# Subsystem Reliability Measurement of (n,k)-Star Graph

Julie Kim
*Department of Computer Engineering*
*California State University Long Beach*
*Long Beach, United State*

***Abstract*—This paper is a *repeat* of measuring the upper and lower bound of reliability of (n,k)-star-graph based on probabilistic fault model and approximation approach. The probabilistic fault model, lower/upper bound, uses combinatorial probability to calculate the reliability of subgraph of the (n,k)-star—graph by implementing the Principle of inclusion (PIE), while the approximation approach calculates the reliability by using disjoint (n-1,k-1)-star-graph, subgraph partition along a fixed dimension. The study shows that these two methods provides very close result. The probabilistic model considers the intersection of only four subgraphs. After our experiment on the two models, we will introduce MTTF as another way to measure the reliability of the (n,k)-star-graph. We will compare and confirm that our system's reliability by MTTF also agree with the probabilistic model. Mean Time to Failure (MTTF) is the time that the subsystems stay in a certain state, which infers the functionality of the entire system.**

***Keywords—(n,k)-Star graph, reliability, reliability of subgraphs, probabilistic fault model, approximate approach.***

## NOTATION

| | |
|---|---|
| $S_{n,k}$ | (n,k)-star graph |
| $S_0$ | State in which a system is fault free, no faulty subsystems |
| $S_i$ | State that the system has i faulty subsystems |
| $\lambda_n$ | Independent fault probability of a node |
| $T_i$ | Meant Time to Failure (MTTF) the system in state $S_i$ |
| p | Fault free probability of each node in the system |
| $S_{n,k}^{n-m,k-m}$ | (n-m, k-m)-substar of $S_{n,k}$ |
| $S_{n,k}^{n-1,k-1}(j,v)$ | Subgraph along one fixed dimension j |
| $R_{n,k}^{n-1,k-1}(p)$ | The probability there is at least one fault-free subgraph along $j^{th}$ dimension |

## I. INTRODUCTION

The Reliability of a system is high when the system can tolerate fault and maintain its functionality as long before it goes into fail state. Reliability is high when such time is long. As multiprocessor system becomes more and more complicated, the chance that such complex system fails increase. The time for the system to stay fault free becomes shorter. The reliability in this paper is used to measure how healthy the entire system is. The health of each subsystem preserves the resilience of the entire system in the present of fault. The entire system fails when all subsystem fails. To keep the system running, the system needs to have at least one fault free subsystem running. The solution is to find a network that has the high fault tolerant and model the system based on the specific network topology. Different kind of graph has been used as network interconnection model to guarantee high reliability to any complex system. Among all the network topology or graph, Star graph has been one of the most attractive networks. (n,k)-Star graph is a generalization of star graph has been proved to be more fault tolerant than some network topology such as Hypercube. (n,k)-Star graph has better proper than other Hypercube such as low degree of nodes, small diameter, symmetrical node and link, and high degree of fault tolerance. It can be recursively decomposed into many subgraphs. Star graph has been used in some fields such as parallel computing and the connection in multiprocessor system. To measure the Reliability, the previous study has proved using probabilistic model to calculate the upper bound of reliability equal the result of Approximation approach [4]. This method includes the intersection of three sub-stars of $S_n$-star graph, which is a special case of (n,k)-Star. Lin and Wang [1] established the lower and upper bound on the (n,k)-star with the intersection of one more subgraph, four. The study has also shown that the results of the three approach of lower/upper bound and Approximation approach merges as the probability of fault free nodes goes low. Latifi and Srimani [3] use MTTF to calculate the reliability of the system based on the number of fault free subsystems having in the entire system using $S_n$ star graph as the network. The study uses Markov Chain to calculate the probability that subsystem or sub-star stays fault free. Recent study has expand on the idea by measuring the system reliability based on subsystem of (n,k)-Star. The last two study discusses extensively about mean time to failure of each sub-star beyond the need of this paper to use MTTF to confirm the result of Lin and Wang [1]. However, the result and data in this paper is used to compare the simulation result. The number of MTTF indicates the time when the system has at least one fault free subsystem, after which the entire system will fail. It is used as the beginning mark of the graph. The graph is the calculation of reliability of the system, where the system has at least on fault free subsystem, or the graph network has at least one fault free sub-star.

The rest of the paper is organized into four sections. Section II discusses about the basic property of (n,k)-Star, comparing it to $S_n$, and how it is used in probabilistic fault model calculation. Section III consists of five parts. The first four parts discuss on topic lower bound, upper bound, approximate approach and mean time to failure. The last part is the discussion on the result obtain from the Java program simulation. Section IV is the conclusion of the paper.

## II. (N,K)-STAR GRAPH NETWORK TOPOLOGY AND BASIC PORPERTY

Graph is used as a model to connection system especially in multiprocessor system. The vertex or node is the same as each processor, and the edges is the same as links. As system needs more and more processors, it is important to connect them all these processors in a way that in case of fault occurred, the entire system can function as long it could before it fails. (n,k)-star has a property that satisfies the resilience purpose of the multiprocessor network. Reference [5] introduces (n,k)-star network as more portable than the special $S_n$ based on the flexible size that the former is than the later one. The (n,k)-star is the generalized version f the n-star graph. The two parameters *n* and *k* can be used to make good choice for the number of nodes in network and for degree/diameter tradeoff. This allow the flexibility in designing network. The n-star has the number of node n! while the (n,k)-star has ( n!)/(n-k)! nodes. For example, $S_4$ has 4! is 24 nodes. $S_{4,2}$ has only 12 nodes [see Figure 1]. With the same degree of network, degree 3 in this case, $S_{4,2}$ provides a smaller number of nodes than $S_4$. Since n-star is a special case, $S_{4,3}$ also has 24 nodes [see Fig. 2]. Reference [1] discusses (n,k)-star can be recursively decomposed in k-1. $S_{4,2}$ can be decomposed in one way since k=2 and k-1=1.

*Definition:* ([1]). The (n,k)-star graph has the vertex set denoted by $V(S_{n,k}) = \{p_1p_2...p_k \mid p_i \in <n>, \text{ and } p_i \neq p_j \text{ for } i \neq j\}$.

1) A vertex $p_1p_2...p_i...p_k$ is adjacent to the vertex $p_ip_2...p_1...p_k$ through and edge of dimension i, where 2<= I <= k.
2) A vertex $p_1p_2...p_i...p_k$ is adjacent to the vertex ǣ$p_1p_2...p_i...p_k$ through an edge of dimension 1, where ǣ $\in <n> \setminus \{ p_1, p_2, ..., p_k \}$

For example, $S_{4,2}$ contain four sub-stars who's each vertex set are {31, 21, 41}, {13, 34, 23}, {14, 34, 24} and {12, 32, 42}. The last digit is the fixed dimension that each sub-star connected to each other. The first set with vertex notation of (31) is connected to the second set with vertex notation of (13), which is the permutation of the first symbol of the string to the fixed dimension along the connection of two sub-star. Another example set four with vertex of (34) is connected set two with vertex of (43), permutation of 4, first string digit, with 3, the fixed second dimension sub-star is partition into. Since k=2, it can be partition into only one dimension. (n,k)-star graph can be decomposed into n copies of (n-1, k-1)-subgraphs along $j^{th}$ position. $S_{4,3}$ can be partition int 4 subgraph along the third dimension as shown in Figure 2. It can also be partition into four subgraph along the second dimension, where the set of vertex for each subgraphs are {214, 314, 413, 213, 312, 412} its second fixed dimension is 1, {321, 421, 124, 324, 423, 123} its second fixed dimension is 2, {231, 431, 134, 234, 432, 132} its second fixed dimension if 3, {241, 341, 143, 243, 342, 142,} its second fixed dimension is 4. Two subgraphs are connected via two nodes that has their permutation of the first string digit and the fixed dimension along which subgraph are connected. Vertex of first set with string symbol of (214) is connected to second set with vertex (124). $S_{4,3}$ has 8 distinct subgraphs partitioned along dimension 2 and 3. In general the number of distinct subgraphs is $\binom{k-1}{m}\binom{n}{m}m!$, where n is 4, k is 3 and m is the number fixed dimension and 1<= m <= k. In this case m is 1 for $S_{4,3}$. For large star graph where smaller subgraphs are need, increase m to reduce the number of nodes in each subgraph. For example, X3X1X, represents a 3-star formed by the set of nodes {23415, 23514, 43512, 43215, 53214, 53412} [4]. The reliability $R_{n,k}^{n-1,k-1}(p)$ is the probability that there exists at least one fault free subgraph of (n-1, k-1)-star graph. We use this reliability to indicate the fault tolerant of the system in the present of faulty nodes. The higher the reliability tell the higher probability that the system has at least one non-faulty

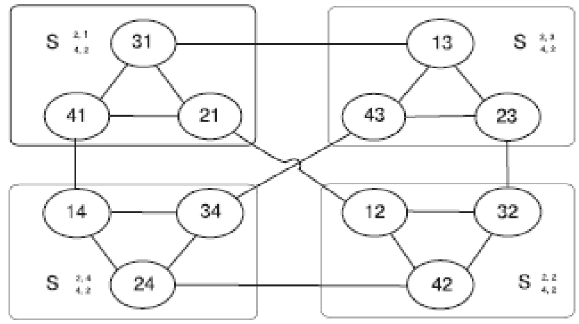


Fig. 1. $S_{4,2}$ (partioned along 2nd dimension)

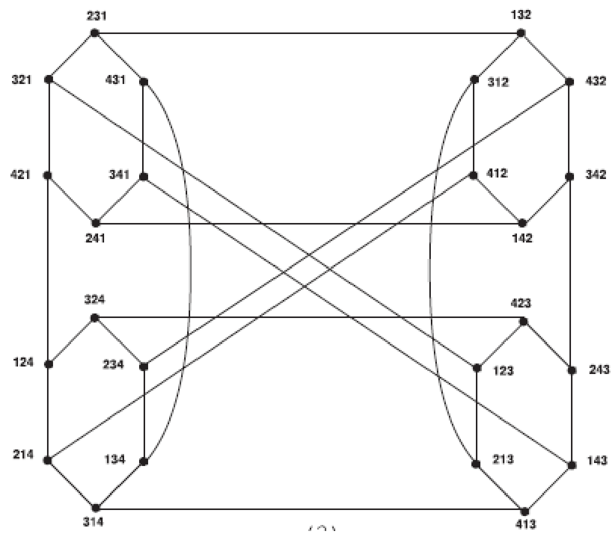


Fig. 2. $S_{4,3}$ (partioned along 3th dimension)

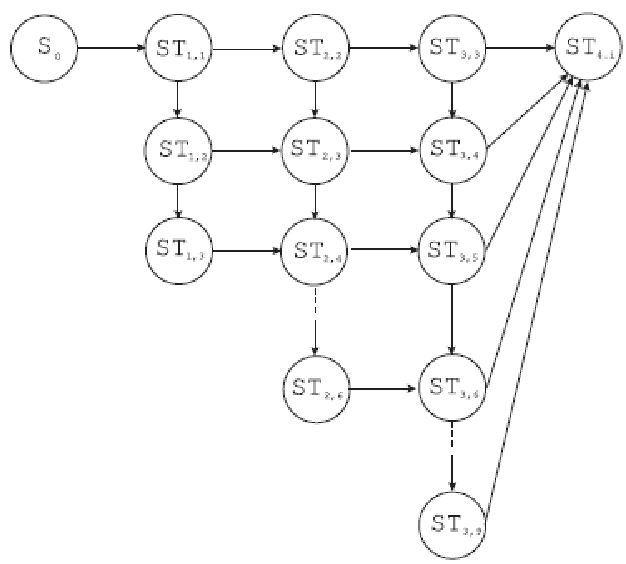

Fig. 3. State transition diagram of $S_{4,2}$

subsystem and the lower the reliability, the lower the probability that there exist any non-faulty subsystem, which mean the system will soon fail. As shown above for $S_{4,3}$, all 8 subgraphs are not disjoint, they are distinct. Some nodes are intersected, sub graph is not mutually exclusive. To make sure the fault free probability (p) of each node is included in calculation only one time, the *principle of Inclusion and Exclusion* (PIE) is applied to the calculation.

## III. Reliability Using Probabilistic Model

In this section we will use the probabilistic model to calculate the Lower Bound Reliability, Upper Bound Reliability, and Approximation approach [1]. Since all the subgraph is not all mutually distinct, the equation derived to calculate the above three implement the PIE to count all the possible intersection of subgraph. As the introduction, this paper will include intersection of four subgraph. The first three section of A, B and C explains the calculation and result of lower/upper bound and approximated reliability, and section C explains the how MTTF confirm the previous part's result. Section E is simulation and F is Discussion.

In the probabilistic fault model, the probability that a subgraph of (n,k)-star graph is operational is expressed in terms of the reliability of individual nodes in the graph. The reliability for (n-m, k-m)-star graph of $S_{n,k}$ is formulated as the union of the probabilistic events that all possible subgraphs are operational. These events are not all mutually disjoint; some subgraph share nodes [1].

### A. *Lower Bound Reliability of (n,k)-Star Graph*

For lower bound, we account for the intersection of 4 subgraphs. For example, $S_{6,5}$ has 720 nodes and string symbol for each vertex of each subgraph contain 5 digits. In the paper, we intersect 4-subgraph that are partition along only one fixed dimension, so the subgraph is (n-1, k-1)-start graph not (n-m, k-m)-star graph where number of fixed dimension *m* could be more than one obtaining subgraph with lesser number of nodes.

*Example:* There are 24 distinct (5,4)-subgraphs of the $S_{6,5}$ by following the formula of $\binom{k-1}{m}\binom{n}{m}m!$ where *m* is 1. As the followings:

$$\begin{array}{llll} X1XXX & XX1XX & XXX1X & XXXX1 \\ X2XXX & XX2XX & XXX2X & XXXX2 \\ X3XXX & XX3XX & XXX3X & XXXX3 \\ X4XXX & XX4XX & XXX4X & XXXX4 \\ X5XXX & XX5XX & XXX5X & XXXX5 \\ X6XXX & XX6XX & XXX6X & XXXX6. \end{array}$$

Each string symbol represent one type of subgraphs partition along dimension 2, 3, 4, and 5. The first column, the six subgraph is partitioned along fixed dimension 2 with each subgraph has different value of 1, 2,.., 6. Column 2 represent another six subgraphs with similar property but partitioned along dimension 3. All types of the intersection among each 4-subgraph is shown in Figure 4. It is initially proved that the arranged intersection of each 4-subgraph based on the permutation of value and fixed dimension of each subgraph counted to 16 types of intersections of 4-subgraph. However, some 4-subgraph proves to be the same intersection type even with different string symbol. For example, the two different set of {X1XXX X2XXX, X3XXX, X4XXX} and {X1XXX, XX1XX, XXX1X, XXXX1} are the same intersection of (4,1)-type [see Figure 4]. There are only 9 intersection type of 4-subgraph as shown in Figure 4. Each string symbol of each circle represent the one subgraph and its partitioned dimension. For example (4,1)-type, all subgraphs are partitioned along dimension 2.

The number of all four subgraph groups is the combination of 4 to choose from 24 which is 10626. The number of groups for each type of intersection is listed in Table II. The equation for Lower Bound Reliability includes all the groups of the 9 intersections types of the 4-subgraph

The general probabilistic fault model-based PIE is:

$$\begin{aligned} R_{n,k}^{n-1,k-1}(p) = & \sum_{i=1}^{n(k-1)} r_i(p) + (-1)^1 \sum_{i<j} r_{(i,j)}(p) \\ & + (-1)^2 \sum_{i<j<l} r_{(i,j,l)}(p) \\ & + (-1)^3 \sum_{i<j<l<q} r_{(i,j,l,q)}(p) + \cdots \\ & + (-1)^{n(k-1)-1} r_{(1,2,\ldots,(k-1)n)}(p). \end{aligned} \quad (1)$$

Given the homogenous node reliability p in an (n,k)-star graph, the lower-bound on Reliability, a function of p, i.e., the probability that there exists a fault-free (n-1, k-1)-subgraph in $S_{n,k}$, is given in equation (2) [1]. Based on equation (1), the Reliability $R_{n,k}^{n-1,k-1}(p)$ is approximate to only four terms as shown in equation (2). In the equation (2), on page five, the first term is the fault free probability of all disjoint or non-intersect subgraph, the second term and third term are the fault free probability of the intersection of pair subgraphs and 3-subgraph respectively. The fourth term is the sum of all the fault free probability of each type of intersection. In column three of Table II, the number of nodes of each type is used to find the fault free probability for one group and multiply to number of groups to find the fault free probability of one type. Equation (2) contains all information from Table II. In the simulation program, we input time unit as hour, each node constant failure rate to find the probability that the system has at least one fault free subgraph.

### B. *Upper Bound Reliability of (n,k)-Star Graph*

Upper-bound reliability is like lower-bound. We consider the intersection of 3-subgraph and apply the PIE principle. Given the homogeneous node reliability p in an (n,k)-star graph $S_{n,k}$, and upper-bound on Reliability is the probability that there exists a fault-free subgraph is equation (3) [1]. The upper-bound is exactly part of lower-part only that upper-bound take into only the first three term, individual subgraph, pair intersection and 3-subgraph fault free probability respectively. With the same simulation method, we input time and constant failure rate.

$$
\begin{aligned}
R_{n,k}^{n-1,k-1}(p) \le\ & n(k-1)p^{\frac{(n-1)!}{(n-k)!}} \\
& - \left\{ \left( k\binom{n}{2} + n\binom{k-1}{2} \right) p^{\frac{2(n-1)!}{(n-k)!}} \right. \\
& \left. + 2\binom{n}{2}\binom{k-1}{2} p^{\sum_{i=1}^{2}(-1)^{i-1}\binom{2}{i}\frac{(n-i)!}{(n-k)!}} \right\} \\
& + \left\{ \left( (k-1)\binom{n}{3} + n\binom{k-1}{3} \right) p^{\frac{3(n-1)!}{(n-k)!}} \right. \\
& + 6\binom{n}{3}\binom{k-1}{3} p^{\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}\frac{(n-i)!}{(n-k)!}} \\
& + 4\binom{n}{2}\binom{k-1}{2} p^{\frac{3(n-1)!-(n-2)!}{(n-k)!}} \\
& \left. + (2n+2k-10)\binom{n}{2}\binom{k-1}{2} p^{t} \right\},
\end{aligned}
$$

where $t = (3(n-1)! - 2(n-2)!)/(n-k)!$. (3)

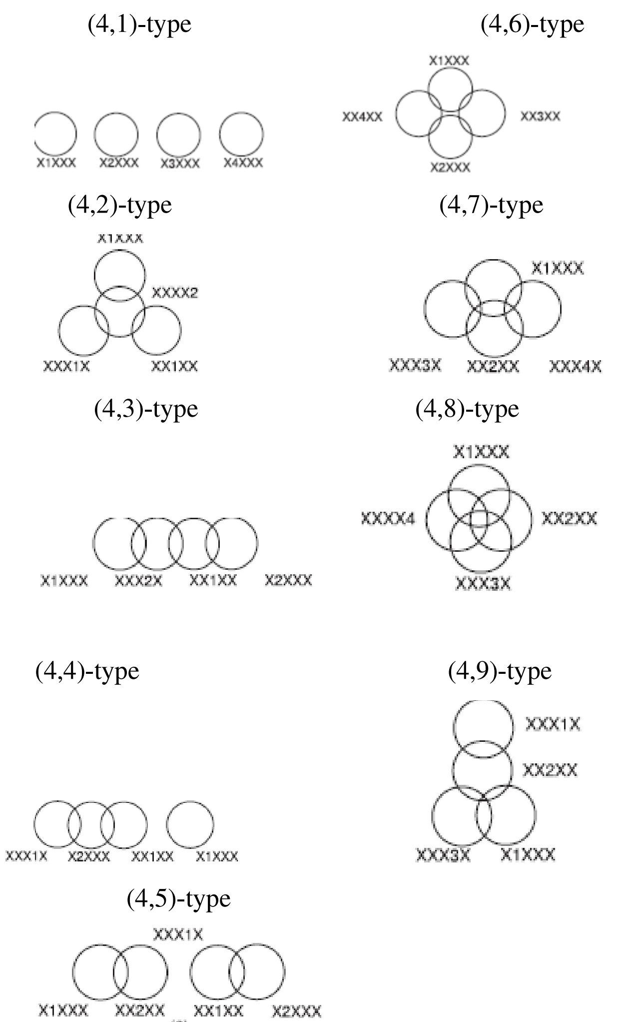


Fig. 4. 4-subgraph groups of $S_{6,5}$

TABLE II
INTERSECTION WAY OF FOUR SUBGRAPHS

| Intersection way | Number of groups | Number of nodes |
|---|---|---|
| (4.1)-type | $\binom{k-1}{4}(\binom{4}{1}\binom{n}{4} + \binom{n}{1})$ | $4\frac{(n-1)!}{(n-k)!}$ |
| (4.2)-type | $2\binom{k-1}{4}\binom{4}{2}\binom{n}{3}\binom{n-1}{3}$ $+\binom{k-1}{4}[\binom{n}{2}(\binom{4}{1} + \binom{4}{3})]$ | $\frac{4(n-1)!-3(n-2)!}{(n-k)!}$ |
| (4.3)-type | $\binom{k-1}{4}\binom{4}{2}\binom{n}{3}\binom{3}{2} \times 2$ $+3\binom{k-1}{4}\binom{4}{3}\binom{n}{2} \times 2$ | $\frac{4(n-1)!-3(n-2)!}{(n-k)!}$ |
| (4.4)-type | $2\binom{k-1}{4}\binom{4}{2}\binom{n}{3}\binom{3}{1}$ $+3\binom{k-1}{4}\binom{4}{3}\binom{n}{2} \times 2$ | $\frac{4(n-1)!-2(n-2)!}{(n-k)!}$ |
| (4.5)-type | $\binom{k-1}{4}\binom{4}{2}\binom{n}{2}$ | $\frac{4(n-1)!-2(n-2)!}{(n-k)!}$ |
| (4.6)-type | $\binom{k-1}{4}\binom{4}{2}\binom{n}{4}\binom{4}{2}$ $+3\binom{k-1}{4}\binom{4}{3}\binom{n}{3}\binom{3}{2}$ $+\binom{k-1}{4}\binom{n}{2}\binom{4}{2}$ | $\frac{4(n-1)!-4(n-2)!}{(n-k)!}$ |
| (4.7)-type | $3\binom{k-1}{4}\binom{4}{3}\binom{n}{4}\binom{4}{2} \times 2$ $+2\binom{k-1}{4}\binom{n}{3}\binom{4}{1}\binom{3}{1}$ $+\binom{k-1}{4}\binom{n}{3}\binom{4}{2} \times 2$ | $\frac{4(n-1)!-5(n-2)!+2(n-3)!}{(n-k)!}$ |
| (4.8)-type | $\binom{k-1}{4}\binom{n}{1}\binom{n-1}{1}\binom{n-2}{1}$ $\binom{n-3}{1}$ | $\sum_{i=1}^{4}(-1)^{i-1}\binom{4}{i}\frac{(n-i)!}{(n-k)!}$ |
| (4.9)-type | $3\binom{k-1}{4}\binom{4}{3}\binom{n}{3}\binom{3}{2} \times 4$ | $4\frac{(n-1)!}{(n-k)!} - 4\frac{(n-2)!}{(n-k)!} + \frac{(n-3)!}{(n-k)!}$ |

## C. *Approximated Reliability of (n,k)-Star Graph*

The Approximation on reliability is computed by ignoring all the intersections all intersection among subgraphs. It starts by computing the fault free probability among the n nonintersection subgraphs. The general (n,k)-star graph has n (n-1, k-1)-star as subgraph partition along one fixed dimensiion as explain above with the example of $S_{4,3}$. The list of all the general distinct subgraph partition along one fixed dimension are in the follwing way:

$\underbrace{X1XX\cdots XX}_{k}$, $\underbrace{X2XX\cdots XX}_{k}$, $\underbrace{X3XX\cdots XX}_{k}$, $\ldots$, $\underbrace{XnXX\cdots XX}_{k}$;
$\underbrace{XX1X\cdots XX}_{k}$, $\underbrace{XX2X\cdots XX}_{k}$, $\underbrace{XX3X\cdots XX}_{k}$, $\ldots$, $\underbrace{XXnX\cdots XX}_{k}$;
...... ........... ................. ..................... ..................
$\underbrace{XXXX\cdots X1}_{k}$, $\underbrace{XXXX\cdots X2}_{k}$, $\underbrace{XXXX\cdots X3}_{k}$, $\ldots$, $\underbrace{XXXX\cdots Xn}_{k}$.

Next calculate the over all fault free probability, including all the disjoint n subgraph along one fixed dimension. Since each n subgraph along one dimension intersect with another n subgraph along another dimension, the PIE will exclude the intersected nodes. In general the Approximate approach calculation is:

$$R_{n,k}^{n-1,k-1}(p) = 1 - \left(1 - p^{\frac{(n-1)!}{(n-k)!}}\right)^{n(k-1)} \tag{4}$$

## D. *Mean Time to Failure (MTTF) Substar in State $S_i$*

Mean Time to Failure is the reliability measurement of the subsystems' resilience under the present of faulty nodes. It measures the time that the number of faulty subsystems will occur. MTTF start with a system with no faulty subgraph, defined as the system is in state $S_0$, with the MTTF of $T_0$. As time goes on, the system enters into a state of having one faulty sub system or subgraph with MTTF in time $T_1$. The system is in state $S_1$, indicating the system contain one faulty subsystem in time $T_1$. The next faulty node occur can be in the same previous faulty subgraph or it will be in different subgraph, which lead to another faulty subgraph occur. The MTTF is the time that system stays in the state of having one faulty subgraph, the value of MTTF is indicated as $T_1$ and the state that the system in is $S_1$. As time passed, when the system ends up having two faulty subgraphs, the system is in state $S_2$. The MTTF value is $T_2$. The higher the probability of having at least one subsystem function under faulty nodes, the higher the reliability the system is. In other words, the longer the time the system can stay in the state of having at least one subsystem operational, the higher the reliability the system is, which indicate the bigger the MTTF, the higher reliability [2]. The state transition diagram is shown in Fig. 3. The diagram moves from left to right indicating increase of faulty subsystem and move vertically from to bottom indicating increase faulty nodes in the same faulty subsystem.

$$\begin{aligned} T_0 &= \int_0^{\infty} e^{-r_0 t} dt = \frac{1}{r_0} = \frac{1}{\frac{n!}{(n-k)!}\lambda_n} \\ T_i &= T_{i-1} + \frac{1}{r_i} \ (1 \le i \le n) \\ r_i &= \lambda_n (n-i) \frac{(n-1)!}{(n-k)!}. \end{aligned} \tag{5}$$

The system state moves from one faulty subsystem, to two, to three and so on. The system fails when there is no non-faulty subsystem. The general formula to computer the MTTF is expression (5) [2]. The MTTF of system having at least one fault free subsystem is the sum of time in which the system having previously less than number of faulty subsystems. The data in Table I shows time in hours when the system has no faulty subsystem for the amount of time $T_0$. The system has one faulty subsystem at time $T_1$. Finally, the system has at least one non-faulty subsystem at time $T_6$. $S_{7,6}$ with 5040 has at least one non-faulty subsystem at time 360 hours. A while after 360 hours, $S_{7,6}$ will no longer operational as it no longer have any fault free subsystem. It enters fail state. MTTF compute the time before the system enter fail states, telling the reliability of the overall system.

$$\begin{aligned} R_{n,k}^{n-1,k-1}(p) \ge \ & n(k-1)p^{\frac{(n-1)!}{(n-k)!}} \\ & - \left\{ \left( k\binom{n}{2} + n\binom{k-1}{2} \right) p^{\frac{2(n-1)!}{(n-k)!}} + 2\binom{n}{2}\binom{k-1}{2} p^{\sum_{i=1}^{2}(-1)^{i-1}\binom{2}{i}\frac{(n-i)!}{(n-k)!}} \right\} \\ & + \left\{ \left( (k-1)\binom{n}{3} + n\binom{k-1}{3} \right) p^{\frac{3(n-1)!}{(n-k)!}} + 6\binom{n}{3}\binom{k-1}{3} p^{\sum_{i=1}^{3}(-1)^{i-1}\binom{3}{i}\frac{(n-i)!}{(n-k)!}} \right. \\ & \quad \left. + 4\binom{n}{2}\binom{k-1}{2} p^{\frac{3(n-1)!-(n-2)!}{(n-k)!}} + (2n+2k-10)\binom{n}{2}\binom{k-1}{2} p^{\frac{3(n-1)!-2(n-2)!}{(n-k)!}} \right\} \\ & - \left\{ \left( \binom{4}{1}\binom{n}{4} + \binom{n}{1} \right) p^{4\frac{(n-1)!}{(n-k)!}} + \left( 2\binom{4}{2}\binom{n}{4}\binom{4}{3} + \binom{n}{2}\left( \binom{4}{1} + \binom{4}{3} \right) \right. \right. \\ & \quad \left. + \binom{4}{2}\binom{n}{3}\binom{3}{2} \times 2 + 3\binom{4}{3}\binom{n}{2} \times 2 \right) p^{\frac{4(n-1)!-3(n-2)!}{(n-k)!}} \\ & \quad + \left( 2\binom{4}{2}\binom{n}{3}\binom{3}{1} + 3\binom{4}{3}\binom{n}{2} \times 2 + \binom{4}{2}\binom{n}{2} \right) p^{\frac{4(n-1)!-2(n-2)!}{(n-k)!}} \\ & \quad + \left( \binom{4}{2}\binom{n}{4}\binom{4}{2} + 3\binom{4}{3}\binom{n}{3}\binom{3}{2} + \binom{n}{2}\binom{4}{2} \right) p^{\frac{4(n-1)!-4(n-2)!}{(n-k)!}} \\ & \quad + \left( 3\binom{4}{3}\binom{n}{4}\binom{4}{2} \times 2 + 2\binom{n}{3}\binom{4}{1}\binom{3}{1} + \binom{n}{3}\binom{4}{2} \times 2 \right) p^{\frac{4(n-1)!-5(n-2)!+2(n-3)!}{(n-k)!}} \\ & \quad + \left( 3\binom{4}{3}\binom{n}{3}\binom{3}{2} \times 4 \right) p^{4\frac{(n-1)!}{(n-k)!}-4\frac{(n-2)!}{(n-k)!}+\frac{(n-3)!}{(n-k)!}} \\ & \quad \left. + \binom{n}{1}\binom{n-1}{1}\binom{n-2}{1}\binom{n-3}{1} p^{\sum_{i=1}^{4}(-1)^{i-1}\binom{4}{i}\frac{(n-i)!}{(n-k)!}} \right\} \binom{k-1}{4} \end{aligned} \tag{2}$$

TABLE I

| $S_{7,k}$ | Number of nodes | $T_0$ | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
|---|---|---|---|---|---|---|---|---|
| $k=3$ | 210 | 480 | 1030 | 1700 | 2530 | 3640 | 5130 | 8640 |
| $k=4$ | 840 | 120 | 260 | 420 | 630 | 910 | 1330 | 2160 |
| $k=5$ | 2520 | 40 | 90 | 140 | 210 | 300 | 440 | 720 |
| $k=6$ | 5040 | 20 | 40 | 70 | 110 | 150 | 220 | 360 |

### E. Simulation

To compare the probabilistic fault model to approximation approach, we construct a high-level programming software to compute the reliability as a function fault free probability (p). Every module or processor has a failure rate, and it accumulate as time passed, increase faulty rate and suppress the reliability of the processor. Expression (6) is the fault free probability used in this simulation.

$$p = 1 - \frac{f}{\frac{n!}{(n-k)!}} = e^{-at}. \tag{6}$$

Below is our code example:

```java
package com.research546;
import java.lang.*;
import java.lang.Math;
public class Main {
    private static int combinationSmall(int nn, int cc) {...}
    private static int factorial(int fac) {...}
    private static int signInSum(int i) {...}
    private static int summation(int iSum, int n, int k) {...}
    private static int factorialDivideFact(int n, int i, int k) {...}
    private static int nodesStarGraph(int n, int k) {...}
    private static int type1GroupsNumber (int n, int k) {...}
    private static int type2GroupsNumber (int n, int k) {...}
    private static int type3GroupsNumber (int n, int k) {...}
    private static int type4GroupsNumber (int n, int k) {...}
    private static int type5GroupsNumber (int n, int k) {...}
    private static int type6GroupsNumber (int n, int k) {...}
    private static int type7GroupsNumber (int n, int k) {...}
    private static int type8GroupsNumber (int n, int k) {...}
    private static int type9GroupsNumber (int n, int k) {...}
    private static double lowerBoundReliability
            (int n, int k, int time, double constnFailRate){...}
    private static double upperBoundReliability
            (int n, int k, int time, double constnFailRate){...}
    private static double approximateReliability
            (int n, int k, int time, double constnFailRate){...}
    private static int MTTF (int n, int k, int i, double constnFailRate) {
        int mttf_0=0; int mttf=0; int t=0;
        while (t<(i+1)) {
            if (t==0)
                mttf_0 = (int)(Math.ceil
                        (1 / (nodesStarGraph(n, k) * constnFailRate)));
            else if (t==1) {
                mttf = mttf_0 +
                        (int)(Math.ceil(1 / (constnFailRate * (n - t) *
                        factorialDivideFact(n, i: 1, k))));
            }
            else {
                mttf = mttf +
                        (int)(Math.ceil (1 / (constnFailRate * (n - t) *
                        factorialDivideFact(n, i: 1, k))));
            }
            t++;
        }
        if (i==0) return mttf_0;
        else return mttf;
    }
```

```java
    private static double upperBoundReliability
            (int n, int k, int time, double constnFailRate){
        double reliability=0; int k1 = k - 1;
        double term1Reliability=0,term2Reliability=0,
                term3Reliability=0;
        //each node fault free probability with constant
        // rate a=constnFailRate at time=t
        double p=Math.exp((-1)*constnFailRate*time);
        int t1 = factorialDivideFact(n, i: 1,k);
        double t1FaultFreeProbability = Math.pow(p,(double)t1);
        term1Reliability = n*(k-1)*Math.pow(p,(double)
                (factorialDivideFact(n, i: 1,k)));
        //term2 calculation
        term2Reliability = (k*combinationSmall(n, cc: 2) +
                n*combinationSmall(k1, cc: 2)) *
                Math.pow(p, (double)
                        (2*(factorialDivideFact(n, i: 1, k))))  +
                (2*combinationSmall(n, cc: 2) * combinationSmall(k1, cc: 2)) *
                        Math.pow(p, (double)summation( iSum: 2, n, k));
        //term3 calculation
        term3Reliability = (k1*combinationSmall(n, cc: 3)
                + n*combinationSmall(k1, cc: 3)) *
                Math.pow(p, (double)(3*factorialDivideFact(n, i: 1, k)))    +
                (6*combinationSmall(n, cc: 3) * combinationSmall(k1, cc: 3)) *
                        Math.pow(p, (double)(summation( iSum: 3, n, k)))    +
                (4*combinationSmall(n, cc: 2) * combinationSmall(k1, cc: 2)) *
                        Math.pow(p,(double)(3*factorialDivideFact(n, i: 1,k)
                                - factorialDivideFact(n, i: 2,k)))          +
                ((2*n+2*k-10) * combinationSmall(n, cc: 2) *
                        combinationSmall(k1, cc: 2)) *
                        Math.pow(p, (double)(3*factorialDivideFact(n, i: 1,k)
                                - 2*factorialDivideFact(n, i: 2,k)));

        System.out.println("term1Reliability= " + term1Reliability);
        System.out.println("term2Reliability= " + term2Reliability);
        System.out.println("term3Reliability= " + term3Reliability);
        reliability = term1Reliability - term2Reliability + term3Reliability;
        return reliability;
    }
    private static double approximateReliability
            (int n, int k, int time, double constnFailRate){
        double reliability=0; int k1 = k - 1;
        //each node fault free probability with
        // constant rate a=constnFailRate at time=t
        double p=Math.exp((-1)*constnFailRate*time);
        reliability = 1 - Math.pow((1 -
                (Math.pow(p, (double)(factorialDivideFact(n, i: 1,k))))),
                (double)(n*k1));
        return reliability;
    }
    public static void main(String[] args) {
        double lowerBound[] = new double[10];
        double upperBound[] = new double[10];
        double approximate[]= new double[10];
        int index = 0;
        for (int i=20; i<66; i=i+5) {...}
        index=0;
        for (int i=20; i<66; i=i+5) {...}
        index=0;
        for (int i=20; i<66; i=i+5) {...}
        index=0;
        System.out.println("\n" + "lowerBound values: ");
        for(index=0; index<10; index++)
            System.out.println(lowerBound[index]);
        System.out.println("\n" +"upperBound values: ");
        for(index=0; index<10; index++)
            System.out.println(upperBound[index]);
        System.out.println("\n"+ "approximate values: ");
        for(index=0; index<10; index++)
            System.out.println(approximate[index]);
        /*MTTF calculation */
        int T0 = MTTF( n: 6, k: 5, i: 0, constnFailRate: 0.001321);
        int T1 = MTTF( n: 6, k: 5, i: 1, constnFailRate: 0.001321);
        int T2 = MTTF( n: 6, k: 5, i: 2, constnFailRate: 0.001321);
        int T3 = MTTF( n: 6, k: 5, i: 3, constnFailRate: 0.001321);
        int T4 = MTTF( n: 6, k: 5, i: 4, constnFailRate: 0.001321);
        int T5 = MTTF( n: 6, k: 5, i: 5, constnFailRate: 0.001321);
        int T6 = MTTF( n: 6, k: 5, i: 6, constnFailRate: 0.001321);
```

For the purpose of checking the accuracy of this program, we simulate the results of Table I and the graph from reference paper [2] and [1] respectively. Also, to avoid any confusion, we use the same constant failure rate for each star network as suggested and the same mission time to try to come up with similar line graph to the reference papers.

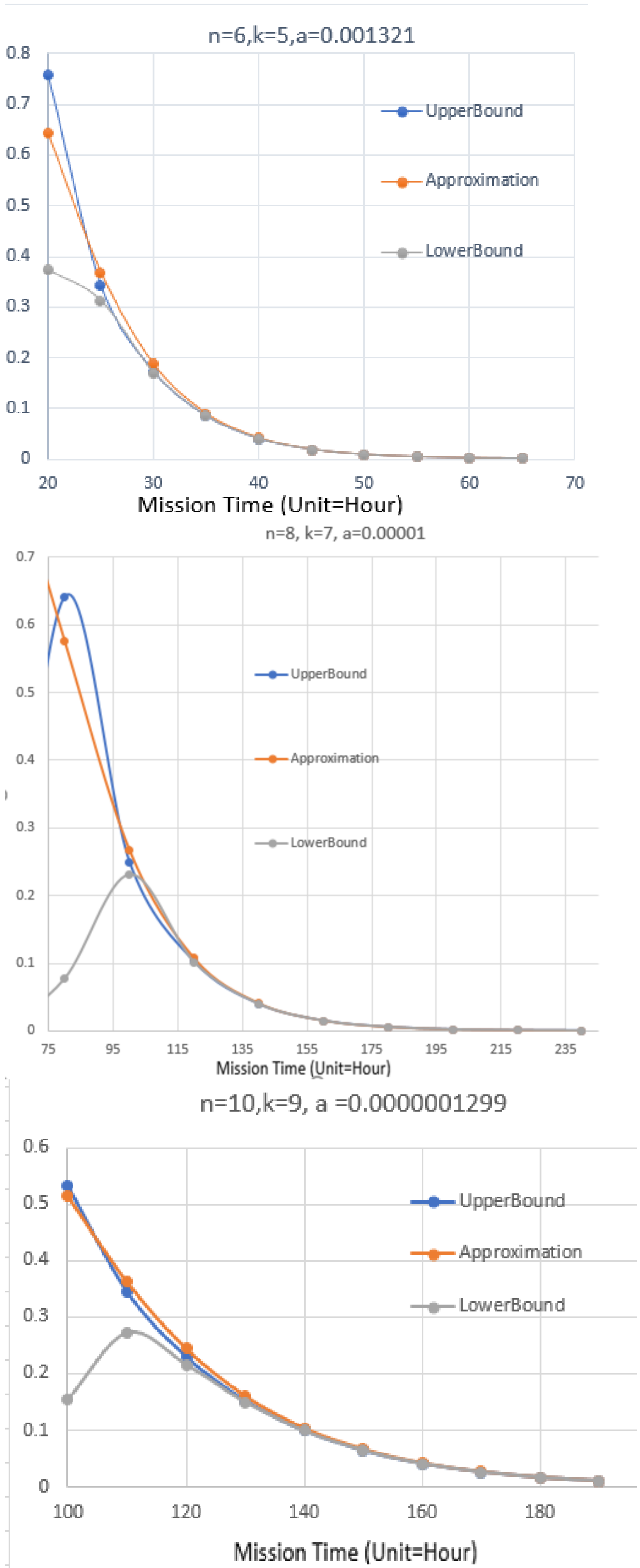


Fig. 5. Reliability of (n-1, k-1)-substar graphs

| | Table IV | | |
|---|---|---|---|
| Time | UpperBound | Approximation | LowerBound |
| 60 | 3.796842193 | 0.908527457 | -13.5008056 |
| 80 | 0.641482294 | 0.576459313 | 0.077238976 |
| 100 | 0.249884061 | 0.267835532 | 0.231277697 |
| 120 | 0.102223132 | 0.107329844 | 0.101603535 |
| 140 | 0.039559117 | 0.040558962 | 0.0395383 |
| 160 | 0.01481774 | 0.014992799 | 0.014817034 |
| 180 | 0.005468087 | 0.005497309 | 0.005468063 |
| 200 | 0.002004911 | 0.002009672 | 0.00200491 |
| 220 | 7.33E-04 | 7.34E-04 | 7.33E-04 |
| 240 | 2.68E-04 | 2.68E-04 | 2.68E-04 |

### F. *Discussion*

This section compares the results from the three plots that shows the merge of both lower and upper bound of probabilistic fault model to approximation [see Fig. 5]. At the beginning as the time, the fault free probability is high, so is the reliability. As time goes by the probabilistic fault model for both lower and upper bout merge to agreement as well as the approximation.

Table IV is the result of our program and is used to plot (8,7)-star graph. At time 60h, both upper and lower bound has the reliability not inside the bound of 0 and 1, it may due to software program error or miss match between type of intersected 4-subgraph number of group listed in table II and the fourth term of lower-bound reliability equation (2). The time after 80h in Table IV are all accurate. Another explanation of the curve down of the either lower-bound or upper-bound of all three plots MTTF. All three plots represent the probability of the system having at least one functional subsystem. Table III shows the time where the system having at least one non-faulty subsystem. Before that time, table III shows the system has two or more non-faulty subsystem, which does not apply to the three plots. The plots of (6,5)-star graph starts with the reliability of the system having at least one non-faulty subsystem at time around 20h, and in Table III confirm that $T_5$ is 20h. It indicates that MTTF of (6,5)-star confirms the accuracy of the probabilistic fault model and the approximation. The MTTF for (8,7)-star graph and (10,9)-star graph is $T_7$=56 and $T_9$=69 respectively. The different of MTTF to the reliability of having at least one non-faulty node in the plots, where (8,7)-star graph is around time 100h and (10,9)-star graph is around time of 110h respectively. The discrepancy might have been either the software program or the inconsistency of number of groups of the intersection of 4-subgraph provided in either table II or upper-bound equation of the reference [1].

## IV. Conclusion

The (n,k)-star graph is resilience, which has been proved and repeat by this paper. The probabilistic fault model computes the intersection of 4-subgraph, and the approximation approach only includes the non-intersected subgraph. We have created program to compute lower-bound, upper-bound and approximate approach to calculate the reliability of a system based on the probability of having at least one non-faulty

Table III

| (n,k)-star | Number of nodes | Fail Rate | T0 | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (6,5)-star | 720 | 0.001321 | 2 | 4 | 6 | 9 | 13 | 20 | | | | |
| (7,6)-star | 5040 | 0.00001 | 20 | 44 | 72 | 107 | 154 | 224 | 363 | | | |
| (8,7)-star | 40320 | 0.00001 | 3 | 6 | 10 | 14 | 19 | 26 | 36 | 56 | | |
| (10,9)-star | 3628800 | 1.3E-07 | 3 | 6 | 9 | 13 | 17 | 22 | 28 | 36 | 47 | 69 |

subsystem. The discrepancy among the three approach goes down as probability of a single to node to be fault goes down. MTTF confirms the time that the system has at least one non-faulty subsystem. The time on the plots that the reliability start to diminish agree with the MTTF of having at least one fault free subsystem. The higher the MTTF, the higher the time on plots should be before the reliability start to diminish. The robustness of the system is indicated by the higher number of MTTF and the higher time t on x-axis at the start of the plot. Although there is discrepancy between the MTTF and the x-axis time t of the last two plots, the data indicate the result is in bound of acceptance.

Our next interest if to explore the application of (n,k)-star graph on NoC. Many network topology has been implemented on NoC such as Crossbar network, Butterfly network and 2D-Mesh network. We would like to explore if (n,k)-star graph could implemented as a fault tolerant network in the chip. This is our next approach from here.

## References


[1] X. Li, S. Zhou, X. Xu, L. Lin and D. Wang, "The Reliability Analysis Based on Subsystems of $(n,k)$ -Star Graph," vol. 65, no. 4, pp. 1700-1709, Dec 2016.

[2] S. Zhou, X. Li, J. Li and D. Wang, "Reliability Assessment of Multiprocessor System Based on $(n,k)$-Star Network," vol. 66, no. 4, pp. 1025-1035, Dec. 2017

[3] K. Fitzgerald, S. Latifi and P. K. Srimani, "Reliability modeling and assessment of the Star-Graph networks," vol. 51, no. 1, pp. 49-57, March 2002.

[4] X. Wu and S. Latifi, "Substar Reliability Analysis in Star Networks," Information Sciences, Volume 178, Issue10, 2008

[5] WK Change and RJ Chen, "The (n,k)-star graph: A Generalization Star Graph," Information Processing Letters, 1995