

Jñānābha, Vol. 56(1P) (2026), 63-73

**A COMPREHENSIVE STUDY OF SOME PETRI NETS WITH THEIR TYPES,
PROPERTIES AND APPLICATIONS**

Riddhi Jangid, Shikha Tripathi and Gajendra Pratap Singh*

School of Computational and Integrative Sciences Jawaharlal Nehru University, New Delhi 110067

Email: riddhijangid97@gmail.com, shikha15sit@jnu.ac.in, gajendra@jnu.ac.in

*Corresponding Author

(Received: August 01, 2025; Revised: February 15, 2026; Accepted: April 15, 2026)

DOI: <https://doi.org/10.58250/jnanabha.2026.561P-7>

Abstract

Modeling a discrete event dynamic system with the help of Petri nets is the main task in the application area. These models must satisfy each and every system design and property in order to be applicable in the real world. What if a system is associated with time? Or what if a system is continuously flowing in nature? What if the resources available are to be utilized minimum for the greatest profit? What if there is a random state change in the system? To answer such difficulties, new functions were introduced with the ordinary Petri nets and this gave rise to many types of Petri nets. Now, the Petri nets convey the time delays associated with the system in Time-dependent Petri nets and the continuous flow of nature has been explained by the continuous and hybrid Petri nets. In this article, we try to show the basics of two different types of Petri nets that have been used in different research areas using their examples. Also, the applicability of Petri nets in diverse areas has been described with a few examples.

2020 Mathematical Sciences Classification: 68Q85, 68M14, 68R10, 68Q10

Keywords and Phrases: Petri nets; Types of Petri nets; Properties of Petri nets; Place/transition nets; Coloured Petri nets; Timed Petri nets; Reachability; Invariants.

1 Introduction

Processes in nature work in a systematic order. Some are complex while some are not. There are many mathematical models designed to solve the complexity of a system and to study the system's behavior theoretically. This helped in saving the resources available and designing a system correctly. Similar thought came into practice when Dr. Carl Adam Petri introduced the concept of Petri net (PN) during his Ph.D. dissertation in 1962. Petri nets (PNs) [11] further proved to be of greater use as it gives precise modeling of the system and its behavioral analysis. They are commonly used in biological networks, communication protocols, chemical systems, and many other areas of computer science [5-8, 14-21].

As per my study, fewer works introducing the basics of types of Petri nets have mostly resulted in ignorance about the wider applications of Petri nets. Many researchers and scholars have worked on the types of Petri nets but have described them by combining two or more two types. It definitely is a good way to approach the application process yet this kind of work doesn't really help a beginner to understand the insights used for the practical

approach. The objective of this work is to understand the basics of two different types of Petri nets so that all the researchers do not have to wander around in search of different scholarly works to find an alternative to their motives. In "Petri nets for Dynamic Event-Driven System Modeling", though J. Wang [22] has explained only time-dependent Petri nets, i.e. Time Petri nets and Stochastic Petri nets along with colored Petri nets, his work motivates one to read about time-dependent and high-level Petri nets in more detail. Though colored and time Petri nets are also discussed by David and Alla [1], they have focused more on hybrid, discrete, and continuous Petri nets in their work "Discrete, continuous, and hybrid Petri nets". Similarly, Time and Petri nets due to [12] is a whole work entirely dedicated to time and Petri nets. The authors have explained the different ways of introducing time with transitions whether it is interval or duration. You'll find glimpses of these works in our work.

We have commenced this work with basic definitions of Petri nets and their modeling example followed by their behavioral properties. Then we dug into the general methods used for analyzing the Petri nets using the coverability graph, reachability tree, etc. Further, we went for defining the basics of types of Petri nets, explaining them with a simple example followed by an application area in the chemical systems in order to make the readers understand the process better. We have also discussed the modeling tools and software we used to model the systems and examples presented in the paper, which may also help the readers model their own applications.

2 Fundamentals of Petri nets

PN is particularly a directed bipartite multigraph [11], that contains two disjoint sets of points/nodes, called place (O) and transition ($\square$), and these points/nodes are connected by directed arc ($\rightarrow$). Arcs are used to connect place node and transition node but in a manner that joins no two places or transitions simultaneously. An arc that is directed from place to transition is the input place of that transition and an arc from the transition to a place is the output place of the transition. We assign tokens ($\bullet$) to the places, which we call marking. These tokens are further used for the execution of Petri nets. Theoretically, we define marking as n -vector $= (M_1, M_2, M_3, \dots, M_n)$, where n is the number of places in the Petri net structure and each $M_i \in \{0, 1, 2, 3, \dots, n\}$. In many cases, we assign some nonnegative integer to the arcs in a Petri net, which we call the weight of the arc. And if weight is not assigned, we consider it 1 by default.

Formally we define Petri net [10] as a 5-tuple $C = (P, T, F, W, M_0)$ where:

$P = \{p_1, p_2, p_3, \dots, p_m\}$: A finite set of places ,

$T = \{t_1, t_2, t_3, \dots, t_n\}$: A finite set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$: A set of arcs,

$W = F \rightarrow \{1, 2, 3, \dots\}$: Weight function assigned with arcs,

$M_0 = P \rightarrow \{0, 1, 2, 3, \dots\}$: The initial marking.

Here $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

2.1 Petri net firing rule

Removing $w(p, t)$ tokens from the input place of a transition and depositing $w(t, p)$ tokens into the output place of the transition causes the state change in a Petri net, which we call firing of that transition. Here, $w(p, t)$ denotes the weight of the arc from place to transition and similarly $w(t, p)$ denotes the weight of the arc from transition to a place. In Fig. 2.1, we have $w(p, t) = w(t, p) = 1$ for all the transitions.

2.2 Execution Rule in a Petri net

The execution of a Petri net is by the firing of transitions and any transition fires if it is enabled. Also, it is important to notice that not every enabled transition fires.

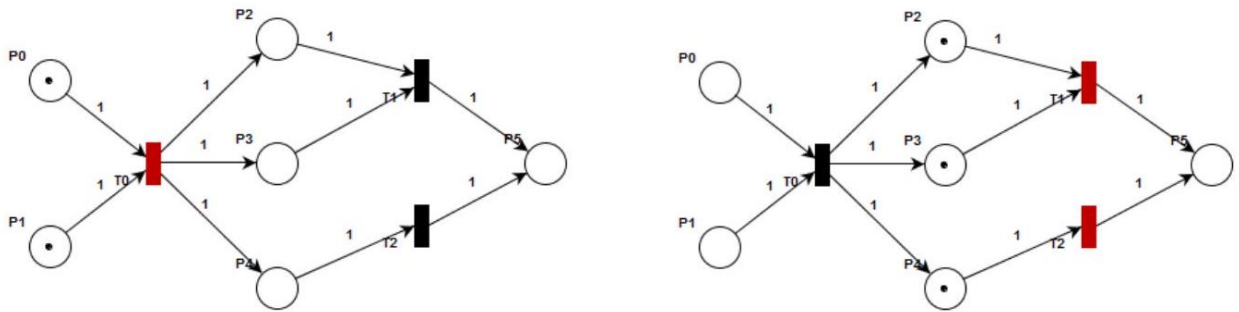


Figure 2.1: (a) Petri net showing enabled transition $T0$ with initial marking $(1,1,0,0,0,0)$
(b) Illustration of the firing of transition $T0$ in the marked Petri net

Here, we have $P = \{P0, P1, P2, P3, P4, P5\}$ is the set of places and $T = \{T0, T1, T2\}$ is the set of transitions. Fig. 2.1. shows marking reached after firing of transition $T0$ and let us suppose now transition $T1$ fires.

Hence, we have a firing sequence $\sigma = T0 \rightarrow T1$ which gives us the markings: $M_0 = (1, 1, 0, 0, 0, 0) \rightarrow M_1 = (0, 0, 1, 1, 1, 0) \rightarrow M_2 = (0, 0, 0, 0, 1, 1)$. Here, M_0 is the initial marking, M_1 is the marking reachable from firing $T0$ and M_2 is the marking reachable from firing $T1$. Hence, we have reachable markings and a random firing sequence that we study for analyzing Petri nets.

2.3. Analysis of a Petri net

2.3.1. Reachability Graph

We analyze a Petri net by constructing a reachability tree [13, 15, 18] from all the possible reachable states of the marked Petri net. In the example above, given is the initial marking $M_0 = (1, 1, 0, 0, 0, 0)$. Now for enabled transitions in the Petri net, only transition $T0$ is enabled and we have the state $M_1 = (0, 0, 1, 1, 1, 0)$ after firing $T0$. After this, we have two enabled transitions. So we consider both the conditions and write both the markings reached from the different transitions, i.e $M_2 = (0, 0, 0, 0, 1, 1)$ after firing of $T1$ while $M_3 = (0, 0, 1, 1, 0, 1)$ after firing of $T2$. And lastly M_2 reaches $M_4 = (0, 0, 0, 0, 0, 2)$ after firing of $T2$ and M_3 reaches $M_4 = (0, 0, 0, 0, 0, 2)$ after firing of $T1$. Hence, we draw the reachability tree as:

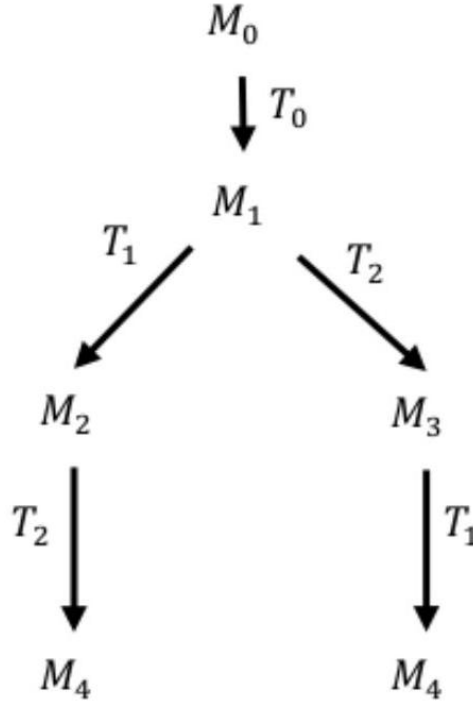


Figure 2.2: The Reachability Tree

3 Types of Petri nets

A. Time-dependent Petri nets

Introducing time variables in the form of interval or duration with Petri nets gives us a variety of possibilities that can be applied to model systems. Somewhere, time was introduced with places, transitions, tokens, or even with the arcs according to the need. Again, this will definitely lead to many different firing rules. So, in order to define a Time Petri net, these are the conditions that need to be fixed.

3.1 Time Petri nets

The most commonly discussed are the Time Petri nets and Timed Petri nets where transitions are connected with time variables. The time variable introduced here is deterministic in nature. Merlin [8] described Time Petri nets by introducing time intervals $[a_t, b_t]$ attached with the transition t such that no transition can fire before a_t time and after b_t time. We call a_t and b_t , earliest firing time of transition t and latest firing time of transition t respectively. Whereas, Ramchandani introduced Timed Petri nets by assigning a time duration d_t to the transition t . The difference, in this case, is that, transition has to fire immediately after being enabled. We will discuss Time Petri nets in this paper. One can refer to [9] for a detailed explanation.

Formally, Time Petri nets are defined as 6-tuple $TPN = (P, T, F, W, M_0, I)$ where

1. $C = (P, T, F, W, M_0)$ is a Petri net and
2. I denotes the Interval function of TPN such that $I : T \rightarrow N \times (N \cup \{\infty\})$ and $\forall t \in T$, with $I(t) = (e(t), l(t))$ such that $e(t) \leq l(t)$, where $t \in T$.

Here, $e(t)$ denotes earliest firing time of t and $l(t)$ denotes latest firing time of t .

Definition 3.1 (p-marking and t-marking). *The p-marking in Time Petri net is defined as the function mapping from set of places to the natural numbers including $\{0\}$. i.e. $M : P \rightarrow \{0, 1, 2, 3, \dots\}$*

And t-marking in a Time Petri net is defined as the function mapping from the set of transitions to $\{R_0^+ \cup \#\}$. i.e. $H : T \rightarrow \{R_0^+ \cup \#\}$.

Definition 3.2 (State). *A state in a Time Petri net is defined as $z = (M, H)$, $z \in TPN$ where M and H represents p-marking and t-marking respectively such that-*

- i. *For all those transitions that are disabled with p-marking, we will take $H(t) = \#$.*
- ii. *For all those transitions that are enabled with p-marking, and the time $H(t)$ is less than or equal to $l(t)$, we have $H(t) \in R_0^+$.*

Also, the initial state is defined as $z_0 = (M_0, H_0)$ where

$$H_0(t) = \begin{cases} 0, & \text{if } t \text{ is enabled in } M_0 \\ \#, & \text{else} \end{cases}.$$

For the state change in a Time Petri net, the first reason is due to the firing of transition.

3.1.1 Execution rule in Time Petri net.

A transition t in a Time Petri net is ready to fire in state (M, H) if and only if it is enabled in the marking $M(t)$ and $H(t) \geq e(t)$. Then the firing of transition t will take place if it is ready to fire.

This will convert state $z = (M, H) \rightarrow z' = (M', H')$ where $M'(t) = M(t) + \Delta t$ and

$$H'(t') = \begin{cases} \#, & \text{if } t \text{ is not enabled in } M' \\ \frac{H(t')}{H(t')}, & \text{if } t \text{ is enabled in } M \text{ and } M' \text{ along with pre-places of } t \text{ and } t' \text{ are disjoint and } t \neq t' \end{cases},$$

for all $t' \in T$.

Hence we say if, after firing of one of the two transitions that are enabled and having at least one common pre-place, another transition is still enabled, then the time is set to zero with that enabled transition.

Another case we need to take care of is t-marking also changes with elapsing of time even when the transition doesn't fire. Consider τ , a positive real number and state $z = (M, H)$ then elapsing of time τ is possible if $\forall t$, when $H(t) \neq \#, H(t) + \tau \leq l(t)$. And this elapsing of time convert the state $z = (M, H) \rightarrow z' = (M', H')$ where $M' = M$ and

$$H'(t) = \begin{cases} H(t) + \tau, & \text{if } t \text{ is enabled in } M' \\ \#, & \text{if } t \text{ is disabled in } M' \end{cases} \forall t \in T.$$

Definition 3.3 (Run). *The sequence $\sigma(\tau) = \tau_0 t_1 \tau_1 \dots \tau_n$ of state transitions, where $\sigma = t_1 t_2 \dots t_n$ is the transition sequence and $\tau = \tau_0 \tau_1 \dots \tau_n$ is the sequence of times having $\tau_i \in R_0^+$ is known as run of.*

Definition 3.4 (Feasible run). *In a time Petri net, if a state $z' = (M', H')$ reachable from state $z = (M, H)$ with the run $\sigma(\tau) = \tau_0 t_1 \tau_1 \dots \tau_n$, then the run is called feasible run from the state z .*

Definition 3.5 (Feasible transition sequence). *The transition sequence $\sigma = t_1 t_2 \dots t_n$ in a time Petri net is feasible if there exists a feasible run $\sigma(\tau) = \tau_0 t_1 \tau_1 \dots \tau_n$ from the state z .*

Definition 3.6 (Reachable state). *The state we reached $z = (M, H)$ from z_0 , whenever there exists a feasible run $\sigma(\tau) = \tau_0 t_1 \tau_1 \dots \tau_n$ is the reachable state from initial state $z_0 = (M_0, H_0)$.*

Let us understand the rules of the firing of Time Petri nets by considering the Petri net-

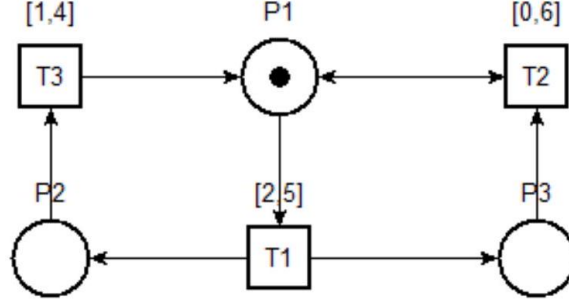


Figure 3.1: A simple example of TPN

In the Petri net, time intervals are associated with the transitions. We have-

$P = \{P_1, P_2, P_3\}$, the set of places and

$T = \{T_1, T_2, T_3\}$, the set of transitions with

$I(T_1) = (2, 5)$, $I(T_2) = (0, 6)$ and $I(T_3) = (1, 4)$.

Initial state/marking, $z_0 = (M_0, H_0) : z_0 = ((1, 0, 0), (0, \#, \#))$.

Now, let us suppose we have attached a clock with every transition that is enabled in the initial marking. Here, only transition T_1 is enabled. Let $\tau_0 = 1.3$ unit of time passes, which changes the state of Petri net into another state, ie $z_0^* = ((1, 0, 0), (1.3, \#, \#))$. We know that the $(T_1) = 2$, hence transition can't fire even though it is enabled. After some more time say $\tau_1 = 1$ time units, the transition is now allowed to fire since $2 \leq 1.3 + 1 = 2.3 \leq 5$. Suppose T_1 fires now. The state that we reach after firing will be $z_1 = ((0, 1, 1), (\#, \#, 0))$. $(\#, \#, 0)$ in z_1 because only transition T_3 is enabled in $(0, 1, 1)$. Proceeding in the same way, we shall have

Transition sequence: $\sigma = T_1 T_3 T_1$

Run: $\sigma(\tau) = 1.3 T_1 2 T_3 2 T_1 1.5$

Feasible run: Run $\sigma(\tau)$ is feasible as we reach z_3 from z_0 by $-z_0 = ((1, 0, 0), (0, \#, \#)) \rightarrow 1.3 \rightarrow z_0^* = ((1, 0, 0), (1.3, \#, \#)) \rightarrow 1, T_1 \rightarrow z_1 = ((0, 1, 1), (\#, \#, 0)) \rightarrow 2 \rightarrow z_1^* = ((0, 1, 1), (\#, \#, 2)) \rightarrow T_3 \rightarrow z_2 = ((1, 0, 1), (0, 0, \#)) \rightarrow 2 \rightarrow z_2^* = ((1, 0, 1), (2, 2, \#)) T_1 \rightarrow z_3 = ((0, 1, 2), (\#, \#, 0))$ and so on.

Hence, $\sigma = T_1 T_3 T_1$ is a feasible transition sequence. And z_3 in this case, is the reachable state because $z_0 \sigma(\tau) \rightarrow z_3$.

Definition 3.7 (State Space). Set of all reachable marking is known as the State Space of TPN.

The properties that can be discussed in this type of Petri net are mainly- liveness, boundedness, and reachability which requires the knowledge of State Space of the Petri net. So, in order to find out the state space, we need to learn some more concepts. Not only the state space, we sometimes are required to find out the minimum or maximum time length that the system requires in reaching a particular state.

Replacing the real positive values τ_i with a variable x_i such that all x_i 's are bounded by some conditions, gives us the Parametric run $(\sigma(x), B_\sigma)$ of σ in TPN with $\sigma(x) = x_0 T_1 x_1 T_2 \dots x_{n-1} T_n x_n$ where B_σ is the set of conditions for all x_i . In continuation,

Definition 3.8 (Parametric State). is defined as (z_σ, B_σ) in TPN where, $z_\sigma = (M_\sigma, H_\sigma)$ such that $M_0 \sigma \rightarrow M_\sigma$ and $H_\sigma(t)$ is the variables' sum.

So, we can conclude: $z_0 \sigma(x) \rightarrow (z_\sigma, B_\sigma)$ and State Space of TPN = $\cup_{\sigma(x)} \{z_{\sigma(x)} \mid x \text{ satisfies } B_\sigma\}$.

Consider the above-discussed example, consider $\sigma = T_1 T_3 T_1$ sequence. Here, $\sigma(x) = x_1 T_1 x_2 T_3 x_3 T_1 x_4$. So, the parametric state will be $(z_\sigma, B_\sigma) = (((0, 1, 2), (\#, \#, 0)), \{2 \leq x_1 \leq 5, 1 \leq x_2 \leq 4, x_1 + x_2 \leq 6, 2 \leq x_3 \leq 5, x_1 + x_2 + x_3 \leq 6, 0 \leq x_3 \leq 4\})$.

Definition 3.9 (Integer Run). Run in which $\tau_i \in \mathbb{N}, \forall i = 0, 1, \dots, n$ is known as Integer Run.

As in the above example, one can easily see that the state z_3 reachable from the Run $\sigma(\tau) = 1.3 T_1 2 T_3 2 T_1 1.5$ is also reachable from $\sigma'(\tau) = 1 T_1 2 T_3 2 T_1 1$. This run $\sigma'(\tau) = 1 T_1 2 T_3 2 T_1 1$ is the Integer run and is solving the purpose that the run $\sigma(\tau) = 1.3 T_1 2 T_3 2 T_1 1.5$ did. Hence, it gives a good result that every feasible transition sequence can be realized by an Integer run.

3.1.2. Reachability Analysis for finite TPN

The reachability graph will be finite if all the reachable p-markings in the net are finite. In order to obtain a reachability graph, it will be difficult to mention all the reachable states. The integer states that we obtained after integer run, where $h(t) \in N, \forall m$ in $t \in T$, carry with themselves all the useful information not only regarding reachability but for other properties as well. Hence, in order to obtain a finite reachability graph, we reduce it by using the idea of "fusing together" all the consecutive edges that represent the elapsing of a single time unit and firing of a single transition. The states obtained in between are removed from the reachability graph. For example, consider an arbitrary path $z_0 t_1 \rightarrow z_1 1 \rightarrow z_1^* 1 \rightarrow z_1^{**} t_2 \rightarrow z_2 \dots$. We will reduce the path to $z_0 t_1 \rightarrow z_1 2, t_2 \rightarrow z_2$. Continuing in the same way, the reachability graph we obtained is the reduced reachability graph for finite TPN.

These are some results and properties that we discussed for finite TPN. One can easily now understand the concepts in case of Infinite TPN from the reference [9].

B. High-level Petri nets

Why High-level Petri nets? Because the tokens in this kind of Petri net carry some information with them, while tokens in Petri nets that we discussed above were all of the same types and represented the state of place by an integer. The aim is just to explain a modeled system in a proper and easier way. The most commonly used in application areas are colored Petri nets.

3.2. Colored Petri nets

Sometimes, there are different systems in the real world that use the same resources. So, it becomes difficult to model such systems practically. So, the need was felt to develop more powerful Net types. Coloured Petri nets [4] solve the purpose. The token assigned in this Petri net with some data values that they carry with themselves, and are called token color. Given a place, all tokens in that place have token colors which belong to a specific type,

which we call a color set of that place. Same color sets can be labelled to the Different places. Marking in these Petri nets is the tokens that are distributed in places. The underlined markings besides the places are the initial marking in these Petri nets. Practically, we can't just put tokens inside the places to show the current marking of colored Petri nets as we did in previous Petri nets because we also need to show the token colors in this case. As a result of introducing token colors and color sets, we will be using a lesser no. of places in comparison to Normal Petri nets. Now, Type of token colors are responsible for the enabling of transitions that are in the input places of the transitions, they can affect the output tokens as well. In order to remove the complexity, now Arc expressions are introduced. These arc expressions specify the collection of the tokens and contain no. of variables that are associated with the transitions. In order to combine many different systems in a system, along with the places one needs to find a way to combine transitions as well, i.e. different actions can be performed by a transition. The variables introduced with the arc expressions will solve our purpose by evaluating different values.

Let's see the formal definition of colored Petri nets, which first requires us to learn what a multi-set is. A set that allows repetition of its elements, different from a set is informally defined as a multi-set. Now,

A colored Petri net [4] is defined as $CPN = (P, T, F, \Sigma, C, N, E, G, I)$ where

$P = \{p_1, p_2, p_3, \dots, p_m\}$: A finite set of places.

$T = \{t_1, t_2, t_3, \dots, t_n\}$: A finite set of transitions.

$F \subseteq (P \times T) \cup (T \times P)$: A set of arcs.

Σ : Set of color sets containing every possible operation and function used in CPN.

C : It is the colour function that is a mapping from places into colours in Σ .

N : It is the node function which is a mapping from F into $(P \times T) \cup (T \times P)$.

E : It is the arc expression function which maps every arc into an expression.

G : It is called the Guard function which is a mapping from every transition to an expression of a particularly Boolean type and,

$I, :$ It is the Initialization function that maps every place into an expression type that is of particularly a multiset over a color set, $C(p)$.

Let us understand CPN more clearly with the help of an example.

We have three courses offered in an institution and the examination sheets of the students collected after the examination are randomly collected. We need to arrange these mark sheets course-wise. The following are the roll numbers of the students according to the course:

Table 3.1: Example of modeling using Coloured Petri nets

S. No.	Course	Roll no. of students
1	B.Sc.(H) Math	2001-2042
2	B.Sc.(H) Statistics	2043-2056
3	B.Sc.(H) Computer Science	2057-2091

We have tried to model the problem using colored Petri nets where we have four places and one transition. Also, we have $\Sigma = \{ \text{Green, Red, Yellow} \}$, arc expression associated with every arc, and color function which distinguishes colors of the tokens by allowing firing in respective transitions.

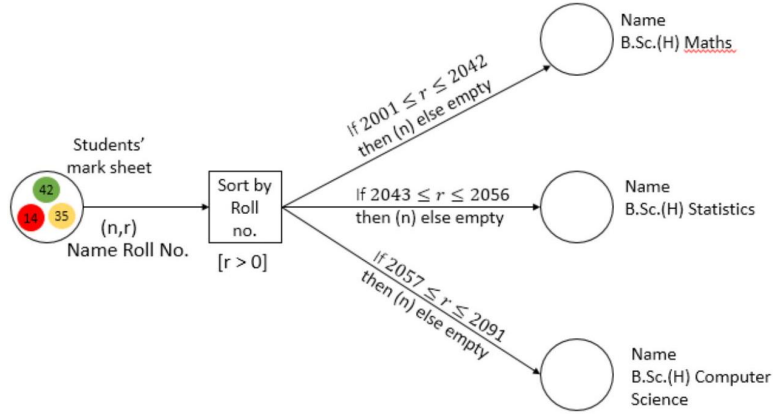


Figure 3.2: An example of a Coloured Petri net graph

4 Application of Petri nets in diverse areas

4.1 Chemical Systems - Lime Cycle

Limestone is a type of Sedimentary rock containing a high percentage of CaCO_3 in it. It has many uses in real-life applications like in constructing houses and buildings. We are going to learn here the Lime Cycle modeling through Petri nets. Consider the Lime Cycle as shown in the figure.

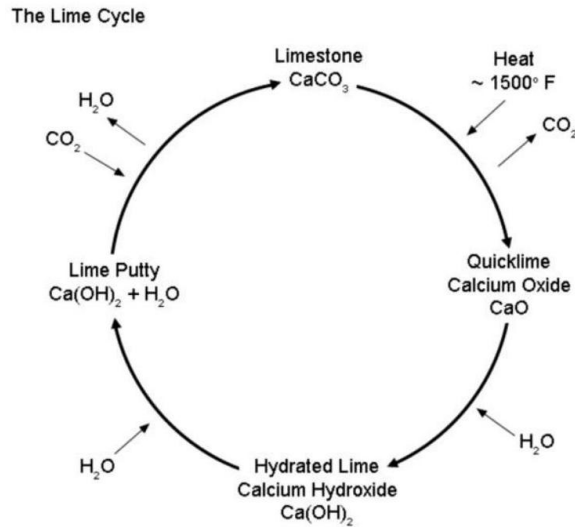


Figure 4.1: Lime cycle in nature

We are trying to show here a simple modeling example in chemical systems that one can use for learning of Petri net applications. Here is the PN modeled in the WoPeD tool for Petri nets:

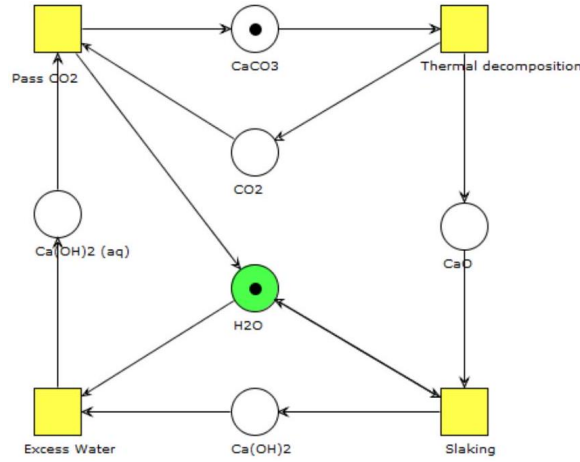


Figure 4.2: Petri net of lime cycle

Here,

$P = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ where the places denote CaCO_3 , Ca(OH)_2 (aq.), CaO , Ca(OH)_2 , CO_2 and H_2O respectively.

And $T = \{T_1, T_2, T_3, T_4\}$ where T_1 denotes the process of Thermal Decomposition, T_2 denotes Slaking, T_3 denotes Excess water and T_4 denotes the Carbon dioxide.

The reachability graph we obtained is-

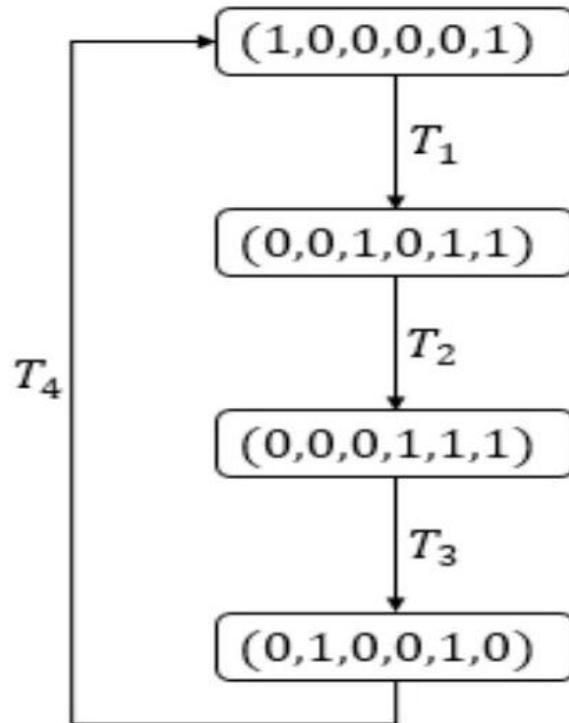


Figure 4.3: Reachability graph of Fig. 4.2.

4.2 Communication System

Consider the elements of the communication system. The elements include the collection of devices that work together in order to establish communication between sender and receiver. The block diagram for the example would be:

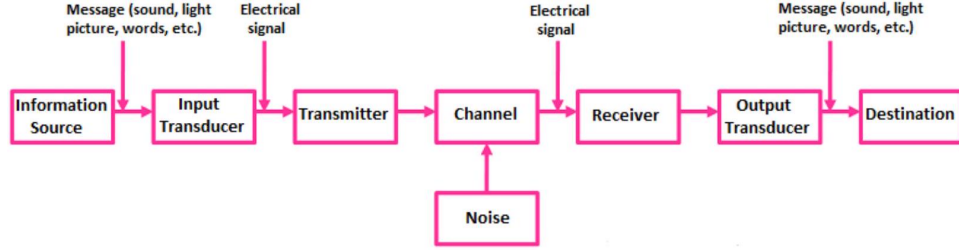


Figure 4.4: Elements of communication systems

Each of these elements has its own functioning that can be studied. Here, we have tried to model a Petri net (on WoPeD: A Petri net tool)[2,3] with the block diagram assuming that the information/message is reached correctly to the destination. We have shown the elements by the set of places and the condition of the message by the set of transitions.

This Petri net model has its own limitations but it can be updated according to the need of the user or situation.

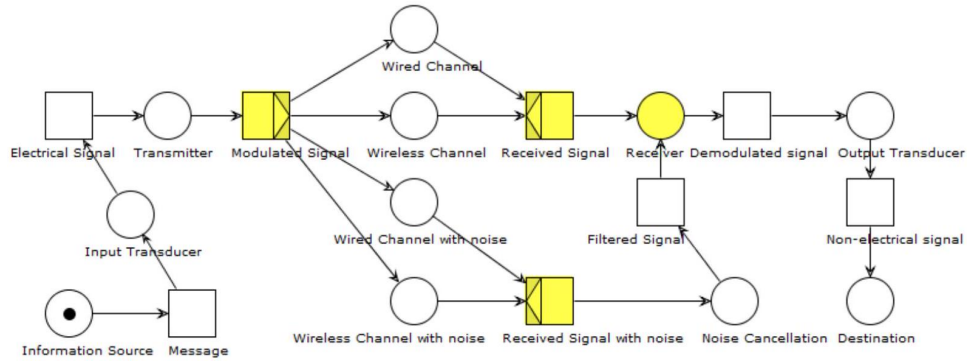


Figure 4.5: Petri net graph modeling of Fig. 4.4.

The model here presents to us the various stages and the possibilities of a communication system. When the message is received by the transmitter, the signal can travel to the receiver either through a wired or wireless channel. If the signal is not interrupted by the noise or the amount of noise is so little that it can entirely be ignored then the receiver can easily decode the message but if the message is accompanied by a constant noise that keeps interrupting the quality of the signal in such a manner that the message can entirely be lost in the process then there is a need to filter the signal to cure the noise and to receive the message correctly. These possibilities need their own individual path to reach the destination and these kinds of Petri net models can help us to reach out to every other possibility and can reduce the amount of error.

The model clearly states the initial marking, $M_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ and the firing of transitions is according to the firing rule. The Petri net analysis gives us the reachability graph:

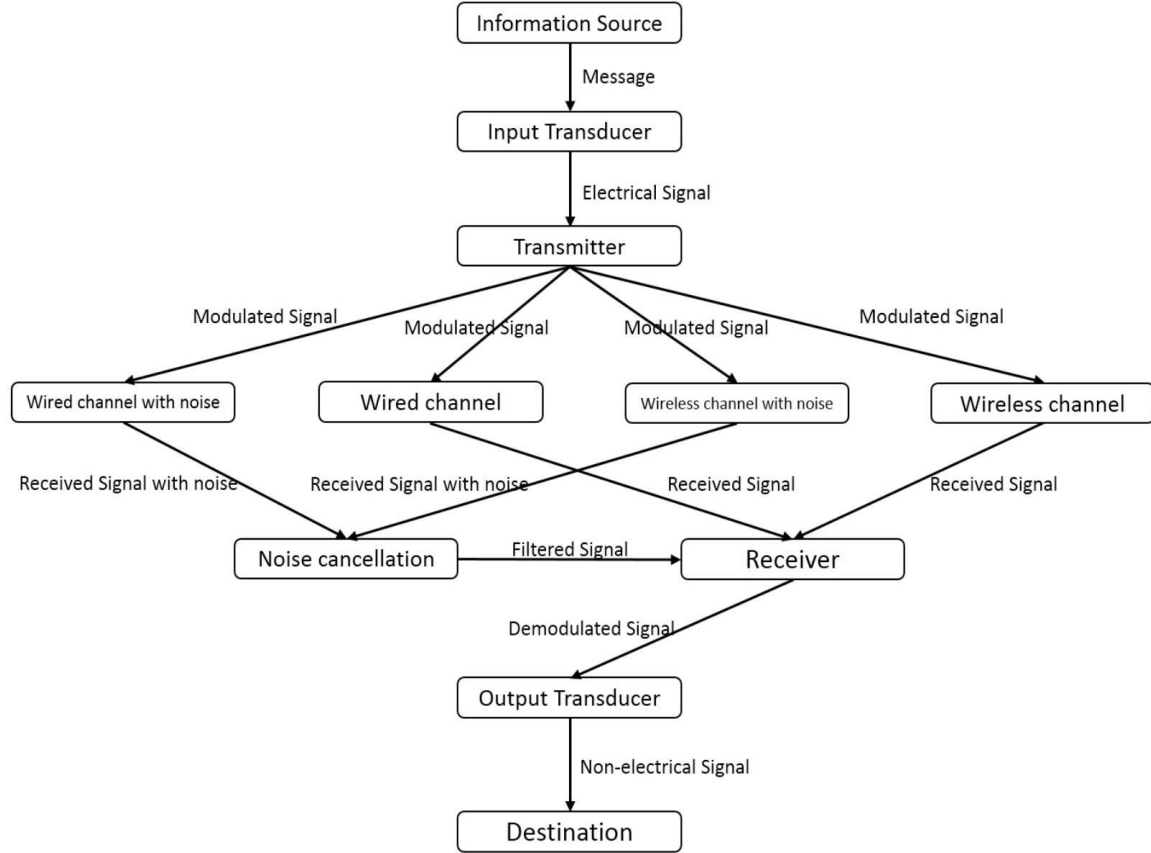


Figure 4.6: Reachability graph of Petri net in Fig. 4.5.

5 Conclusion, Discussion, and Outlook

There have been many mathematical models proposed to solve systems. Petri nets introduced in the 1960s changed the older methods and introduced some new modeling tools for the upcoming generations of mathematicians and scientists. The different types of Petri nets open the different ways in which this modeling tool can be used in our day-to-day life. It has definitely made the process easier to understand and easier to respond to. The types of Petri nets we compiled in this dissertation are a try to provide excess to, all the researchers and readers, the types in just one place which was not easily available earlier. We have discussed the application areas in this paper which is a better way to understand the topic and can definitely help through the application processes in the field. Even after my hard try, I might have missed a lot here in this work; as Petri nets, though interesting yet, is a wide area to work on. I hope this work extends some major help to the researchers.

Acknowledgement. The second author is thankful to DBT-BIC project ID No. BT/PR40251/BTIS/137/11/2021, Govt. of India, for their support and all the authors are expressing their deep gratitude to anonymous reviewers and editors for their valuable suggestions and comments.

References

- [1] R. David and H. Alla, *Discrete, continuous, and hybrid Petri nets*, **1** (2010), 17–130, Springer, Berlin.
- [2] N. J. Dingle, W. J. Knottenbelt and T. Suto, PIPE2: a tool for the performance evaluation of generalised stochastic Petri Nets, *ACM SIGMETRICS Performance Evaluation Review*, **36**(4) (2009), 34–39.
- [3] T. Freytag and M. Sanger, WoPeD—An Educational Tool for Workflow Nets, *BPM (Demos)*, **31** (2014).
- [4] K. Jensen, *Coloured Petri nets: basic concepts, analysis methods and practical use*, **1**, Springer Science & Business Media (1996).
- [5] R. Jangid and G. P. Singh, Mathematical Modeling using Fuzzy Petri Nets and its Classifications, in *Advances in Mathematics: Theory, Methods & Applications*, River Publishers (2025), 89–99.

- [6] R. Jangid and G. P. Singh, On minimal Crisp Boolean Petri net generating Boolean Petri net, *Proceedings of the Jangjeon Mathematical Society*, **28**(4) (2025), 585–598.
- [7] S. Kansal, G. P. Singh and M. Acharya, On Petri nets generating all the binary n-vectors, *Scientiae Mathematicae Japonicae*, **71**(2) (2010), 209–216.
- [8] S. Kansal, G. P. Singh and M. Acharya, 1-Safe Petri nets generating every binary n-vector exactly once, *Scientiae Mathematicae Japonicae*, **74**(1) (2011), 29–36.
- [9] M. A. Marsan, Stochastic Petri nets: an elementary introduction, in *European Workshop on Applications and Theory in Petri Nets*, Springer, Berlin, Heidelberg (1988), 1–29.
- [10] T. Murata, Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, **77**(4) (1989), 541–580.
- [11] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall PTR, 1981.
- [12] L. Popova-Zeugmann, Time petri nets, in *Time and Petri nets*, Springer, Berlin, Heidelberg, (2013), 31–137.
- [13] G. P. Singh, Applications of Petri nets in electrical, electronics and optimizations, in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE (2016), 2180–2184.
- [14] G. P. Singh and A. Gupta, A Petri net analysis to study the effects of diabetes on cardiovascular diseases, in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE (March, 2019), 481–488.
- [15] G. P. Singh and S. K. Singh, Petri net recommender system for generating of perfect binary tree, *International Journal of Knowledge and Systems Science (IJKSS)*, **10**(2) (2019), 1–12.
- [16] G. P. Singh, M. Jha and M. Singh, Modeling the mechanism pathways of first line drug in Tuberculosis using Petri nets, *International Journal of System Assurance Engineering and Management*, **11**(2) (2020), 313–324.
- [17] G. P. Singh, M. Jha and M. Singh, Petri net modeling of clinical diagnosis path in tuberculosis, in *Advances in Interdisciplinary Research in Engineering and Business Management*, Springer, Singapore (2021), 401–412.
- [18] G. P. Singh, S. K. Singh and M. Jha, Existence of forbidden digraphs for crisp boolean Petri nets, *International Journal of Mathematical, Engineering and Management Sciences*, **5**(1) (2020), 83–95.
- [19] G. P. Singh, On Petri Net Generating all the Standard Basis for n-Dimensional Euclidean Space, *Conference Proceeding*.
- [20] R. N. Verma, M. Z. Malik, G. P. Singh and N. Subbarao, Identification of key proteins in host-pathogen interactions between Mycobacterium tuberculosis and Homosapiens: A systematic network theoretical approach, *Healthcare Analytics*, **2** (2022), 100052.
- [21] R. N. Verma, M. Malik, N. Subbarao, G. P. Singh and D. N. Sinha, Entamoeba histolytica HM-1: IMSS gene expression profiling identifies key hub genes, potential biomarkers, and pathways in Amoebiasis infection: A systematic network meta-analysis, *Bioscience Reports*, 2022.
- [22] J. Wang, Petri Nets for Dynamic Event-Driven System Modeling, *Handbook of Dynamic System Modeling*, **1** (2007).