}(t)$ can be computed after finding the number of possible places to locate a (16×16) submesh. The (16×16) submesh can be located at $(\frac{32}{16})^2(60 - 1) = 236$ different places. The system reliability for the example under consideration is thus given as

$$R_{sys}(t) = R_{64}(t) \cdot R_{32}(t) \cdot R_{16}(t). \tag{11}$$

Algorithm for the Multi-Task Reliability Computation

Let the number of disjoint places where a $(j_i \times j_i)$ submesh can be located in an $(N \times N)$ mesh is denoted as ρ_i.

1. Arrange the jobs in decreasing order of submesh size. Let $\{j_1,\ j_2,\ ...,j_k\}$ be the ordered list.
2. Compute ρ_i's for $1 \leq i \leq k$, using

$$\rho_i = (\frac{j_{i-1}}{j_i})^2(\rho_{i-1} - 1),\ \text{for } 1 < i \leq k, \tag{12}$$

   where $\rho_1 = (\frac{N}{j_1})^2$.
3. Calculate the reliabilities, $R_{j_i}(t)$'s for all j_i's using the ρ_i's, and equation (8).
4. The system reliability $R_{sys}(t)$ is finally calculated from

$$R_{sys}(t) = \prod_{l=1}^{k} R_{j_l}(t). \tag{13}$$

V. AVAILABILITY MODEL

In this section, we propose a methodology for predicting system availability using a simple Markov chain (MC). The objective is to find the probability of a working submesh in a repairable environment. Task allocation is assumed to be done using the TDBS (a $(2^i \times 2^i)$ submesh is allocated for an incoming job). Construction of a detailed MC representing all possible states for finding any rectangular submesh with the perfect recognition or the FS scheme is almost impossible. The technique can be best explained by first considering an example of a small system, and then extending it to a generalized technique.

Consider a 16-node (4×4) mesh where the task requirement is a (2×2) submesh. A (2×2) submesh can be located in a fault-free (4×4) mesh at $(\frac{4}{2})^2 = 4$ different locations with the TDBS. The system is considered operational as long as all the nodes in at least one of the four locations are working. Intuitively, one might suggest developing a MC with a vectorial notation, where each element of the vector could represent the number of working nodes at each of the four locations. This has two drawbacks. First, the MC becomes overly complex when the number of elements in the vector increase. Second, the single repair facility for the whole system cannot be subdivided among the elements of the vector. Thus, it is essential to integrate the working nodes for proper representation of the repair rate.

The functional states of the system could have w working nodes, where $4 \leq w \leq 16$. Note that all the states with w working elements may not provide a (2×2) working submesh. Moreover, the submesh should also be identifiable by the TDBS. Let p denote a working state, $4 \leq p \leq 16$, where the working nodes provide at least one identifiable fault-free (2×2) submesh.

The state transition rates can be derived as follows. Normally, the transition from a state $(p+1)$ to p would be at a rate $(p+1)\lambda$, for $4 \leq p < 16$. But we are considering the task requirements of a working submesh. Therefore a state containing p functional nodes may or may not be a working state depending on the location of the nodes. The system goes to a failed state if the configuration of the working nodes does not form a submesh of the required size. So the transition from a working state $(p+1)$ to the next working state p due to the failure of an element may not be at a rate $(p+1)\lambda$. Let α_{p+1} be the probability with which it goes to the next working configuration. The values of α varies from state to state. Thus the transition rate from the working state p to the next working state would be $p\alpha_p\lambda$. The system can move from a working state p to a failed state due to the following conditions:

(a) The failure of a node such that a fault-free submesh of the required size cannot be found at any of the locations defined by the TDBS. The rate at which the system fails due to this factor is $(1-\alpha_p)p\lambda$.

(b) The system can fail at a rate $\alpha_p p\lambda(1-C)$ due to imperfect coverage.

(c) The system goes to a failed state at a rate of $(1-r)\mu$ due to imprecise repair.

In order to quantify the failure rates, we need to determine the values of α_p's. Computation of α_p's for the example under consideration is described next. Let (i, j, k, l) be a 4-tuple working state that represents the number of nodes in the four identifiable locations in state p, where $i+j+k+l = p$ and $i \vee j \vee k \vee l = 4$ ($\vee$ denotes the 'OR' operation). The number of nodes in the four locations at state $(p+1)$ could be either $(i+1, j, k, l)$ or $(i, j+1, k, l)$ or $(i, j, k+1, l)$ or $(i, j, k, l+1)$. The transition rate from state $(p+1)$ to state p is either $(i+1)\lambda$ or $(j+1)\lambda$ or $(k+1)\lambda$ or $(l+1)\lambda$. A recursive function Υ representing the total number of ways in which a particular state can be reached, is given as

$$\Upsilon(i,j,k,l) = \Upsilon(i+1,j,k,l)(i+1) + \Upsilon(i,j+1,k,l)(j+1) + \Upsilon(i,j,k+1,l)(k+1) + \Upsilon(i,j,k,l+1)(l+1). \quad (14)$$

The recursion terminates for $\Upsilon(4,4,4,4)$. $\Upsilon(4,4,4,4) = 1$ and $\Upsilon(i,j,k,l) = 0$ when $i,j,k,l > 4$. We are not taking into account the transition from p to $(p+1)$ due to repair because here we are only concerned with finding the transition rate due to a processor failure.

Let Φ and Θ be two quantities defined as: $\Phi = \sum \Upsilon(i,j,k,l)$ for all possible i,j,k, and l that satisfy the task requirement, and $\Theta = \sum \Upsilon(i,j,k,l)$ for all possible i,j,k, and l that do or do not satisfy the task requirements. α_p is thus equal to Φ/Θ. For the example under consideration, when $p = 15, 14, 13$, all possible combinations satisfy the task requirements. So $\alpha_{16}, \alpha_{15}, \alpha_{14} = 1$. For the next state transition, *i.e.* from 13 to 12, $\alpha_{13} = 0.86$. So with a rate $(0.86 \times 13\lambda C) = 11.17\lambda C$, the system goes to a working state, and with $(13\lambda - 11.17\lambda C)$, it goes to a failed state. Other state transition rates are shown in Figure 3 after simplification.

We assume on-line repair process. When a node fails, the repair process is invoked with a rate $r\mu$. The coverage factor r denotes the probability that the fault is detected and is successfully repaired. The repair process of a node that is required to maintain the configuration as demanded by the task is given priority. This facilitation improves the system availability. Thus there are no transitions from one failed state to another failed state.

For the general case, consider an $(N \times N)$ mesh with a task requirement of $(n \times n)$ submesh, where n and N are powers of 2, and $n \leq N$. Let ρ denotes the number of locations where an $(n \times n)$ submesh can be found in the $(N \times N)$ mesh using the TDBS. $\rho = (\frac{N}{n})^2$. We denote the the working elements in state p as a vector of length ρ and is represented as,

$$\overline{X}_\rho^p = [X_1, X_2, ..., X_i, ..., X_\rho], \quad \sum_{i=1}^{\rho} X_i = p,$$

Fig. 3. Markov Chain for the (4×4) System

Fig. 4. Markov Chain for the $(N \times N)$ System

where X_i represents the number of working nodes in location i. The function Υ can be expressed as

$$\Upsilon(\overline{X}_\rho^p) = \sum_{i=1}^{\rho} \Upsilon(\overline{X}_\rho^{p+1_i})(X_i + 1) \tag{15}$$

where,

$$\overline{X}_\rho^{p+1_i} = [X_1, X_2, ..., X_i + 1, ..., X_\rho].$$

The above expression represents that state $\overline{X}_\rho^p$ could have been preceeded by any of the working configurations, $\overline{X}_\rho^{p+1_i}$, for $1 \leq i \leq \rho$. The transition from $\overline{X}_\rho^{p+1_i}$ to $\overline{X}_\rho^p$ is at a rate $(X_i + 1)\lambda$, which accounts for the term $(X_i + 1)$ in Equation (15). λ is ignored as we eventually take a ratio of two functions involving the same order of λ's. Note that $\Upsilon = 1$ if $X_i = n$ for all i and $\Upsilon = 0$ if $X_i > n$ for any i. Θ_p represents the summation of Υ for all possible combinations of $\overline{X}_\rho^p$ (working or failed), and is given by

$$\Theta_p = \sum_{\substack{X_1, X_2, ..X_\rho = 0 \\ X_1 + X_2 + .. + X_\rho = p}}^{n^2} \Upsilon(\overline{X}_\rho^p). \tag{16}$$

In the above expression, the components (X_i's) of vector $\overline{X}_\rho^p$ can take any value from 0 to n^2 as long as the sum of the working nodes in all the ρ locations is p.

Φ_p denotes the summation of Υ for all possible $\overline{X}_\rho^p$'s *that satisfy the task requirement.* Excluding all the failed states from Θ_p, we get Φ_p. Thus,

$$\Phi_p = \Theta_p - \sum_{\substack{X_1, X_2, ..X_\rho = 0 \\ X_1 + X_2 + .. + X_\rho = p}}^{n^2 - 1} \Upsilon(\overline{X}_\rho^p). \tag{17}$$

The generalized structure of the MC is shown in Figure 4. There are $(N^2 - n^2 + 1)$ working states in the MC, each with a corresponding failure state. The total number of states is thus $2(N^2 - n^2 + 1)$. The number of states could be high for large systems when n is small. As p varies from N^2 down to n^2, the computation time of α increases. On the other hand, the value of α becomes extremely small after the failure of a few nodes. The state probabilities of the working states resulting after a few failures are almost negligible. We can therefore truncate the low probable states in order to reduce the computational time without sacrificing accuracy. For example, the error introduced by truncating all the states for which $\alpha \leq 0.1$ is of the order of 10^{-5} for a 256-node system. Depending upon the accuracy required, one may select a suitable value of α for state truncation.

For small systems, α_p's can be computed using HARP [14]. A fault-tree can be developed based on the task requirement. We can obtain all possible distributions by processing the fault-tree using HARP. The probabilities of each of these distributions can be computed for small systems. A similar method is described in [20], but this method becomes tedious and restrictive for large systems.

Other variations of repair assumptions can be also analyzed by modifying the Markov chain. One such assumption is the restoration of all faulty elements by an off-line repair process when the system goes to an F state. In such cases, there will be a transition from all the F states to the initial state with a rate $r\mu'$, where μ' denotes the global repair rate. The Markov chain can then be solved to obtain the state probabilities.

VI. RESULTS AND DISCUSSION

A. Simulation

We conduct extensive simulation to validate the analytical models and compare the bounds. An $(N \times M)$ mesh is simulated as a matrix of the same size whose elements are either 1 (working) or 0 (failed). Initially all the elements are set to 1 representing a fault-free mesh. The failure and repair instants are computed assuming an exponential distribution of node failure and repair time, respectively. Faults are injected by setting an element to 0. Repair is done on a FCFS basis by changing a 0 to 1. Availability of the required fault-free submesh is checked at specified intervals. The search for a fault-free submesh is done as per the underlying allocation policy. We search the complete $(N \times M)$ mesh exhaustively for perfect recognition. The TDBS and FS algorithms are implemented as described in [4, 5] for finding the required submesh. The 95% confidence interval was estimated to be within 3% of the mean.

B. Model Validation

The analytical results are compared with the simulation results to demonstrate the validity of the models. Figure 5 shows the submesh reliability with perfect recognition ability for a (100×80) mesh. The solid lines represent the lower bound obtained from the analytical model, and the dotted lines reflect the results obtained through simulation. 25% degradation refers to a mesh with 6000 working nodes. Thus, the configurations of (80×75), (75×60), and (50×40) reflect degradation of 25%, 44%, and 75%, respectively.

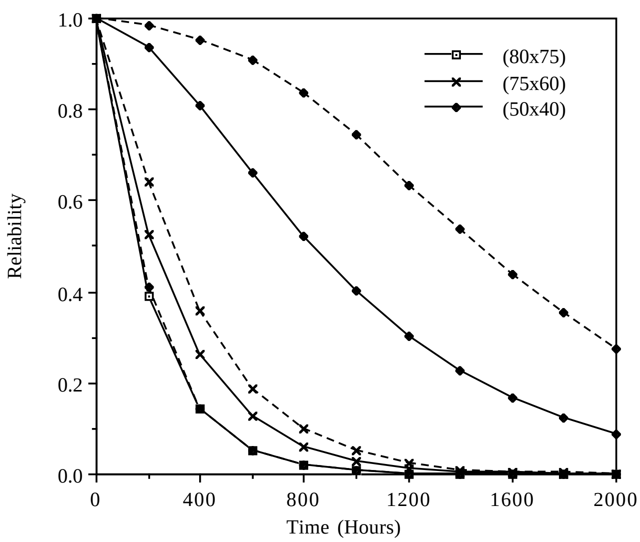

Fig. 5. Submesh Reliability with Perfect Recognition for a (100x80) Mesh.
C=0.99, λ=0.000001

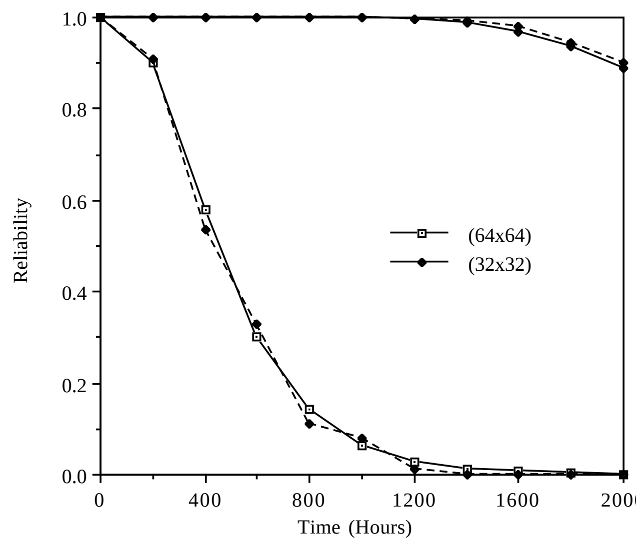

Fig. 6. Submesh Reliability with TDBS for a (128x128) Mesh
C=1.0, λ=0.000001

The proposed analytical model predicts almost exact reliability when the degradation is within 25%. Tight lower bounds are obtained for up to 50% degradation. However, deviation between analysis and simulation increases for higher degradation. This discrepancy is attributed to the approximation included in aligning the working nodes. The expanding row/column technique needs more than the required number of consecutive nodes to ensure proper alignment. For low degradation, the number of additional nodes is less and thus the model gives almost exact or a tight lower bound for the system reliability. For more than 50% degradation, the reliabilities of the additional nodes ($\frac{N}{2}m$ or $\frac{M}{2}n$) become a significant factor.

System reliability with the TDBS for a (128×128) system is shown in Figure 6. Figure 7 illustrates the reliability of a (100×80) mesh with the FS allocation policy. Solid lines refer to the analytical results while the dotted lines reflect the simulation results. System reliability is plotted for two different degradations with both the allocation policies. The analytical results match closely with the results obtained via simulation since no approximations are made in the analysis. A small difference in the results is due to the rounding off errors in the simulation while computing the failure instants of the nodes.

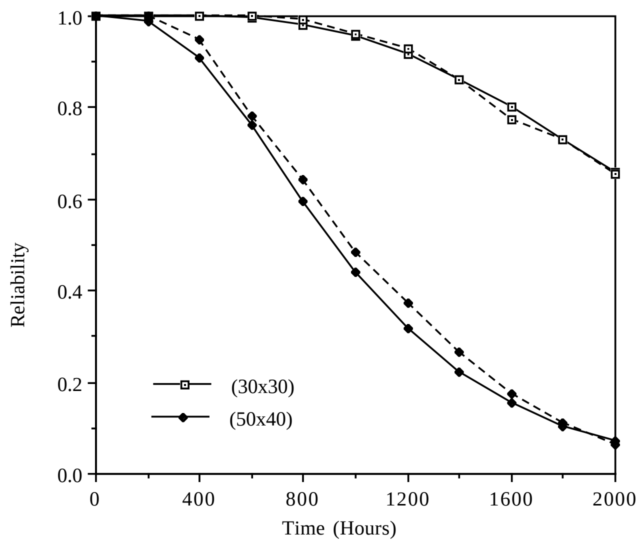

Fig. 7. Submesh Reliability with FS Scheme for a (100x80) Mesh
C=1.0, λ=0.000001

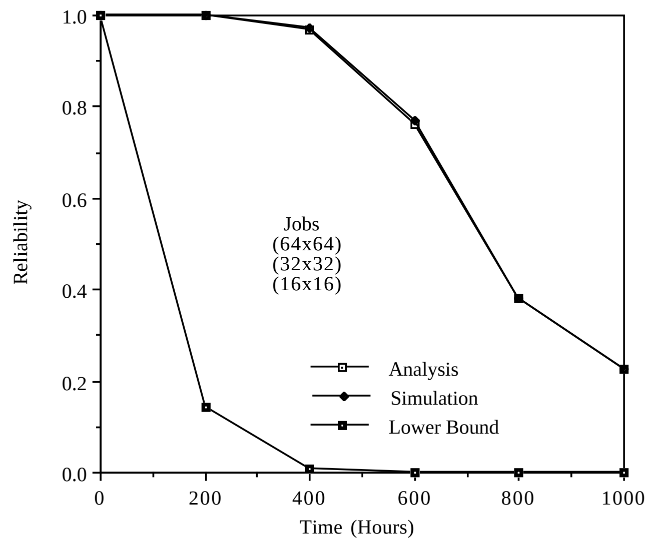

Fig. 8. Multitask Reliability of a (256x256) System
C=1.0, λ=0.000001

Multitask reliability of a (256×256) system with three jobs allocated employing the TDBS is depicted in Figure 8. The job sizes are (64×64), (32×32), and (16×16). A working submesh of size (128×128) is enough for accommodating the three jobs. An absolute lower bound can be obtained by computing the reliability of a (128×128) submesh. An estimation of the reliability, obtained using the methodology presented in section IV, is shown in the figure. The simulation result is also plotted to validate the analysis and compare the tightness of the bound.

Validation of the availability model follows next. Here, results are reported for a (16×16) mesh (System-A) and a (32×32) mesh (System-B). The task requirements are submeshes of size (16×16) and (32×32) in System-A and System-B, respectively. For System-A, the values of α_p's for the parameters shown in Table I are $\alpha_{256} = \alpha_{255} = \alpha_{254} =$

1.00, $\alpha_{253} = 0.904$, $\alpha_{252} = 0.760$, $\alpha_{251} = 0.610$, $\alpha_{250} = 0.476$, $\alpha_{249} = 0.365$, $\alpha_{248} = 0.276$, $\alpha_{247} = 0.207$, $\alpha_{246} = 0.155$, $\alpha_{245} = 0.115$, $\alpha_{244} = 0.085$, and so on. Similarly, the values of α_p's for the 1024-node system are $\alpha_{1024} = \alpha_{1023} = \alpha_{1022} = 1.00$, $\alpha_{1021} = 0.906$, $\alpha_{1020} = 0.764$, $\alpha_{1019} = 0.617$, $\alpha_{1018} = 0.485$, $\alpha_{1017} = 0.374$, $\alpha_{1016} = 0.285$, $\alpha_{1015} = 0.216$, $\alpha_{1014} = 0.163$, $\alpha_{1013} = 0.123$, $\alpha_{1012} = 0.092$, and so on. We have truncated the states when the value of α falls below 0.1. The states $245 \leq p \leq 256$ are considered for System-A, and for System-B, the states $1013 \leq p \leq 1024$ are analyzed. Note that the probabilities of state 245 (for System-A) and state 1013 (for System-B) are very low. A transition probability of less than 0.1 from these states would be almost negligible. This was the basis of our state truncation. However, other suitable truncation range can be used depending upon the desired accuracy.

Table I. Availability Prediction of Two Systems

System–A: (16×16) *mesh*, (8×8) *submesh*, λ=0.00001, μ=0.01, C=r=0.90

System–B: (32×32) *mesh*, (16×16) *submesh*, λ=0.00001, μ=0.01, C=r=0.95

Time	System A		System B	
Hours	Analysis	Simulation	Analysis	Simulation
0	1.000	1.000	1.000	1.000
200	0.963	0.964	0.928	0.930
400	0.952	0.953	0.891	0.897
600	0.948	0.949	0.866	0.867
800	0.946	0.947	0.849	0.840
1000	0.946	0.947	0.835	0.819
1200	0.946	0.947	0.825	0.811
1400	0.945	0.946	0.818	0.801
1600	0.945	0.946	0.811	0.792
1800	0.945	0.946	0.807	0.786
2000	0.945	0.946	0.801	0.785

Numerical results from analysis and simulation are shown for System-A and System-B in Table I. The observation window is limited to 2000 hours where availability of both the systems attain steady state. The difference between the analysis and simulation is within 3%.

C. Comparison and Discussion

System reliability comparisons with the perfect recognition, FS, and TDBS are plotted in Figures 9 and 10. A (64×64) size mesh is considered. Reliability of a job that requires a submesh with side lengths as a power of 2 is shown in Figure 9. Figure 10 depicts the system reliability with a (20×20) submesh requirement (not powers of 2). Comparison of the three cases indicates that the reliability increases with the submesh recognition ability. Reliability with the perfect recognition ability is therefore an absolute upper bound. FS and TDBS allocations are indistinguishable when the submesh requirement is a square with side lengths as powers of 2 (Figure 9). Reliability with the FS scheme is better than that of TDBS when the job size is not restricted to have side lengths as powers of 2 (Figure 10). In this case, the TDBS searches for a (32×32) mesh whereas the FS scheme searches for a (20×20) mesh. An efficient task allocation scheme therefore improves both performance and dependability.

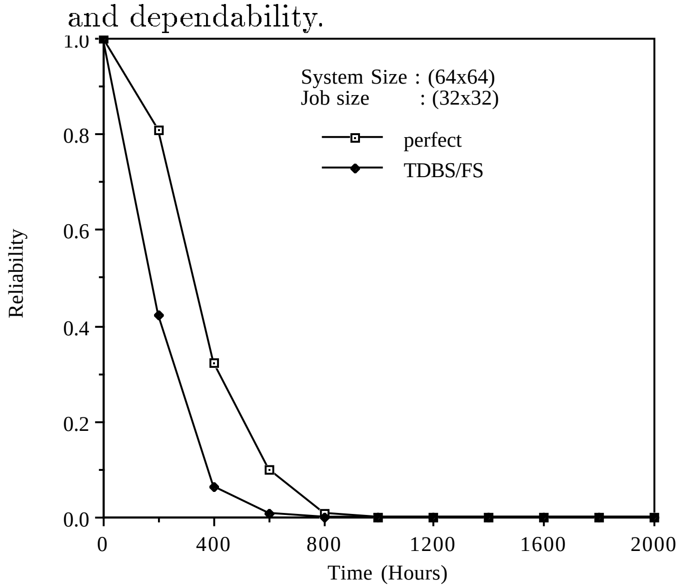

Fig. 9. Submesh Reliability Comparison with Cubic Job Size

C=1.0, λ=0.00001

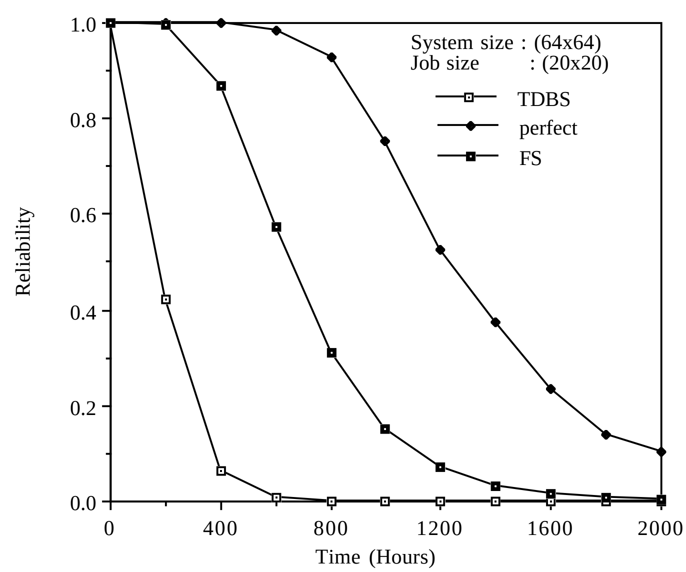

Fig. 10. Submesh Reliability Comparison with Non-Cubic Job Size

C=1.0, λ=0.00001

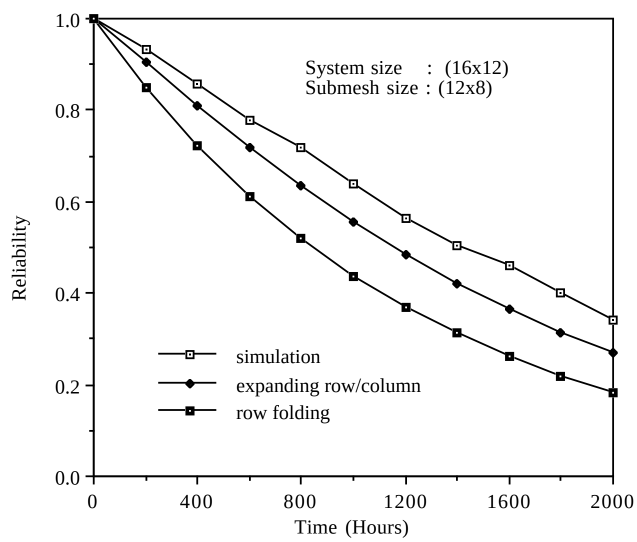

Fig. 11. Submesh Reliability with 50% Degradation
C=1.0, λ=0.00001

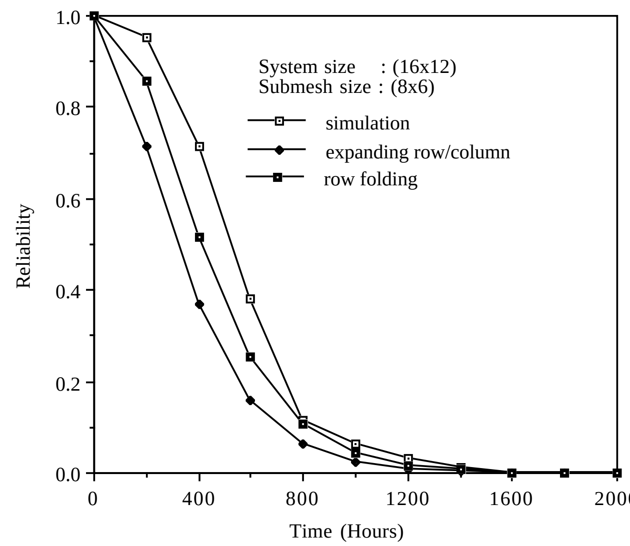

Fig. 12. Submesh Reliability with 75% Degradation
C=1.0, λ=0.00001

System reliability with perfect submesh recognition, using the expanding row/column technique, is compared with the row folding approach in Figures 11 and 12. System reliability with 50% degradation is plotted in Figure 11. It was mentioned earlier that the expanding row/column technique provides tight bounds up to 50% degradation. For low degradation (0-50%), it is observed that the results with the expanding row/column technique are much closer to the simulation results than the results obtained using the row-folding method. The row folding method provides better results than the expanding row/column technique for higher degradation as is evident from Figure 12 (75% degradation). One can use any one of the techniques depending upon the degree of degradation. It is however unlikely that a system would be operated with more that 50% degradation. Thus, the expanding row/column technique is useful for most practical cases.

Mesh topology can be converted to a torus by connecting the peripheral nodes via "wrap-around" links (like Illiac IV and MPP). We simulated a mesh and a torus of size identical to that of Touchstone Delta (33×16). System reliabilities with 50% degradation (a (22×12) submesh) and 26% degradation (a (28×14) submesh) are obtained and compiled in Table II. The results indicate that the reliability of a mesh can be considerably improved by converting it into a torus. This is because the wrap around connection provides more choices for finding a submesh. It is therefore advantageous to convert a mesh to a torus since it provides better reliability while preserving the architectural features of a mesh.

Table II. Reliability Comparison of Mesh and Torus

System size: (33×16), λ=0.00001, C=0.95

Time	(22 × 12) Submesh		(28 × 14) Submesh	
Hours	Mesh	Torus	Mesh	Torus
0	1.000	1.000	1.000	1.000
200	0.815	0.947	0.552	0.937
400	0.635	0.899	0.288	0.833
600	0.456	0.802	0.172	0.645
800	0.271	0.708	0.084	0.439
1000	0.159	0.530	0.047	0.271
1200	0.104	0.415	0.034	0.162
1400	0.074	0.317	0.017	0.112
1600	0.059	0.231	0.012	0.073
1800	0.034	0.171	0.000	0.031
2000	0.025	0.140	0.000	0.013

VII. CONCLUDING REMARKS

This paper introduces different techniques for finding single or multiple submesh reliability in mesh connected systems. Three different models are described to find the probability of a working submesh. The first one is based on perfect submesh recognition ability, and the other two utilize the underlying allocation schemes (TDBS and FS) for finding a submesh. Due to the complex nature of the problem, we provide an approximation method, called *row/column expansion*, to compute system reliability with perfect submesh recognition. This technique provides a good estimate of the system reliability for up to 50% degradation. An alternate approach, called *row folding*, is shown to be suitable for higher degradation. Exact reliability models with the TDBS and FS allocation schemes are derived. For availability computation, a Markov model solution technique is proposed. State transitions in the MC are determined by considering the configuration required for the task. Using this approach, the number of states in the MC is limited to $2(N^2 - n^2 + 1)$, where $(N \times N)$ is the original system size, and $(n \times n)$ is the required submesh size. The proposed approach also provides means of attacking complex problems for which detailed

MC solutions are not feasible. Simulation results are provided to validate the analytical techniques.

The results indicate that the system dependability is not only affected by the node failures but also dependent on the allocation policies. An allocation scheme with better submesh recognition ability demonstrate higher reliability. Thus this work may encourage researchers to investigate further development and analysis of allocation policies from a dependability standpoint.

A number of interesting but intricate research issues to be addressed in future include, exact modeling of mesh reliability with perfect submesh recognition, modeling of multitask reliability based on perfect recognition or FS allocation scheme, extension of the model for higher dimensions, modeling of reconfigurable meshes, and integrating performance with dependability to evaluate performability.

REFERENCES

[1] G. Zorpette, *et. al.*, "Reinventing the Machine," IEEE Spectrum, pp. 28-41, Sept. 1992.

[2] J. C. Laprie and A. Costes, "Dependability: A Unifying Concept for Reliable Computing," FTCS-12, pp. 18-21, June 1982.

[3] S. Rai and D. P. Agrawal, "Distributed Computing Network Reliability," *Tutorial Textbook*, IEEE Computer Soc. Press, 1990.

[4] K. Li and K. H. Cheng, "Job Scheduling in a Partitionable Mesh Using a Two Dimensional Buddy System Partitioning Scheme," IEEE Trans. on Parallel and Distributed Systems, pp. 413-422, Oct. 1991.

[5] P. J. Chuang and N. F. Tzeng, "An Efficient Submesh Allocation Strategy for Mesh Computer Systems," Int. Conf. on Distributed Computing Systems, pp.256-263, May 1991.

[6] M. J. Serrano and B. Parhami, "Optimal Architectures and Algorithms for Mesh-Connected Parallel Computers with Separable Row/Column Buses," IEEE Trans. on Parallel and Dist. Systems, pp. 1073-1080, Oct. 1993.

[7] C. W. H. Lam, H. F. Li, and R. Jayakumar, "A Study of Two Approaches for Reconfiguring Fault-Tolerant Systolic Arrays," IEEE Trans. on Computers, pp. 833-844, June 1989.

[8] M. Chean and J. A. B. Fortes, "The FUSS Approach to Hardware Reconfiguration for Fault-Tolerant Processor Arrays," IEEE Trans. on Computers, pp. 564-571, April 1990.

[9] J. A. B. Fortes and C. S. Raghavendra, "Gracefully Degradable Processor Arrays," IEEE Trans. on Computers, pp. 1033-1044, Nov. 1985.

[10] N. L. Benitez and J. A. B. Fortes, "Detailed Modeling and Reliability Analysis of Fault-Tolerant Processor Arrays," IEEE Trans. on Computers, pp. 1193-1200, Sept. 1992.

[11] C. R. Das and J. Kim, "A Unified Task-Based Dependability Model for Hypercube Computers," IEEE Trans. on Parallel and Distributed Systems, pp. 312-324, May 1992.

[12] J. Blake and K. S. Trivedi, "Reliability Analysis of Interconnection Networks Using Hierarchical Composition," IEEE Trans. on Reliability, pp. 111-120, April 1989.

[13] C. R. Das, P. Mohapatra, et. al., "An Availability Model for MIN-Based Multiprocessors," IEEE Transactions on Parallel and Distributed Systems, pp. 1118-1129, Oct. 1993.

[14] J. B. Dugan, K. S. Trivedi, et al, "The Hybrid Automated Reliability Predictor," AIAA Journal of Guidance, Control and Dynamics, Vol. 9, No. 3, pp. 319-331, May-June 1986.

[15] R. Sahner and K. S. Trivedi, "Reliability Modeling using SHARPE," IEEE Trans. on Reliability, pp. 186-193, June, 1987.

[16] Y. Zhu, "Efficient Processor Allocation Strategies for Mesh-Connected Parallel Computers," Journal of Parallel and Distributed Computing, 16, pp. 328-337, 1992.

[17] A. A. Salvia and W. C. Lasher, "2-Dimensional Consecutive-k-out-n:F Models," IEEE Trans. on Reliability, pp. 382-385, Aug. 1990.

[18] T. K. Boehme, A. Kossow, and W. Preuss, "A Generalization of Consecutive k-out-of-n: F Systems," IEEE Trans. on Reliability, pp. 451-457, Sept. 1992.

[19] C. Derman, G. J. Lieberman, S. M. Ross, "On the Consecutive k-of-n:F System," IEEE Trans. on Reliability, pp. 57-63, April 1982.

[20] W. E. Smith and K. S. Trivedi, "Dependability Evaluation of a Class of Multi-Loop Topologies for Local Area Networks," IBM Journal of Research and Development, Vol. 33, No. 5, pp. 511-523, Sept. 1989.