Nuclear Engineering and Technology 52 (2020) 560–567

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

Performance evaluation of safety-critical systems of nuclear power plant systems

Pramod Kumar [a, *], Lalit Kumar Singh [b], Chiranjeev Kumar [a]

[a] Department of Computer Science & Engineering, Indian Institute of Technology (ISM), Dhanbad, India
[b] Department of Computer Science & Engineering, Indian Institute of Technology (BHU), Varanasi, India

ARTICLE INFO

Article history:
Received 12 February 2019
Received in revised form
19 August 2019
Accepted 20 August 2019
Available online 23 August 2019

Keywords:
Performance
Safety critical systems
Petri nets
Markov chain

ABSTRACT

The complexity of safety critical systems of Nuclear Power Plant continues to increase rapidly due its transition from analog to digital systems. It has thus become progressively more imperative to model these systems prior to their implementation in order to meet the high performance, safety and reliability requirements. Timed Petri Nets (TPNs) have been widely used to model such systems for non-functional analysis. The paper presents a novel methodology for the analysis of the performance metrics using PN modeling. The paper uses the isomorphism property of the TPNs and the Markov chains for the performance analysis of the safety critical systems. The presented methodology has been validated on a Shutdown System of a Nuclear Power Plant.

1. Introduction

safety-critical control systems (SCCS) like nuclear power plants (NPP), aircrafts, medical systems etc. are targeted to fulfill high performance, reliability and safety requirements, as the failure of such systems may cause huge economy loss, threat to lives, loss of goal-oriented mission, extensive ecological damage, etc. Therefore, these systems must be modeled prior to their implementation for the assessment of the above dependability metrics.

The Instrumentation and Control (I&C) of a NPP should be appropriately planned, designed, constructed and maintained in order to enable the human operator to take judicious action during abnormal operations. Thus, I&C along with the human operator form the central link for the safe and efficient operation of the NPP. Different logic circuits ensure the protection and safety of the NPP in case of any abnormal conditions. Some of the important I&C logic circuits providing the protection and engineered safety system performance in NPP are: Emergency shutdown system, Initiation of auxiliary feedwater system, Steamline isolation, Isolation of normal feedwater lines, and Initiation of safety injection system.

Performance of any SCS can be described by four elements namely: capability, efficiency, reliability and availability [1]. The performance of safety critical control systems has to be simulated with the proper operating conditions. The simulators guarantee the NPP performance to match the “as-designed” plant. However, “fine-tuning” the simulator is time consuming, expensive and introduces technological challenges. Moreover, simulation can be termed as an art instead of an exact science. As per International Atomic Energy Agency (IAEA) recommendations, the following desirable characteristics should be possessed by any NPP I&C simulator for performance measurement:

- The error of the simulator instrument should be less than or equal to that of the transducer, comparable meter and related instrument system of the reference plant [2].
- The principal energy and mass balances shall be satisfied. The values like steady state, full power and automatic control operation as computed by the simulator shall not drift by more than ±2% over 1 h time period [2].
- The critical parameters values computed by simulator shall be within ±2% with respect to the reference plant [3].
- The non-critical parameters values relevant to plant operation shall be within ±10% with the reference plant. The response of the simulator resulting from any action (operator, no operator and improper operator), automatic controls and inbuilt operating characteristics shall be realistic within the limit of the performance criteria [4].

* Corresponding author.
E-mail addresses: pramod.16dr000212@cse.ism.ac.in (P. Kumar), lalit.rs.cse@iitbhu.ac.in (L.K. Singh), chiranjeev@iitism.ac.in (C. Kumar).

https://doi.org/10.1016/j.net.2019.08.018

Acronyms and abbreviations

TPN	Timed Petri Nets
SC(C)S	Safety Critical (and Control) Systems
NPP	Nuclear Power Plants
I&C	Instrumentation and Control
IAEA	International Atomic Energy Agency
SDS	Shutdown System
PN	Petri Nets
UML	Unified Modeling Language
FAV	Fast Acting Valves
LC	Logic Condition
DTMC	Discrete Time Markov Chain

Notations

p_n	Places
t_n	Transitions
p_1	Trip parameters deviates
p_2	Creates LC
p_3	Holds LC
p_4	Restores LC
p_5	Relays energize to close the vent valves
p_6	Redundant information of FAV in closed state
p_7	Vent valves closed
p_8	FAV closed
p_9	FAV opened
p_{10}	Relays de-energize to open the vent valves
p_{11}	Redundant information of FAV in open state
p_{12}	Opens the Vent valves
t_1	Send signal to create LC and to energize the relays to close the vent valves
t_2	Triggers signal to hold LC in created state
t_3	Triggers signal to restore LC and de-energize the relays to open the vent valves
t_4	Triggers to close the vent valves
t_5	Triggers to open all FAV
t_6	Triggers to close all FAV
t_7	Triggers to open the vent valves

- There should not be a difference between the response of the simulator and the reference plant [4].

The performance of a SCCS using a simulator requires setting up of proper operating conditions as far as practicable. Also, I&C maintenance department maintains and updates the documentation relating to I&C equipment and the systems from time to time. So, the simulator vendor must have the updated knowledge of the system. In addition to this, full-scope NPP training simulators are very expensive (may cost in multi-million dollar). The simulators for NPP performance measuring still provide a simulation of the power plant and are not capable of substituting the actual plant expertise.

Paying attention to the above said issues, there is a need of devising a methodology which is capable of assessing the performance of the SCCS irrespective of the above existing issues. Deterministic as well as stochastic models are capable for performance evaluation of any system. But due to some unrealistic assumptions like mission completion times, mission arrival times and synchronization level etc. the deterministic models fail to accurately calculate the performance. However, if the above said constraints are available deterministic models can estimate the performance very accurately. For the case of stochastic models the above said constraints can be very easily calculated using any of the existing probability distribution functions. This paper contributes a novel approach to address the problem of performance assessment of a SCCS using a simulator and presents an effective methodology for the performance quantification of SCS of NPP systems using state space modeling using isomorphism property of the TPN and the Markov chains. SCS of NPP are designed and developed using standards and are smaller in size as compared with other software systems. Use of TPN for performance evaluation, task execution times, task arrival times, etc. are usually determined by probability distribution functions. Moreover in case of SCCS, the synchronization among the tasks can also be modeled because such systems are very small in size and hence there state space is very small. The benefit of using TPN is its capability of predicting the performance at the early stages of system development when all the system characteristics are not known and well understood. This predicted value saves significant effort and avoid delay in system development. In addition to this, use of TPN allows the incorporation of timing information into the analysis-a necessity for real-time life critical system analysis [5]. The most critical function of software can be determined using PN which can then be augmented with fault tolerance facilities and to determine the conditions which must be incorporated into the run-time tests associated with these facilities such as watchdog timers and acceptance tests in recovery blocks. Thus, this technique is useful for life critical applications, where performability requirements are very high.

The organization of the paper is as follows. In the following section, existing approaches with their shortcomings have been briefly recalled. Section III discusses the PN modeling process and its basic terminology. Section IV presents the performance analysis and its evaluation framework. The case study: Shutdown System (SDS-2) and its PN model is discussed in Section V. Section VI show the calculation and validation of the performance of SDS-2. Section VII concludes the paper.

2. Related works

SCCS of NPP always need attention due their critical function. These safety systems are designed and developed using standards and are smaller in size. Also, these are kept simple and have minimal interfaces [6]. Researchers have proposed many approaches for performance evaluation based on response time, throughput and resource utilization for software systems. Both the performance prediction and measurement are dealt in these approaches. Performance prediction deals with the expected performance of a system to avoid performance problems when system is in implementation stage, which can lead to substantial costs for redesigning the system architecture again. The Performance measurement analyses the observable performance of implemented and running systems to understand its performance properties, to determine their maximum capacity, identify performance-critical components, and to remove performance bottlenecks [7]. Performance of any system can be defined as the total effectiveness of a system including availability, throughput and individual response time. Some recent studies [8–14] show that researchers are focusing on both reliability and performance of software application of SCS. However, no validation has been shown to prove the effectiveness of the approach.

Lalit et al. [9] proposed an approach for system modeling and design verification of instrumentation and control systems of NPP. The paper estimates the reliability of embedded unit of NPP. It also

presents a performance estimation approach. However, the performance has not been evaluated using any data set. It is necessary to do performance analysis of SCS for its dependability.

Lalit et al. [10] presented methodology for modeling and prediction of performability of SCS using PN. The methodology is demonstrated on a case study of SCS of NPP. It addresses the dynamic modeling of Test Facility of a SCS used in NPP. However, the methodology relies completely on Time NET tool for the calculation and hence cannot consider the component interfaces correctly. Also, the presented method does not allow more than one simultaneously enabled generally distributed transition.

Liu et al. [13] proposed a deterministic stochastic Petri net model for performance evaluation of a subsea blown-out prevention system. In this approach, the system is partitioned into two subsystems: (i) mechanical system (ii) computer-based system, in order to obtain the availability and reliability of the system. Component failure rate and their repair time on the overall system performance are also analyzed. However, the authors assume that failure rate of the component is constant which does not hold true in case of SCS. Moreover, SCS of NPP are designed and developed using standards and are smaller in size as compared with other software. Also, they are kept simple and have minimal interfaces with other components. Therefore, such software confronts very less number of failures during testing or operational phase. Because of non-sufficiency of failure data and unrealistic assumptions, the existing models are incapable to quantify the performance of such software.

Wang et al. [15] extended Cheung's [16] work and combined performance and reliability analysis in order to support different architectural styles. However, they rely on the operational profile and testing data or the software architecture's intuition to for the prediction of the performance.

Merseguer et al. [17] establishes an association between Unified Modeling Language (UML) and labeled generalized stochastic Petri nets (LGSPN) in two steps. Each UML state machine diagram is converted into the corresponding LGSPN and then, the Petri nets thus obtained are coupled according to the information given in the UML sequence and use case diagrams that also contain the performance specifications. The exploitation of the compositional technique yields the final PN model. However, the limitation of such an approach lies in the assumption of infinite resources by disregarding the hardware influence.

Sharma and Trivedi [18] propose an architecture-based unified hierarchical model for software performance, reliability, security and cache behavior prediction using discrete DTMC. However, the Markov model suffers from state-space explosion problem [19]. Also, it is difficult to model concurrency of control flow using a DTMC model. Paying attention to the shortcomings of the different modeling techniques stated above for the performance analysis there exists a gap of a methodology which is robust in nature. The presented approach for performance evaluation overcomes the entire stated shortcoming.

3. Modeling with Petri Nets

Petri Nets [20] are directed bipartite graphical and mathematical modeling tool used for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic. Mathematically, PN can be defined by a 5-tuple, $PN = \{P, T, F, W, M_0\}$ where $P = \{p_1, p_2, p_3, ..., p_m\}$ signifies a finite set of places, $T = \{t_1, t_2, ..., t_n\}$ finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs, $W : F \rightarrow \{1, 2, 3, \ldots\}$ is weight function, $M_0 : P \rightarrow \{0, 1, 2, \ldots\}$ is the initial marking, and $P \cap T = \phi$ and $P \cup T \neq \phi$.

The dynamic behavior of a SCS is modeled using the execution of a process, represented by firing of the subsequent transition. The movements of tokens in the net represent the changes in the system state. PN firing rules are as follows:

1. A transition T is enabled at a marking M if and only if when $\forall\ p \in^{\circ} t,\ M(p) \geq W(p,t)$, where $p \in^{\circ} t$ signifies the input place of t, $M(p)$ is the number of tokens in place p, $W(p,t)$ is the weight of the arc from p to t.
2. Only an enabled transition can fire.
3. Firing of a transition:
   i. Removes a token from each of its input places; and
   ii. Deposits a token to each of its output places.

Fig. 1 shows the execution of a PN modeled system. Fig. 1a shows that one token is available with places A so transition T_1 is enabled. After firing T_1, the net changes to the one shown in Fig. 1b. The firing of transition T_1 removes the token from places A, and deposits them to the output place B and C. In Fig. 2b, place C is having one token. So, transition T_2 is enabled. (It should be noted that place B is also having one token and connected to transition T_3. But transition T_3 is disabled at this point because one of its input places D is not having any token in it.) Firing of transition T_2 changes the marking to the one shown in Fig. 1c. Now, at this point transition T_3 is enabled and its firing leads to returning the net to its initial configuration i.e. Fig. 1a.

Reachability: The dynamic property of a system cannot be studied without reachability. A marking M_n is reachable from another marking M_1 if there is a sequence of firings transforming M_1 to M_n. A firing sequence of transition is represented by $\sigma = M_1 t_1 M_2 t_2 M_3 \ldots tn\ M_n$ or simply $\sigma = t_1 t_2 \ldots tn$.

Isomorphism [21]: Two transition systems $A_1 = (S_1, E_1, T_1)$ and $A_2 = (S_2, E_2, T_2)$ where S, E and T respectively denote the state, event and transition are said to be isomorphic iff there exists two bijections, $\beta : S_1 \rightarrow S_2$ and $\eta : E_1 \rightarrow E_2$ such that:

$$1)\quad \forall\ (s,e,s')\ \varepsilon\ T_1 \qquad (\beta(s),\ \eta(e),\ \beta(s'))\varepsilon\ T_2$$
$$2)\quad \forall\ (s,e,s')\ \varepsilon\ T_2 \quad \left(\beta^{-1}(s),\ \eta^{-1}(e),\ \beta^{-1}(s')\right)\varepsilon\ T_1$$

4. Performance analysis and evaluation

This section addresses the performance analysis and its evaluation. In order to analyze and evaluate the performance of any system modeled using PN it needs to be consistent. Ramamoorthy

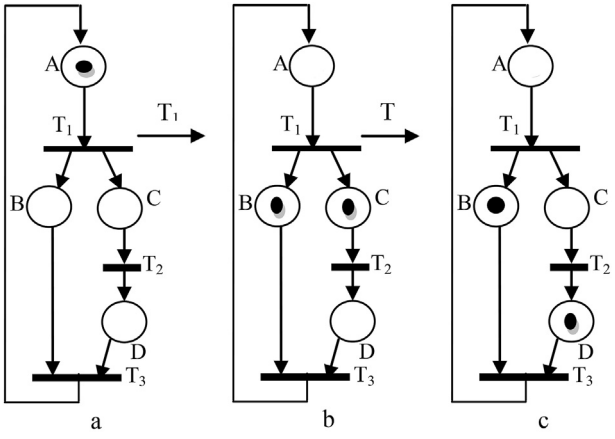

Fig. 1. Petri Net execution.

and Gary [22] classified the PN systems as consistent and inconsistent. Consistent PNs were further classified as decision-free PN, safe persistent PN and general PN. *Decision-free PNs* are those which have exactly one input and exactly one output arc connected to each place i.e. $\forall p \in P, |^{\circ}p| = |p^{\circ}| = 1$. *Safe persistent PNs* are those which are safe for all the reachable markings. Unlike decision-free nets, it may have any number of input or output arcs to or from a place i.e. $\exists p \in P, |^{\circ}p| > 1$ *or* $|p^{\circ}| > 1$. *General PNs* are consistent nets having reachable markings. But, unlike decision-free and safe persistent PNs, the firing of a transition may disable some other transitions.

Reliability and safety are the fundamental basis for SCS. Improvement in reliability and safety of a SCS enhances its performance. So, while evaluating for performance we should consider the factors which can breach reliability and safety of SCS. Deadlock, nonliveness, stability and boundedness are some of the critical aspects that can breach reliability and safety. So, these aspects should be considered while analyzing performance.

If a marked trap is present in a siphon, then it is not a potential deadlock, and if no siphon is a potential deadlock, a PN is free from deadlock.

Let us denote *M(S)* as the token count in *S*, then the siphon *S* is considered to be a potential deadlock iff,

$$d(S) = min\{M(S) \mid \mathrm{M} \in \mathrm{R}(M_0)\} = 0 \tag{1}$$

It is difficult to solve (1) due to the large number of reachable markings available in practical applications. So, we define *D(S)* as

$$D(S) = min\{M(S) | M = M_0 + IY, M \geq 0, Y \leq 0\} \tag{2}$$

Where, $M = M_0 + IY$ represents the state equation, *I* represents the incidence matrix and *M* and *Y* represent the real vectors. Equation (2) is a type of linear programming problem and so, it can be solved in a polynomial time. All the reachable marking satisfy this state equation but its reverse does not hold true

$$\Rightarrow D(S) \leq d(S) \tag{3}$$

Therefore, $S : D(S) > 0$ is not a potential deadlock.

Therefore, if a PN contains a marked trap or $D(S) > 0$ it is free from deadlock.

Liveness: The property of liveness is related to absence of potential deadlock i.e. any marked PN is live if and only if it has no any potential deadlock [20].

Boundedness, Stability and Steady State Analysis:

A PN (N, M_0) is said to be bounded if $M_p \leq k \forall\ p$ and $M \in R(M_0)$ i.e. the token count in each place does not exceed a finite number *k* for any marking reachable from M_0. The boundedness property guarantees that there will be no overflow in any place or buffers irrespective of any firing sequence.

A PN (N, M_0) is safe if the following condition (4) holds true:

$$\forall M \in R(N, M_0),\ \forall p \in P, M(p) \leq 1 \tag{4}$$

A PN (N, M_0) is stable if it is bounded for any feasible firing sequence and is said to be steady if condition (5) holds true.

$$(\Delta M(t)) / \Delta t = 0, \quad Where \ \ \Delta t = t - t_0 \tag{5}$$

Let, a timed PN $N = (P, T, F, W, D)$ be a union of two nets namely N_1 and N_2 , where $N_1 = (P, T, F, W)$ and $N_2 = (P_\varepsilon, T_\varepsilon, F_\varepsilon, W_\varepsilon)$ such that $P \cap P_\varepsilon = T \cap T_\varepsilon = F \cap F_\varepsilon = \varnothing$. If N_1 is a strongly connected, general and pure PN and N_2 is a PN such that $P_\varepsilon = \{P_i, P_f\}$, $T_\varepsilon = \{t_\varepsilon\}$, $F_\varepsilon \subseteq P_\varepsilon \times T_\varepsilon$, $W_\varepsilon : F_\varepsilon \rightarrow \{1\}$, then the timed PN is stable.

For steady state analysis we need to prove that PN (N, M_0) can be in steady states.

Lemma. *PN* (N, M_0) *can be in steady states.*

Proof. *For any PN,* $M(t) = M(t_0) + [N].\sigma(\Delta t)$, *where* $\sigma(\Delta t) = \sigma(t) - \sigma(t_0)$ *is the firing count vector between* t_0 *and* t.

We have, $\Delta M = [N] \cdot \sigma(\Delta t)$. Dividing both sides by Δt we get,

$$\Delta M / \Delta t = [N] \cdot \sigma(\Delta t)\ ) / \Delta t = [\sigma(t) - \sigma(t_0)] / t - t_0$$

Since, PN is consistent therefore firing sequence leads the system to move from M_0 to *M* i.e. $[N] \cdot \sigma = 0$

$$\Rightarrow \frac{\sigma(t) - \sigma(t_0)}{\Delta t} = \Xi \cdot \sigma : [N] \cdot \sigma(\Delta t) = 0$$

$$\therefore \exists \frac{\Delta M}{\Delta t} = 0 \tag{6}$$

This proves that PN (N, M_0) can be in steady state.

We can also compute the steady-state probability by transforming the PN model into its equivalent Markov Chain using the reachability graph. We define an infinitesimal generator $Q = [q_{ij}]$, such that $(i \neq j)$ and q_{ij} denotes the transition rate from state S_i to S_j. For no transition $q_{ij} = 0$. Then the diagonal elements will be the negative sum of the elements in the respective row, i.e.

$$q_{ii} = \sum_{j=1}^{j=n,\, j \neq i} q_{ij} \tag{7}$$

Let steady-state probability is $\mathrm{A} = (\alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n)$. Then

$$\begin{cases} A \times Q = 0 \\ \sum_{i=0}^{n} \alpha_i = 1 \end{cases} \tag{8}$$

Equations (7) and (8) need to solved for computing the steady-state probability.

Performance Analysis: We can estimate the performance of a system, if we can find the average sojourn time i.e total time spent in the system by a token before the system reaches to the initial marking. The approach for performance evaluation is shown in Fig. 2. It consists of seven phases. The detailed methodology of the performance analysis of a case study is shown Section VI.

5. Case study: SDS-2 and ITS PN model

NPP is a type SCS which is operated and maintained under strict rules and regulations in order to achieve high reliability and the required performance. The Atomic Energy Regulatory Board of the respective country establishes these rules and regulations. The NPP

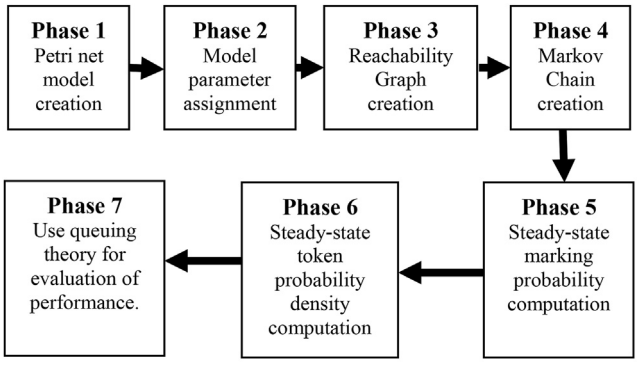

Fig. 2. Performance Evaluation framework.

is equipped with multiple safety systems that keep it safe under any attacks. Shutdown systems are a special type of safety systems which are exclusively included in the plant to alleviate the hazardous consequences of plant failure by automatically shutting it down. SDS-2 injects gadolinium nitrate into the NPP reactor to instantly stop the nuclear reaction. SDS-2 has been taken as a case study for the demonstration of performance analysis methodology. The block diagram of SDS-2 has been shown in Fig. 3.

SDS-2 is a processor based system responsible for quickly terminating the reactor operation. High pressure helium is released by the four fast acting valves (FAV) to inject gadolinium nitrate solution into moderator. Gadolinium nitrate solution absorbs the neutrons and stops the nuclear chain reaction. The FAVs ensure their opening with high reliability and on demand, as they are air-to-close and spring to open. The gadolinium nitrate poison tanks are cylindrical in shape and are mounted on the outer fence of reactor vault. Poison is injected into the moderator using the nozzle of the poison tank. A stainless steel pipe is used to connect each poison tank to a horizontal in core injection tube nozzle which spans the calandria and is immersed in the moderator. When the poison injection in initiated, the helium pressure transfers the gadolinium nitrate poison to the calandria and the ball, which sits at the top of the poison tank, falls to the poison tank bottom. The ball sits at the bottom position of the gadolinium nitrate poison tank outlet and prevents the release of high pressure helium to the calandria. Apart from the manual initiation, there are 9 parameters which can initiate SDS-2 [23], as shown in Table 1.

5.1. Petri Net Model of SDS-2

The SDS-2 failure is catastrophic in nature. However, it is equipped with several components like sensors, actuators, logic and dedicated human machine interface to meet its objective. But we will not consider the hardware failures because the failure rates of these are of the 10^{-6} order. The SDS-2 system consists of two vent valves on both the lines of FAV, which are always in open state in order to release the pressure in the line to avoid spurious injection of poison. Whenever any of the nine parameters, shown in Table 1 deviates from its normal limit a token is deposited at place p_1. After the trip parameter deviate the transition t_1 sends signal to create LC and to energize the relays to close the vent valves. The LC gets created in place p_2, and gets restored in place p_4. Place p_5 relays energize to close the vent valves. Opening of the FAV (place p_9) pressurizes the gadolinium poison to get injected into the moderator. After successful injection of the poison the LC gets restored, which closes the FAV followed by opening the vent valves. It must be noted that we have kept redundant information in place $\boldsymbol{p}_{11}$ in order to track the state of the FAV due to its criticality for safety. The FAV must be in open state to trip the reactor and must be in closed state for its normal operating conditions. The modeled PN for SDS-2 is 1-bounded so a default weight of arcs is 1 throughout. Moreover, a 1-bounded PN qualifies the safeness criteria [20]. The PN modeling is explained in Section III. We have used timed PN to model the SDS-2 process as shown in Fig. 4.The description of places and the transitions are given in notations. FAV must be in open state to trip the reactor and must be in the closed state for the normal operation conditions.

6. Performance evaluation and validation

1) *Deadlock and Liveness Analysis:* Consider the PN model of the poison injection system shown in Fig. 4. The delays of the transitions are deterministically given as per the design specification. We have used timed PN for the modeling. We run the modeled PN of SDS-2 on Time Net tool [24]. It has three minimal siphons $S_1 = \{p_8, p_9\}$, $S_2 = \{p_6, p_{11}\}$, $S_3 = \{p_1\}$ and three marked trap $T_1 = \{p_6, p_{11}\}$, $T_2 = \{p_8, p_9\}$ and $T_3 = \{p_4\}$. We can see that S_1 and S_2 are also marked trap but S_3 do not contain any trap. Solving Equation (2) with $S = S_1$ and $S = S_2$ we find that $D(S_1) = 1$ and $D(S_2) = 1$, which proves that our PN model is deadlock free.

As, the net contains no any potential deadlock it satisfies the liveness condition shown in Section-IV.

2) *Stability, Boundedness and Steady State Analysis:* The PN model of the poison injection SDS-2 shown in Fig. 4 shows that the maximum token count for any place is not more one for all the markings reachable from the initial marking M_0 i.e., $M_0 \leq 1$. So the model is stable. Also, as the model is one-bounded, it implies safe also. Furthermore, we see that $\Delta M/\Delta t = 0$. Therefore, from equation (6), it implies the system is steady too.

Thus, the designed PN model of the SDS-2 system satisfies all the performance metrics. These metrics if not satisfied will affect the reliability of the system resulting in performance degradation.

3) Performance *Analysis:* The performance evaluation framework has been shown in Fig. 2 of Section IV. It consists of seven

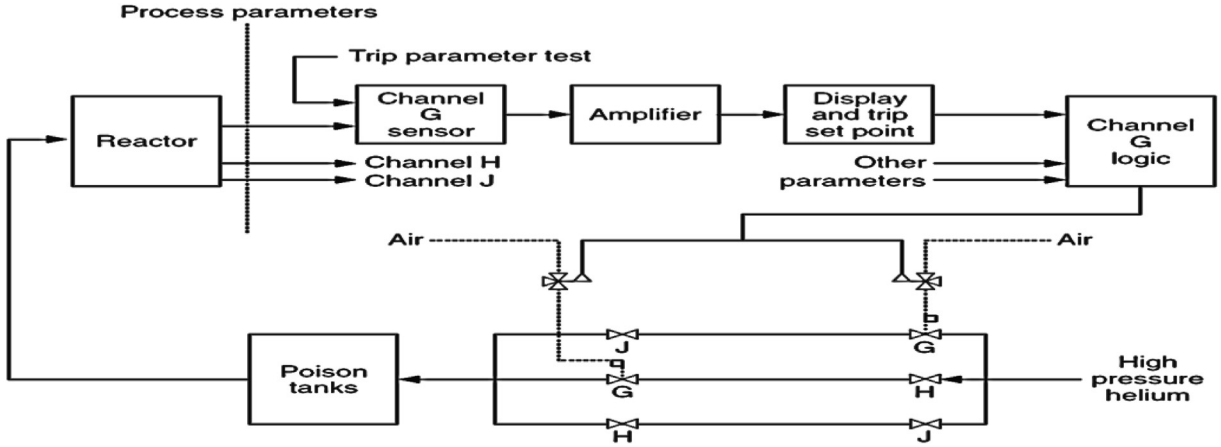

Fig. 3. Block diagram of shutdown system 2.

Table 1
SDS-2 Trip parameters and detectors used

S. No.	Trip Parameter	Detector
1	Neutron Power	Vertical in-core detectors
2	Rate Log Neutron Power	Ion Chambers
3	Heat transport system flow	Differential Pressure Transmitter
4	Heat transport system pressure	Pressure Transmitter
5	Reactor building Pressure	Differential Pressure Transmitter
6	Steam generator Level	Differential Pressure Transmitter of each steam generator
7	Steam generator Feedline Pressure	Pressure Transmitter on Individual Feedlines
8	Moderator level	Differential Pressure Transmitter
9	Low pressurizer level	Differential Pressure Transmitter

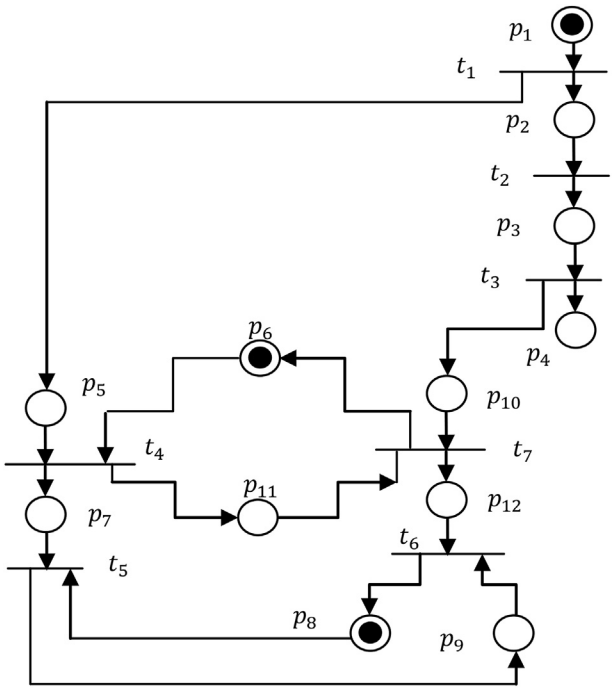

Fig. 4. Petri net model for SDS-2.

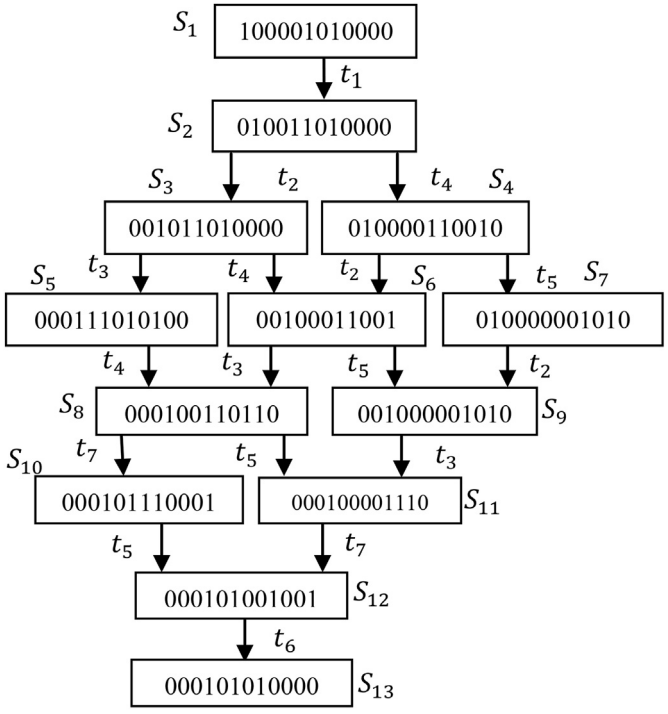

Fig. 5. Reachability Graph of the PN model of SDS-2.

phases. We will proceed step by step in section to calculate the performance of SDS-2 system.

a) *Phase 1:* Petri net model creation.

The modeling process for PN has been explained in Section III and the PN model for SDS-2 is shown in Section V.

b) *Phase 2:* Model Parameter Assignment.

As, SDS-2 is a real-time system, we keep the delays in the transitions as per the specification. We use Time Net tool for the PN model creation and computation of the transitions firing rates which are given in Table 2.

c) *Phase 3:* Reachability Graph Creation.

We have explained reachability in Section III. The reachability graph creation using PN model is well explained in the paper [20]. From the PN model shown in Fig. 4, the equivalent reachability graph is created and is shown in Fig. 5.

Table 2
Firing rate of transitions.

t_1	t_2	t_3	t_4
4.22	5.21	4.81	3.98
t_5	t_6	t_7	
4.77	3.98	5.20	

d) Phase *4:* Markov Chain Creation

The Markov chain of timed PN is shown in Fig. 6. It is drawn using the reachability graph of the corresponding PN model shown in Fig. 5.

e) *Phase 5:* Steady-State Marking Probability Computation.

The steady-state marking probability is computed by solving equations (7) and (8). The infinitesimal generator $Q = [q_{ij}]$, such that $(i \neq j)$ and q_{ij} denoting the transition rate from state S_i to S_j is shown in equation (9). Equation (10) is the resultant equation.

Solving equation (10), we get the steady-state probability as follows: $\alpha_1 = 9.7 \times 10^{-4}$, $\alpha_2 = 4.462 \times 10^{-4}$, $\alpha_3 = 2.68 \times 10^{-3}$, $\alpha_4 = 1.7848 \times 10^{-3}$, $\alpha_5 = 3.2398 \times 10^{-4}$, $\alpha_6 = 9.71 \times 10^{-4}$, $\alpha_7 = 1.642 \times 10^{-3}$, $\alpha_8 = 5.82 \times 10^{-4}$, $\alpha_9 = 2.91 \times 10^{-3}$, $\alpha_{10} = 6.402 \times 10^{-3}$, $\alpha_{11} = 3.24174 \times 10^{-3}$, $\alpha_{12} = 0.119504$, $\alpha_{13} = 0.97$

f) *Phase 6:* Steady-State Token Probability Density Computation.

The steady-state token probability density function computes the probability of having a certain number of tokens in a place in a steady-state. As the PN model is stable and bounded i.e. the maximum token count for any place is not more one for all the markings reachable from the initial marking M_0. The steady-state token probability density function can be computed from steady-state probability values as shown in Table 3.

$$Q = \begin{bmatrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} & S_{11} & S_{13} & S_{13} \\ S_1 & q_{11} & t_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_2 & 0 & q_{22} & t_2 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_3 & 0 & 0 & q_{33} & 0 & t_3 & t_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_4 & 0 & 0 & 0 & q_{44} & 0 & t_2 & t_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_5 & 0 & 0 & 0 & 0 & q_{55} & 0 & 0 & t_4 & 0 & 0 & 0 & 0 & 0 \\ S_6 & 0 & 0 & 0 & 0 & 0 & q_{66} & 0 & t_3 & t_5 & 0 & 0 & 0 & 0 \\ S_7 & 0 & 0 & 0 & 0 & 0 & 0 & q_{77} & 0 & t_2 & 0 & 0 & 0 & 0 \\ S_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{88} & 0 & t_7 & t_5 & 0 & 0 \\ S_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{99} & 0 & t_3 & 0 & 0 \\ S_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{1010} & 0 & t_5 & 0 \\ S_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{1111} & t_7 & 0 \\ S_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{1212} & t_6 \\ S_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_{1313} \end{bmatrix} \tag{9}$$

$$\left.\begin{array}{l} t_1 \cdot \alpha_1 = t_6 \cdot \alpha_{13} \\ (t_2 + t_4) \cdot \alpha_2 = t_1 \cdot \alpha_1 \\ (t_3 + t_4) \cdot \alpha_3 = t_2 \cdot \alpha_2 \\ (t_2 + t_5) \cdot \alpha_4 = t_4 \cdot \alpha_2 \\ t_4 \cdot \alpha_5 = t_3 \cdot \alpha_3 \\ (t_3 + t_5) \cdot \alpha_6 = t_2 \cdot \alpha_4 \\ t_2 \cdot \alpha_7 = t_5 \cdot \alpha_4 \\ (t_7 + t_5) \cdot \alpha_8 = t_4 \cdot \alpha_5 + t_3 \cdot \alpha_6 \\ t_3 \cdot \alpha_9 = t_5 \cdot \alpha_6 + t_2 \cdot \alpha_7 \\ t_5 \cdot \alpha_{10} = t_7 \cdot \alpha_8 \\ t_7 \cdot \alpha_{11} = t_5 \cdot \alpha_8 + t_3 \cdot \alpha_9 \\ t_6 \cdot \alpha_{12} = t_5 \cdot \alpha_{10} + t_7 \cdot \alpha_{11} \\ 0 \cdot \alpha_{13} = t_6 \cdot \alpha_{12} \\ \sum_{i=1}^{13} \alpha_i = 1 \end{array}\right\} \tag{10}$$

g) *Phase 7:* Use Queuing Theory for the Evaluation of Performance.

Let us consider the PN model as a black box system and define:

λ = Average number of tokens arriving in the system per unit time
$\overline{T}$ = Average time spent in the system i.e. sojourn time
$\overline{N}$ = Average number of tokens in the system

Then, using Little's Law $\overline{N} = \lambda \overline{T}$ (11)

The value of $\overline{N}$ can be calculated by summing up the steady-state probability density function obtained in Table 3. We will not include the probability density value of the initial place p_1 because it will initialize the whole process. Hence, $\overline{N}$ is given by $\overline{N} = \sum_{2}^{12} \mu_i$, where μ_i denote the individual steady-state probability density value of the places.

Therefore, the average number of tokens in the system is,

$$\overline{N} = 3.06 \tag{12}$$

Initially the places p_1, $p_6$and p_8 contain one token. Therefore, the average arrival rate of tokens can be calculated by multiplying the steady-state probability density values of these places to their corresponding transition rate and then summing them up.

$$\lambda = (p_1 \cdot t_1) + (p_6 \cdot t_4) + (p_8 \cdot t_5)$$

$$\Rightarrow \lambda = 8.649746545 \tag{13}$$

Using Equations (11)–(13) we can find the average time spent in the system.

$$\overline{T} = \frac{\overline{N}}{\lambda} = \frac{3.06}{8.649746545}$$

$$\Rightarrow \overline{T} = 0.3537 \tag{14}$$

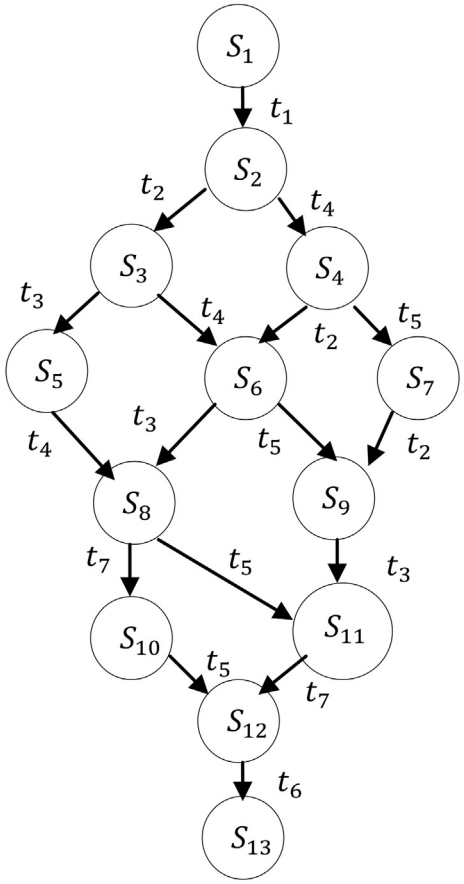

Fig. 6. Markov Chain of the PN model of SDS-2.

Table 3
Steady-state token probabilities density values.

$P(1\ \textbf{Token in}\ p_1) = 9.7 \times 10^{-4}$
$P(1\ \textbf{Token in}\ p_2) = \alpha_1 + \alpha_4 + \alpha_7 = 0.013126816$
$P(1\ \textbf{Token in}\ p_3) = \alpha_3 + \alpha_6 + \alpha_9 = 6.56 \times 10^{-3}$
$P(1\ \textbf{Token in}\ p_4) = \alpha_5 + \alpha_8 + \alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{13} = 0.99250012$
$P(1\ \textbf{Token in}\ p_5) = \alpha_2 + \alpha_3 + \alpha_5 = 0.006363$
$P(1\ \textbf{Token in}\ p_6) = \sum_{1}^{3} \alpha_i + \alpha_5 + \alpha_{10} + \alpha_{12} + \alpha_{13} = 0.99276978$
$P(1\ \textbf{Token in}\ p_7) = \alpha_4 + \alpha_6 + \alpha_8 + \alpha_{10} = 9.74 \times 10^{-3}$
$P(1\ \textbf{Token in}\ p_8) = \sum_{1}^{6} \alpha_i + \alpha_8 + \alpha_{10} + \alpha_{13} = 0.98415711$
$P(1\ \textbf{Token in}\ p_9) = \alpha_7 + \alpha_9 + \alpha_{11} + \alpha_{12} = 0.019744156$
$P(1\ \textbf{Token in}\ p_{10}) = \alpha_5 + \alpha_8 + \alpha_{11} = 4.14772 \times 10^{-3}$
$P(1\ \textbf{Token in}\ p_{11}) = \alpha_4 + \alpha_7 + \alpha_8 + \alpha_9 + \alpha_{11} = 0.0111314872$
$P(1\ \textbf{Token in}\ p_{12}) = \alpha_{10} + \alpha_{12} = 0.0183524$

Therefore, on average a token spends 0.3537 units of time in the system. In other words the modeled PN of SDS-2 takes 0.3537 seconds to inject gadolinium nitrate poison to trip the nuclear reactor in case of emergency. It shows the average delay of the SDS 2 system.

Experimentation Validation: A healthy SDS-2 system is supposed to inject poison in the nuclear reactor on deviation of any of the trip parameters given in Table 1. However, before injecting the poison a proper communication takes place between different components of the SDS2. The communication between transitions involve reading message, sending message, send and receive acknowledgement; all having exponentially distributed execution times. The message needs to be send within a fixed timeout time. Therefore the timeout transition is not exponentially distributed. Timeout is denoted by a random variable having Erlangian probability density function. The communication involves cyclic redundancy check (CRC) as well. The trip value communicated to the system is represented by $\boldsymbol{p}_1$ having a Poisson rate η. Notice that the probability ρ of no token being in the $\boldsymbol{p}_1$ place $P_{\boldsymbol{p}_1}$ is the probability that the subsystem is busy and cannot accept new messages. Therefore, the actual throughput of the of SDS 2 system becomes $\eta(1 - \rho)$.

Consider the SDS 2 communication network system has a 9600 baud line with a 5% of error probability and 1024 bit packets. Then, the performance of the shutdown system can be analyzed for the transition rates given in Table 4. The average delay is calculated for the system when the observation is made that system is busy or packet acknowledgement is lost or on-hold, it has exactly one message in it. In this case, the average delay can be calculated using Little's Law, N = ST, where S is the throughput rate. This delay gives no weight to the blocked messages. For the throughput values given in Table 4, the average delay in the SDS 2 communication network for poison injection would be 0.3662 s. Hence, comparing this delay with (14), the experimental results are validated.

7. Conclusion

This paper focuses on the performance evaluation of safety critical real-time system of NPP using timed PN. The proposed technique has a potential to address the challenges and limitations of the existing techniques that are discussed in section II. The methodology discussed here estimates the time required by the SDS-2 for successful injection of poison to trip the NPP. The technique is useful for the validation of the design of SCCS of NPP. Mathematical modeling using timed PN has been used to model the SDS-2 system of NPP. The performance evaluation framework consists of seven phases which is well explained in Section VI. In future we plan to extend our existing modeling techniques to capture the dynamic relationships between components, such as redundancy and dependency for performance and other dependability analysis.

Table 4
Firing rate in the communication network of SDS 2.

Transition	Throughput Rate (firing/sec)
SEND, SEND ACK	9.375
MSG DROP, ACK DROP	3.91
CRC OK, ACK OK	74.22
TIMEOUT	1.000

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.net.2019.08.018.

References

[1] M. Modarres, M.P. Kaminskiy, V. Krivtsov, Reliability Engineering and Risk Analysis: a Practical Guide, CRC press, 2016.
[2] Nuclear Power Plant Simulators for Use in Operator Training, U.S. Nuclear Regulatory Commission, 1981.
[3] Modern Instrumentation and Control for Nuclear Power Plants: a Guidebook, International Atomic Energy Agency, 1999.
[4] W.C. Lipinski, Nuclear Power Plant Instrumentation and Control—A Guidebook, International Atomic Energy Agency, Vienna, Austria, 1984.
[5] N.G. Leveson, J.L. Stolzy, Safety analysis using Petri nets, IEEE Trans. Softw. Eng. 3 (1987) 386–397.
[6] P. Kumar, L.K. Singh, C. Kumar, Suitability analysis of software reliability models for its applicability on NPP systems, Qual. Reliab. Eng. Int. 34 (8) (2018) 1491–1509.
[7] Heiko Koziolek, Performance evaluation of component-based software systems: a survey, Perform. Eval 67 (8) (2010) 634–658.
[8] S.S. Gokhale, K. S Trivedi, Reliability prediction and sensitivity analysis based on software architecture, in: 13th IEEE International Symposium on Software Reliability Engineering, 2002, pp. 64–75.
[9] L.K. Singh, G. Vinod, A.K. Tripathi, Design verification of instrumentation and control systems of NPP, IEEE Trans. Nucl. Sci. 61 (2014) 921–930.
[10] L.K. Singh, G. Vinod, A.K. Tripathi, Modeling and prediction of performability of safety critical computer based systems using Petri nets, in: 2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops, Dallas, TX, 2012, pp. 85–94.
[11] K. Goseva- Popstojanova, K.S. Trivedi, Failure correlation in software reliability models, IEEE Trans. Reliab. 49 (1) (2000) 37–48.
[12] R.A. Sahner, K.S. Trivedi, A. Puliafito, Performance and Reliability Analysis of Computer Systems: an Example-Based Approach Using the SHARPE Software Package, Springer Science & Business Media, 2012.
[13] Z. Liu, Y. Liu, B. Cai, X. Li, X. Tian, Application of Petri nets to performance evaluation of subsea blowout preventer system, ISA Trans. 54 (2015) 240–249.
[14] L.K. Singh, H. Rajput, Dependability analysis of safety critical real-time systems by using Petri nets, IEEE Trans. Control Syst. Technol. 26 (2017) 415–426.
[15] W.L. Wang, D. Pan, M.H. Chen, Architecture-based software reliability modeling, J. Syst. Softw. 79 (2006) 132–146.
[16] L. Cheung, R. Roshandel, N. Medvidovic, L. Golubchik, Early prediction of software component reliability, in: IEEE 30th International Conference on Software Engineering, 2008, pp. 111–120.
[17] J. Merseguer, J. Campos, S. Bernardi, S. Donatelli, A compositional semantics for UML state machines aimed at performance evaluation, in: 6th IEEE International Workshop on Discrete Event Systems, 2002, pp. 295–302.
[18] V.S. Sharma, K.S. Trivedi, Quantifying software performance, reliability and security: an architecture-based approach, J. Syst. Softw. 80 (2007) 493–509.
[19] P. Kumar, L.K. Singh, C. Kumar, An optimized technique for reliability analysis of safety-critical systems: a case study of nuclear power plant, Qual. Reliab. Eng. Int. 35 (2019) 461–469.
[20] T. Murata, Petri nets: properties analysis and applications, Proc. IEEE 77 (1989) 541–580.
[21] L. Bernardinello, G. De Michelis, K. Petruni, S. Bigna, On the synchronic structure of transition systems, in: Structures in Concurrency Theory, Springer, London, 1995, pp. 69–84.
[22] C.V. Ramamoorthy, G.S. Ho, Performance evaluation of asynchronous concurrent systems using Petri nets, IEEE Trans. Softw. Eng. 5 (1980) 440–449.
[23] T.L. Chu, G. Martinez-Guridi, M. Yue, P. Samanta, G. Vinod, J. Lehner, Workshop on Philosophical Basis for Incorporating Software Failures into a Probabilistic Risk Assessment, Technical Report, BNL-90571-2009-IR, Brookhaven National Laboratory, 2009.
[24] C. Lijie, T. Tao, Z. Xianqiong, E. Schnieder, Verification of the safety communication protocol in train control system using colored Petri net, Reliab. Eng. Syst. Saf. 100 (2012) 8–18.