Simulator for Software Project Reliability Estimation

Sanjana
Department of Computer Science and Applications,
Kurukshetra University, Kurukshetra, Haryana (INDIA)
Sanjana.taya@gmail.com

Dr. P.K. Suri
(Dean, Faculty of Engg. & Technology, Professor & Chairman)
Department of Computer Science and Applications
Kurukshetra University, Kurukshetra, Haryana (INDIA)
pksuritf25@yahoo.com

Abstract— **Several models are there for software development processes, each describing approaches to a variety of tasks or activities that take place during the process. Without project management, software projects can easily be delivered late or over budget. With large numbers of software projects not meeting their expectations in terms of functionality, cost or delivery schedule, effective project management appears to be lacking.IEEE defines reliability as "the ability of a system to perform its required function under stated conditions for a specified period of time. To most software project managers, reliability is equated to correctness that is number of bugs found and fixed. The purpose is to develop a simulator for estimating the reliability of the software project using PERT approach keeping in view the criticality index of each task.**

Keywords- Software Project Reliability; SoftwareProject Network; Criticality Index; Simulation, PERT; Warshall's Algorithm; Transmission Path

I. INTRODUCTION

In recent era of corporate management confusion and reorganization, in addition to the time and cost issues in software project management, reliability have become even more critical. The traditional project management models with ever-present budget overruns and late completion of the project have reduced the concept of credibility and injured the productivity and hence unreliability in transmission path between specific nodes in the network.

Shelbi Joseph et al. [1] proposed an algorithm for determining reliability of software based systems by integrating both hardware reliability and software reliability. The open source software data is made use of and the methodology for evaluating the software reliability involves identifying a fixed number of packages at the start of the time and defining the failure rate based on the failure data for these preset number of packages. The defined function of the failure rate is used to arrive at the software reliability model. Mou Dasgupta et al. [2] proposed two linear time complexity algorithms for approximate assessment and the enhancement of the reliability of the given networks. The proposed techniques basically identify the node-pairs having lower reliability, insert communication links in them and calculate the increase in reliability on insertion iteratively until the satisfactory reliability is achieved.

S Chatterjee et al. [3] proposed a sequential Bayesian approach for estimating reliability growth or decay of software. It also showed the variation of parameter over a time, as new failure data become available. The usefulness of method is demonstrated with some real life data.P.K. Suri et al. [4] proposed a simulator for estimating the reliability of communication network and to avoid problems in communication networks like topology delay, congestion, throughput, flow-control etc. and to establish an effective communication among nodes which requires network to be reliable. Network reliability is based on whether the link between two nodes operates properly. Hui Kin-Ping [5] presented different techniques for network reliability estimation like combinatorial sampling; permutation sampling, tree cut and merge, importance sampling etc. They also estimated the difference in reliability of two very similar networks.

.Cancela H et al. [6] described Monte Carlo simulation to be useful in the evaluation of K-terminal-reliability of large communication systems because the exact algorithms are extremely time consuming. This paper shows that the well-known series-parallel reductions can be incorporated in the recursive variance reduction simulation method, leading to a more efficient estimator, as demonstrated by experimental results. Cavers J. et al. [7] discussed the problem of calculating the reliability of a communication network. The definition of failure is generalized from simple disconnection to the event that terminal capacity falls below some threshold, and simple methods are developed for bounding the probability of such failures and for approximating the mean time between failures.

The success of any project is very much dependent upon the quality of the planning, scheduling and controlling of various phases of the project. Program evaluation and review technique is an aid to management in expediting and controlling the resources to meet the scheduled completion date of the projects which involve high degree of uncertainty. In Research & Development projects, or in Social Projects which are defined as "Process Projects", where learning is an important outcome, the cause-effect relationship is not so well established. In such situations, the PERT approach is useful, because it can accommodate the variation in event completion times, based on an expert committee’s estimates. Activity durations are estimates of the actual time required and there is liable to be a significant amount of uncertainty associated with the actual durations. To incorporate uncertainty in the scheduling process is to apply the critical path scheduling process and then analyze the results from probabilistic perspective. Using expected activity durations and critical path scheduling, critical activities can be determined. The critical path is then used to analyze the duration of the project incorporating the uncertainties of activities duration along the critical path. The expected project duration is equal to the sum of the expected durations of the activities along the critical path.

If to, tm, tp are the optimistic, most likely and pessimistic activity durations denoted by To[I], Tm[I], Tp[I] respectively for each activity I, then the mean and standard deviation for the respective activity are computed as:

$$\text{Mu}[I] = (\text{To}[I] + 4\text{Tm}[I] + \text{Tp}[I])/6$$

$$\text{Sigma}[I] = (\text{Tp}[I] - \text{To}[I])/6$$

The use of these optimistic, most likely and pessimistic estimates stems from the fact that these are thought to be easier for managers to estimate subjectively.

II. PROPOSED MODEL

In research and development projects or in social projects which are having high degree of uncertainties in their outcomes. In such situations, the PERT approach is useful, because it can accommodate the variation in event completion times, based on an expert committee’s estimates. A simulator is developed for software project estimating the reliability of activities involved in the project and is generalized to estimate the reliability of any network project. The simulator is developed using high level programming language.

ALGORITHM DESCRIPTION

Terms and Notations Used

N	: Total no of activities
M	: Total no of nodes
RUNS	: Number of Simulation Runs
S [I]	: Starting node for I th activity
F [I]	: Finishing node for I th activity
MU [I]	: Mean Time of I th activity defined by probability distribution
SIGMA [I]	: Standard Deviation of I th activity defined by probability distribution
TIME [I]	: Data Transmission Time of I th activity
CRI [I]	: Criticality Index of I th activity

REL [I]	: Reliability of I th activity
WEI [I]	: Weighted Reliabilities of I th activity

PRO_REL_SIM Algorithm

```
1. a)  Read the total number of activities M
   b)  Read the total number of nodes N
   c)  Read the number of simulation runs RUNS
   d)  Set I to 1
       Repeat while (I<=N)
       [Read time estimates, start node, finish node and reliability R [I] for each I th activity]
       Read To[I], Tm[I], Tp[I],S[I], F[I], R[I]
       Increment I by 1
       End Loop
2.     Set I to 1
       Repeat while (I<=N)
       [Compute activity duration time MU [I] and standard deviation SIGMA [I] using beta distribution]
       MU [I] = (To [I] +4Tm [I] +Tp [I])/6
       SIGMA [I] = (Tp [I]-To [I])/6
       Increment I by 1
       End Loop
3.     Set I to 1
       Repeat while (I<= RUNS)
       Repeat while (I<=N)
       [Compute data transmission time for I th activity using Box-Muller transformation]
   a)  Generate a pair of pseudorandom numbers (r1, r2) from random number generator using different seed
       values.
   b)  Compute S= (SQRT (-2*log (r1))*COS (2*3.1415*r2))
   c)  Compute TIME [I] =S*SIGMA [I] +MU [I]
   d)  Traverse forward pass for each transmission path.
   e)  Traverse backward pass for each transmission path.
   f)  Update criticality index counter only for those communication paths which become critical during
       simulation runs.
       Increment I by 1
       End Loop
       Increment I by 1
       End Loop
4.     Calculate criticality index CRI [I] of each transmission path for each Ith activity.
5.     Calculate initial reliability REL [I] of each transmission path for each activity using simple RAND
       function.
6.     Calculate Weighted reliabilities WEI [I] of each data transmission path by multiplying the Initial
       reliability with the criticality index matrix.
7.     Generate minimal path sets through warshalls algorithm
8.     Stop
```

III. CASE STUDY

Consider a network project with 13 activities and 11 terminals as shown in the fig 1 labeled with Time estimates To, Tm, Tp corresponding to each activity and used to calculate the value of MU and SIGMA respectively. Box-Muller transformation is used to generate the duration of data transmission for each activity from the probability distribution for each activity such as MU [I] and SIGMA [I] for Ith activity. A PERT approach presents a graphic illustration of a project as a network diagram consisting of numbered nodes representing milestones in the project linked by labeled vectors representing tasks involved in the project Here source node is 1 and sink node is 11. The details of the activities to be carried out for project is shown with the help of network diagram shown in fig 1.A network project include the following activities as shown in table 1.

Activity-id	Activity-Description
1-2	Create schedule
1-3	Buy hardware product
2-4	Involved in Programming
3-5	Product Installation
5-4	Dummy activity
5-7	Write user manual
5-6	Conversion of system files
4-8	Testing
6-9	Dummy activity
7-9	Provide user Training
8-10	Test system
10-9	Dummy activity
9-11	User test

Table1: Activities description involved in project

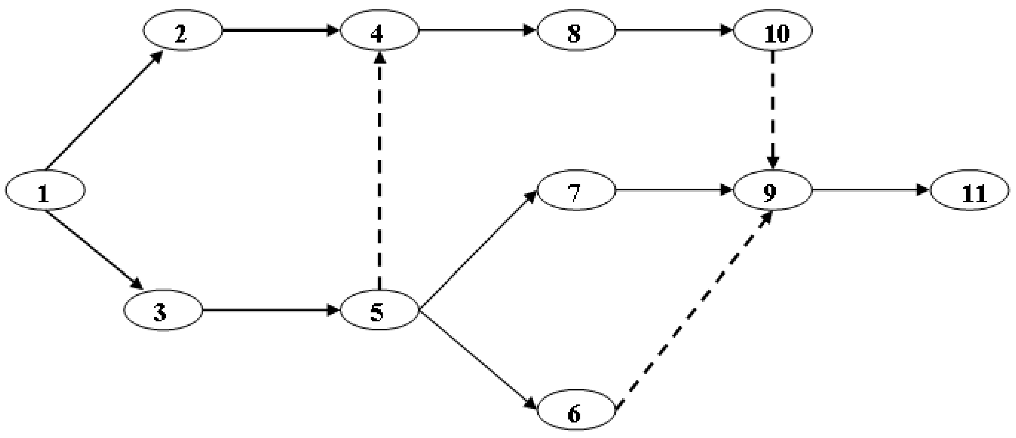


Figure1: Software Project Network Diagram

- Numbered rectangles are nodes represent events or milestones.
- Directional arrow represents dependent tasks that must be completed sequentially like 1,2,4,8 and10
- Diverging arrow directions like 1-2 & 1-3 indicate possibly concurrent tasks.
- Dotted lines indicate dependent tasks that don't require resources called dummy activity. for e.g. linking nodes 6 and 9 indicates that the system files must be converted before the user test can take place, but that the resources and time required to prepare for the user test (writing the user manual and user training) are on another path.

Description of activities for particular activity-id is shown in table 2.

Activity-id	Activity No.	Activity-Description	Time Estimates (To, Tm, Tp)
1-2	1	Create schedule	2,5,14
1-3	2	Buy hardware product	7,10,13
2-4	3	Programming	3,12,15
3-5	4	Product Installation	1,3,7
5-4	5	Dummy activity	0,0,0
5-7	6	Write user manual	2,2,8
5-6	7	Conversion of system files	1,6,7
4-8	8	Testing	2,6,14
6-9	9	Dummy activity	0,0,0
7-9	10	Provide user Training	4,13,16
8-10	11	Test system	4,4,10
10-9	12	Dummy activity	0,0,0
9-11	13	User test	4,5,10

Table 2: Description of activities along with their ids

Assumptions:

- All tasks are performed independently.
- Initial reliability of all the tasks is randomly generated using simple RAND function
- Weighted reliability will be calculated for all those tasks which are being traversed while moving from one node to another node.
- Traverse using forward and backward pass for each transmission path.
- Transmission time is calculated using Box-Muller Transformation
- Once the project is started, it is not certain whether the project would be completed or not.

Implementation:

Output obtained for the simulator developed for software project shown in figure1 for different runs 1000, 5000, 10000, 15000, 20000 etc. is given below:

For 1000 runs

Critical activities are   2   4   6 10 13

Reliability matrix for each activity

```
0    0.29  0.7  0    0    0    0    0    0    0    0
0    0     0    0.26 0    0    0    0    0    0    0
0    0     0    0    0.49 0    0    0    0    0    0
0    0     0    0    0    0    0    0.64 0    0    0
0    0     0    0.69 0    0.16 0.42 0    0    0    0
0    0     0    0    0    0    0    0    0.51 0    0
0    0     0    0    0    0    0    0    0.88 0    0
0    0     0    0    0    0    0    0    0    0.42 0
0    0     0    0    0    0    0    0    0    0    0.04
0    0     0    0    0    0    0    0    0.12 0    0
0    0     0    0    0    0    0    0    0    0    0
```

Criticality Index Matrix

```
 0   0.528 0.546 0     0     0   0     0     0     0     0
 0   0     0     0.528 0     0   0     0     0     0     0
 0   0     0     0     0.546 0   0     0     0     0     0
 0   0     0     0     0     0   0     0.539 0     0     0
 0   0     0     0.017 0     0   0.538 0     0     0     0
 0   0     0     0     0     0   0     0     0     0     0
 0   0     0     0     0     0   0     0     0.538 0     0
 0   0     0     0     0     0   0     0     0     0.539 0
 0   0     0     0     0     0   0     0     0     0     1
 0   0     0     0     0     0   0     0     0.539 0     0
 0   0     0     0     0     0   0     0     0     0     0
```

Weighted Reliability Matrix

```
 0   0   0   0.15312 0.3822 0   0       0       0       0       0
 0   0   0   0       0      0   0       0.14014 0       0       0
 0   0   0   0.00833 0      0   0.26362 0       0       0       0
 0   0   0   0       0      0   0       0       0       0.34496 0
 0   0   0   0       0      0   0       0.37191 0.22596 0       0
 0   0   0   0       0      0   0       0       0       0       0.51
 0   0   0   0       0      0   0       0       0       0       0.88
 0   0   0   0       0      0   0       0       0.22638 0       0
 0   0   0   0       0      0   0       0       0       0       0
 0   0   0   0       0      0   0       0       0       0       0.12
 0   0   0   0       0      0   0       0       0       0       0
```

Total transmission paths are 20

Transmission path reliability matrix

```
    |1  2  3   4       5      6  7        8       9       10       11
Node|
    ___________________________________________________________________
  1 |0  0  0  0.15312 0.3822 0  0        0.75411 0.60816 0.49808  0.61808
  2 |0  0  0  0       0      0  0        0.14014 0.36652 0        0
  3 |0  0  0  0.00833 0      0  0.26362  0       0       0.35329  0.47329
  4 |0  0  0  0       0      0  0        0       0       0.34496  0.46496
  5 |0  0  0  0       0      0  0        0.37191 0.22596 0        0
  6 |0  0  0  0       0      0  0        0       0       0        0.51
  7 |0  0  0  0       0      0  0        0       0       0        0.88
  8 |0  0  0  0       0      0  0        0       0.22638 0        0
  9 |0  0  0  0       0      0  0        0       0       0        0
  10|0  0  0  0       0      0  0        0       0       0        0.12
  11|0  0  0  0       0      0  0        0       0       0        0
```

For 5000 runs

Critical activities are 2 4 6 10 13

Reliability matrix for each activity

```
0   0.002 0.52  0     0     0     0      0     0     0     0
0   0     0     0.74  0     0     0      0     0     0     0
0   0     0     0     0.49  0     0      0     0     0     0
0   0     0     0     0     0     0      0.59  0     0     0
0   0     0     0.9   0     0.89  0.45   0     0     0     0
0   0     0     0     0     0     0      0     0.02  0     0
0   0     0     0     0     0     0      0     0.11  0     0
0   0     0     0     0     0     0      0     0     0.24  0
0   0     0     0     0     0     0      0     0     0     0.1
0   0     0     0     0     0     0      0     0.55  0     0
0   0     0     0     0     0     0      0     0     0     0
```

Criticality Index Matrix

```
0   0.5412 0.545 0      0     0   0      0      0      0      0
0   0      0     0.5412 0     0   0      0      0      0      0
0   0      0     0      0.545 0   0      0      0      0      0
0   0      0     0      0     0   0      0.5518 0      0      0
0   0      0     0.0192 0     0   0.5338 0      0      0      0
0   0      0     0      0     0   0      0      0      0      0
0   0      0     0      0     0   0      0      0.5338 0      0
0   0      0     0      0     0   0      0      0      0.5518 0
0   0      0     0      0     0   0      0      0      0      1
0   0      0     0      0     0   0      0      0.5518 0      0
0   0      0     0      0     0   0      0      0      0      0
```

Weighted Reliability Matrix

```
0   0   0   0.001082 0.2834 0  0        0        0       0        0
0   0   0   0        0      0  0        0.408332 0       0        0
0   0   0   0.009408 0      0  0.261562 0        0       0        0
0   0   0   0        0      0  0        0        0       0.325562 0
0   0   0   0        0      0  0        0.49662  0.24021 0        0
0   0   0   0        0      0  0        0        0       0      0.51
0   0   0   0        0      0  0        0        0       0      0.88
0   0   0   0        0      0  0        0        0.13243 0        0
0   0   0   0        0      0  0        0        0       0        0
0   0   0   0        0      0  0        0        0       0      0.12
0   0   0   0        0      0  0        0        0       0        0
```

Total transmission paths are 20

Transmission Path Reliability Matrix

```
    |1 2 3   4        5      6  7          8        9       10      11
Node|
    ____________________________________________________________________
  1 |0 0 0 0.001082 0.2834 0  0       0.78002 0.52361 0.32664 0.876644
  2 |0 0 0 0        0      0  0       0.40833 0.54076 0       0
  3 |0 0 0 0.009408 0      0  0.26156 0       0       0.33497 0.371562
  4 |0 0 0 0        0      0  0       0       0       0.32556 0.875562
  5 |0 0 0 0        0      0  0       0.49662 0.24021 0       0
  6 |0 0 0 0        0      0  0       0       0       0       0.02
  7 |0 0 0 0        0      0  0       0       0       0       0.11
  8 |0 0 0 0        0      0  0       0       0.13243 0       0
  9 |0 0 0 0        0      0  0       0       0       0       0
  10|0 0 0 0        0      0  0       0       0       0       0.55
  11|0 0 0 0        0      0  0       0       0       0       0
```

For 10000 runs

Critical activities are 1 3 8 11 12 13

Reliability Matrix for each activity

```
0   0.14  0.94 0    0    0    0      0     0     0    0
0   0     0    0.2  0    0    0      0     0     0    0
0   0     0    0    0.33 0    0      0     0     0    0
0   0     0    0    0    0    0      0.01  0     0    0
0   0     0    0.34 0    0.92 0.31   0     0     0    0
0   0     0    0    0    0    0      0     0.18  0    0
0   0     0    0    0    0    0      0     0.5   0    0
0   0     0    0    0    0    0      0     0     0.08 0
0   0     0    0    0    0    0      0     0     0    0.65
0   0     0    0    0    0    0      0     0.64  0    0
0   0     0    0    0    0    0      0     0     0    0
```

Criticality Index Matrix

```
0   0.5394 0.5473 0      0      0   0      0      0      0      0
0   0      0      0.5394 0      0   0      0      0      0      0
0   0      0      0      0.5473 0   0      0      0      0      0
0   0      0      0      0      0   0      0.5509 0      0      0
0   0      0      0.0196 0      0   0.5364 0      0      0      0
0   0      0      0      0      0   0      0      0      0      0
0   0      0      0      0      0   0      0      0.5364 0      0
0   0      0      0      0      0   0      0      0      0.5509 0
0   0      0      0      0      0   0      0      0      0      1
0   0      0      0      0      0   0      0      0.5509 0      0
0   0      0      0      0      0   0      0      0      0      0
```

Weighted Reliability Matrix

```
0  0  0  0.075516 0.51446 0  0        0        0        0        0
0  0  0  0        0       0  0.11018  0        0        0        0
0  0  0  0.006468 0       0  0.177012 0        0        0        0
0  0  0  0        0       0  0        0        0        0.005509 0
0  0  0  0        0       0  0        0.187306 0.166284 0        0
0  0  0  0        0       0  0        0        0        0        0.18
0  0  0  0        0       0  0        0        0        0.5      0
0  0  0  0        0       0  0        0        0.044072 0        0
0  0  0  0        0       0  0        0        0        0        0
0  0  0  0        0       0  0        0        0        0        0.64
0  0  0  0        0       0  0        0        0        0        0
```

Total Transmission Paths are 20

Transmission Path Reliability Matrix

```
    |1  2  3   4      5      6   7       8       9        10       11
Node|
    ____________________________________________________________________
  1 |0 0  0 0.07551 0.51446 0 0      0.70176 0.68074 0.081025 0.821025
  2 |0 0  0 0       0       0 0      0.11018 0.15425 0        0
  3 |0 0  0 0.00646 0       0 0.1770 0       0       0.011977 0.65177
  4 |0 0  0 0       0       0 0      0       0       0.005509 0.64559
  5 |0 0  0 0       0       0 0      0.18730 0.16628 0        0
  6 |0 0  0 0       0       0 0      0       0       0        0.18
  7 |0 0  0 0       0       0 0      0       0       0        0.5
  8 |0 0  0 0       0       0 0      0       0.04407 0        0
  9 |0 0  0 0       0       0 0      0       0       0        0
  10|0 0  0 0       0       0 0      0       0       0        0.64
  11|0 0  0 0       0       0 0      0       0       0        0
```

For 15000 runs

Critical Activities are   2   4   6 10 13

Reliability Matrix for each activity

```
0   0.61  0.08  0    0    0     0     0    0     0     0
0   0     0     0.78 0    0     0     0    0     0     0
0   0     0     0    0.5  0     0     0    0     0     0
0   0     0     0    0    0     0     0.52 0     0     0
0   0     0     0.41 0    0.98  0.54  0    0     0     0
0   0     0     0    0    0     0     0    0.73  0     0
0   0     0     0    0    0     0     0    0.58  0     0
0   0     0     0    0    0     0     0    0     0.08  0
0   0     0     0    0    0     0     0    0     0     0.59
0   0     0     0    0    0     0     0    0.23  0     0
0   0     0     0    0    0     0     0    0     0     0
```

Criticality Index Matrix

```
0  0.536 0.55033 0       0       0 0       0      0       0       0
0  0     0       0.53673 0       0 0       0      0       0       0
0  0     0       0       0.55033 0 0       0      0       0       0
0  0     0       0       0       0 0       0.5483 0       0       0
0  0     0       0.0192  0       0 0.53933 0      0       0       0
0  0     0       0       0       0 0       0      0       0       0
0  0     0       0       0       0 0       0      0.53933 0       0
0  0     0       0       0       0 0       0      0       0.54833 0
0  0     0       0       0       0 0       0      0       0       1
0  0     0       0       0       0 0       0      0.54833 0       0
0  0     0       0       0       0 0       0      0       0       0
```

Weighted Reliability Matrix

```
0  0  0  0.327407 0.044027 0 0        0        0       0        0
0  0  0  0        0        0 0        0.4277   0       0        0
0  0  0  0.0096   0        0 0.269667 0        0       0        0
0  0  0  0        0        0 0        0        0       0.285133 0
0  0  0  0        0        0 0        0.224817 0.29124 0        0
0  0  0  0        0        0 0        0        0       0        0.73
0  0  0  0        0        0 0        0        0       0        0.58
0  0  0  0        0        0 0        0        0.04386 0        0
0  0  0  0        0        0 0        0        0       0        0
0  0  0  0        0        0 0        0        0       0        0.23
0  0  0  0        0        0 0        0        0       0        0
```

Total Transmission Paths are 20

Transmission Path Reliability Matrix

```
    |1 2 3   4      5      6 7         8        9      10       11
Node|
    ___________________________________________________________________
  1 |0 0 0 0.3274 0.04402 0  0       0.26884 0.31271 0.061254 0.842541
  2 |0 0 0 0      0       0  0       0.4277  0.47156 0        0
  3 |0 0 0 0.0096 0       0  0.26966 0       0       0.294733 0.524733
  4 |0 0 0 0      0       0  0       0       0       0.385133 0.515133
  5 |0 0 0 0      0       0  0       0.22481 0.26868 0        0
  6 |0 0 0 0      0       0  0        0      0       0        0.73
  7 |0 0 0 0      0       0  0       0       0       0        0.58
  8 |0 0 0 0      0       0  0       0       0.04386 0        0
  9 |0 0 0 0      0       0  0       0       0       0        0
  10|0 0 0 0      0       0  0       0       0       0        0.23
  11|0 0 0 0      0       0  0       0       0       0        0
```

For 20000 runs

Critical Activities are 1 3 8 11 12 13

Reliability Matrix for each activity

```
0   0.02 0.73 0     0    0    0    0    0     0     0
0   0    0    0.002 0    0    0    0    0     0     0
0   0    0    0     0.29 0    0    0    0     0     0
0   0    0    0     0    0    0    0.85 0     0     0
0   0    0    0.95  0    0.85 0.88 0    0     0     0
0   0    0    0     0    0    0    0    0.57  0     0
0   0    0    0     0    0    0    0    0.35  0     0
0   0    0    0     0    0    0    0    0     0.44  0
0   0    0    0     0    0    0    0    0     0     0.91
0   0    0    0     0    0    0    0    0.39  0     0
0   0    0    0     0    0    0    0    0     0     0
```

Criticality Index Matrix

```
0  0.53735 0.55065 0      0       0  0      0      0      0      0
0  0       0       0.5374 0       0  0      0      0      0      0
0  0       0       0      0.55065 0  0      0      0      0      0
0  0       0       0      0       0  0      0.5487 0      0      0
0  0       0       0.0195 0       0  0.5395 0      0      0      0
0  0       0       0      0       0  0      0      0      0      0
0  0       0       0      0       0  0      0      0.5393 0      0
0  0       0       0      0       0  0      0      0      0.5487 0
0  0       0       0      0       0  0      0      0      0      1
0  0       0       0      0       0  0      0      0.5487 0      0
0  0       0       0      0       0  0      0      0      0      0
```

Weighted Reliability Matrix

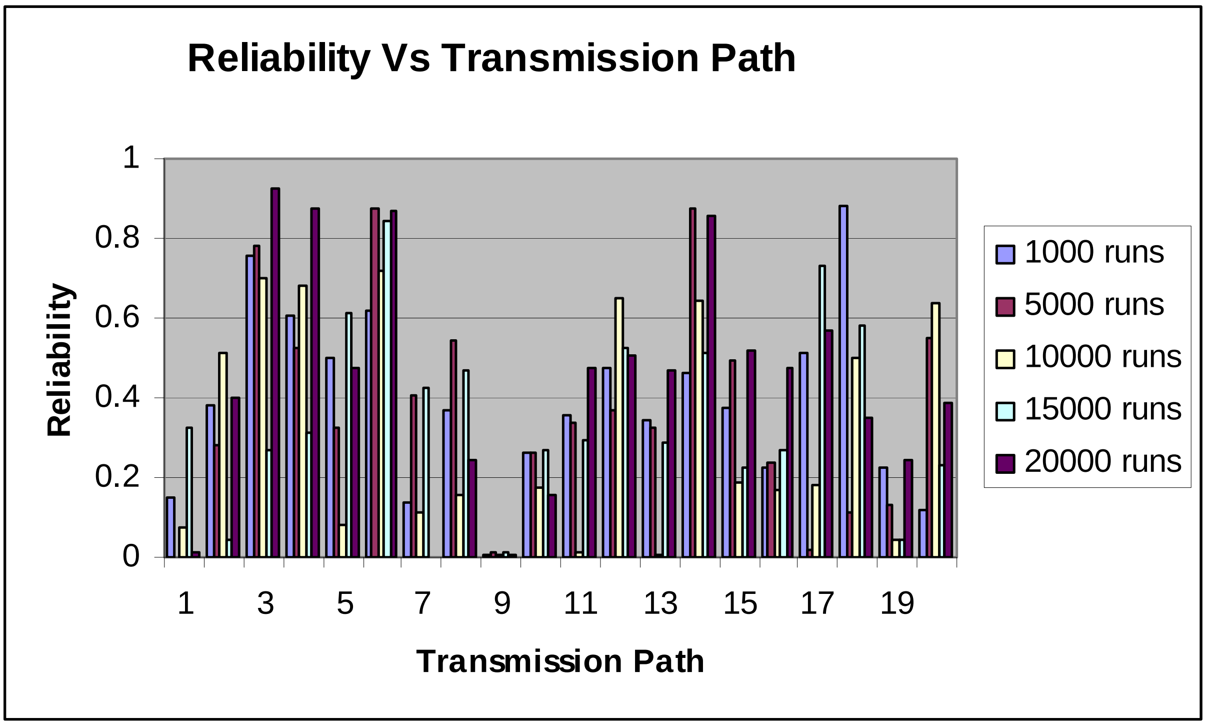
```
0  0  0 0.010748 0.401974 0 0        0        0       0        0
0  0  0 0        0        0 0        0.001097 0       0        0
0  0  0 0.005655 0        0 0.156455 0        0       0        0
0  0  0 0        0        0 0        0        0       0.466395 0
0  0  0 0        0        0 0        0.521265 0.47476 0        0
0  0  0 0        0        0 0        0        0       0        0.57
0  0  0 0        0        0 0        0        0       0        0.35
0  0  0 0        0        0 0        0        0.24142 0        0
0  0  0 0        0        0 0        0        0       0        0
0  0  0 0        0        0 0        0        0       0        0.39
0  0  0 0        0        0 0        0        0       0        0
```

Total Transmission Paths are 20

Transmission Path Reliability Matrix

```
    |1  2 3   4       5      6    7       8      9       10        11
Node|
     ___________________________________________________________________
  1 |0 0 0 0.01074 0.4019 0  0       0.92323 0.87673 0.477143 0.867143
  2 |0 0 0 0       0      0  0       0.00109 0.24252 0        0
  3 |0 0 0 0.00565 0      0  0.15645 0       0       0.47205  0.506455
  4 |0 0 0 0       0      0  0       0       0       0.466395 0.856395
  5 |0 0 0 0       0      0  0       0.52126 0.47476 0        0
  6 |0 0 0 0       0      0  0       0       0       0        0.57
  7 |0 0 0 0       0      0  0       0       0       0        0.35
  8 |0 0 0 0       0      0  0       0       0.24148 0        0
  9 |0 0 0 0       0      0  0       0       0       0        0
  10|0 0 0 0       0      0  0       0       0       0        0.39
  11|0 0 0 0       0      0  0       0       0       0        0
```

Graph corresponding to Reliability and Transmission Path for Software Project



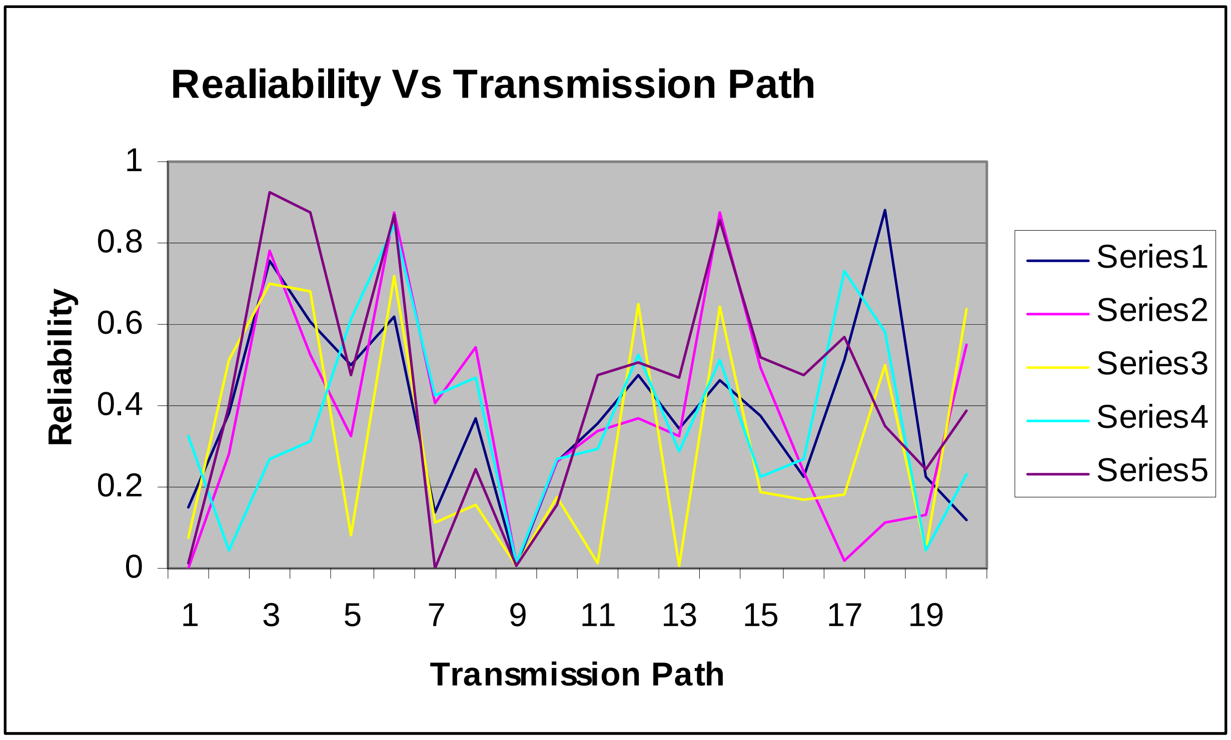


Figure2: Reliability Vs Transmission Path Graphs

IV. RESULTS AND DISCUSSION

Figure 2 shows the graphs between Reliability and the Transmission Path for Simulator developed in order to estimate the reliability of software project under assumptions. It shows the graph for the simulator when it is not certain whether the project would be completed or not. The graph here shows that the maximum reliability is for the tasks between nodes 1 to 8 and is 0.92323.This means that tasks involved between nodes 1 to 8 would be most reliable.

Thus simulator developed is helpful in selecting the most reliable task. The simulators developed here is executed for different simulation runs i.e. for 1000, 5000,10000,15000,20000 runs etc and graph between reliability and transmission path is plotted among which the most reliable task can be selected.

V. CONCLUSION

The simulator developed is executed with 13 links and 11 terminals and their expected time is computed by making use of mean time and standard deviation with the help of Box-Muller Transformation. Mean Time and Standard Deviation are computed using time estimates: optimistic, most likely and pessimistic time with the help of beta-distribution. On keeping track of how many times an activity comes on critical path for different simulation runs, we compute criticality index. The computed criticality index is then multiplied with initial reliability for each task to determine the weighted reliabilities. In actual sense, weighted reliability depends upon the criticality index of that activity. The weighted reliabilities are then used to generate minimal path sets using Warshall’s algorithm with the help of which we determine the most reliable task in the software project. The simulator developed is generalized to estimate the reliability of any software project provided with time estimates.

VI.REFERENCES

[1] Shelbi Joseph ,Shouri P.V ,Jagathy Raj V. P “A Model for Reliability Estimation of Software based Systems by Integrating Hardware and Software” Special Issue on Computational Science - New Dimensions & Perspectives (1):26–29, 2011.

[2] Mou Dasgupta, G.P.Biswa “Reliability Measurement and Enhancement of the Communication Networks”, International Journal of Computer Applications 1(9):18–25, February 2010, Number 9Article 3.

[3] S Chatterjee, S S Alam and R B Mishra “sequential Bayesian Technique: an alternative approach for software reliability estimation”, published in sadhana vol. 34, part2, April 2009.

[4] P.K. Suri, Bharat Bhushan “simulator for network reliability estimation”, International Journal of Computer Sciences and Network Security (IJCSNS), Korea on July 2008.

[5] Hui, Kin-Ping “network reliability estimation”, Software Performance, 5th International Workshop on Software and Performance, Aug 2005.

[6] Cancela, H El Khadiri, M “Series-Parallel reductions in Monte Carlo network-reliability evaluation”, Reliability, Jun 1998 Volume: 47 Issue: 2 on page(s): 159 – 164.

[7] Cavers, J.; Carleton Univ., Ottawa, Ont., Canada “Cutest Manipulations for Communication Network Reliability Estimation”, Communications, IEEE Transactions on June 1975, and Issue: 6 page 569-575.

[8] Jalote Pankaj, Software project management in practice. Addison-Wesley. ISBN 0201737213, 2002.