Chapter 5.
Petri Nets (PN)

Mary Ann Lundteigen   Marvin Rausand

RAMS Group
Department of Mechanical and Industrial Engineering
NTNU

(Version 0.1)

NTNU – Trondheim
Norwegian University of
Science and Technology

Learning Objectives

The main learning objectives associated with these slides are to:

1. Explain the main attributes and concepts associated with Petri Nets
2. Explain how Petri Nets may be used for system reliability analysis
3. Discuss some pros and cons for the application of Petri Nets

The slides include topics from Chapter 5 in **Reliability of Safety-Critical Systems: Theory and Applications**. DOI:10.1002/9781118776353.

Outline of Presentation

About Petri nets

Petri Nets is a well known approach in many application areas, but has not until recently got a high status within the field of system reliability.

- Petri Nets was introduced by *Carl Adam Petri* in the 1960s
- It has bee widely used for analysis of telecommunication, software engineering, and transportation
- Petri Nets are now referenced by IEC 61508 as a suitable approach for reliability analysis
- IEC 62551 defines terminology and gives requirements for the use of Petri Nets

We will use the abbreviation PN for Petri nets in the following slides.

Places, transitions, and arcs

A PN constiute two basic nodes: *places*, drawn as circles, and *transitions*, drawn as bars.

- ☞ Places: Circles used to model *local states* or conditions (e.g., failed or functioning).
- ☞ Transitions: Bars used to model *local events* (e.g., failure or restoration).
- ☞ Arcs: Directed arcs that links places and transitions.

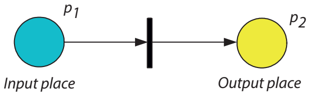


Figure: Input places vs output places

Tokens and firing

Tokens (black circles) are dynamic elements used to illustrate the system state at a certain point in time.

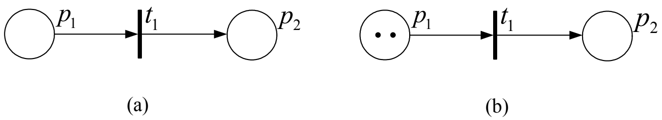


Figure: Simple PN, without (a) and with (b) tokens

Note that:

- A PN with tokens is called a *marked PN*
- Firing is the event where one or more tokens are moved from one place to another

Tokens and firing

Tokens (black circles) are dynamic elements used to illustrate the system state at a certain point in time.

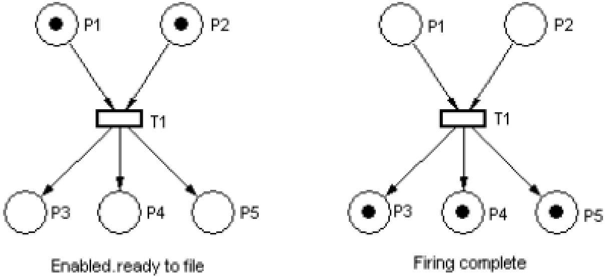


Fig 1

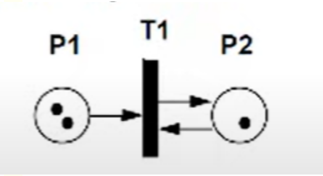


Fig 2

Note that:

- A PN with tokens is called a *marked PN*
- Firing is the event where one or more tokens are moved from one place to another

Multiplicity and enabling

☞ Multiplicity (or weight): A digit (e.g., 1, 2 etc) assigned to an arc, and which represents the number of tokens the arc delivers at a time.

It may be remarked that:

- When multiplicity is 1, it is not specified in the PN
- A transition is enabled (ready for firing) if the number of tokens in each of its input places is equal to or greater than the multiplicity of the associated arcs.

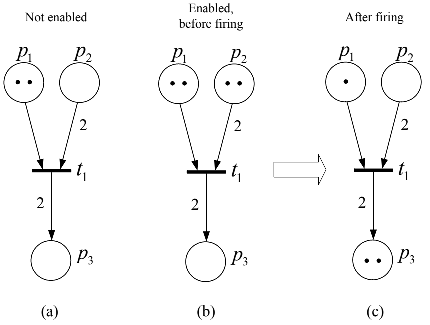
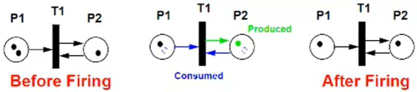


Figure: PN before and after firing



Inhibitor arc

☞ Inhibitor arc: A directed arc with by a small circle, and whose main purpose is to block the output transition if the number of tokens is in the input places are equal to or higher than the multiplicity (weight) of the arc.

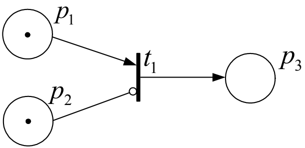


Figure: PN with inhibitor arc

Figure above: Transition t_1 is inhibited as long as there is a token in place p_2.

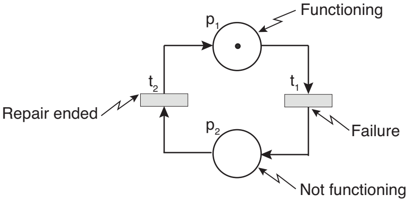
Transition types

Transitions may be immidiate (when enabling conditions are fulfilled), or timed with a delay > 0. The delays may be *deterministic* or *stochastic*, as shown in the table below:

	Type of transition in PN			
	Deterministic		***Stochastic***	
	Delay is zero	*Delay is d*	*Exponentially or geometrically distributed*	*Arbitrary distributed*
Parameter		d	λ	
Symbol				

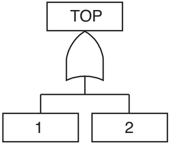
State of an item

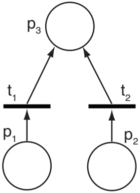
A PN may be used to model the functioning and failure of a single item, as shown below:



Fault tree

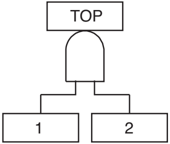
A PN may be used to model an OR gate in a fault tree, as shown below:

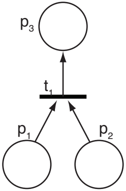




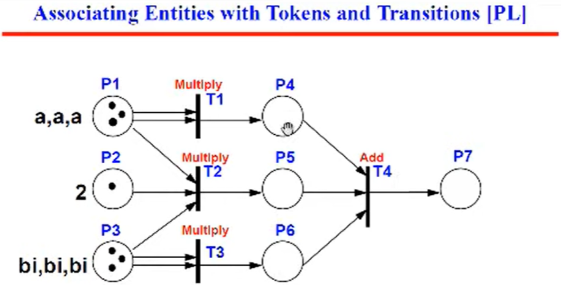
Fault tree

A PN may be used to model an AND gate in a fault tree, as shown below:





Associating Entities w/ Tokens and Transitions

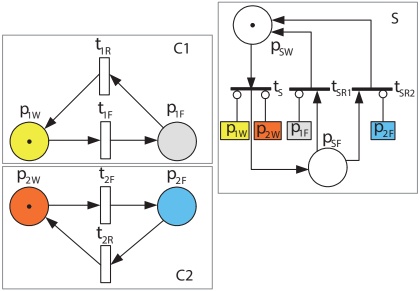


- Complex Multiplication: $(a+bi)(a+bi) = a^2 - b^2 + 2abi$

Modularization of PNs

It may be feasible to modularize the PN, by modeling each functional block (in a reliability block diagram). A module can include two parts:

- *Intrinsic part*: Modules that describes the behavoir of the items in the module
- *Extrinsic part*: Module that links the intrinsic part to system properties (e.g., failure, repair, working)



Note: Notations are as defined on previous slides. Note that i is the channel

Helpful Resource

1. https://www.youtube.com/watch?v=GCsVxWh995o