Accelerated Reliability Analysis for Self-Healing SONET Networks *

Hakki C. Cankaya, V. S. S. Nair
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX. 75275
email : {candan,nair}@seas.smu.edu

Abstract

Recently, a parametric State Reward Markov Model (SRMM/p) has been developed for the reliability and availability analysis of self-healing SONET mesh networks [2]. In this paper, we investigate the factors that affect the run-time complexity of the model presented in [2]. In order to accelerate the reliability and availability analysis, we present an approach that aggregates a set of states in the model based on 2-phase hypoexponential distribution. A comparison of the original and the reduced model, with respect to run-time complexity and accuracy, is carried out by applying the models for the analysis of few complex networks.

1 Introduction

As telecommunication networks become faster and more critical to social life as well as to business, there has been an increased demand for higher reliability and availability. In order to meet the reliability requirements, service providers use a number of techniques including fault avoidance, fault removal, fault-tolerance, and fault-forecasting and combinations there of [9]. In this context, the objective of survivability analysis is to provide quantitative measures for the network's capability to tolerate failures and to provide continuous service.

Markov models are being widely used for reliability modeling of complex systems as they can capture the probabilistic behavior of the system with attention to the design details. Particularly, Markov models allow coverage to be included in the evaluation process. This feature becomes important for fault-tolerant systems where coverage factor plays a critical role [8]. In a previous study, we have introduced a special type of Markov model, called parametric State Reward Markov Model (SRMM/p), for the reliability and availability analysis of self-healing SONET mesh networks [2]. One limitation of the model was the high run-time complexity. One observes that this is contributed mainly by large disparity between transition rates among various states in the model. In this paper, we present an approach to circumvent this problem by aggregating some of the states.

Section 2 briefly presents the SRMM/p model and discusses the factors that affect the run-time complexity of the model. In section 3, we present a methodology to accelerate analysis using a reduced model. We give some experimental results in section 4 and conclude the paper in section 5.

The notation used in the paper is given in the following:

* This research has been partly supported by Nortel Inc.

Notation

$M(l,h,m)$	model with parameters
l	model parameter representing the number of stages
h	model parameter representing the performance threshold
m	model parameter representing the number of partially functioning states
W	set of functioning states
Q	set of restoration states
F	set of failure states
λ	failure rate
Θ	average restoration rate
μ	repair rate
c	coverage for successful recovery
e	number of links in the network
Si	full functioning state at stage i in the model
$Ki.j$	j^{th} level partial functioning state at stage i in the model
Fi	failure state at stage i in the model
QSi	restoration state at stage i for the failure that occurred at full functioning state $S(i-1)$
$QKi.j$	restoration state at stage i for the failure that occurred at state $K(i-1).j$
$s(l,m)$	function of the number of states in original model $M(l,h,m)$
$s'(l,m)$	function of the number of states in reduced model $M(l,h,m)$
$g(\Theta,\lambda,\mu)$	step-size function for the analysis of the original model
$g'(\Theta,\lambda,\mu)$	step-size function for the analysis of the reduced model

2 The State Reward Markov Model

For the reliability and availability analysis of self-healing SONET mesh networks, we have introduced the paramet-

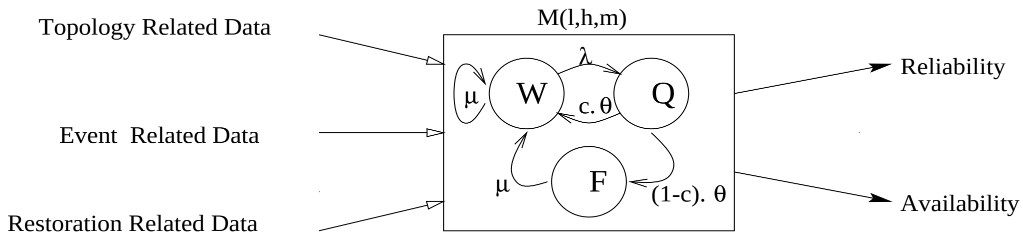

Figure 1: White-box Representation of the Model

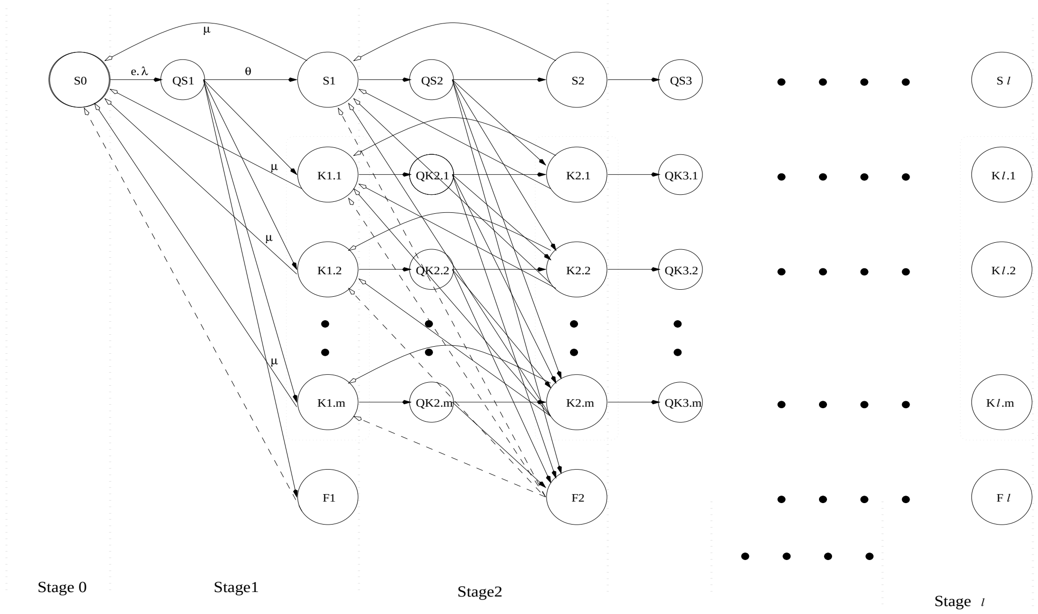

Figure 2: The Multi Stage Parametric State Reward Markov Model

ric State Reward Markov Model, denoted by SRMM/p and illustrated in Figure 1 [2]. Model parameters, which are given by the user, determine the size and the output accuracy of the model. Other than the model parameters, the user needs to provide a set of input data. This input data set includes information about approximate network topology, failure and repair events, and restoration capability which altogether determine the goodness of the network [4]. The model has functioning states (denoted by W), restoration states (denoted by Q), and failure states (denoted by F). The rate of transitions from functioning states to restoration states are given by failure rate (denoted by λ). Restoration rates (denoted by θ) with the coverage (denoted by c) for successful recovery designate the rate of transitions from restoration states to either working or failure states. The rate of transitions from failure states to functioning states and amongst the different levels of functioning states is given by the repair rate (denoted by μ). A symbolic representation, $M(l, h, m)$, is used for describing the model with its parameters; where "l" denotes the number of consecutive link failures that the user wants the model to consider. More than "l" number of failures puts the system in a failure state automatically, even though it might be recoverable. Since varying levels of the system performance are considered by the model, "h" represents the lower bound of the functioning performance (threshold) below which the system is considered malfunctioning. Parameter "m" represents the number of performance levels considered between full functioning level and threshold level. These performance levels are represented by partially functioning states in the model. The solution of the model is used to generate the numerical values of the survivability metrics: reliability and availability [3]. The comprehensive model showing all the states and transitions is given in Figure 2 [2].

2.1 Run-time complexity of the model

For the numerical solution of the transient behavior of the SRMM/p, we use Runge-Kutta method with adaptive step-size control [13]. In adaptive step-size control, the algorithm observes its progress at each step and adjusts the step-size according to the accuracy required. It increases the step-size as far as the predetermined accuracy allows and completes the integration with fewer number of steps than it would without any increase in step-size. Obviously, this adaptive step-size control feature results in faster integration for the required integration duration.

In SRMM/p, transition rates determine the transient behavior of the system. As the ratios between these rates change drastically, the transient status of the system alters faster. This activity affects the Kolmogorov-equations by decreasing the iteration interval $(t + \Delta t)$ or, in other words, increases the rate of change in the system. Therefore, it diminishes the potential gain in run-time through the adaptive step-size control of the method. Figure 3 depicts the relationship between logarithmic proportions of transition rates versus the number of steps to integrate the given duration. From the figure, we can see that when the logarithmic proportion of transition rates exceeds a certain value, the number of steps increases drastically. Since the number of steps is one of the main factors determining the run-time, we see the similar relationship between logarithmic proportions of transition rates and run-time as shown in Figure 4.

The number of states is a function of parameter "l" and parameter "m" in the M(l, h, m) representation, and is denoted by $s(l, m)$. Parameter "l" represents the number of

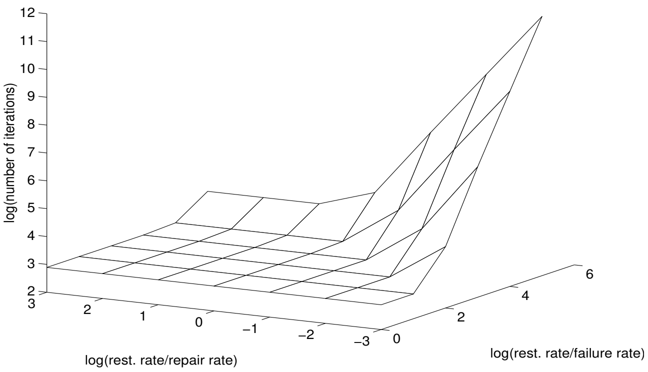

Figure 3: The effect of transition ratios on number of iterations

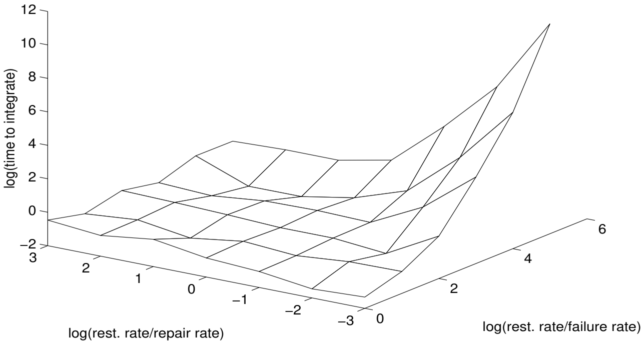

Figure 4: The effect of transition ratios on run-time

stages in the model each of which accommodates a consecutive failure. Parameter "m" represents the number of partially functioning states in each stage. It can be shown that

$$s(l, m) = m(2l - 1) + 3l + 4. \quad (1)$$

A pictorial representation of $s(l, m)$ is given in Figure 5. Since the number of states in the model directly determines number of equations, it constitutes one of the factors which affect the run-time.

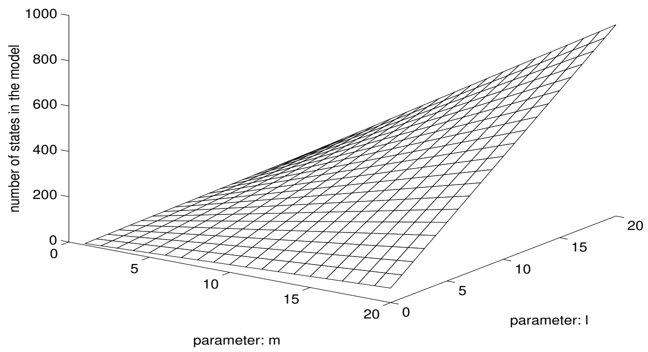

Figure 5: The effect of model parameters on the number of states

In today's standard high speed telecommunication networks, the logarithmic ratio between average restoration rate "$\hat{\Theta}$" and failure rate "λ" is determined to be high [10] [12] [11].

$$\log \frac{\hat{\Theta}}{\lambda} \geq 5 \quad (2)$$

Moreover, as the technology progresses towards decreasing the failure rate and shortening the restoration time, this ratio will tend to be bigger and result in longer run-time. In addition to that, we have a similar high logarithmic ratio between average restoration rate "$\hat{\Theta}$" and repair rate "μ" [10] [12] [11].

$$\log \frac{\hat{\Theta}}{\mu} \geq 3 \quad (3)$$

However, at least in this case the technology aims to make the ratio smaller by trying to develop fast repair activities. Therefore, the primary objective here is to alleviate the effect of these high ratios of transition rates by aggregating some of the states in the model. This will not only bring down the ratios of transition rates but also reduce the number of states in the model. Of course some penalty need to be paid in terms of loss in accuracy. However, we show in the paper that this complexity reduction can be achieved with minimal loss in accuracy.

3 Accelerated Analysis

We have investigated few approaches that might help to accelerate the analysis of the model in terms of run-time. For example, decomposition has always been one of the most efficient approaches used to deal with the complexity of any kind. In this technique, the system is dissected into its constituents each of which can be analyzed by a separate method in an easier way than the whole and the results of each constituent are combined to get the evaluation of the entire system [5] [15] [17] [6]. However, not every system can be dissected easily. Even though it could be possible, the behavior of the constituents might be totally different in isolation than in the system. If all these conditions are satisfied in a system then it is called a completely decomposable system. However, there are still some systems which fulfill the conditions partially and give satisfactory approximations to the real results with an acceptable level of accuracy. These systems are called nearly decomposable systems and encountered more frequently than the completely decomposable systems in real life. Behavioral decomposition in Markov models is a near-complete decomposition. This decomposition technique is based on dissecting the Markov model with respect to the relative magnitude of the state-transition rates. All fast transitions and their related states are isolated and analyzed for steady-state behavior by assuming that these states reach equilibrium condition faster in comparison to the slow transitions. The results of this analysis can be used in an approximate behavioral (transient or steady-state) analysis of the rest of the model.

In our case, due to the nature of the SRMM/p, transitions flow in an alternating sequence of functional and restoration states with failure and restoration rates, respectively. As we mentioned previously and showed in (2) and (3), the average restoration rate "$\hat{\Theta}$" is much bigger when compared to failure rate "λ", and repair rate "μ". Therefore, we cannot group a set of states in the SRMM/p where intra-group transitions are far different in magnitude than inter-group transitions. This hinders us to use the behavioral decomposition towards the reduction of run-time complexity in our model.

Another approach to alleviate the problem of long run-time is to use a technique called importance sampling [7]. In this approach, failure events are accelerated by changing the probability measures; however, the output values need to be adjusted accordingly by using heuristic methods [14]. Choosing a good heuristic is difficult for specific systems [1]. Further, even though the approach accelerates the run-time, it needs more complex analysis of the results due to the correlations induced [16].

The concept of aggregation in Markov models has been in use for couple of decades. In this approach, states of the model are fused in order to decrease the model size in number of states and transitions. In our case, possible aggregation of working (W) and restoration (Q) states and their corresponding transitions might help to reduce the ratio of different transition rates in the model. Both failure-time and restoration-time distributions are exponentially distributed in the model and there is only one transition, denoted by "λ" in Figure 6, from a working state to a restoration state. These features of the model allow us to aggregate the working and restoration states by using 2-phase hypoexponential distribution [18]. After the aggregation process, the working and restoration states are reduced into a new state, named "C" in Figure 7, with a new transition whose average rate is "α", where $\alpha = \Theta + \lambda$.

Figure 8 shows the reduced version of the original model, which was given in Figure 2, after the application of the state aggregation to the entire SRMM/p. The reduced model simplifies the run-time complexity from two different points of view. One is the reduction in the ratio between transition rates in the model. The other is the reduction in number

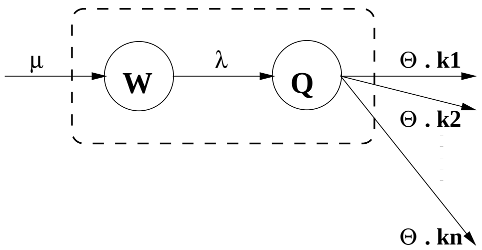

Figure 6: Aggregation of working and restoration states

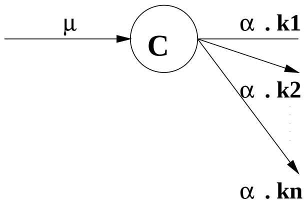

Figure 7: Aggregated state with altered rates

of states in the model. We define two functions "g" and "g'" to quantify the effect of transition rates on the step-size for the original and the reduced models, respectively. These are functions of the transition rates. The higher values of both functions cause longer run-time by increasing the number of steps in the integration of both the original and the reduced models.

$$g(\Theta, \lambda, \mu) = max(|\log \frac{\Theta}{\lambda}|, |\log \frac{\Theta}{\mu}|, |\log \frac{\mu}{\lambda}|) \qquad (4)$$

$$g'(\Theta, \lambda, \mu) = |\log \frac{\Theta + \lambda}{\mu}| \qquad (5)$$

We can observe the gain in run-time complexity through the model reduction by comparing the Figure 9 and 10 which depict the "g" and "g'" for both models. Obviously, the reduced model is less sensitive to the rate of change in logarithmic ratios of the transition rates than the original model. This is due to the fact that the function for the reduced model does not have the highest ratio which is between failure and the restoration rates.

The other simplification obtained by the model reduction is the decrease in number of states in the model. So the differential equation system has fewer number of unknowns to integrate which results in faster run-time. Functions $s(m, l)$ and $s'(m, l)$ give the number of states in the original and the reduced model in terms of model parameters, respectively.

$$s(m, l) = m(2l - 1) + 3l + 4 \qquad (6)$$

$$s'(m, l) = ml + 3l + 1 \qquad (7)$$

4 Experimental Results

In this section, we present experimental results to support our claim on the complexity reduction achieved through the model reduction. In order to answer these questions, we conduct experiments on two sample networks. The first sample network is based on a real LATA which is known as the

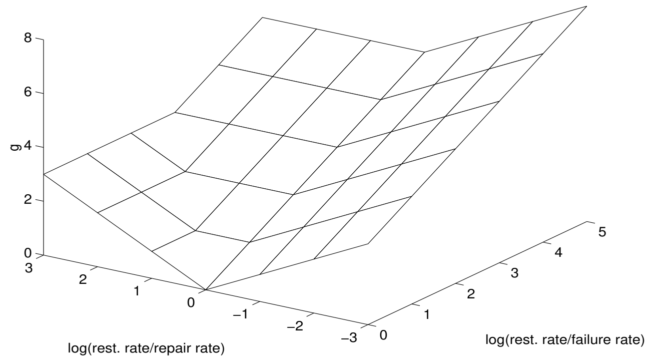

Figure 9: $g(\Theta, \lambda, \mu)$ for the original model

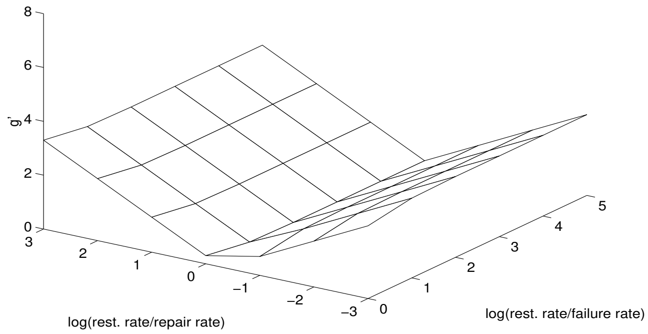

Figure 10: $g'(\Theta, \lambda, \mu)$ for the reduced model

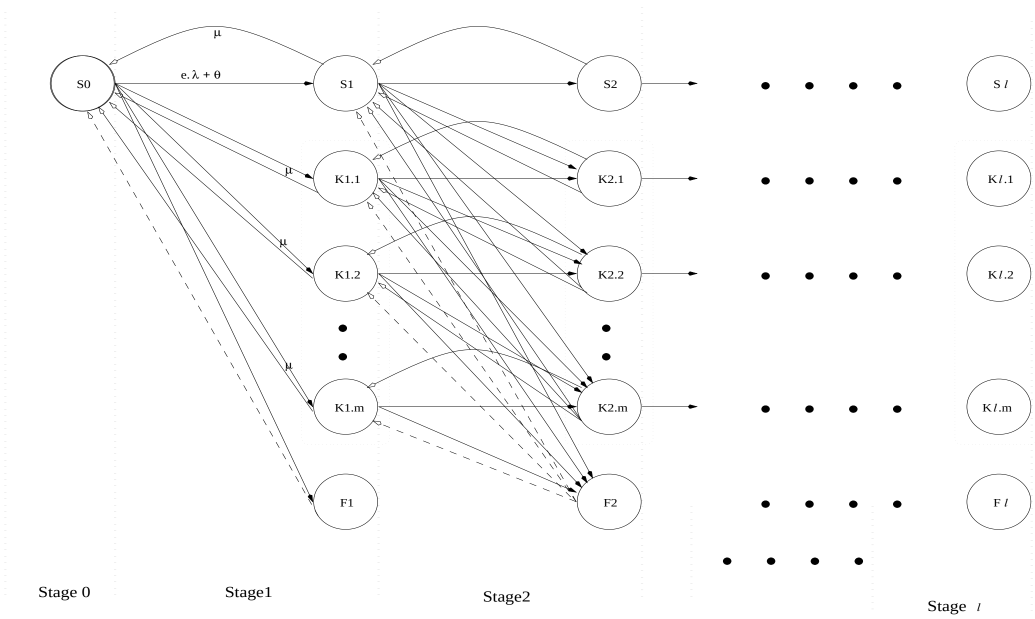

Figure 8: The Reduced Model

the "New Jersey" network [4] [19]. Figure 11 illustrates this sample network with two numbers associated with each link, representing the working capacity and the spare capacity in terms of STS-1. The second sample network is known as the "US" network which is illustrated in Figure 12 [4].

For both experiments, we used the model "M(3,0.5,4)" as shown in Figure 13 and considered consecutive link failures. With the proposed approximation, the original model is reduced into the one shown in Figure 14.

After running both experiments with original and reduced models, we tabulate the results in Table 1 and Table 2. Each table gives the number of iteration steps versus simulated analysis period for the numerical solution of both the original and the reduced models. Since the number of steps for original and reduced models drastically diverge as the simulated analysis period increases in both experiments, it becomes obvious that the model reduction approximation is even more beneficial for longer analysis in terms of number of steps in integration.

The gain in number of steps through the reduced model reflects equally on the run-time. Table 3 and Table 4 tabulates the actual run-times for both experiments. From the numbers, it is obvious that the approach accelerates the model in terms of run-time.

On the other hand, if we look at what we pay for the benefits of the approach in terms of the accuracy of the results, fortunately we do not see as much difference in survivability results as we do in run-time. Moreover, the results from the original and the reduced models do not diverge. As we can see in Table 5, availability differs only $\%2.2 * 10^{-6}$ between models for the "New Jersey" network. The same table shows similar close relationship in availability for the "US" network. Furthermore, time dependent results of both models in each experiment are really close to each other. For example, Figure 15 illustrates the non-divergent reliability results of both models for the "New Jersey" network. Figure 16 gives the similar results in reliability for the "US"

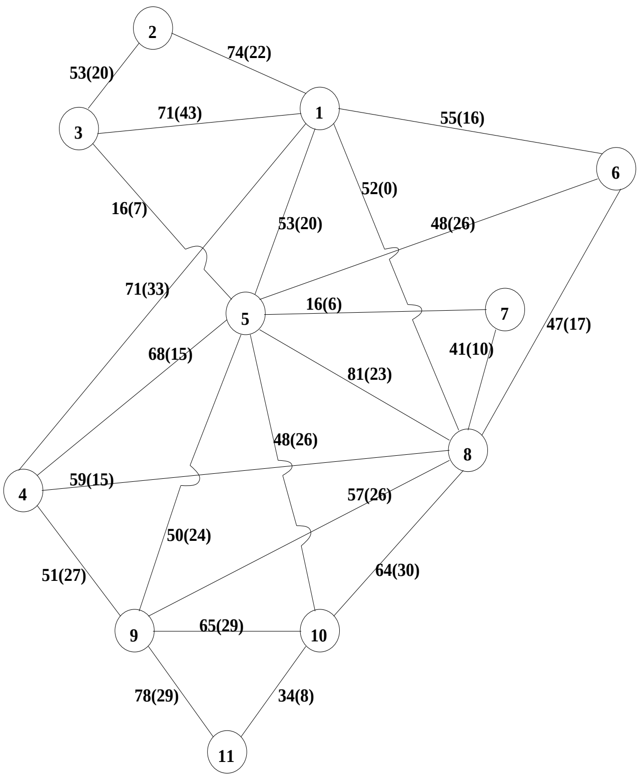

Figure 11: The New Jersey network

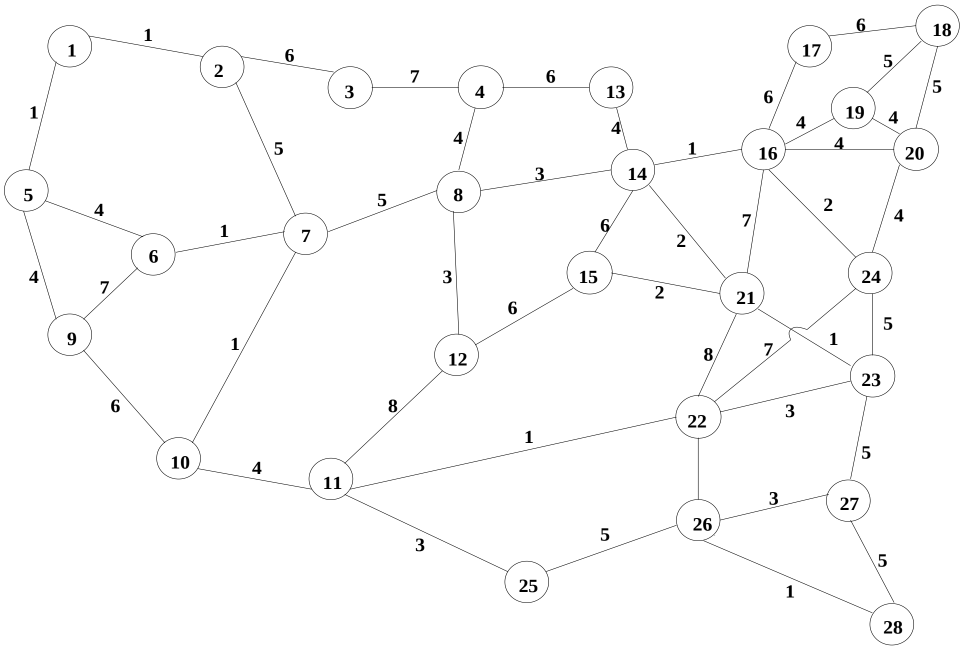

Figure 12: The US network

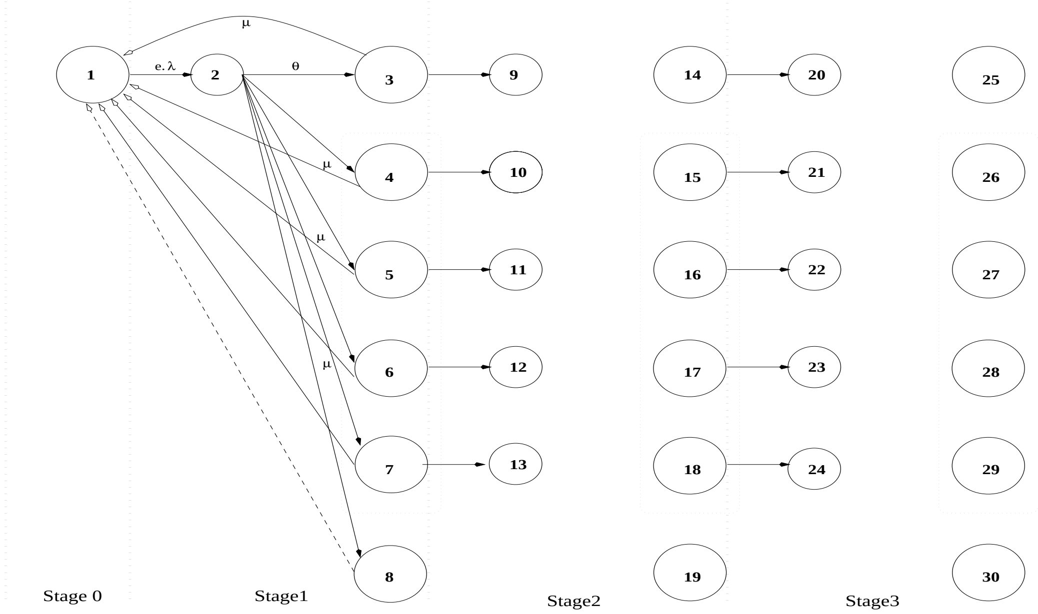

Figure 13: Original M(3,0.5,4)

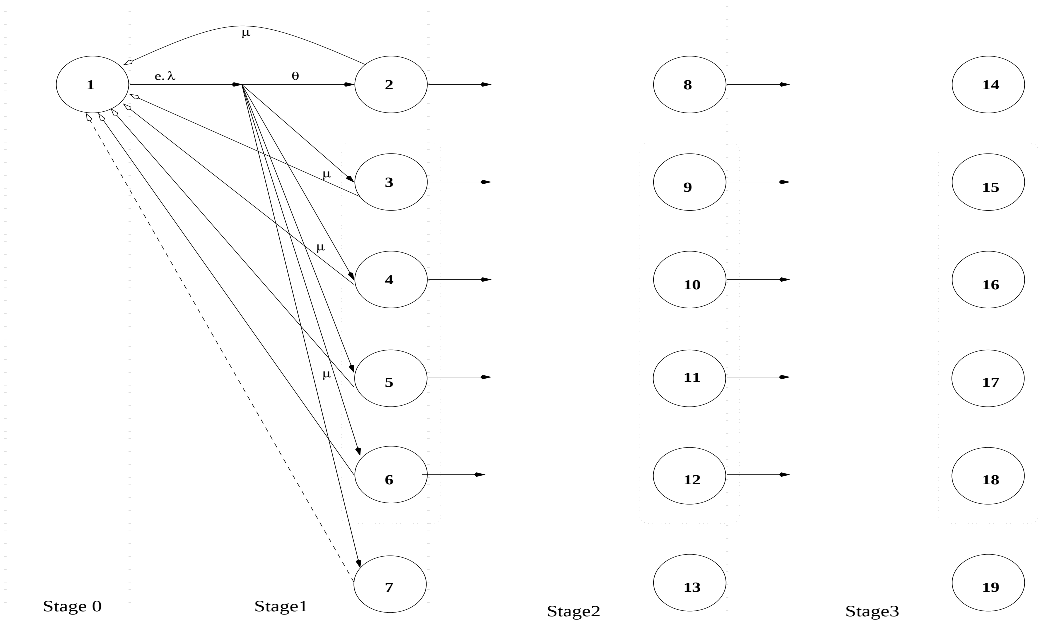

Figure 14: Reduced M(3,0.5,4)

Table 1: Number of steps for the "New Jersey" network

Simulated analysis period (hour)	Number of steps	
	Original Model (30 State)	Reduced Model (19 State)
1	1295	18
2	2586	19
3	3878	22
4	5168	24
5	6460	25
6	7752	26
7	9043	27
8	10335	28
9	11626	29
10	12917	32
20	25831	72
50	77485	183
100	129141	368

Table 2: Number of steps for the "US" network

Simulated analysis period (hour)	Number of steps	
	Original Model (30 State)	Reduced Model (19 State)
1	821	18
2	1636	21
3	2452	25
4	3267	27
5	4082	29
6	4897	31
7	5714	32
8	6530	33
9	7346	34
10	8162	34
20	16320	63
50	40791	152
100	68014	301

Table 3: Run-Time for the “New Jersey” network

Simulated analysis period (hour)	Run-Time (hr:min:sec)	
	Original Model (30 State)	Reduced Model (19 State)
1	00:03:49	00:00:00.23
2	00:08:56	00:00:00.21
3	00:15:27	00:00:00.24
4	00:23:10	00:00:00.26
5	00:32:25	00:00:00.28
6	00:51:64	00:00:00.29
7	00:57:11	00:00:00.30
8	01:16:21	00:00:00.58
9	01:26:40	00:00:00.45
10	01:58:19	00:00:00.70
20	02:48:32	00:00:04
50	06:25:11	00:00:08
100	46:38:46	00:00:19

Table 4: Run-Time for the “US” network

Simulated analysis period (hour)	Run-Time (hr:min:sec)	
	Original Model (30 State)	Reduced Model (19 State)
1	00:00:42	00:00:00.21
2	00:01:42	00:00:00.23
3	00:03:05	00:00:00.28
4	00:04:51	00:00:00.30
5	00:07:13	00:00:00.32
6	00:09:35	00:00:00.35
7	00:12:30	00:00:00.52
8	00:16:25	00:00:00.61
9	00:20:27	00:00:00.71
10	00:24:11	00:00:00.57
20	00:38:03	00:00:03
50	01:42:23	00:00:07
100	12:15:22	00:00:17

Table 5: Availability for the “New Jersey” and the “US” network

Experimental Networks	Availability	
	Original Model (30 State)	Reduced Model (19 State)
“New Jersey”	0.999833389953	0.999833410915
“US”	0.9843426772	0.9843461500

network. If we sum up all the pros and cons of the new model-reduction approximation, we conclude that it does not only simplify the run-time complexity in a very considerable amount, but it is also a close enough approximation.

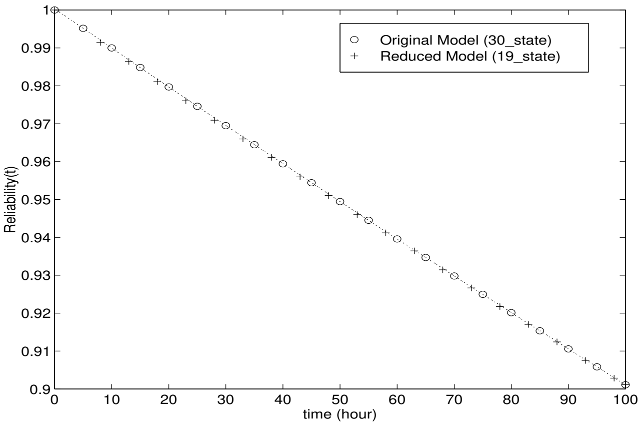

Figure 15: Reliability of the “New Jersey” network

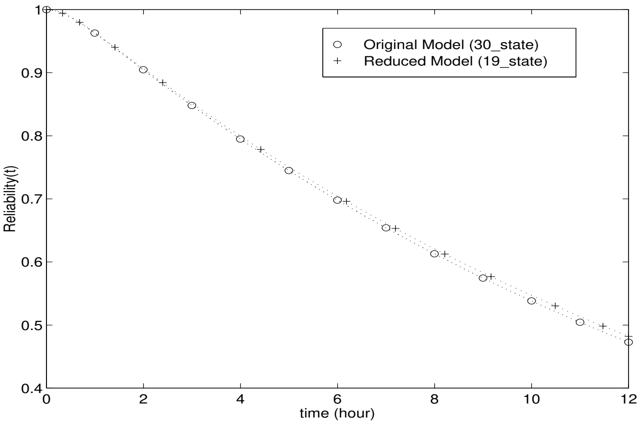

Figure 16: Reliability of the “US” network

5 Conclusion

In this paper, we investigated the run-time complexity of the parametric State Reward Markov Model (SRMM/p) that we have proposed for the reliability analysis of self-healing SONET mesh networks. With the observation that the disparity in the transition rates among various states in the model is the main contributing factor to the run-time complexity, we developed a solution based on state aggregation. This approach considerably reduces the run-time by decreasing the number of iterations and equations in the behavioral analysis of the underlying Markov model (SRMM/p). These claims were verified by experimental results. In the examples we chose, the run-time complexity drops by the order of thousands while the accuracy loss is only of the order of 10^{-6}.

References

[1] Bratley, P. *A Guide to Simulation.* Springer-Verlag, 1983.

[2] Cankaya, H. C., and Nair, V. S. S. Reliability and availability evaluation of self-healing SONET mesh networks. In *IEEE Proceedings of GLOBECOM'97* (1997), vol. 1, IEEE Com. Soc., pp. 252–256.

[3] Cankaya, H. C., and Nair, V. S. S. Survivability metrics for self-healing SONET mesh networks. In *Proceedings of ISCIS'XII* (1997), vol. 1, Bogazici University, Bogazici University Press, pp. 269–276.

[4] Chow, C. E., Bicknell, J., and Syed, S. Performance analysis of fast link restoration algorithms. *Journal of Communication Systems 8* (1995), 325–345.

[5] Courtois, P. J. *Decomposability: Queuing and Computer System Applications*, 1 ed., vol. 1. Academic Press, 1977.

[6] Geist, R., and Trivedi, K. S. Reliability estimation of fault-tolerant systems: Tools and techniques. *IEEE Computer* (July 1990), 52–61.

[7] Juneja, S., and Shahabuddin, P. Fast simulation of markovian reliability/availability models with general repair policies. In *Proceedings of the Twenty-Second Int'l Symp. on Fault Tolerant Computing* (1992), IEEE Press, pp. 150–159.

[8] Lee, D., and Abraham, J. A numerical technique for the hierarchical evaluation of large, closed fault-tolerant systems. In *Proc. of Second Int. Working Conference on Dependable Computing for Critical Applications* (Tucson, Arizona, 1991), pp. 1–8.

[9] Nieuwenhuis, L. J. M., Misser, H. S., Hawker, I., Donachie, S. S., Ravera, M., Balzaretti, S., and Ralenius, T. Reliability engineering for future telecommincation networks and services. In *IEEE Proceedings of GLOBECOM'93* (1993), IEEE, pp. 686–691.

[10] on Network Survivability Performance, W. G. A Technical Report on Network Survivability Performance. Tech. Rep. T1A1.2/93-001R3, Technical Committee T1, 1993.

[11] on Network Survivability Performance, W. G. A Technical Report on Enhanced Analysis of FCC-Reportable Service Outage Data. Tech. Rep. T1A1.2/95-001R4, Technical Committee T1, 1995.

[12] on Network Survivability Performance, W. G. Draft Supplement to Technical Report No.24 on Network Survivability Performance. Tech. Rep. T1A1.2/97-001R1, Technical Committee T1, 1997.

[13] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. *Numerical Recipes: The Art of Scientific Computing*, 1 ed., vol. 1. Cambridge University Press, 1986.

[14] Pritsker, A. A. *Intoduction to Simulation and SLAM II.* John Wiley & Sons, 1986.

[15] Rieders, M. *State Space Decomposition for Large Markov Chains*, 1 ed., vol. 1. 1996, ch. 33, pp. 587–590.

[16] Ripley, B. D. *Stochastic Simulation.* John Wiley & Sons, 1987.

[17] Sahner, R. A., Trivedi, K. S., and Puliafito, A. *Performance and Reliability Analysis of Computer Systems*, first ed. Kluwer Academic Publishers, 1996.

[18] Trivedi, K. S. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, third ed. Prentice-Hall, 1994.

[19] Yang, C. H., and Hasegawa, S. FITNESS: failure immunization technology for network service survivability. In *IEEE Proceedings of GLOBECOM'88* (November 1988), pp. 1549–1554.