

Reliability Analysis of Self-Healing Network using Discrete-Event Simulation

Thara Angskun¹, George Bosilca¹, Graham Fagg¹, Jelena Pješivac–Grbović¹, and Jack Dongarra²

¹Department of Computer Science, The University of Tennessee, Knoxville

²University of Tennessee, Oak Ridge National Laboratory and University of Manchester
{angskun, bosilca, fagg, pjesa, dongarra}@cs.utk.edu

Abstract

The number of processors embedded on high performance computing platforms is continuously increasing to accommodate user desire to solve larger and more complex problems. However, as the number of components increases, so does the probability of failure. Thus, both scalable and fault-tolerance of software are important issues in this field.

To ensure reliability of the software especially under the failure circumstance, the reliability analysis is needed. The discrete-event simulation technique offers an attractive alternative to traditional Markovian-based analytical models, which often have an intractably large state space. In this paper, we analyze reliability of a self-healing network developed for parallel runtime environments using discrete-event simulation. The network is designed to support transmission of messages across multiple nodes and at the same time, to protect against node and process failures. Results demonstrate the flexibility of a discrete-event simulation approach for studying the network behavior under failure conditions and various protocol parameters, message types, and routing algorithms.

1 Introduction

Recently, several high performance computing platforms have been installed with more than 10,000 CPUs, such as Blue-Gene/L at LLNL, BGW at IBM and Columbia at NASA [1]. Unfortunately, as the number of components increases, so does the probability of failure. To satisfy the requirement of such a dynamic environment (where the available number of resources is fluctuating), a scalable and fault-tolerance communication framework is required. A self-healing network [2] is developed to facilitate the communication framework requirement. The network is built on top of a scalable and fault-tolerant communication protocol [3]. The network can be used as a basis for constructing

higher level runtime environments for parallel applications that exhibit the two properties required for large scale applications: scalability and fault tolerance. The basic ability of the self-healing network is to efficiently transfer messages across multiple nodes while protecting against node or process failures. The reliability analysis estimates the probability that a system will be operational during a particular time interval. Such analysis is necessary to understand reliability of a network under the failure conditions.

There are several existing analytical methods for reliability analysis. These methods may be classified into two classes: non-state, space-based methods (a.k.a. combinatorial models) and state, space-based methods. Each of them has different advantages and limitations. There are 3 popular techniques [4, 5] using the non-state, space-based methods: a reliability block diagram, a reliability graph and a fault tree. These models capture conditions that make a system fail in terms of relationship between components. The series-parallel reliability block diagram (RBD) [6] is probably the oldest technique for reliability analysis. All components are represented in blocks. All blocks are combined together in series and/or parallel. Unfortunately, only some systems can be mapped into a RBD. The reliability graph [5] (a.k.a non-series-parallel block diagram) is a directed graph where the edges represent a component. Such graphs constitute a superset of the RBD. The fault tree [7, 8] is represented as a tree-like topology where the root of the tree is a failure event of the whole system and the branches represent failures of an individual component. All of the above mentioned non-state, space-based models assume stochastic independence among failures. However, this assumption may not hold in practice. Most of the state, space-based methods, such as Markov reward model and Petri Nets, are based on Markovian (or semi-Markovian) models [9, 10]. The Markov reward model [11] is a Markov model with the reward assigned to all states and transitions. Each submodel is linked by mathematical expressions. Petri Nets [12, 13] consist of places, transitions and directed arcs. Places may contain tokens, which can move from place to

place when each transition becomes enabled. All of the state, space-based model are subject to an intractably large state space.

Discrete-event simulation (DES) [14] is a method to model the behavior of a system in response to designated events as time progresses. It offers an alternative to traditional analytical models as it can capture details of the system and illustrate the influence of various factors. In this paper, we analyze reliability of the self-healing network using the discrete-event simulation technique. We study the influence of various factors on the network behavior in failure circumstances.

The rest of the paper is organized as follows. Section 2 introduces the self-healing network and the simulation. The simulation results and analysis are given in section 3, followed by conclusions and future work in section 4.

2 Overview

In this section, we provide a brief overview of the self-healing network, which was designed to support parallel runtime environment. A similar protocol is currently used in FT-MPI [15]. The reliability analysis concepts and details of the simulation are also mentioned in this section.

2.1 Self-Healing Network (SHN)

SHN [2] was designed to support transfer of messages across multiple nodes efficiently, while protecting against node or process failures. It was built on top of a scalable and fault-tolerant protocol (SFTP) [3] and is capable of automatically recovering itself. The SFTP is based on a k -ary sibling tree topology used to develop a self healing tree topology. The k -ary sibling tree topology is a k -ary tree, where k is number of fan-out ($k \geq 2$), and the nodes on the same level (same depth on the tree) are linked together using a ring topology. The tree is primarily designed to allow scalability for broadcast and multicast operations that are typically required during parallel application startup, input redirection, control signals and termination. The ring is used to provide a well-understood secondary path for transmission when the tree is damaged during failure conditions (simplest multi-path extension).

Fig. 1(a) illustrates an example of the binary ($k=2$) sibling tree. Each node needs to know the contact information of at most $k+3$ neighbors (i.e. parent, left, right and their k children). The number of neighbors is kept to a minimum to reduce the state management load on each node. In both the tree and the ring topologies neighbor's address can be computed locally. When a node attempts to send a broadcast message to all of its children and detects that a child has died, it will reroute the message to all children of that child. This is done using an encapsulation technique. The

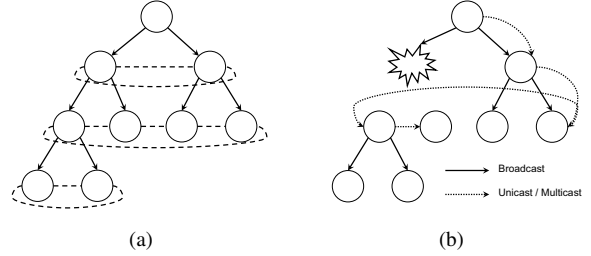


Figure 1. (a) Binary sibling tree topology. (b) Broadcast messages rerouting in failure circumstances.

node will encapsulate the broadcast message into a multicast message and send to its grandchildren. The grandchildren will decapsulate the multicast packet and continue to forward the broadcast message as shown in Fig. 1(b). Alternatively, if a node tries to send a multicast (or unicast) message and detects that the next hop neighbor died, it will choose the next hop from other valid neighbor nodes which have the highest priority (the lowest estimated hop count to the destinations [3]).

2.2 SHN Reliability Simulation

This section presents fundamental concepts of reliability analysis as well as the description of the simulation.

2.2.1 Reliability Analysis Concepts

The reliability of the SHN is defined as its ability to maintain operation over a period of time t , i.e., the reliability, $R(t) = Pr(\text{the network is operational in } [0, t])$. The SHN is “operational”, if it can successfully deliver messages from sources to the alive destination(s) even when some nodes in the routing path die. The probability density function (pdf) which associated with lifetime of the network can be configured in the simulation. Currently, the simulation supports only exponential [16] and Weibull [17] distribution. The cumulative distribution function (cdf), $F(t)$ can be defined as

$$F(t) = \int_0^t f(t)dt$$

where $f(t)$ is the pdf. Reliability function, hazard function and mean time between failures are often used in reliability analysis and can be determined from the pdf and cdf.

The reliability function (or survival function), $R(t)$ is the probability that the SHN survives to time t . It can be defined as

$$R(t) = 1 - F(t).$$

The simulation assumes that there is no failure at the initial time, i.e., $t = 0$, $R(0) = 1$. The hazard function, $h(t)$ is the

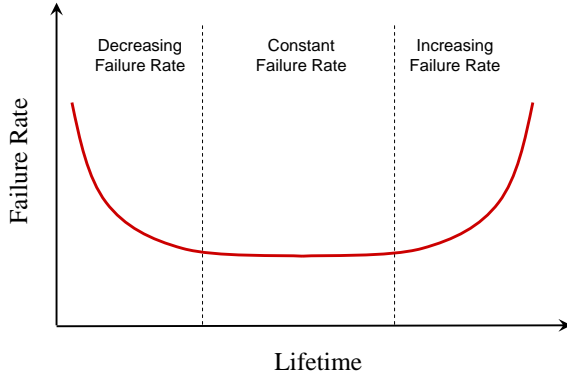


Figure 2. Bathtub Curve

failure rate of the network. The $h(t)$ is defined by

$$h(t) = \frac{f(t)}{R(t)}.$$

The failure rate in practice has a bathtub shape [18]. The hazard function of SHN is also assumed to change as the bathtub curve, which consists of 3 phases: decreasing failure rate (burn in), constant failure rate and increasing failure rate (wearing out) as shown in Fig. 2. Both distributions supported in the simulation are enough to model the bathtub curve. The constant failure rate can be modeled with the exponential distribution, while the decreasing and increasing failure rate can be modeled with the Weibull distribution. The mean time between failure (MTBF) is defined to be the average (or expected) lifetime of the network. The MTBF is defined by

$$MTBF = \int_0^{\infty} R(t)dt.$$

If a failure rate is λ , the pdf of exponential distribution (for $t > 0$) is given by

$$f(t) = \lambda e^{-\lambda t}.$$

The pdf of Weibull distribution is given by

$$f(t) = \beta \alpha^{-\beta} t^{\beta-1} \exp \left[-\left(\frac{t}{\alpha}\right)^{\beta} \right]$$

where α is the scale parameter and β is the shape parameter. The exponential distribution is actually a special case of Weibull distribution where $\beta = 1$ and $\alpha = \frac{1}{\lambda}$. The associate functions of exponential and Weibull distribution can be summarized in Table 1. The Γ denotes the gamma function where $\Gamma(n)$ ¹ is defined as

$$\Gamma(n) = \int_0^{\infty} e^{-x} x^{n-1} dx.$$

¹If n is an integer, then $\Gamma(n) = (n-1)!$.

Table 1. Associate characteristic functions of distributions

Characteristics	Distribution	
	Exponential	Weibull
CDF, F(t)	$1 - e^{-\lambda t}$	$1 - e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$
Reliability Function, R(t)	$e^{-\lambda t}$	$e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$
Hazard Function, h(t)	λ	$\beta \alpha^{-\beta} t^{\beta-1}$
MTBF	$\frac{1}{\lambda}$	$\alpha \Gamma\left(1 + \frac{1}{\beta}\right)$

A more general, 3-parameter form of the Weibull includes an additional waiting time parameter μ (sometimes called a shift or location parameter). The formulas for the 3-parameter Weibull can be easily obtained from these formulas by substituting occurrences of t by $(t - \mu)$.

2.2.2 Simulation Description

The fan-out (k) of a k -ary sibling tree topology can be configured in the simulation. It can be varied from $k=2$ (binary sibling tree) to $k=N-1$ (flat tree), where N is total number of nodes.

The unicast messages are simulated by sending messages from all possible sources (S) to all possible destinations (D), where $S \neq D$. Thus, there are $N \times (N-1)$ simulation cases for normal circumstances. During the failure mode, the failed nodes (F) are obtained from combinations of all possible nodes (N), i.e., $\binom{N}{F}$, where the sources and the destination nodes $\notin F$. Hence, there are $\binom{N-2}{F}$ simulation cases for each unicast transmission.

The total number of simulation cases of unicast message transmission (T_u) for N nodes of the k -ary sibling tree with F failed nodes is given by

$$T_u = N \times (N-1) \times \binom{N-2}{F} = \frac{N!}{(N-F-2)!F!}.$$

The transmission of unicast messages is considered successful if the messages can reach the destination. This means that the network can deliver messages even in the presence of failures in the routing path. If there are S_u success cases, the percent of average number of successes (P_u) is defined by

$$P_u = \left(\frac{S_u}{T_u}\right) \times 100.$$

Due to the fact that the multicast routing protocol is based on multi-destination unicast [3], the percent of average number of successes (P_m) of multicast messages to D destinations in the N nodes of k -ary sibling tree with F failed

Table 2. Percent average number of success for broadcast

Algorithm	T_{bn}	P_{bn}
Basic	$T_{b1} = \frac{2 \times (N-1)!}{(N-F-2)!F!}$	$P_{b1} = \left(\frac{S_{b1}}{T_{b1}}\right) \times 100$
Up-Down	$T_{b2} = \frac{(N-1)!}{(N-F-2)!F!}$	$P_{b2} = \left(\frac{S_{b2}}{T_{b2}}\right) \times 100$
Spanning Tree	$T_{b3} = \frac{N!}{(N-F-2)!F!}$	$P_{b3} = \left(\frac{S_{b3}}{T_{b3}}\right) \times 100$

nodes can be estimated by

$$P_m \approx \left(\frac{P_u}{100}\right)^D \times 100.$$

The broadcast messages are simulated by sending messages from all possible sources (S) to the rest of the tree. During the failure mode, the failed nodes (F) are obtained from combinations of all possible nodes (N), i.e., $\binom{N}{F}$, where the sources $\notin F$. The broadcast operation is considered successful if there are no orphan nodes. An alive node may become an orphan if it becomes unreachable due to network bisectioning. There are three broadcast routing algorithms [2]. The first algorithm (called basic) is done by sending messages to a root of the tree and the root will forward the messages to the rest of the tree. The second algorithm (called up-down) is similar to the first algorithm but it allows a node between a source and a root of the tree to send messages to their children after they send the messages to their parent, i.e., the overhead of sending messages to the root of the tree in the first algorithm is hidden in the second algorithm. The third algorithm is based on a logical spanning tree from the source. The spanning tree creation is based on the breadth first search with a graph coloring technique. If there are S_{b1}, S_{b2} and S_{b3} success cases for the three broadcast algorithms, the percent of average number of success for each algorithm is defined in Table 2.

3 Simulation Results and Analysis

The results were obtained by simulating all possible cases described in the Simulation Description section (2.2.2). We analyzed the effect the dead nodes have on the percent success rate of different networks, as well as the effect the fan-out of the tree and the lifetime distribution have on the network characteristics.

3.1 Effect of dead nodes

Fig. 3 presents the effect of a few dead nodes to the reliability of the network to deliver multicast messages. The

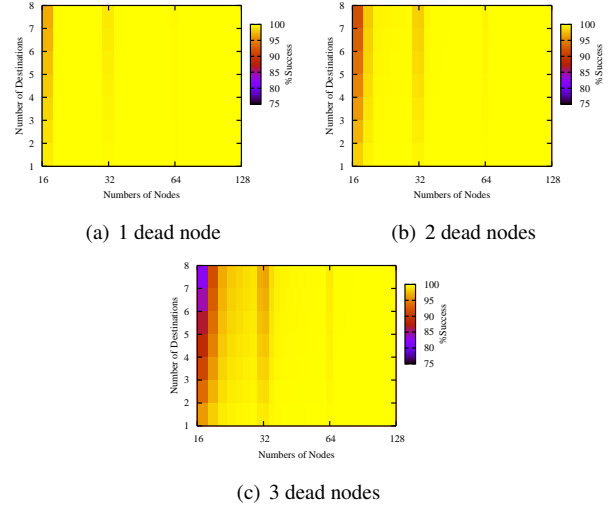


Figure 3. Dead Node Effect on Multicast

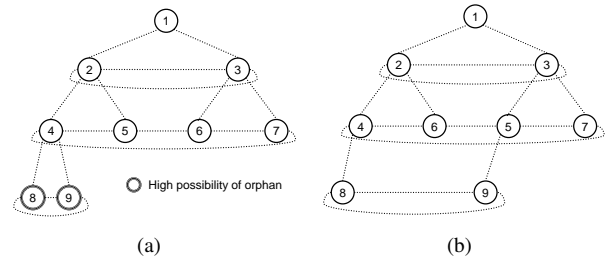


Figure 4. (a) High probability of orphan nodes (b) Rearranged nodes

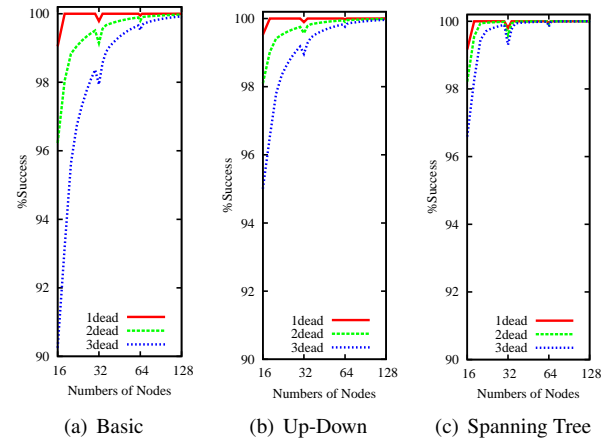


Figure 5. Effect of Dead Nodes on Broadcast

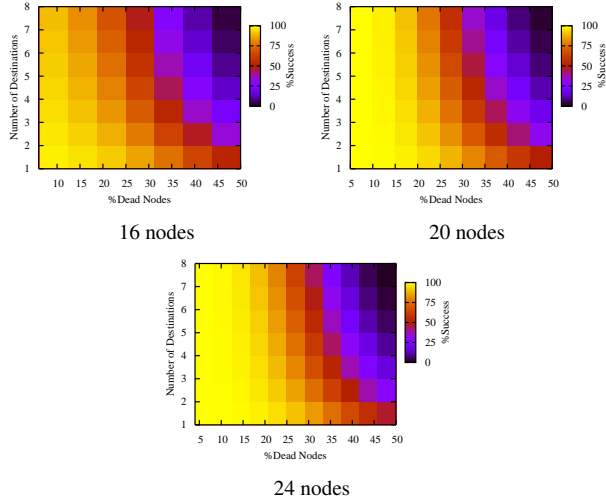


Figure 6. Number of Node Effect on Multicast

unicast is a special case of multicast where the number of destinations is 1. The percent average of success delivery (P_m) is inversely proportional to the number of nodes for every different number of dead node and it is also inversely proportional to the number of destinations of the multicast messages (i.e., $P_m \propto N^{-1} \propto D^{-1}$.) If the number of nodes is 2^k , where $k \in N$, the reliability of the network decreases as the probability of nodes becoming orphans increases. This situation occurs when the number of nodes in the last level (highest depth) of the tree is less than or equal to k , where k is fan-out as shown in figure 4(a).

There are two solutions to solve this drawback. The first solution is to provide the grandparent of the last level with the contact information of the last level in order to be able to recover the tree. The second solution is to re-arrange the nodes in the tree to reduce the possibility of orphan nodes as shown in the Fig 4(b).

Fig. 5 presents the effect of dead nodes to the operations of broadcast message delivery. It shows that the percent average of success delivery (P_b) is inversely proportional to the number of nodes for every different number of dead node (i.e. $P_b \propto N^{-1}$) as well as the effect of high possibility orphan nodes. The figure also shows that the different broadcast routing algorithms also affect the percent average of success. Not only is the spanning tree from source routing algorithm the fastest among them [2], but it is also the most reliable. On the other hand, the basic algorithm is the worst in terms of both efficiency and reliability.

3.2 Effect of number of nodes

Fig. 6 illustrates that although the percentage of dead nodes are the same in different numbers of nodes, the re-

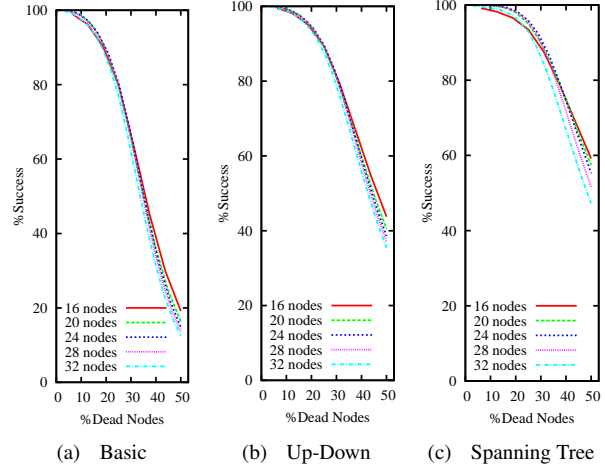


Figure 7. Number of Node Effect on Broadcast

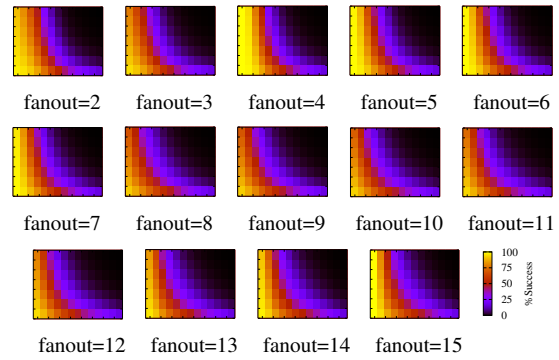


Figure 8. Fanout Effect on Multicast

liability of multicast operations in the self-healing network is significantly different. On the other hand, the reliability of broadcast operations is only marginally different as shown in Fig. 7. This figure also shows that in the spanning tree case, even in the presence of 50% of the dead nodes, up to 60% of broadcast operations survive without any recovery. It illustrates that the k -ary sibling tree is not only suitable in terms of scalability², but also suitable in terms of reliability for supporting parallel runtime environments.

3.3 Effect of fan-out

Fig. 8 illustrates the effect of the number of fanouts on a multicast message operation with 16 nodes. The x-axis for each fanout is percentage of dead nodes (between 6.25% and 87.5%). The y-axis is the number of destinations (between 1 and 8), and the color represents the percent average

²The number of nodes does not reflect the scalability of the SHN. It was limited by simulation time, e.g., the simulation time of 50%dead-32 nodes broadcast (144×10^9 cases) is 3 days on an AMD 2.2 GHz

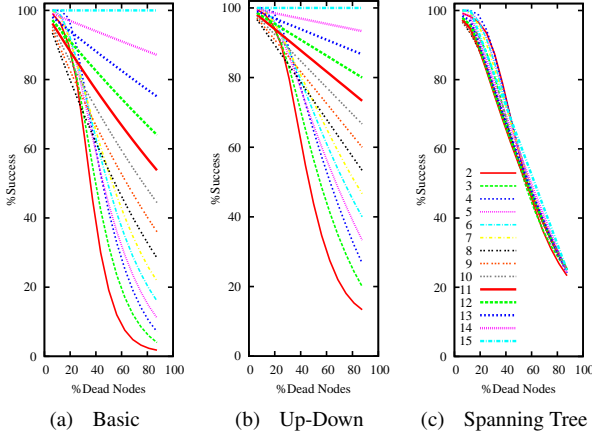


Figure 9. Fanout Effect on Broadcast

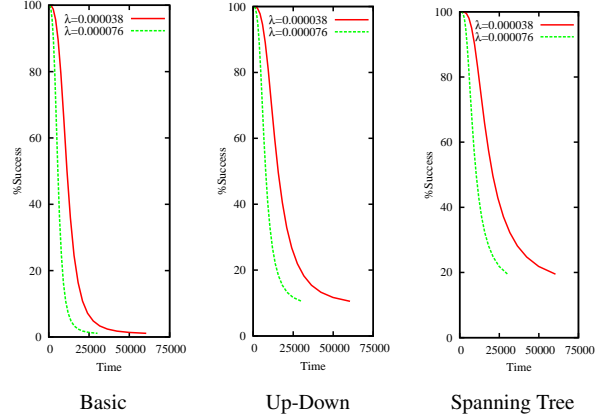


Figure 11. Exponential Dist. on Broadcast

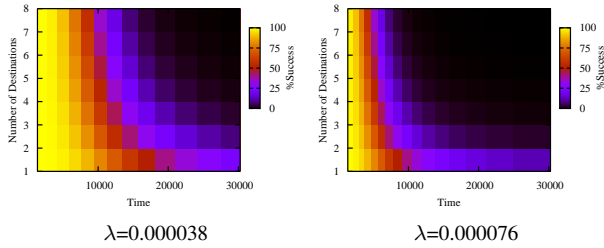


Figure 10. Exponential Dist. on Multicast

Table 3. Exponential parameters

MTBF	26280 hours	13140 hours
$h(t)$	3.8×10^{-5}	7.6×10^{-5}
$R(t)$	$e^{-3.8 \times 10^{-5}t}$	$e^{-7.6 \times 10^{-5}t}$

of success (P_m). For each number of node configuration, the best fanout in terms of reliability is different. For example, in the 16 node configuration, the 4-ary sibling tree is the best. There are two different patterns of results of fanout effected by broadcast message operations as shown in Fig. 9. There are only marginally different reliabilities for each number of fanouts of the spanning tree algorithm, while the reliability of basic and up-down is significant for different fanout numbers. These different reliability properties are caused by the simulation assumption that the source of broadcast and the source and destination of unicast messages are always alive. Thus the root of the tree (node 1) is always alive for basic and up-down, but it is not true for the spanning tree. This implies that if the logical root node can be run on a stable platform, it could significantly improve overall reliability.

3.4 Exponential Lifetime Distribution

The exponential lifetime distribution assumes that the age of network (uptime) has no effect on how likely the network is to fail, i.e., the network is equally likely to fail at any moment during its lifetime, regardless of how old it is. The failure rate (λ) is known to be a constant (the second phase of the bathtub curve).

If we assume that MTBF of the networks are three years (26,280 hours) and 1.5 years (13,140 hours), the hazard and reliability functions can be calculated as shown in Table 3. The percent average number of success with exponential lifetime distribution on multicast and broadcast operations is shown in Fig. 10 and Fig. 11, respectively.

3.5 Weibull Lifetime Distribution

The Weibull lifetime distribution assumes that the hazard function is time dependent. The hazard function is dependent on the value of β as shown in Fig. 12(a).

If $\beta < 1$, the hazard function is the decreasing function (infant mortality or burn in) i.e. the older it is, the less likely it fails (the first phase of the bathtub curve).

If $\beta = 1$, the age has no effect. The lifetime has exponential distribution.

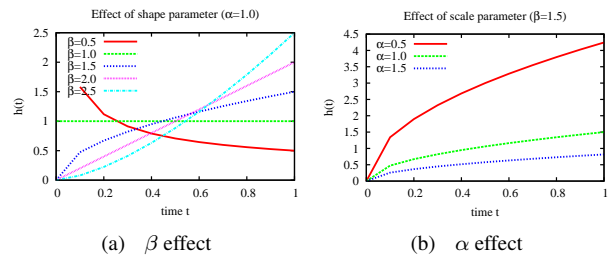


Figure 12. Effects of shape (β) and scale (α)

Table 4. Weibull ($\alpha = 26280$) parameters

β	$h(t)$	$R(t)$
0.5	$0.5 \times 26280^{-0.5} t^{-0.5}$	$e^{-\left(\frac{t}{26280}\right)^{0.5}}$
1.0	3.8×10^{-5}	$e^{-\left(\frac{t}{26280}\right)}$
1.5	$1.5 \times 26280^{-1.5} t^{0.5}$	$e^{-\left(\frac{t}{26280}\right)^{1.5}}$
2.0	$2.0 \times 26280^{-2.0} t$	$e^{-\left(\frac{t}{26280}\right)^{2.0}}$
2.5	$2.5 \times 26280^{-2.5} t^{1.5}$	$e^{-\left(\frac{t}{26280}\right)^{2.5}}$

Table 5. Weibull ($\beta = 1.5$) parameters

α	$h(t)$	$R(t)$
13140	$1.5 \times 13140^{-1.5} t^{0.5}$	$e^{-\left(\frac{t}{13140}\right)^{1.5}}$
39420	$1.5 \times 39420^{-1.5} t^{0.5}$	$e^{-\left(\frac{t}{39420}\right)^{1.5}}$

If $\beta > 1$, the hazard function is the increasing function (wearing out) i.e. the older it is, the more likely it is to fail. It is the third phase of the bathtub curve. If $1 < \beta < 2$, the hazard function is concave (increasing at a decreasing rate). On the other hands, the hazard function is convex (increasing at an increasing rate), if $\beta > 2$

Fig. 13 and Fig. 14 illustrate the effect of β parameters in the Weibull lifetime distribution to the percent average of success of multicast and broadcast operations. It assumes $\alpha = 26, 280$ and various β values as shown in Table 4.

Fig. 12(b) shows the effects of the characteristic life (α) on the failure rate, which affects the spread (scale) of the distribution. Fig. 15 and Fig. 16 illustrate the effect of α parameters in the Weibull lifetime distribution to the percent average of success of multicast and broadcast operations. It assumes $\beta = 1.5$ and α values as shown in Table 5.

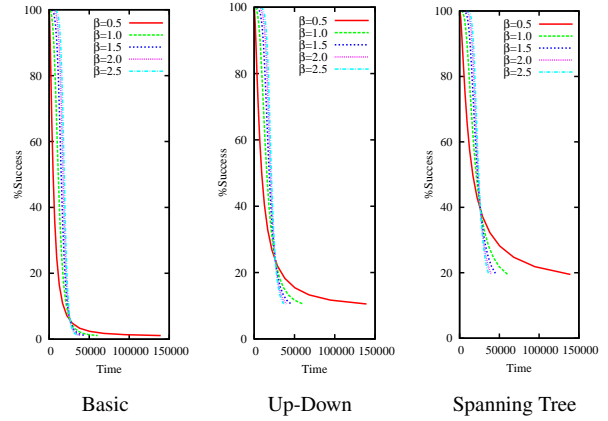


Figure 14. Weibull on Broadcast ($\alpha=26280$)

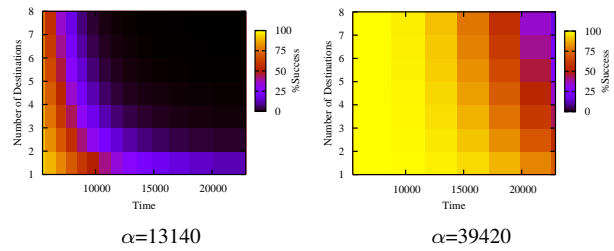


Figure 15. Weibull Dist. on Multicast ($\beta=1.5$)

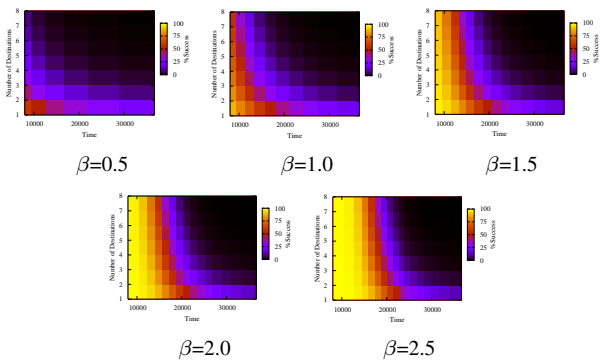


Figure 13. Weibull on Multicast ($\alpha=26280$)

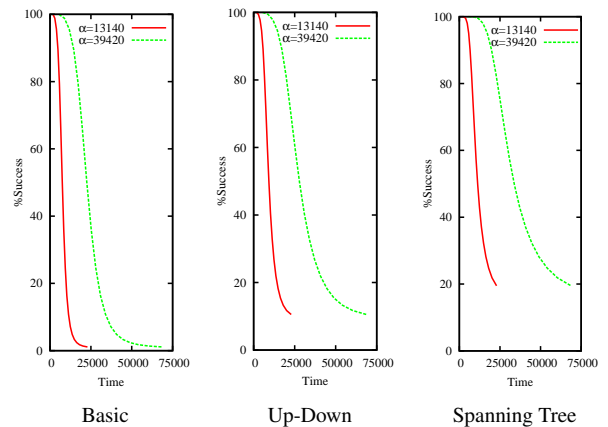


Figure 16. Weibull Dist. on Broadcast ($\beta=1.5$)

4 Conclusions and Future Work

This paper presents reliability analysis of a self-healing network developed for scalable and fault-tolerant, parallel runtime environments. The network is designed to support the transmission of messages across multiple nodes while it also protects against node and process failures. The results show that the spanning tree broadcast algorithm is the most reliable among several broadcast algorithms.

There are several improvements that we plan for the near future. The complexity of simulation need to be reduced. Expanding the simulation to support other lifetime distributions, such as Lognormal, can improve the accuracy of the analysis. Additionally, we plan to run the self-healing network on large-scale platforms and obtain the actual failure data to compare the empirical and theoretical distributions.

Acknowledgment This material is based upon work supported by “Los Alamos Computer Science Institute (LACSI)”, funded by Rice University Subcontract No. R7B127 under Regents of the University Subcontract No. 12783-001-05 49

References

- [1] Jack J. Dongarra, Hans Meuer, and Erich Strohmaier. TOP500 supercomputer sites. *Supercomputer*, 13(1):89–120, 1997.
- [2] Thara Angskun, Graham E. Fagg, George Bosilca, Jelena Pjesivac-Grbovic, and Jack J. Dongarra. Self-healing network for scalable fault tolerant runtime environments. In *Proceedings of 6th Austrian-Hungarian workshop on distributed and parallel systems*, Innsbruck, Austria, 2006. Springer-Verlag.
- [3] Thara Angskun, Graham E. Fagg, George Bosilca, Jelena Pjesivac-Grbovic, and Jack Dongarra. Scalable fault tolerant protocol for parallel runtime environments. In *Recent Advances in PVM and MPI*, number 4192 in LNCS, pages 141–149. Springer, 2006.
- [4] Kishor S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. John Wiley & Sons, Inc., New York, 2002.
- [5] Robin A. Sahner, Kishor S. Trivedi, and Antonio Puliafito. *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*. Kluwer Academic Publishers, New York, 1996.
- [6] Wendai Wang, James M. Loman, Robert G. Arno, Pantelis Vassiliou, Edward R. Furlong, and Doug Ogden. Reliability block diagram simulation techniques applied to the IEEE std. 493 standard network. *IEEE Transactions on Industry Applications*, 40(3):887–895, 2004.
- [7] Joanne Bechta Dugan. Fault trees and imperfect coverage. *IEEE Transactions on Reliability*, 38(2):177–185, 1989.
- [8] Robin A. Sahner and Kishor S. Trivedi. Modeling with extended fault trees. In *Fifth IEEE International Symposium on High Assurance Systems Engineering*, pages 238–246, Albuquerque, NM, US, 2000.
- [9] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*. Springer-Verlag, 1999.
- [10] Robin A. Sahner and Kishor S. Trivedi. A hierarchical, combinatorial-markov model of solving complex reliability models. In *ACM '86: Proceedings of 1986 ACM Fall joint computer conference*, pages 817–825, Dallas, Texas, US, 1986. IEEE Press.
- [11] Boudewijn R. Haverkort and Kishor S. Trivedi. Specification and generation of markov reward models. *Discrete-Event Dynamic Systems: Theory and Applications*, 3:219–247, 1993.
- [12] Manish Malhotra and Kishor S. Trivesi. Dependability modeling using petri nets. *IEEE Trans. on Reliability*, 44(3):428–440, 1995.
- [13] Antonio Puliafito, Miklos Telek, and Kishor S. Trivedi. The evolution of stochastic petri nets. In *World Congress on Symtems Simulation (WCSS'97)*, Singapore, 1997.
- [14] Lawrence M. Leemis and Stephen K. Park. *Discrete-Event Simulation: A First Course*. Prentice Hall, 2005.
- [15] Graham E. Fagg, Edgar Gabriel, George Bosilca, Thara Angskun, Zizhong Chen, Jelena Pjesivac-Grbovic, Kevin London, and Jack Dongarra. Extending the mpi specification for process fault tolerance on high performance computing systems. In *Proceedings of the International Supercomputer Conference (ISC) 2004*, Heidelberg, Germany, 2006. Primeur.
- [16] Narayanaswamy Balakrishnan and Asit P. Basu. *The Exponential Distribution: Theory, Methods and Applications*. Gordon and Breach Publishers, 1995.
- [17] Alan Plait. The Weibull distribution with tables. *Industrial Quality Control*, 19(5):17–26, 1962.
- [18] Ming Xie and Chin Diew Lai. Reliability analysis using an additve Weibull model with bathtub-shaped failure rate function. *Reliability Engineering and System Safety*, 52(1):87–93, 1995.