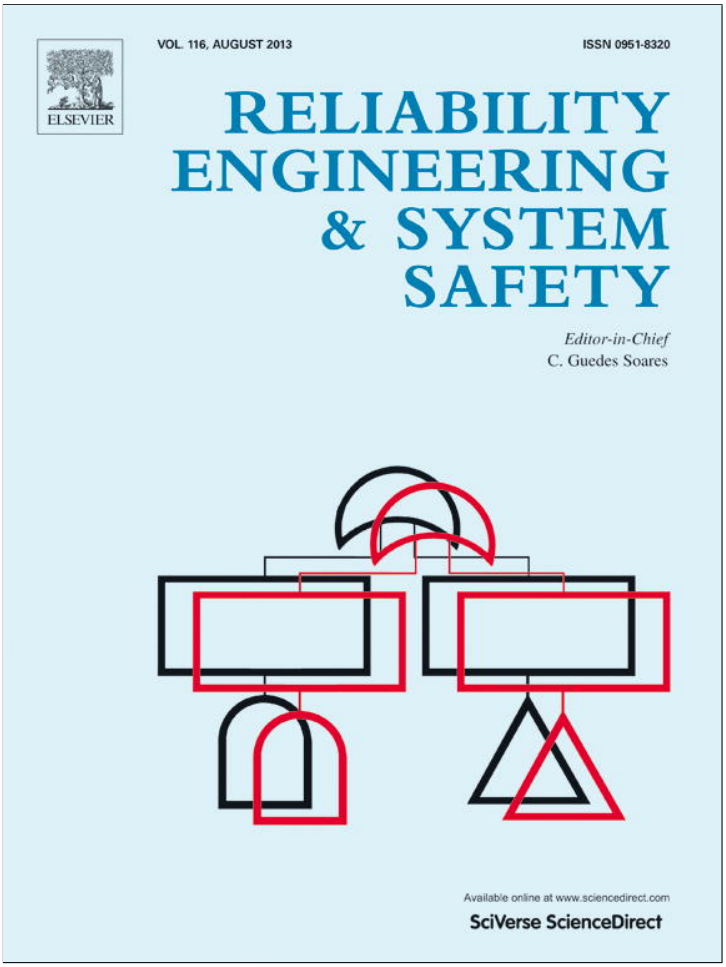

Reliability Engineering and System Safety 116 (2013) 109–125

Contents lists available at SciVerse ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

System resiliency quantification using non-state-space and state-space analytic models ☆

Rahul Ghosh [a,*], DongSeong Kim [b], Kishor S. Trivedi [c]

[a] IBM Software Group, Durham, NC 27703, USA
[b] Department of Computer Science and Software Engineering, University of Canterbury, Christchurch 8140, New Zealand
[c] Department of Electrical and Computer Engineering, Duke University, NC 27708, USA

ARTICLE INFO

Article history:
Received 20 October 2011
Received in revised form
22 November 2012
Accepted 23 December 2012
Available online 6 February 2013

Keywords:
Analytic models
Changes
Design envelope
Non-state-space models
Resiliency
State-space models

ABSTRACT

Resiliency is becoming an important service attribute for large scale distributed systems and networks. Key problems in resiliency quantification are lack of consensus on the definition of resiliency and systematic approach to quantify system resiliency. In general, resiliency is defined as the ability of (system/person/organization) to recover/defy/resist from any shock, insult, or disturbance [1]. Many researchers interpret resiliency as a synonym for fault-tolerance and reliability/availability. However, effect of failure/repair on systems is already covered by reliability/availability measures and that of on individual jobs is well covered under the umbrella of performability [2] and task completion time analysis [3]. We use Laprie [4] and Simoncini [5]'s definition in which resiliency is the persistence of service delivery that can justifiably be trusted, when facing changes. The changes we are referring to here are beyond the envelope of system configurations already considered during system design, that is, beyond fault tolerance. In this paper, we outline a general approach for system resiliency quantification. Using examples of non-state-space and state-space stochastic models, we analytically–numerically quantify the resiliency of system performance, reliability, availability and performability measures w.r.t. structural and parametric changes.

1. Introduction

For large scale commercial systems (e.g., telephone networks, cloud), resiliency is a key feature to achieve the requirements set by service level agreements. This paper presents a general approach for resiliency quantification of such systems using both non-state-space and state-space analytic models. Two key problems in performing such analysis are: (1) lack of consensus on the definition of resiliency and (2) lack of systematic approach for quantifying the system resiliency measures. The term resiliency is used in many different fields and its definitions are diverse. In general, resiliency can be defined as the ability of (system/person/organization) to recover/defy/resist from any shock, insult, or disturbance [1]. Dearnley [6] defined resiliency of database systems as the ability to return to a previous state after the occurrence of some event or action which may have changed that state. The terms related to the concept of resiliency are privacy, security and integrity. Najjar et al. [7] defined network resiliency as the probability of no disconnection in a family of regular graph network topologies. Resiliency is also used in other application domains such as aviation systems [8], cryptographic protocols [9], network subject to failures [10,11,7,12], computer network [13] and content distribution networks (CDN) under distributed denial of service (DDoS) attacks [14], wireless sensor networks [15], water resources systems [16], and data center [17]. Many researchers interpret resiliency as a synonym for fault-tolerance but the effects of using fault tolerance can be captured by traditional dependability measures such as reliability, availability, maintainability, safety and so on. DeBardeleben et al. [18] defined resiliency as the ability of a system to keep applications running and maintain an acceptable level of service in the face of transient, intermittent, and permanent faults. However, the effect of failure/repair on systems is fully covered by reliability/availability measures and on individual jobs is well covered under the umbrella of performability [2] and task completion time analysis [3]. Engelman et al. [19] proposed to use fundamental models (reliability and performance modeling) and resiliency supporting models (such as failure prediction, checkpointing, rejuvenation scheduling, etc.) to model and improve resiliency of high performance computing (HPC). Sterbenz et al. [20] defined resiliency as the combination of trustworthiness (dependability,

☆This research was supported in part by IBM Research and the US National Science Foundation under Grant NSF-CNS-08-31325.

* Corresponding author. Tel.: +1 919 452 0646.
E-mail addresses: rahul.ju.etce@gmail.com, rg51@ee.duke.edu, rghosh@us.ibm.com (R. Ghosh), dongseong.kim@canterbury.ac.nz (D. Kim), kst@ee.duke.edu (K.S. Trivedi).

0951-8320/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ress.2012.12.023

security, and performability) and tolerance (survivability, disruption tolerance, and traffic tolerance). There are demands to establish standards for resiliency in taxonomy, mechanisms and models [21]. Among many definitions, we use Laprie [4] and Simoncini [5]'s definition in which resiliency is the persistence of service delivery that can justifiably be trusted, when facing *changes*. We use this notion of *change*(*s*) in resiliency quantification. Changes can be: (i) increase/decrease in workload/faultload, (ii) increase/decrease in system capacity, (iii) change in system structure/configuration, (iv) occurrence of security attacks and (v) occurrence of accidents and disasters (e.g., earthquakes, flooding). Note that changes in system states due to dynamic redundancy features such as failure detection followed by (automated) reconfiguration are already included in traditional fault-tolerance and reliability/availability. The change we are referring to here is beyond the envelope of system configuration already considered during system design, i.e., beyond fault-tolerance.

Clearly, notion of resiliency should be beyond fault-tolerance and reliability/availability/performability. We view it as the transient system behavior after the change occurs; not just the steady state system behavior before and after the change. Resiliency quantification of performance and dependability attributes of systems can be carried out by using model representation techniques such as: (i) non-state-space models (e.g., reliability block diagrams, fault trees), (ii) state-space models (e.g., continuous time Markov chains (CTMC), stochastic Petri nets), (iii) hierarchical and fixed-point iterative models, (iv) simulation and hybrid models [22,23], among others [24]. In this paper, we extend our previous work [25] in several ways: (i) showing resiliency quantification for composite performability state-space model, (ii) quantifying resiliency of system reliability using phased-mission system (PMS) analysis for non-state-space models and (iii) analyzing the transient impacts of both parametric and structural changes. We present general steps for resiliency quantification using stochastic analytic models and subsequently describe how this approach can be applied for two real system examples: (a) emergency core cooling system (ECCS) of a boiler water reactor (BWR) and (b) a telephone switching system. For the ECCS, we quantify resiliency of system reliability using non-state-space models when parametric and structural changes are applied. For the telephone switching system, we quantify resiliency of performance, availability and performability measures when parametric changes are applied.

Rest of the paper is organized as follows. Section 2 describes a traditional reliability model using a fault tree and traditional performance, availability and performability models using continuous time Markov chains (CTMC). In Section 3, we show how resiliency analysis is different from the traditional dependability analysis and describe the general steps for resiliency quantification. Numerical results for resiliency quantification of real case studies are also presented in Section 3. Finally, we conclude this work and outline future avenues of research in Section 4.

2. Traditional performance and dependability analysis

We describe performance, reliability, availability and performability models for real system examples using both non-state-space and state-space models. All models are exercised in SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) [26] and numerical results are shown for different output measures.

2.1. Reliability analysis using non-state-space model

We analyze the reliability of emergency core cooling system (ECCS) of a boiling water reactor (BWR) using a fault tree [27]. A detailed physical description of the fault tree can be found in [27]; while a simplified version is described in [28]. Fig. 1 shows the fault tree of the ECCS under normal conditions.

Model output: From Fig. 1, reliability of the overall system is given by

$$R_{eccs} = 1 - Pr[\overline{A} \cap (\overline{B} \cup (\overline{C} \cap \overline{D} \cap \overline{E} \cap \overline{F}))] \quad (1)$$

where, $X=1$ ($X \in \{A,B,C,D,E,F\}$) denotes the event that the component X is up.

Numerical results: Solution methods for such fault tree models have been implemented in many software packages such as SHARPE [26]. Fig. 2 shows reliability of ECCS as a function of time, under normal conditions. Mean time to failure (MTTF) of each component was assumed to be 1000 h. SHARPE input file for reliability analysis of ECCS is shown in Fig. 3.

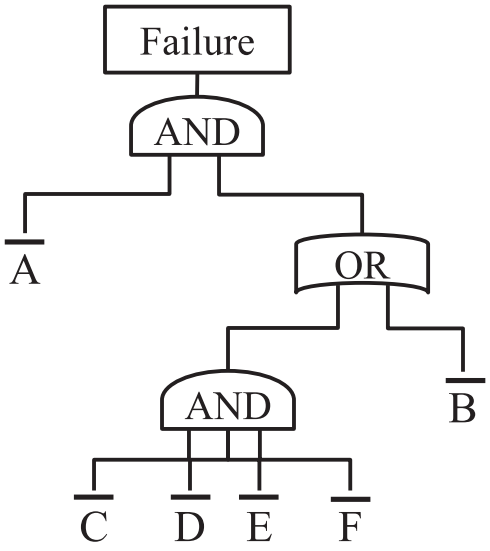

Fig. 1. Fault tree of emergency core cooling system (ECCS) of a boiling water reactor (BWR).

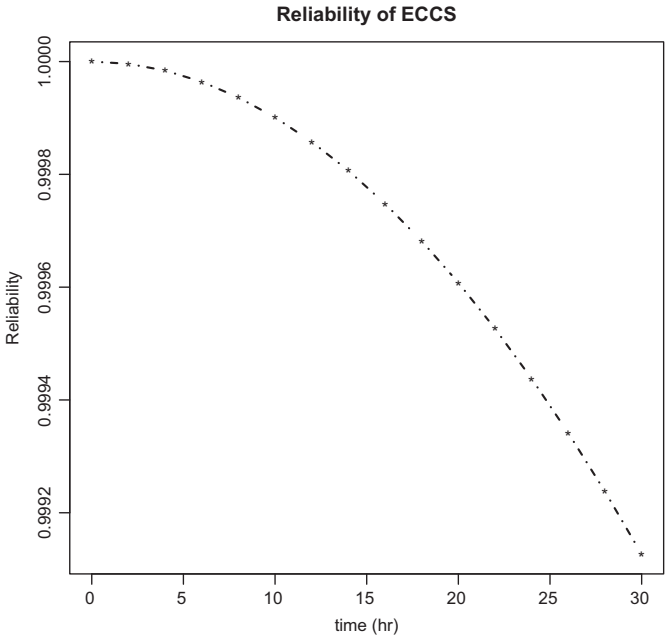

Fig. 2. Reliability of ECCS under normal conditions.

```
1.  format 8
2.
3.
4.  bind
5.  a_x 0.001
6.  b_x 0.001
7.  c_x 0.001
8.  d_x 0.001
9.  e_x 0.001
10. f_x 0.001
11. T 30
12. end
13. * model description
14. ftree X1
15. basic a exp(a_x)
16. basic b exp(b_x)
17. basic c exp(c_x)
18. basic d exp(d_x)
19. basic e exp(e_x)
20. basic f exp(f_x)
21. and CDEF c d e f
22. or CDEFB CDEF b
23. and top a CDEFB
24. end
25. * output measure
26.
27. func reliability(t)\
    1-tvalue(t;X1)
28.
29. loop
30. t,0,T,2
31. expr reliability(t)
32. end
33.
34. end
```

Fig. 3. SHARPE input file for reliability analysis of ECCS (Fig. 2), under normal conditions.

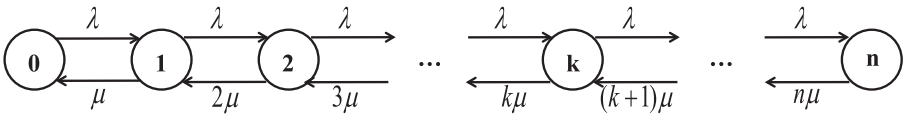

Fig. 4. Performance model of telephone switching system.

2.2. Performance, availability and performability analysis using state-space models

We consider a telephone switching system consisting of n channels [29]. This sub-section describes analytic models for performance, availability and performability measures. We describe the mathematical details to obtain the steady state and transient measures from these models and use SHARPE to show numerical results.

2.2.1. Performance model

We assume that the call arrival process is homogenous Poisson with rate λ. Call service times are assumed to be independent and exponentially distributed with rate μ. The performance model is then a homogenous CTMC as shown in Fig. 4. State index k of the CTMC in Fig. 4 denotes the number of busy channels. Steady state probability vector $\boldsymbol{\pi}$ for the CTMC in Fig. 4 is denoted by

$$\boldsymbol{\pi} = [\pi_0, \pi_1, \ldots, \pi_n] \tag{2}$$

Let the generator matrix of the CTMC in Fig. 4 be $\boldsymbol{Q_{perf}}$. After solving the system of linear equations

$$\boldsymbol{\pi Q_{perf}} = 0 \tag{3}$$

with $\sum_{k=0}^{n} \pi_k = 1$, we can compute the steady state probabilities. Let the transient state probability vector $\boldsymbol{\pi(t)}$ of the CTMC in Fig. 4 be denoted by

$$\boldsymbol{\pi(t)} = [\pi_0(t), \pi_1(t), \ldots, \pi_n(t)] \tag{4}$$

and the time derivative of the transient state probability vector by

$$\frac{\boldsymbol{d\pi(t)}}{\boldsymbol{dt}} = \left[\frac{d\pi_0(t)}{dt}, \frac{d\pi_1(t)}{dt}, \ldots, \frac{d\pi_n(t)}{dt}\right] \tag{5}$$

Then, the Kolmogorov differential equation for transient analysis can be written as

$$\frac{\boldsymbol{d\pi(t)}}{\boldsymbol{dt}} = \boldsymbol{\pi(t) Q_{perf}} \tag{6}$$

with the initial state probability vector

$$\boldsymbol{\pi(0)} = [\pi_0(0), \pi_1(0), \ldots, \pi_n(0)] \tag{7}$$

By solving Eq. (6) with the initial state probability vector in Eq. (7), we can obtain the transient state probabilities.

Model outputs: Different performance measures can be computed using the Markov reward approach [29]. Using a reward rate r_k for state k, we can compute the steady state performance measure as steady state expected reward rate, given by: $\sum_{k=0}^{n} r_k \pi_k$. Similarly, transient performance measure can be computed as: $\sum_{k=0}^{n} r_k \pi_k(t)$. In this paper, we focus only on call blocking probability. When all n channels are busy, an incoming call will be blocked. This is represented by state n and call blocking probability is given as the probability of being in state n. Thus, by assigning a reward rate of 1 to state n and a reward rate 0 to all other states, steady state and transient call blocking probabilities can be obtained as π_n and $\pi_n(t)$ respectively.

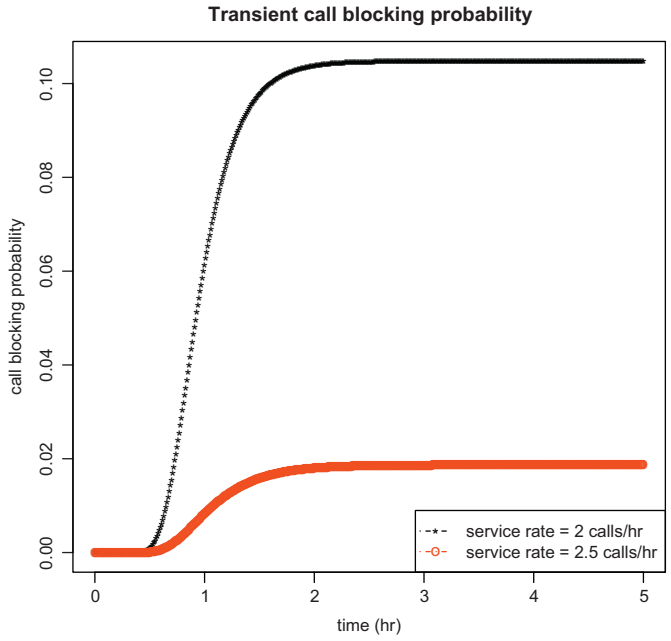

Fig. 5. Transient call blocking probability of the telephone switching system.

Numerical results: Fig. 5 shows the transient call blocking probability of a switching system with 50 channels. We assumed that at $t=0$, no channels were busy. Keeping the call arrival rate same ($\lambda = 100$ calls/h), we show the effects of varying call service rates ($\mu = 2$ and 2.5 calls/h) on call blocking probability. SHARPE input file for transient performance analysis of switching system is shown in Fig. 6.

2.2.2. Availability model (without fault detection/reconfiguration delay)

Availability model takes into account failure–repair behavior of the channels. We assume that times to channel failure and

```
format 8

bind
lambda 100
mu    2
n 50
t 0
t_init 0
t_final 5.0
time_step 0.01
end

* model description
markov perf(n,lambda)
loop i,0,n-1
$(i) $(i+1) lambda
$(i+1) $(i) (i+1)*mu
end

end
$(0) 1
end

* output measure
func Pb_per(n,lambda, t) \
tvalue(t;perf,$(n);n,lambda)

bind  t     t_init
while (t <= t_final)
bind  block_prob \
 Pb_per(n,lambda,t)
expr  t
expr  block_prob
bind  t     (t+time_step)
end

end
```

Fig. 6. SHARPE input file for transient performance analysis of switching system.

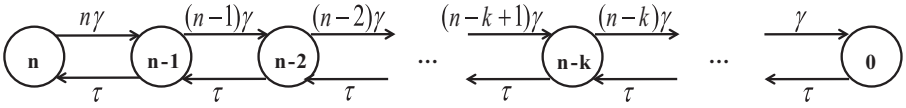

Fig. 7. Availability model of switching system without fault detection/reconfiguration delay.

repair are exponentially distributed with mean $1/\gamma$ and $1/\tau$ respectively. After a channel failure, repair of the failed channel is started immediately (i.e., with instantaneous fault detection). We also assume that a single repair facility is shared by all the channels. The availability model is then a homogenous CTMC as shown in Fig. 7. State index k of the CTMC in Fig. 7 denotes the number of non-failed channels. Steady state probability vector $\boldsymbol{\phi}$ for the CTMC in Fig. 7 is denoted by

$$\boldsymbol{\phi} = [\phi_0, \phi_1, \ldots, \phi_n] \tag{8}$$

Generator matrix for the CTMC in Fig. 7 is denoted by $\boldsymbol{Q_{avail}}$. After solving the system of linear equations

$$\boldsymbol{\phi Q_{avail}} = 0 \tag{9}$$

with $\sum_{k=0}^{n} \phi_k = 1$, we can compute the steady state probabilities. Transient state probability vector $\boldsymbol{\phi(t)}$ for the CTMC in Fig. 7 is given by

$$\boldsymbol{\phi(t)} = [\phi_0(t), \phi_1(t), \ldots, \phi_n(t)] \tag{10}$$

and the time derivative of the transient state probability vector by

$$\frac{\boldsymbol{d\phi(t)}}{\boldsymbol{dt}} = \left[\frac{d\phi_0(t)}{dt}, \frac{d\phi_1(t)}{dt}, \ldots, \frac{d\phi_n(t)}{dt}\right] \tag{11}$$

Thus, the Kolmogorov differential equation for transient analysis as

$$\frac{\boldsymbol{d\phi(t)}}{\boldsymbol{dt}} = \boldsymbol{\phi(t) Q_{avail}} \tag{12}$$

with the initial state probability vector

$$\boldsymbol{\phi(0)} = [\phi_0(0), \phi_1(0), \ldots, \phi_n(0)] \tag{13}$$

By solving Eq. (12) with the initial state probability vector in Eq. (13), we can compute the transient state probabilities.

Model outputs: Using Markov reward approach as described earlier, we can compute different steady state and transient measures from the availability model. In this paper, we only focus on switching system unavailability. When all n channels have failed, the system will be unavailable to a new incoming call. This is represented by state 0 and unavailability is given as the probability of being in state 0. Thus, by assigning a reward rate 1 to state 0 and reward rate 0 to all other states, steady state and transient unavailability can be obtained as ϕ_0 and $\phi_0(t)$ respectively.

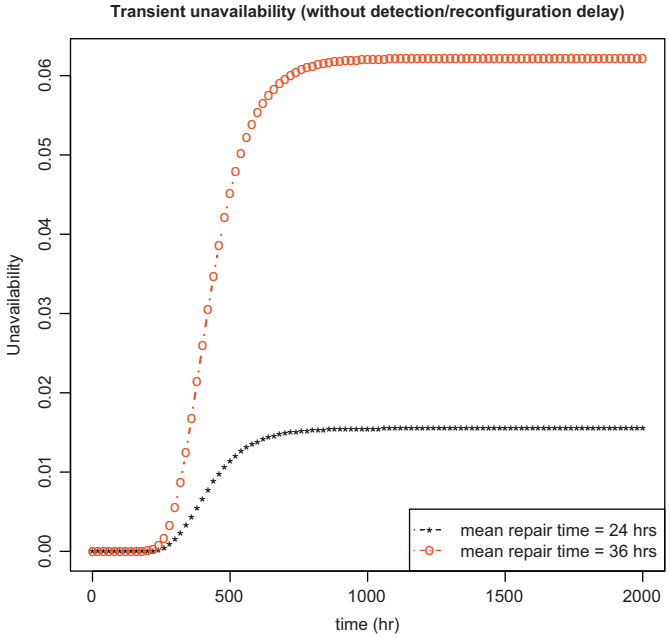

Fig. 8. Transient unavailability of switching system (without fault detection/reconfiguration delay).

Numerical results: Fig. 8 shows the transient unavailability of a switching system with 50 channels. We assumed that at $t=0$, all channels were 'UP'. Keeping the mean time to failure (MTTF) of a channel fixed ($1/\gamma = 100$ h), we show the effects of varying mean time to repair (MTTR) ($1/\mu = 24$ and 36 h) on unavailability. SHARPE input file for this analysis is shown in Fig. 9.

2.2.3. Availability model (with fault detection/reconfiguration delay)

Availability model shown in Fig. 7 was developed under the assumption that a fault in switching system is detected instantaneously and repair process starts immediately after the failure.

```
1.  format 8
2.
3.  bind
4.  gamma 1/100
5.  tau   1/24
6.  n 50
7.  t 0
8.  t_init 0
9.  t_final 2000
10. time_step 20
11. end
12.
13. * model description
14. markov avail(n, gamma)
15. loop i,n,1,-1
16. $(i) $(i-1) i*gamma
17. $(i-1) $(i) tau
18. end
19. end
20. $(n) 1
21. end
22.
23. * output measure
24.
25. func  UAt(n, gamma, t) \
     tvalue(t;avail,0;n,gamma)
26.
27. bind  t     t_init
28. while (t <= t_final)
29. bind  unavail  UAt(n, gamma,t)
30. expr  t
31. expr  unavail
32. bind  t     (t+time_step)
33. end

34. end
```

Fig. 9. SHARPE input file for transient availability analysis of switching system (w/o detection/reconfiguration delay).

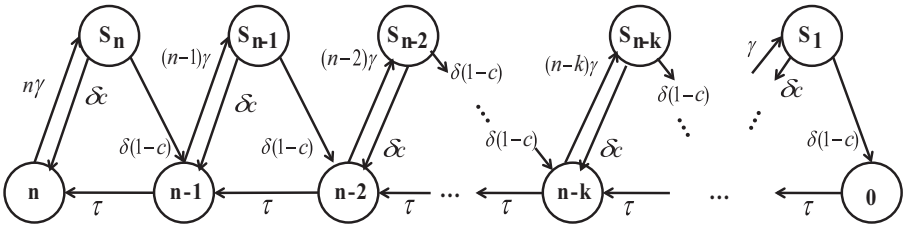

Fig. 10. Availability model of switching system with fault detection/reconfiguration delay.

This assumption is relaxed in the availability model shown in Fig. 10. We assume, there are n channels and MTTF of a channel is $1/\gamma$. After a channel failure, the system goes to a detection/reconfiguration state, denoted by S_k, where k is the number of non-failed channels before the system goes to detection/reconfiguration state. Time to fault-detection/reconfiguration is assumed to be exponentially distributed with mean $1/\delta$. After detection/reconfiguration, channel failure is either recovered with probability c (coverage factor) or the fault stays within the system with probability $(1-c)$. A failed channel which cannot be recovered by detection/reconfiguration is finally repaired with MTTR $1/\tau$. Steady state probability vector $\boldsymbol{\psi}$ for the CTMC in Fig. 10 is denoted by

$$\boldsymbol{\psi} = [\psi_n, \psi_{S_n}, \ldots, \psi_0] \tag{14}$$

Generator matrix for the CTMC in Fig. 10 is denoted by $\boldsymbol{Q}_{a\_detect}$. After solving the system of linear equations

$$\boldsymbol{\psi Q}_{a\_detect} = 0 \tag{15}$$

with $\sum_{k=0}^{n} \psi_k + \sum_{k=1}^{n} \psi_{S_k} = 1$, we can compute the steady state probabilities. Let the transient state probability vector $\boldsymbol{\psi(t)}$ of the CTMC in Fig. 10 be denoted by

$$\boldsymbol{\psi(t)} = [\psi_n(t), \psi_{S_n}(t), \ldots, \psi_0(t)] \tag{16}$$

and the time derivative of the transient state probability vector by

$$\frac{\boldsymbol{d\psi(t)}}{\boldsymbol{dt}} = \left[\frac{d\psi_n(t)}{dt}, \frac{d\psi_{S_n}(t)}{dt}, \ldots, \frac{d\psi_0(t)}{dt}\right] \tag{17}$$

Then, the Kolmogorov differential equation for transient analysis is

$$\frac{\boldsymbol{d\psi(t)}}{\boldsymbol{dt}} = \boldsymbol{\psi(t) Q}_{a\_detect} \tag{18}$$

with the initial state probability vector $\boldsymbol{\psi(0)}$

$$\boldsymbol{\psi(0)} = [\psi_n(0), \psi_{S_n}(0), \ldots, \psi_0(0)] \tag{19}$$

By solving Eq. (18) with the initial state probability vector in Eq. (19), we can compute the transient state probabilities.

Model outputs: Using Markov reward approach as described earlier, we can compute steady state and transient unavailability. We assume that detection/reconfiguration states are considered to be down if the sojourn times in those states, are longer than a pre-defined threshold t_{th}. In a detection/reconfiguration state, probability that the sojourn time is longer than t_{th} is given by $e^{-\delta t_{th}}$. So, we assign reward rate $e^{-\delta t_{th}}$ to the detection/reconfiguration states, reward rate 1 to state 0 and reward rate 0 to all other states. Thus, steady state unavailability is given by

$$UA_{a\_detect} = e^{-\delta t_{th}} \sum_{k=1}^{n} \psi_{S_k} + \psi_0 \tag{20}$$

Transient unavailability is given by

$$UA_{a\_detect}(t) = e^{-\delta t_{th}} \sum_{k=1}^{n} \psi_{S_k}(t) + \psi_0(t) \tag{21}$$

Numerical results: Fig. 11 shows the transient unavailability of a 50 channel switching system with fault detection/reconfiguration delays. We assumed that at $t=0$, all channels were 'UP'. Keeping the channel MTTF ($1/\gamma = 100$ h), MTTR ($1/\tau = 24$ h) and mean detection/delay ($1/\delta = 5$ s) fixed, we show the effects of varying t_{th} ($t_{th} = 1$ and 10 s) on unavailability. Coverage factor c was assumed to be 0.95. SHARPE input file for this analysis is shown in Fig. 12.

2.2.4. Performability model (without fault detection/reconfiguration delay)

Performability model takes into account both performance and failure–repair behavior of the channels. We assume that time to failure of a channel is exponentially distributed with rate γ. Channel fault is detected instantaneously and a failed channel is repaired with rate τ. Inter-arrival times and service times of a call are exponentially distributed with rates λ and μ respectively. The performability model is then a homogenous CTMC as shown in Fig. 13. State index of the CTMC in Fig. 13 is denoted by (i,j), where i is the number of non-failed channels and $j(\leq i)$ denotes

the number of ongoing calls in the system. Note that channels which are in use as well as those which are free can fail with corresponding failure rates. Steady state probability vector $\boldsymbol{\theta}$ for the CTMC in Fig. 13 is denoted by

$$\boldsymbol{\theta} = [\theta_{(0,0)}, \theta_{(1,0)}, \ldots, \theta_{(n,n)}] \tag{22}$$

After solving the system of linear equations

$$\boldsymbol{\theta Q_{perform}} = 0 \tag{23}$$

with $\sum_{i=0}^{n} \sum_{j=0}^{i} \theta_{(i,j)} = 1$, we can compute the steady state probabilities. Transient state probability vector $\boldsymbol{\theta(t)}$ for the CTMC in Fig. 13 is denoted by

$$\boldsymbol{\theta(t)} = [\theta_{(0,0)}(t), \theta_{(1,0)}(t), \ldots, \theta_{(n,n)}(t)] \tag{24}$$

After solving the system of equations

$$\frac{\boldsymbol{d\theta(t)}}{\boldsymbol{dt}} = \boldsymbol{\theta(t) Q_{perform}} \tag{25}$$

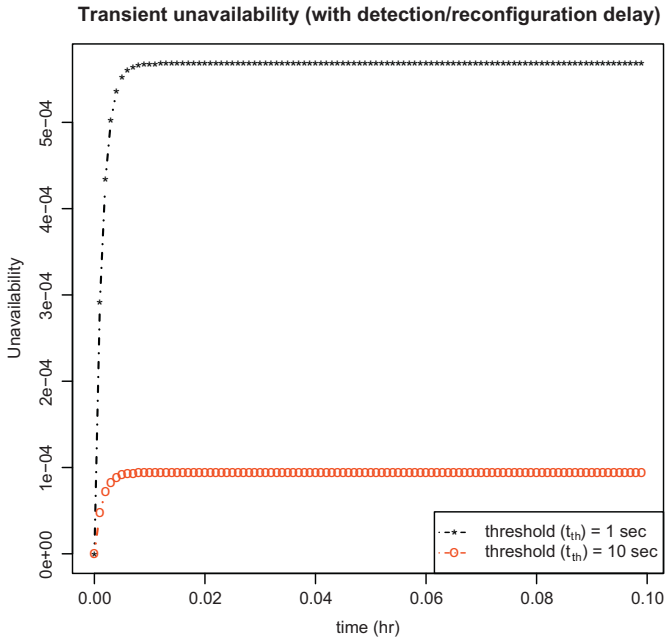

Fig. 11. Transient unavailability of switching system (with fault detection/reconfiguration delay).

with the initial condition

$$\boldsymbol{\theta(0)} = [\theta_{(0,0)}(0), \theta_{(1,0)}(0), \ldots, \theta_{(n,n)}(0)] \tag{26}$$

we can compute the transient state probabilities.

Model outputs: Two key measures from the performability model are: (i) total call blocking probability and (ii) total call dropping probability. We can compute the steady state and transient values of call blocking and dropping probabilities using Markov reward approach. To compute blocking probability, we attach a reward rate 1 to states of the form (i,i) and reward rate 0 is assigned to all other states. Values of steady state and transient call blocking probabilities are: $\sum_{i=0}^{n} \theta_{(i,i)}$ and $\sum_{i=0}^{n} \theta_{(i,i)}(t)$ respectively. To compute dropping probability, we attach a reward rate $(j\gamma/\lambda)$ to state (i,j). Values of steady state and transient call dropping probabilities are: $\sum_{i=0}^{n} \sum_{j=0}^{i} (j\gamma/\lambda)\theta_{(i,j)}$ and $\sum_{i=0}^{n} \sum_{j=0}^{i} (j\gamma/\lambda)\theta_{(i,j)}(t)$ respectively.

Numerical results: Fig. 14(a) and (b) respectively shows the transient call blocking and dropping probabilities of a switching system with five channels, in the presence of failure and repair. We assumed that at $t=0$, no channels were busy and all channels were 'UP'. MTTF and MTTR of a channel were assumed to be 1000 h and 24 h respectively. Keeping the call arrival rate fixed ($\lambda = 10$ calls/h), we show the effects of varying call service rates ($\mu = 2$ and 2.5 calls/h) on call blocking and dropping probabilities. SHARPE input file for this analysis is shown in Fig. 15.

2.2.5. Performability model (with fault detection/reconfiguration delay)

Fig. 16 shows the CTMC of performability model with a non-zero fault detection/reconfiguration delay. State indices of the CTMC in Fig. 16 are denoted by: (a) (i,j), where i is the number of non-failed channels and $j(\leq i)$ denotes the number of ongoing calls in the system and (b) (S_p,q), where p is the number of non-failed channels before the system goes to detection/reconfiguration state and q is the number of ongoing calls. In detection/reconfiguration state, we assume that a new call is not accepted. Call arrival and service rates are assumed to be λ and μ respectively. Channel failure and repair rates are assumed to be γ and μ respectively. Rate of fault detection/reconfiguration is assumed to be δ. A detected fault can be recovered with probability c. Steady state probability vector $\boldsymbol{\omega}$ for the CTMC in Fig. 16 is denoted by

$$\boldsymbol{\omega} = [\omega_{(0,0)}, \omega_{(S_1,0)}, \ldots, \omega_{(n,n)}] \tag{27}$$

```
1.  format 8
2.
3.  bind
4.  gamma 1/100
5.  tau   1/24
6.  n 50
7.  t 0
8.  t_init 0
9.  t_final    0.1
10. time_step  0.001
11. c     0.95
12. delta 720
13. t_th  10/3600
14. end
15.
16. * model description
17. markov avail_det(n, gamma)
18. loop i,n,1,-1
19.
20. * failure arcs
21. $(i)  S_$(i) i*gamma
22. S_$(i) $(i) delta*c
23. S_$(i) $(i-1) delta*(1-c)
24. * repair
25. $(i-1) $(i) tau
26. end
27. end
28. $(n) 1
29. end
30.
31. * output measure
32. func  UAt(n, gamma, t)\
    tvalue(t;avail_det,0;n,gamma)+ \
    sum(k, 1, n, ^(-delta*t_th)* \
    tvalue(t; avail_det, S_$(k); n,
    gamma))
33.
34. bind  t     t_init
35.
36. while (t <= t_final)
37. bind  unavail  UAt(n, gamma,t)
38. expr  t
39. expr  unavail
40. bind  t     (t+time_step)
41. end
42. end
```

Fig. 12. SHARPE input file for transient availability analysis of switching system (with detection/reconfiguration delay).

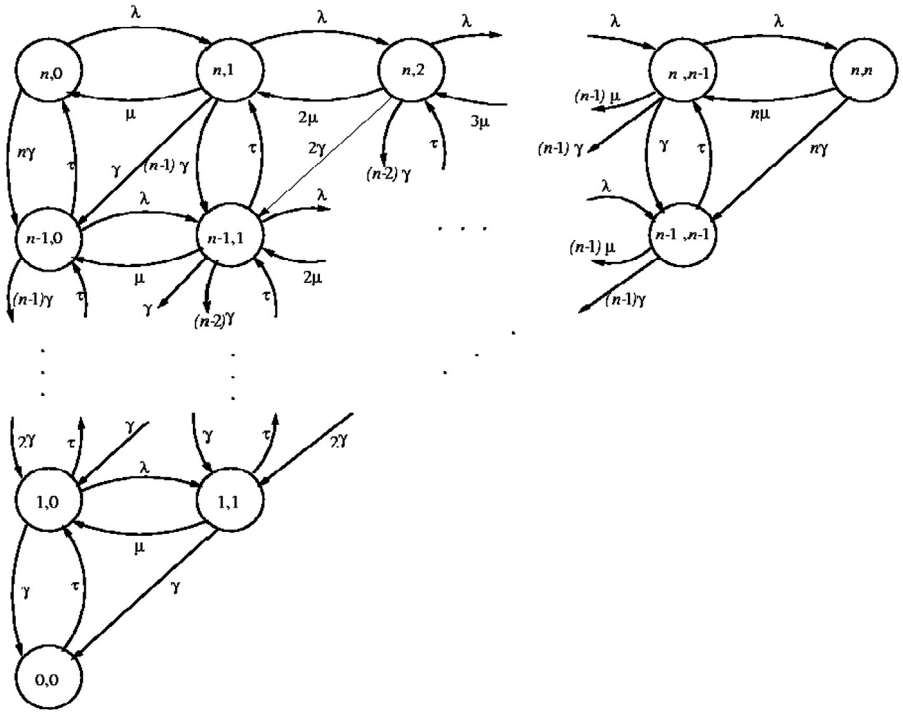

Fig. 13. Performability model of switching system (without fault detection/reconfiguration delay).

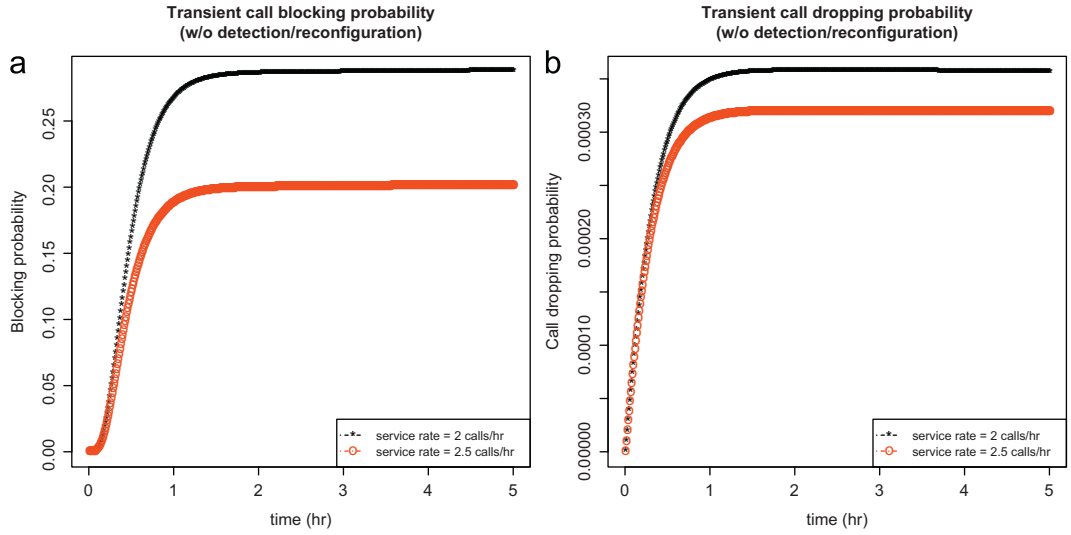

Fig. 14. (a) Transient call blocking probability and (b) transient call dropping probability of switching system from performability model (without detection/reconfiguration delay).

After solving the system of linear equations

$$\boldsymbol{\omega Q}_{pa\_detect} = 0 \tag{28}$$

with

$$\sum_{i=0}^{n}\sum_{j=0}^{i}\omega_{(i,j)} + \sum_{p=1}^{n}\sum_{q=0}^{p-1}\omega_{(S_p,q)} = 1$$

we can compute the steady state probabilities. Transient state probability vector $\boldsymbol{\omega}(t)$ for the CTMC in Fig. 16 is denoted by

$$\boldsymbol{\omega}(t) = [\omega_{(0,0)}(t), \omega_{(S_1,0)}(t), \ldots, \omega_{(n,n)}(t)] \tag{29}$$

After solving the system of equations

$$\frac{d\boldsymbol{\omega}(t)}{dt} = \boldsymbol{\omega}(t)\boldsymbol{Q}_{pa\_detect} \tag{30}$$

with the initial condition

$$\boldsymbol{\omega}(0) = [\omega_{(0,0)}(0), \omega_{(1,0)}(0), \ldots, \omega_{(n,n)}(0)] \tag{31}$$

we can compute the transient state probabilities.

Model outputs: Steady state call blocking probability is given by

$$T_{b\_detect} = \sum_{i=0}^{n}\omega_{(i,i)} + \sum_{p=1}^{n}\sum_{q=0}^{p-1}\omega_{(S_p,q)} \tag{32}$$

```
format 8
bind
lambda      10
mu    2
n 5
gamma 1/1000
tau   1/24
t 0
t_init 0
t_final 5
time_step 0.01
end

* model description
markov pa(lambda, gamma)
  loop i,n,1,-1
    loop j,0, i-1

        * Definition of the performance part of the model
        $(i)_$(j) $(i)_$(j+1) lambda
        $(i)_$(j+1) $(i)_$(j) (j+1)*mu

        * Definition of the availability part of the model
        $(i)_$(j) $(i-1)_$(j) (i-j)*gamma
        $(i-1)_$(j) $(i)_$(j) tau

        * Diagonal arc for example from (n,2) to (n-1,1)
        $(i)_$(j+1) $(i-1)_$(j) (j+1)*gamma
     end
  end
end
$(n)_0  1
end

* output measures
func Pblock_performa(t,lambda, gamma) sum(i,0,n, \
tvalue(t;pa,$(i)_$(i);lambda, gamma))
func Pdrop_performa(t,lambda, gamma) sum(i,0,n, sum(j,0,i, \
(j*gamma/lambda)*tvalue(t;pa,$(i)_$(j);lambda, gamma)))


while (t <= t_final)
    bind    block_prob      Pblock_performa(t, lambda, gamma)
    bind    drop_prob       Pdrop_performa(t, lambda, gamma)
    expr    t
    expr    block_prob
    expr    drop_prob
    bind    t       (t+time_step)
end
end
```

Fig. 15. SHARPE input file for transient performability analysis (w/o detection/reconfiguration delay).

Transient call blocking probability is given by

$$T_{b\_detect}(t) = \sum_{i=0}^{n} \omega_{(i,i)}(t) + \sum_{p=1}^{n} \sum_{q=0}^{p-1} \omega_{(S_p,q)}(t) \tag{33}$$

Steady state call dropping probability is given by

$$T_{d\_detect} = \sum_{i=0}^{n} \sum_{j=0}^{i} (j\gamma/\lambda)\omega_{(i,j)} \tag{34}$$

Transient call dropping probability is given by

$$T_{d\_detect}(t) = \sum_{i=0}^{n} \sum_{j=0}^{i} (j\gamma/\lambda)\omega_{(i,j)}(t) \tag{35}$$

Numerical results: Fig. 17(a) and (b) shows respectively the transient call blocking and dropping probabilities of a switching system with five channels, in the presence of failure and repair. We assumed that at $t=0$, no channels were busy and all channels were 'UP'. MTTF and MTTR of a channel were assumed to be 1000 h and 24 h respectively. Keeping the call arrival rate fixed ($\lambda = 10$ calls/h), we show the effects of varying call service rates ($\mu = 2$ and 2.5 calls/h) on call blocking and dropping probabilities. SHARPE input file for this analysis is shown in Fig. 18.

3. Proposed resiliency quantification approach

We outline the general steps for resiliency quantification of a system:

(1) Construct a stochastic analytic model of a given system to find measure(s) of interest. Example of such a model can be performance, reliability, availability or performability model of the system.

(2) Determine the system measures of interest under normal conditions from the model developed in step (1) above. In this step, we compute transient or steady state values of measures of interest, without any application of changes.

(3) Apply change(s) to the system. Changes can be classified into two broad categories: (a) parametric changes and (b) structural changes. Parametric changes can be enforced by increasing

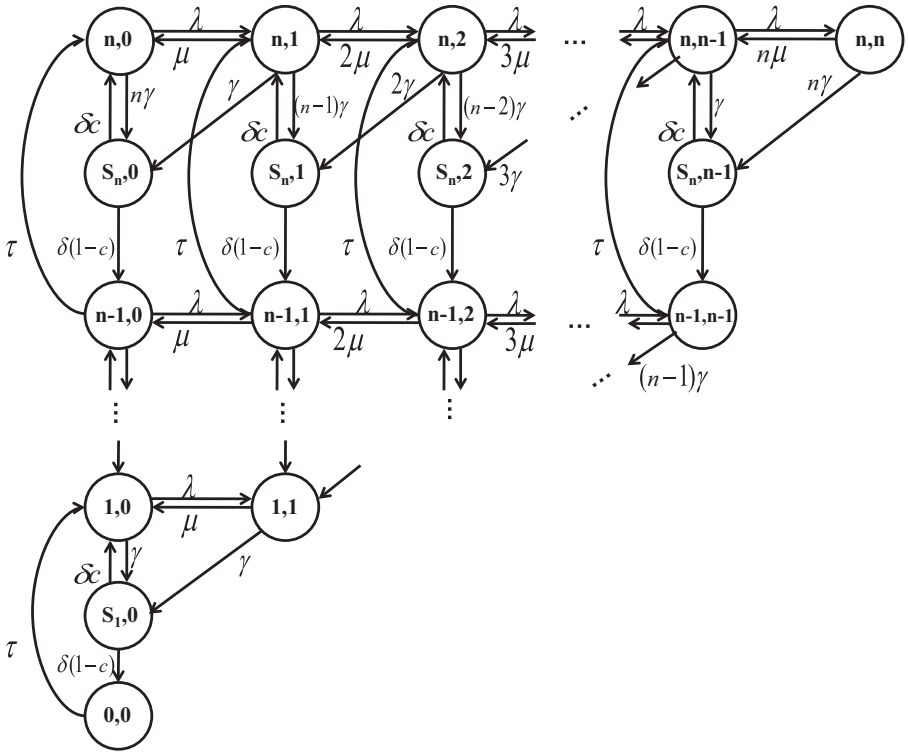

Fig. 16. Performability model of switching system (with fault detection/reconfiguration delay).

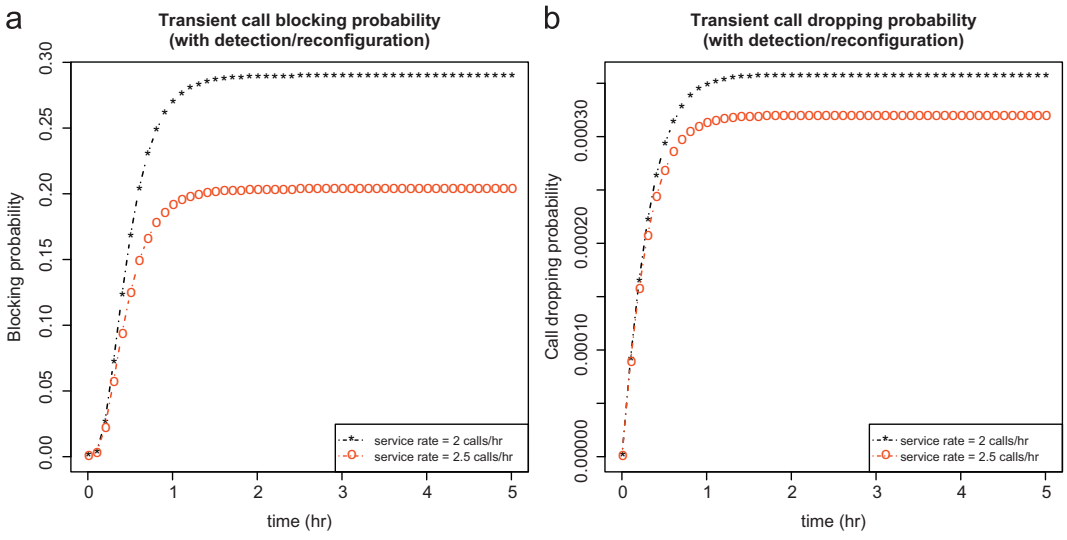

Fig. 17. (a) Transient call blocking and (b) transient call dropping probability from performability model (with detection and reconfiguration).

(or decreasing) the value(s) of input parameter(s) of the model. Examples of such changes are variation of call arrival rates, hardware/software failure rates. In case of state-space models, structural changes can be enforced by adding (or removing) states and/or transitions in the model. In case of non-state-space model, such as ECCS, an example of structural change is a change in system configuration (the logic model).

(4) Analyze the transient behavior of the system model to compute the transient measures after applying the change(s). Initial probabilities for this transient analysis are obtained from the steady state/transient probabilities as computed from the normal behavior of the system model in the step (2). After the change is applied, transient response of the performance, reliability, availability and performability measures quantify the resiliency of the system.

We explain how this approach can be applied for the ECCS and the telephone switching system examples. Reliability model of ECCS and performance, availability and performability models of switching system have been already shown in Section 2. In this section, we show the mathematical steps for resiliency quantification. For ECCS, we show the impact of both parametric and structural changes while for the telephone switching system, we focus only on parametric changes. For mathematical convenience, we summarize meanings of different superscripts in Table 1, as used in rest of the paper. We use SHARPE [26], to obtain the numerical results.

```
format 8
bind
lambda 10
mu    2.5
n 5
gamma 1/1000
tau 1/24
c     0.95
delta 0.9
t_th  1/3600
t_init     0
t_final    5
t     t_init
time_step  0.1
end

* model description
markov pa_detect(lambda, gamma)
  loop i, n, 1, -1
    loop j, 0, i-1
      $(i)_$(j)        $(i)_$(j+1)     lambda
      $(i)_$(j+1)      $(i)_$(j)      (j+1)*mu
      $(i)_$(j)        S_$(i)_$(j)    (i-j)*gamma
      S_$(i)_$(j)      $(i)_$(j)      delta*c
      S_$(i)_$(j)      $(i-1)_$(j)    delta*(1-c)
      $(i-1)_$(j)      $(i)_$(j)      tau
      $(i)_$(j+1)      S_$(i)_$(j)    (j+1)*gamma
    end
  end
end
$(n)_0  1
end

* output measure
func Pblock_trans(t,lambda, gamma)     sum(i,0,n, \
tvalue(t;pa_detect,$(i)_$(i);lambda, gamma)) + sum(i, 1, n, \
sum(j, 0, i-1, tvalue(t; pa_detect, S_$(i)_$(j); lambda, gamma)))

func Pdrop_trans(t,lambda, gamma)      sum(i,0,n, sum(j,0,i \
(j*gamma/lambda)*tvalue(t;pa_detect,$(i)_$(j);lambda, gamma)))

while (t <= t_final)
        bind    block_prob      Pblock_trans(t, lambda, gamma)
        bind    drop_prob       Pdrop_trans(t, lambda, gamma)
        expr    t
        expr    block_prob
        expr    drop_prob
        bind    t       (t+time_step)
end
end
```

Fig. 18. SHARPE input file for transient performability analysis (w/detection/reconfiguration delay).

Table 1
Meanings of different superscripts used.

Superscripts	Descriptions
'param'	Denotes resiliency analysis with parametric changes; used in non-state-space reliability models
'struct'	Denotes resiliency analysis with structural changes; used in non-state-space reliability models
'o'	*Old* values of parameters, generator matrix, state probabilities *before the change is applied*; used in performance and availability models
'w'	*New* values of parameters, generator matrix, state probabilities *after the change is applied*; used in performance and availability models
'op'	*Old* value of call arrival rate, generator matrix, state probabilities *before the change is applied*; used in performability model for resiliency of *blocking probability*
'wp'	*New* value of call arrival rate, generator matrix, state probabilities *after the change is applied*; used in performability model for resiliency of *blocking probability*

3.1. Resiliency quantification for ECCS

To quantify resiliency of ECCS, we map the problem onto what is known as phased-mission system (PMS) model in the reliability literature [27,28]. In a PMS model, a system carries out a specific mission which can be divided into consecutive time periods (phases). In each phase, the system needs to accomplish a specific task and the system configuration, the phase duration, and failure/repair rates of the hardware/software components may change from phase to phase. Such changes in system configuration (i.e., structural change) and/or in component failure/repair rates (i.e., parametric change) make the PMS model convenient for resiliency analysis. We describe resiliency quantification of ECCS reliability under parametric and structural changes.

In the PMS model, cumulative distribution function (CDF) of time to failure of a component K_j in the pth phase can be written

as [28]

$$F_{K_{jp}}(t) = 1 - \left[\exp\left(-\sum_{i=1}^{p-1} v_{ji} T_i\right)\right] \exp(-v_{jp} t) \tag{36}$$

where v_{ji} is the failure rate of component K_j in phase i, T_i is the phase duration for phase i, $t \in [0, T_p]$ and $1 \le i \le p$. Observe that $F_{K_{jp}}(t)$ is the probability that the component K_j fails at some time instant t_f, where $0 \le t_f \le t + \sum_{i=1}^{p-1} T_i$. The probability mass at the beginning of phase p is the probability that the component has already failed at a time instant within first $(p-1)$ phases and is given by setting $t=0$ in the above formula. In Eq. (36), the term in square bracket is the probability that the component K_j has survived in the first $(p-1)$ phases.

While Eq. (36) provides the CDF of time to failure of individual component in different phases, overall unreliability of the PMS can be computed from sum of disjoint phase products (SDPP) [28]. If PE_i be the event that a PMS is down in phase i, overall reliability of the PMS is given by

$$R_{pms} = 1 - Pr\left[\bigcup_{i=1}^{p} PE_i\right] \tag{37}$$

ECCS has three phases [27,28] and durations of three phases are assumed to be T_1, T_2 and T_3 respectively.

3.1.1. Resiliency quantification of ECCS reliability upon parametric changes

Each phase of ECCS is represented through the fault tree shown in Fig. 1. We quantify the resiliency of reliability of ECCS w.r.t. the change in failure rate of component A. During phase 1, let the failure rate of component A be v_{a_1} and let $A_1^{(v_{a_1})} = 1$ denote the event that the component A is up in phase 1. We assume that failure rates of other components remain same through all phases and $X_i = 1 (X \in \{B,C,D,E,F\})$ denotes the event that the component X is up in i-th phase. The event that the PMS is down in phase 1 is given by

$$PE_1^{(param)} = Pr[\overline{A_1^{(v_{a_1})}} \cap (\overline{B_1} \cup (\overline{C_1} \cap \overline{D_1} \cap \overline{E_1} \cap \overline{F_1}))] \tag{38}$$

In phase 2, we change the failure rate of component A to v_{a_2}. Let $A_2^{(v_{a_2})} = 1$ denote the event that the component A is up in phase 2. The event that PMS is down in phase 2, is given by

$$PE_2^{(param)} = Pr[\overline{A_2^{(v_{a_2})}} \cap (\overline{B_2} \cup (\overline{C_2} \cap \overline{D_2} \cap \overline{E_2} \cap \overline{F_2}))] \tag{39}$$

In phase 3, we again bring back the failure rate of component A to v_{a_1}. Let $A_3^{(v_{a_1})} = 1$ denote the event that the component A is up in phase 3. The event that PMS is down in phase 3, is given by

$$PE_3^{(param)} = Pr[\overline{A_3^{(v_{a_1})}} \cap (\overline{B_3} \cup (\overline{C_3} \cap \overline{D_3} \cap \overline{E_3} \cap \overline{F_3}))] \tag{40}$$

Thus, overall reliability of the system at the end of phase 3 is given by

$$R_{pms}^{(param)} = 1 - Pr[PE_1^{(param)} \bigcup PE_2^{(param)} \bigcup PE_3^{(param)}] \tag{41}$$

Observe that the CDF of time to failure of component A in third phase is given by

$$F_{A_3}(t) = 1 - [\exp(-(v_{a_1} T_1 + v_{a_2} T_2))] \exp(-v_{a_1} t) \tag{42}$$

Numerical results: Fig. 19 shows the reliability of ECCS with and without parametric change. We assumed three phases and duration of each phase was 10 h. For parametric change, during the three phases, MTTFs of component A were assumed to be: (i) 1000 h during first phase ($t=0$ to $t=10$), (ii) 50 h during second phase ($t=10$ to $t=20$) and (iii) 1000 h during third phase ($t=20$ to $t=30$). In case of normal system behavior (without any change), we assumed MTTF of A to be 1000 h during all three phases ($t=0$ to $t=30$). Reliability of the system without parametric change decreases smoothly whereas the reliability of the system with parametric change drops sharply in the beginning of second phase, when the change (increase in failure rate of node A) is applied. SHARPE input file for resiliency analysis of ECCS under parametric change is shown in Fig. 20.

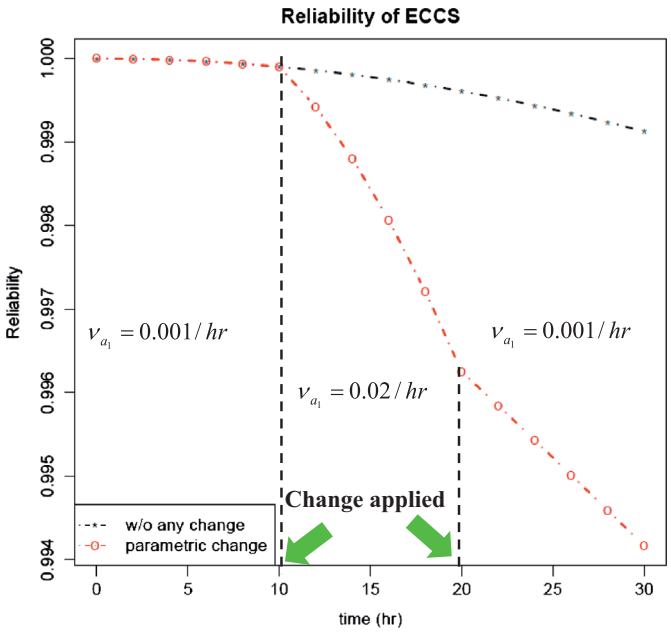

Fig. 19. Reliability of ECCS with and without parametric change.

3.1.2. Resiliency quantification of ECCS reliability upon structural changes

ECCS can also have different fault tree structures during its three phases [27,28] as shown in Fig. 21. We quantify the resiliency of reliability of ECCS w.r.t. such structural changes. We assume that failure rates of all components remain same through all phases and $X_i = 1 (X \in \{A,B,C,D,E,F,G,H\})$ denotes the event that the component X is up in ith phase. The event that the PMS is down in phase 1, is given by

$$PE_1^{(struct)} = Pr[\overline{A_1} \cap (\overline{B_1} \cup (\overline{C_1} \cap \overline{D_1} \cap \overline{E_1} \cap \overline{F_1}))] \tag{43}$$

The event that PMS is down in phase 2, is given by

$$PE_2^{(struct)} = Pr[\overline{B_2} \cup (\overline{X} \cap \overline{Y})] \tag{44}$$

where $\overline{X} = \overline{G_2} \cup \overline{C_2} \cup (\overline{D_2} \cap \overline{E_2} \cap \overline{A_2} \cap \overline{F_2})$ and $\overline{Y} = \overline{H_2} \cup \overline{D_2} \cup (\overline{C_2} \cap \overline{E_2} \cap \overline{A_2} \cap \overline{F_2})$. The event that PMS is down in phase 3, is given by

$$PE_3^{(struct)} = Pr[((\overline{G_3} \cup \overline{C_3}) \cap (\overline{H_3} \cup \overline{D_3}))] \tag{45}$$

Thus, overall reliability of the system at the end of phase 3 is given by

$$R_{pms}^{(struct)} = 1 - Pr[PE_1^{(struct)} \bigcup PE_2^{(struct)} \bigcup PE_3^{(struct)}] \tag{46}$$

The computational algorithm for such expressions for PMS models is developed in [28] and implemented in SHARPE; we omit the details here.

Numerical results: Fig. 22 shows the reliability of ECCS with and without structural change. We assumed three phases and duration of each phase was 10 h. Structural changes were applied at ends of phase 1 ($t=10$) and phase 2 ($t=20$). In case of normal system behavior (without any change), we assumed that fault tree structure remained the same during all three phases ($t=0$ to $t=30$) as shown in Fig. 21(a). Reliability of the system without structural change decreases smoothly whereas the reliability of

```
format 8

bind
a_x 0.001
b_x 0.001
c_x 0.001
d_x 0.001
e_x 0.001
f_x 0.001
**new a_x
a_x_new 0.02
T_x1 10
T_x2 10
T_x3 10
end

* model \
  description
ftree X1
basic a exp(a_x)
basic b exp(b_x)
basic c exp(c_x)
basic d exp(d_x)
basic e exp(e_x)
basic f exp(f_x)
and CDEF c d e f
or CDEFB CDEF b
and top a CDEFB
end

ftree X2
basic a \
  exp(a_x_new)
basic b exp(b_x)
basic c exp(c_x)
basic d exp(d_x)
basic e exp(e_x)
basic f exp(f_x)
and CDEF c d e f
or CDEFB CDEF b
and top a CDEFB
end

ftree X3
basic a exp(a_x)
basic b exp(b_x)
basic c exp(c_x)
basic d exp(d_x)
basic e exp(e_x)
basic f exp(f_x)
and CDEF c d e f
or CDEFB CDEF b
and top a CDEFB
end

pms X123
1 X1 T_x1
2 X2 T_x2
3 X3 T_x3
end

func \
  reliability(t) \
  1-tvalue(t;X123)

loop
t,0,T_x1+T_x2+T_x3,2
expr reliability(t)
end

end
```

Fig. 20. SHARPE input file for resiliency analysis of ECCS under parametric change.

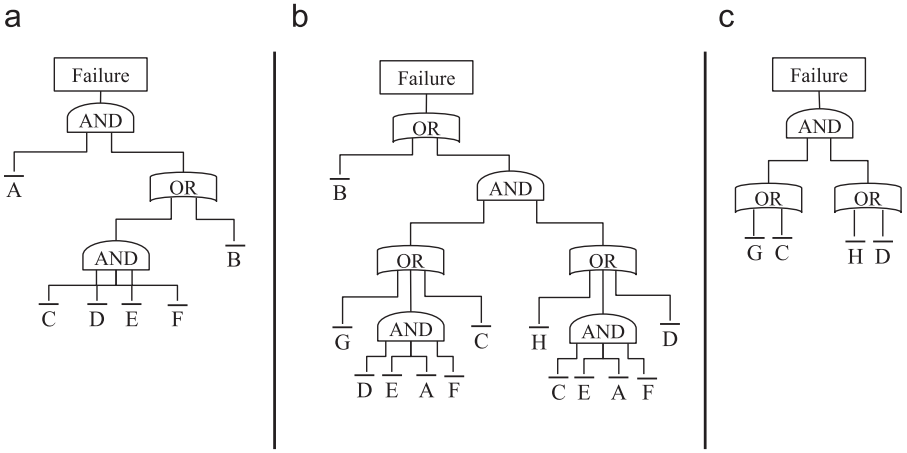

Fig. 21. Fault trees of ECCS with different structures during three phases.

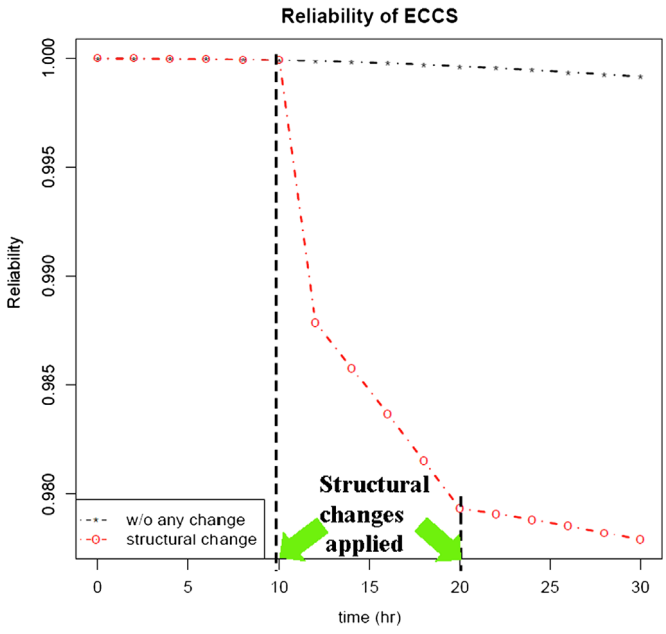

Fig. 22. Reliability of ECCS with and without structural change.

the system with structural change drops sharply when the change is applied ($t=10$ and $t=20$). SHARPE input file for resiliency analysis of ECCS under structural change is shown in Fig. 23.

3.2. Resiliency quantification for telephone switching system

3.2.1. Resiliency quantification of system performance measure

We quantify the resiliency of call blocking probability as obtained from the performance model described in Fig. 4.

Study the normal behavior of the system before the change is applied. We assume that, the initial call arrival rate is $\lambda^{(o)}$. Under this condition, steady state probability vector $\boldsymbol{\pi}^{(o)}$ for the CTMC in Fig. 4 is denoted by

$$\boldsymbol{\pi}^{(o)} = [\pi_0^{(o)}, \pi_1^{(o)}, \ldots, \pi_n^{(o)}] \tag{47}$$

When $\lambda = \lambda^{(o)}$, let the generator matrix of the CTMC be $\boldsymbol{Q}_{\boldsymbol{perf}}^{(\boldsymbol{o})}$. After solving the system of linear equations

$$\boldsymbol{\pi}^{(\boldsymbol{o})}\boldsymbol{Q}_{\boldsymbol{perf}}^{(\boldsymbol{o})} = 0 \tag{48}$$

with $\sum_{k=0}^{n} \pi_k^{(o)} = 1$, we can compute the steady state probabilities. Thus, steady state blocking probability (P_b) can be obtained as the value of $\pi_n^{(o)}$.

Apply a change to the system. Call arrival rate is set to $\lambda^{(w)}$.

```
format 8

bind
a_x 0.001
b_x 0.001
c_x 0.001
d_x 0.001
e_x 0.001
f_x 0.001
a_y 0.001
b_y 0.001
c_y 0.001
d_y 0.001
e_y 0.001
f_y 0.001
g_y 0.001
h_y 0.001
a_z 0.001
b_z 0.001
c_z 0.001
d_z 0.001
e_z 0.001
f_z 0.001
g_z 0.001
h_z 0.001
T_x 10
T_y 10
T_z 10
end
* Model description
ftree X
basic a exp(a_x)
basic b exp(b_x)
basic c exp(c_x)
basic d exp(d_x)
basic e exp(e_x)
basic f exp(f_x)
and CDEF c d e f
or CDEFB CDEF b
and top a CDEFB
end

ftree Y
basic a exp(a_y)
basic b exp(b_y)
basic c exp(c_y)
basic d exp(d_y)
basic e exp(e_y)
basic f exp(f_y)
basic g exp(g_y)
basic h exp(h_y)
and DEAF d e a f
or GDEAFC g DEAF c
and CEAF c e a f
or HCEAFD h CEAF d
and and1 GDEAFC HCEAFD
or top b and1
end
ftree Z
basic c exp(c_z)
basic d exp(d_z)
basic h exp(h_z)
basic g exp(g_z)
or or1 g c
or or2 h d
and and1 or1 or2
end

pms XYZ
1 X T_x
2 Y T_y
3 Z T_z
end

* Output measure
func reliability(t) \
   1-tvalue(t;XYZ)
loop \
   t,0,T_x+T_y+T_z,2
expr reliability(t)
end

end
```

Fig. 23. SHARPE input file for resiliency analysis of ECCS under structural change.

Study the transient behavior of the system after the change is applied. When $\lambda=\lambda^{(w)}$, let the transient state probability vector $\boldsymbol{\pi}^{(w)}(t)$ of the CTMC in Fig. 4 be denoted by

$$\boldsymbol{\pi}^{(w)}(t)=[\pi_0^{(w)}(t),\pi_1^{(w)}(t),\ldots,\pi_n^{(w)}(t)] \tag{49}$$

Let us denote the new generator matrix as $\boldsymbol{Q}_{perf}^{(w)}$ and the time derivative of the transient state probability vector by

$$\frac{d\boldsymbol{\pi}^{(w)}(t)}{dt}=\left[\frac{d\pi_0^{(w)}(t)}{dt},\frac{d\pi_1^{(w)}(t)}{dt},\ldots,\frac{d\pi_n^{(w)}(t)}{dt}\right] \tag{50}$$

Thus, the Kolmogorov differential equation for transient analysis is

$$\frac{d\boldsymbol{\pi}^{(w)}(t)}{dt}=\boldsymbol{\pi}^{(w)}(t)\boldsymbol{Q}_{perf}^{(w)} \tag{51}$$

For this transient analysis, initial state probability vector is obtained from the steady state probabilities before the change is applied ($\lambda=\lambda^{(o)}$). Thus, initial state probability vector is given by

$$\boldsymbol{\pi}^{(w)}(0)=[\pi_0^{(o)},\pi_1^{(o)},\ldots,\pi_n^{(o)}] \tag{52}$$

By solving Eq. (51) with the initial probability vector in Eq. (52), we compute the transient call blocking probability after the change is applied.

Numerical results: Fig. 24 shows the resiliency of call blocking probability of a switching system with 50 channels. At $t=0$, call arrival rate is changed from 80 calls/h to 100 calls/h. Call arrival rate is again brought back to 80 calls/h at $t=5$. Fig. 24 shows that a switching system with higher service rate is more resilient as the relative change in call blocking probability is lower when call arrival rate is changed. SHARPE input file for Fig. 24 is shown in Fig. 25.

3.2.2. Resiliency quantification of system availability (without fault detection/reconfiguration delays)

We quantify the resiliency of unavailability as computed from the CTMC in Fig. 7. Mathematically we describe resiliency analysis built around this model.

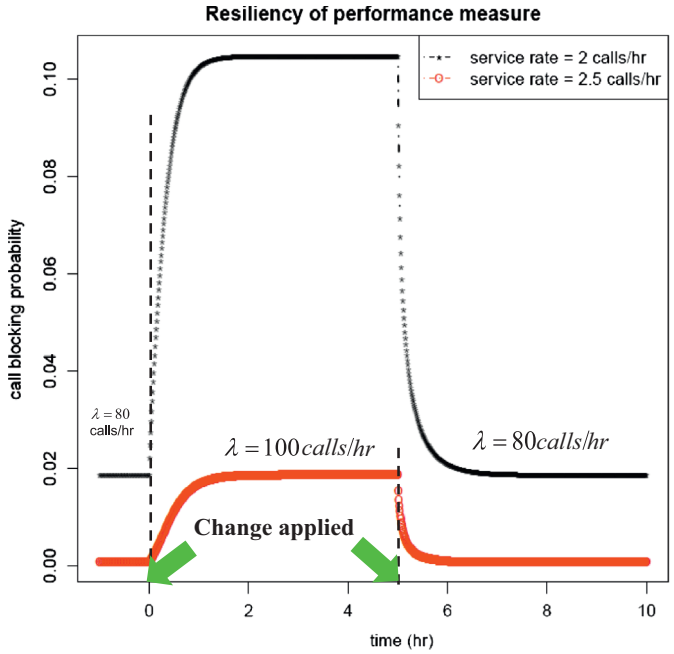

Fig. 24. Resiliency of call blocking probability from performance model.

Study the normal behavior of the system before the change is applied. We assume that, initial channel failure rate is $\gamma^{(o)}$. Under this condition, steady state probability vector $\boldsymbol{\phi}^{(o)}$ is first computed in the usual way by solving the steady state balance equations.

Apply a change to the system. Channel failure rate is set to $\gamma^{(w)}$.

Study the transient behavior of the system after the change is applied. Now, the transient state probability vector $\boldsymbol{\phi}^{(w)}(t)$ is computed by solving the Kolmogorov differential equations with the initial state probability vector $\boldsymbol{\phi}^{(o)}$.

Numerical results: Fig. 26 shows the resiliency of unavailability for a switching system with 50 channels. At $t=0$, channel failure rate is changed from 0.001/h to 0.01/h. Channel failure rate is

```
format 8

bind
lambda_init 80
lambda_new 100
mu    2
nb 50
t 0
t_init 0
t_final 5.0
time_step 0.01
end

* model description
markov perf(n,lambda)
loop i,0,n-1
$(i) $(i+1) lambda
$(i+1) $(i) (i+1)*mu
end
end
$(0) 1
end

markov \
    perf_trans(n,lambda_new,\
    lambda_init)
loop i,0,n-1
$(i) $(i+1) lambda_new
$(i+1) $(i) (i+1)*mu
end
end
loop i,0,n
* Steady-state probability\
    of perf model
$(i) prob(perf,$(i); \
    n,lambda_init)
end
end

markov
    perf_trans2(n,lambda_new, \
    lambda_init, t_change)
loop i,0,n-1
$(i) $(i+1) lambda_init
$(i+1) $(i) (i+1)*mu
end
end
loop i,0,n
* Transient probability of \
    perf_trans model
$(i) tvalue(t_change; \
    perf_trans,$(i);n,lambda_new, \
    lambda_init)
end
end

func Pb_res(n,lambda_new, \
    lambda_init,t) \
    tvalue(t;perf_trans,$(n); \
    n,lambda_new, lambda_init)

func Pb_res2(n,lambda_new, \
    lambda_init,t,t_change) \
    tvalue(t;perf_trans2,$(n); \
    n,lambda_new, lambda_init,
    t_change)

bind   t     t_init

while (t <= t_final)
bind   block_prob \
    Pb_res(nb,lambda_new,lambda_init,t)
expr   t
expr   block_prob
bind   t     (t+time_step)
end

bind   t_change   t_final
bind   t     t_final
bind   time  time_step

while  (time <= t_final)
bind   block_prob Pb_res2(nb, \
    lambda_new,lambda_init,time, \
    t_change)
bind   t     (t+time_step)
expr   t
expr   block_prob
bind   time  (time+time_step)
end

end
```

Fig. 25. SHARPE input file for Fig. 24.

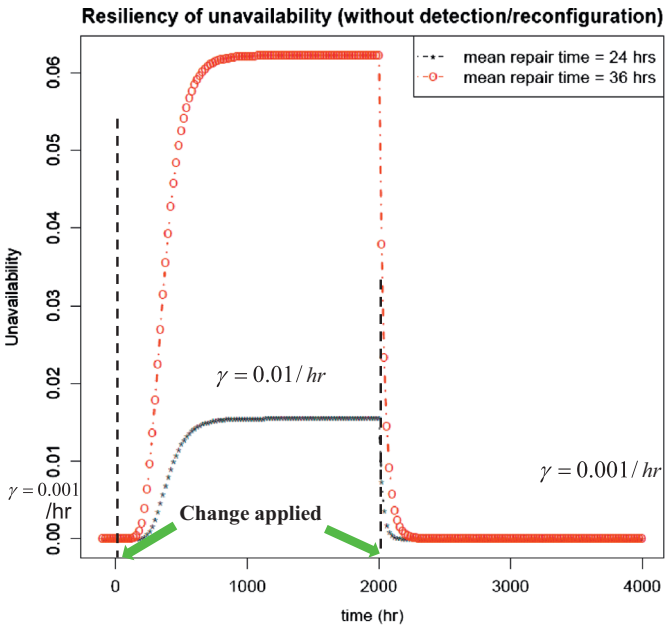

Fig. 26. Resiliency of unavailability of switching system (without fault detection/reconfiguration delays).

again brought back to 0.001/h at $t=2000$. Fig. 26 shows that a switching system with higher repair rate is more resilient as the relative change in unavailability is lower when channel failure rate is changed. SHARPE input file for Fig. 26 is shown in Fig. 27.

3.2.3. *Resiliency quantification of system availability (with fault detection/reconfiguration delay)*

We quantify the resiliency of unavailability as computed from the CTMC in Fig. 10.

Study the normal behavior of the system before the change is applied. We assume that, initial channel failure rate is $\gamma^{(o)}$. Under this condition, steady state probability vector $\boldsymbol{\psi}^{(\mathbf{o})}$ is computed by solving the steady state balance equations. The steady state unavailability is then computed by

$$UA^{(o)}_{a\_detect} = e^{-\delta t_{th}} \sum_{k=1}^{n} \psi^{(o)}_{S_k} + \psi^{(o)}_{0} \tag{53}$$

Apply a change to the system. Channel failure rate is set to $\gamma^{(w)}$.

Study the transient behavior of the system after the change is applied. Now, the transient state probability vector $\boldsymbol{\psi}^{(\mathbf{w})}(\boldsymbol{t})$ is computed by solving the Kolmogorov differential equation with the initial state probability vector $\boldsymbol{\psi}^{(\mathbf{0})}$.

Numerical results: Fig. 28 shows the resiliency of unavailability with of a 50 channel switching system with fault detection/

```
1.  format 8
2.
3.  bind
4.  gamma_init  1/1000
5.  gamma_new   1/100
6.  tau    1/24
7.  nc 50
8.  t 0
9.  t_init 0
10. t_final 2000
11. time_step 20
12. end
13.
14. * model description
15. markov avail(n,gamma)
16. loop i,n,1,-1
17.    $(i) $(i-1) i*gamma
18.    $(i-1) $(i) tau
19. end
20. end
21. $(n)  1
22. end

23. markov \
    avail_trans(n,gamma_new,\
    gamma_init)
24. loop i,n,1,-1
25. $(i) $(i-1) i*gamma_new
26. $(i-1) $(i) tau
27. end
28. end
29. loop i,0,n
30. $(i) prob(avail,$(i);n,
31. gamma_init)
32. end
33. end
34.
35. markov
    avail_trans2(n,gamma_new, \
    gamma_init, t_change)
36. loop i,n,1,-1
37. $(i) $(i-1) i*gamma_init
38. $(i-1) $(i) tau
39. end
40. end
41. loop i,0,n
42. $(i)  tvalue(t_change; \
    avail_trans,$(i);n, gamma_new, \
    gamma_init)
43. end
44. end

45. *output measures
46. func UAt(n, gamma_new, gamma_init,\
    t) tvalue(t;avail_trans,0;n, \
    gamma_new, gamma_init)

47. func UAt2(n, gamma_new, \
    gamma_init, t, t_change) \
    tvalue(t; avail_trans2, 0; n, \
    gamma_new, gamma_init, t_change)

48. bind   t    t_init
49.
50. while (t <= t_final)
51. bind    unavail \
    UAt(nc, gamma_new,gamma_init,t)
52. expr   t
53. expr   unavail
54. bind   t    (t+time_step)
55. end

56. bind   t_change   t_final
57. bind   t    t_final
58. bind   time  time_step
59.
60. while  (time <= t_final)
61. bind    unavail      UAt2(nc, \
    gamma_new, gamma_init,time, \
    t_change)
62. bind   t    (t+time_step)
63. expr   t
64. expr   unavail
65. bind   time  (time+time_step)
66. end

67. end
```

Fig. 27. SHARPE input file for Fig. 26.

reconfiguration delay. At $t=0$, channel failure rate is changed from 0.001/h to 0.01/h. Channel failure rate is again brought back to 0.001/h at $t=0.1$. MTTR of a channel was assumed to be 24 h through out the entire period ($t=0$ to $t=0.2$). Fig. 28 shows that a switching system with longer value of detection/reconfiguration threshold (t_{th}) is more resilient as the relative change in unavailability is lower when channel failure rate is changed.

3.2.4. Resiliency quantification of system performability measure (without detection/reconfiguration delay)

We quantify the resiliency of total call blocking probability from the performability model shown in Fig. 13. Mathematical steps for such analysis are described here.

Study the normal behavior of the system before the change is applied. We assume that, initial call arrival rate is $\lambda^{(op)}$. Steady state probability vector $\boldsymbol{\theta}^{(\boldsymbol{op})}$ is first computed by solving the steady state balance equations. The steady state call blocking probability is given by

$$T_b = \sum_{i=0}^{n} \theta_{(i,i)}^{(op)} \tag{54}$$

Apply a change to the system. Call arrival rate is set to $\lambda^{(wp)}$.

Study the transient behavior of the system after the change is applied. Now, the transient state probability vector $\boldsymbol{\theta}^{(\boldsymbol{wp})}(\boldsymbol{t})$ is computed by solving the Kolmogorov differential equations with the initial state probability vector $\boldsymbol{\theta}^{(\boldsymbol{op})}$.

Numerical results: Fig. 29 shows the resiliency of total call blocking probability from performability model with five channels. At $t=0$, call arrival rate is changed from 5 calls/h to 10 calls/h. Call arrival rate is again brought back to 5 calls/h at $t=5$. Through out our analysis, we assumed channel failure rate to be 0.001/h and mean repair time to be 24 h. Telephone switching system with higher service rate is more resilient as the relative change in call blocking probability is lower when call arrival rate is increased.

3.2.5. Resiliency quantification of system performability measure (with detection/reconfiguration delay)

We describe mathematical steps to quantify the resiliency of total call blocking probability from the performability model shown in Fig. 16.

Study the normal behavior of the system before the change is applied. We assume that initial call arrival rate is $\lambda^{(op)}$. Steady state probability vector $\boldsymbol{\omega}^{(\boldsymbol{op})}$ for such condition is computed by solving the steady state balance equations. The steady state call blocking probability is given by

$$T_{b\_detect} = \sum_{i=0}^{n} \omega_{(i,i)}^{(op)} + \sum_{p=1}^{n} \sum_{q=0}^{p-1} \omega_{(S_p,q)}^{(op)} \tag{55}$$

Apply a change to the system. Call arrival rate is set to $\lambda^{(wp)}$.

Study the transient behavior of the system after the change is applied. Now, the transient state probability vector $\boldsymbol{\omega}^{(\boldsymbol{wp})}(\boldsymbol{t})$ is

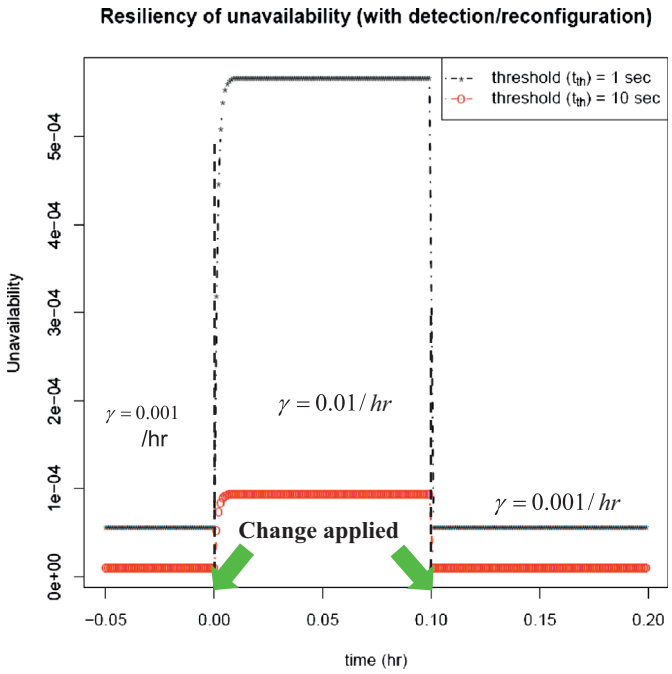

Fig. 28. Resiliency of unavailability of switching system (with detection/reconfiguration delay).

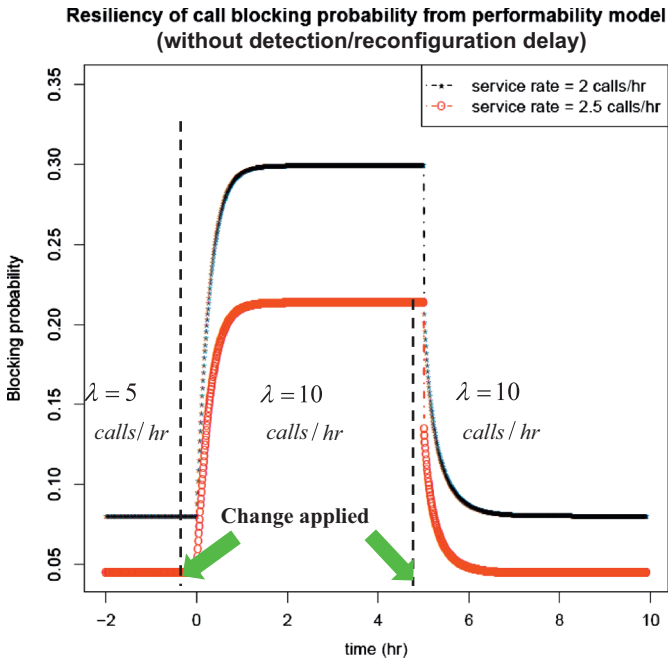

Fig. 29. Resiliency of total call blocking probability from performability model (without detection/reconfiguration delay).

computed by solving the Kolmogorov differential equations with the initial state probability vector $\omega^{(op)}$.

Numerical results: Fig. 30 shows the resiliency of total call blocking probability from performability model with five channels, when detection/reconfiguration delay is non-zero. At $t=0$, call arrival rate is changed from 5 calls/h to 10 calls/h. Call arrival rate is again brought back to 5 calls/h at $t=5$. Through out our analysis, we assumed channel failure rate to be 0.001/h and mean repair time to be 24 h. Telephone switching system with higher service rate is more resilient as the relative change in call blocking probability is lower when call arrival rate is increased.

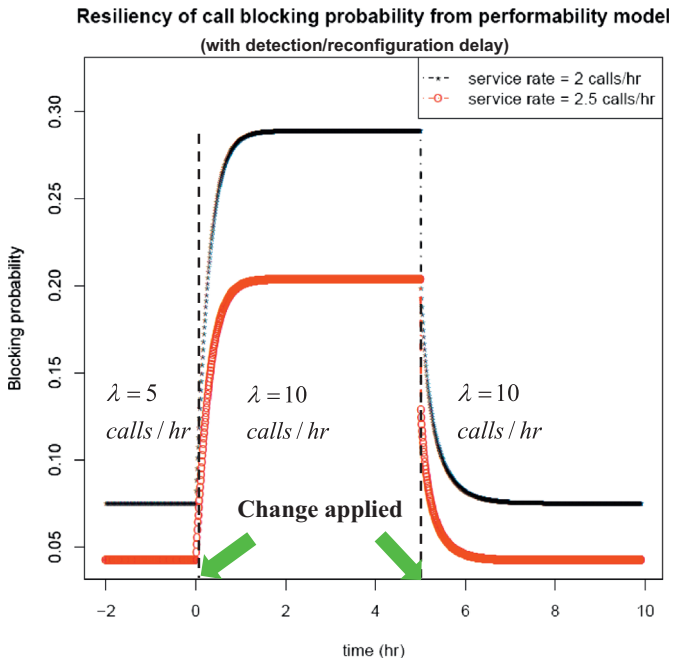

Fig. 30. Resiliency of total call blocking probability from performability model (with detection/reconfiguration delay).

4. Conclusions and future work

In this paper, we have presented a systematic approach for resiliency quantification using non-state-space and state-space based stochastic analytic models. We have shown the resiliency of: (i) system reliability of emergency core cooling system (ECCS) of a boiling water reactor (BWR) using non-state-space models and (ii) system performance, availability and performability measures of telephone switching system using state-space models. Impact of both structural and parametric changes on system resiliency are shown. For the examples shown, computations of different resiliency metrics, as described in our other work [30], are left as future research.

Acknowledgment

Authors were supported in part under a 2010 IBM Faculty Award and in part by NSF under Grant NSF-CNS-08-31325. This work was completed during Rahul's internship at IBM T.J. Watson Research Center.

References

[1] Resilience 〈http://en.wikipedia.org/wiki/Resilience〉.
[2] Haverkort B, Marie R, Rubino G, Trivedi KS, editors. Performability modeling tools and techniques. Wiley; 2001.
[3] Kulkarni VG, Nicola VF, Trivedi KS. The completion time of a job on multi-mode systems. Advances in Applied Probability 1987;19:932–54.
[4] Laprie JC. From dependability to resilience. In: DSN; 2008.
[5] Simoncini L. Resilient computing: an engineering discipline. In: IPDPS; 2009.
[6] Dearnley PA. An investigation into database resilience. The Computer Journal 1976;19(2):117–21.
[7] Najjar WA, Gaudiot J-L. Network resilience: a measure of network fault tolerance. IEEE Transactions on Computers 1990;39(2):174–81.
[8] Chialastri A, Pozzi S. Resilience in the aviation system. In: SAFECOMP; 2008.
[9] Frankel Y, Gemmell P, MacKenzie PD, Yung M. Optimal resilience proactive public-key cryptosystems. In: FOCS; 1997.
[10] Cholda P, Tapolcai J, Cinkler T, Wajda K, Jajszczyk A. Quality of resilience as a network reliability characterization tool. IEEE Network 2009;23(2):11–9.
[11] Liu G, Ji C. Scalability of network-failure resilience: analysis using multi-layer probabilistic graphical models. IEEE/ACM Transactions on Network 2009;17(1):319–31.

[12] Touvet F, Harle D. Network resilience in multilayer networks: a critical review and open issues. In: ICN; 2001.
[13] Sharafat AR, Fallah MS. A measure of resilience against denial of service attacks in computer networks. Computer Systems Science and Engineering 2002;17(4/5):259–67.
[14] Lee K-W, Chari S, Shaikh A, Sahu S, Cheng P-C. Improving the resilience of content distribution networks to large scale distributed denial of service attacks. Computer Networks 2007;51(10):2753–70.
[15] Alarifi A, Du W. Diversify sensor nodes to improve resilience against node compromise. In: SASN; 2006.
[16] Kjeldsen T, Rosbjerg D. Choice of reliability, resilience and vulnerability estimators for risk assessments of water resources systems. Hydrological Sciences Journal 2004;49(5):755–67.
[17] Khalil Y, Elmaghraby A, Kumar A. Evaluation of resilience for data center systems. In: ISCC; 2008.
[18] Debardeleben N, Laros J, Daly J, Scott S, Engelmann C, Harrod B. High-end computing resilience: analysis of issues facing the HEC community and path-forward for research and development. Position paper; 2010.
[19] Engelmann C, Leangsuksun C. Modeling techniques towards resilience. In: National HPC workshop on resilience; 2009.
[20] Sterbenz JP, et al. Resilience and survivability in communication networks: strategies, principles, and survey of disciplines. Elsevier Computer Networks 2010;54(8):1245–65.
[21] Leangsuksun C. A call for standardization of resilience in high performance computing. In: Workshop on resilience in HPC; 2010.
[22] Tuffin B, Choudhary PK, Hirel C, Trivedi KS. Simulation versus analytic-numeric methods: a petri net example. In: VALUETOOLS; 2007.
[23] Trivedi KS, Kim DS, Roy A, Medhi D. Dependability and security models. In: DRCN; 2009.
[24] Nicol DM, Sanders WH, Trivedi KS. Model-based evaluation: from dependability to security. IEEE Transactions on Dependable and Secure Computing 2004;1(1):48–65.
[25] Trivedi KS, Ghosh R, Kim DS. An analytic approach for resiliency quantification of systems. In: MMR; 2011.
[26] Trivedi KS, Sahner R. SHARPE at the age of twenty-two. ACM Sigmetrics Performance Evaluation Review 2009;36(4):52–7.
[27] Burdick GR, Fussell JB, Rasmuson DM, Wilson JR. Phased mission analysis: a review of new developments and an application. IEEE Transactions on Reliability 1977;R-26(1):43–9.
[28] Ma Y, Trivedi KS. An algorithm for reliability analysis of phased-mission systems. Reliability Engineering and System Safety 1999;66(2):157–70.
[29] Trivedi KS. Probability and statistics with reliability, queuing and computer science applications; 2001. Wiley.
[30] Ghosh R, Longo F, Naik VK, Trivedi KS. Quantifying resiliency of IAAS cloud. In: SRDS RACOS; 2010.