

# Critical Systems

---

Paulo Maciel

Centro de Informática - UFPE

# Objective

---

- Learn methods and techniques for assessing, modeling and evaluation of critical systems.

# Requirements

---

- Basic statistics and probability
- Discrete event systems

# Program

---

- Dependability
- Real Time Systems

# Program

---

## ■ Dependability

- History
- Basic concepts and terminology
- Background
- Reliability data analysis
- Detection and recovering mechanisms, and fault tolerance
- Coherent systems
- Operational and failure modes

# Program

## ■ Dependability

- Combinational models: RBD, FT, RG
  - Structural and logic functions
  - Analysis methods
  - Modeling
- CTMC modeling
- SPN modeling
- Hierarchical and heterogeneous modeling

# Program

---

## ■ Real Time Systems

- Characteristics and requirements
- Classification
- Task allocation and scheduling
- Performance measures for real time systems
- Models
  - Timed process algebras
  - Timed Petri nets
- Analysis, verification and estimation

# Methodology

---

- Expositive classes
- Lab. classes



# Evaluation

---

- Problem solving
- Homework

# Basic bibliography

- **Dependability Modeling.** Paulo Maciel. Kishor S. Trivedi, Rivalino Matias and Dong Kim. In: Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions ed. Hershey, Pennsylvania: IGI Global, 2011. Book Chapter.
- **Reliability, Maintainability and Risk: Practical methods for engineers,** David J Smith 8th edition, Elsevier. 2011.
- **Reliability: Probabilistic Models and Statistical Methods,** Lawrence M. Leemis, 2<sup>nd</sup> Edition, ISBN: 978-0-692-00027-4, 2009.
- **Uma Introdução às Redes de Petri e Aplicações.** MACIEL, P. R. M.; LINS, R. D.; CUNHA, Paulo Roberto Freire. Sociedade Brasileira de Computação, 1996. v. 1. 213 p.
- **Modelling with Generalized Stochastic Petri Nets,** Marsan, A., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G., *Wiley Series in Parallel Computing*, 1995.
- **Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications,** Second Edition, **Gunter Bolch, Stefan Greiner, Hermann de Meer,** Kishor S. Trivedi, WILEYINTERSCIENCE, 2007.
- **Probability and Statistics with Reliability, Queueing, and Computer Science Applications,** Trivedi. K., *2nd edition, Wiley*, 2002.
- **Fundamental Concepts of Computer System Dependability,** A. Avizienis, J. Laprie, B. Randell, IARP/IEEE-RAS Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments – Seoul, Korea, May 21-22, 2001

# Dependability

---

Dependability of a computing system is the ability to deliver service that can justifiably be trusted.

The service delivered by a system is its behavior as it is perceived by its user(s).

A user is another system (physical, human) that interacts with the former at the service interface.

The function of a system is what the system is intended for, and is described by the system specification.

[Laprie, J. C. (1985)].

# Dependability

In early 1980s Laprie coined the term dependability for encompassing concepts such reliability, availability, safety, confidentiality, maintainability, security and integrity etc [Laprie, J. C. (1985)].

Dependable Computing and Fault Tolerance: Concepts and terminology. In Proc. 15th IEEE Int. Symp. on Fault-Tolerant Computing, (pp. 2-11).



Jean Claude Laprie



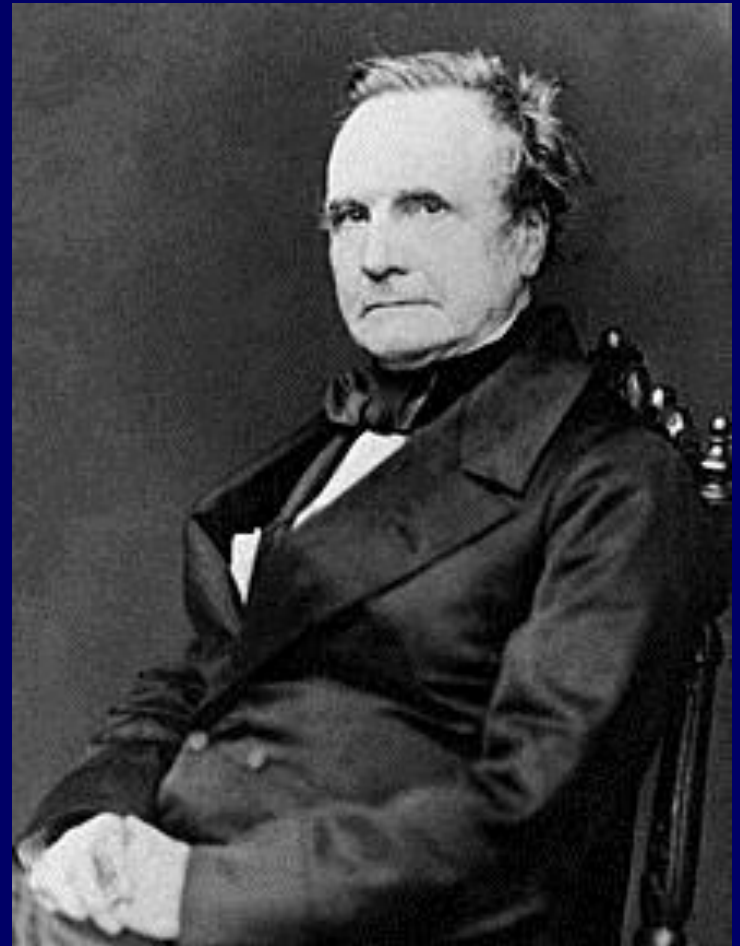
# **A BRIEF HISTORY**

# A Brief History

Dependability is related to disciplines such as reliability and fault tolerance.

The concept of dependable computing first appeared in 1820s when Charles Babbage undertook the enterprise to conceive and construct a mechanical calculating engine to eliminate the risk of human errors. In his book, "On the Economy of Machinery and Manufacture", he mentions "The first objective of every person who attempts to make any article of consumption is, or ought be, to produce it in perfect form'.

" (Blischke, W. R. & Murthy, D. N. P. (Ed.) 2003).



Charles Babbage in 1860

# A Brief History

---

In the nineteenth century, reliability theory evolved from probability and statistics as a way to support computing maritime and life insurance rates.

In early twentieth century methods had been applied to estimate survivorship of railroad equipment [Stott, H. G. (1905)] [Stuart, H. R. (1905)].

# A Brief History

---

The first IEEE (formerly AIEE and IRE) public document to mention reliability is “Answers to Questions Relative to High Tension Transmission” that summarizes the meeting of the Board of Directors of the American Institute of Electrical Engineers, held in September 26, 1902.

[Answers to Questions Relative to High Tension Transmission. (1904). Transactions of the American Institute of Electrical Engineers, XXIII, 571-604.]

In 1905, H. G. Stott and H. R. Stuart: discuss “Time-Limit Relays and Duplication of Electrical Apparatus to Secure Reliability of Services at New York and at Pittsburg.

In these works the concept of reliability was primarily qualitative.

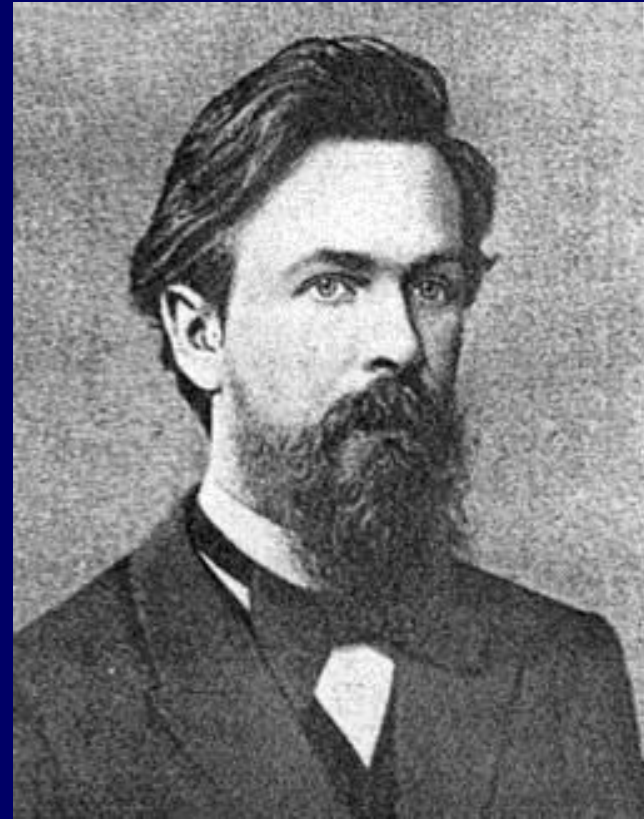


# A Brief History

In 1907, A. A. Markov began the study of an important new type of chance process.

In this process, the outcome of a given experiment can affect the outcome of the next experiment.

This type of process is now called a Markov chain [Ushakov, I. (2007)]



Andrei A. Markov

# A Brief History

In 1910s, A. K. Erlang studied telephone traffic planning problems for reliable service provisioning [Erlang, A. K. (1909)].

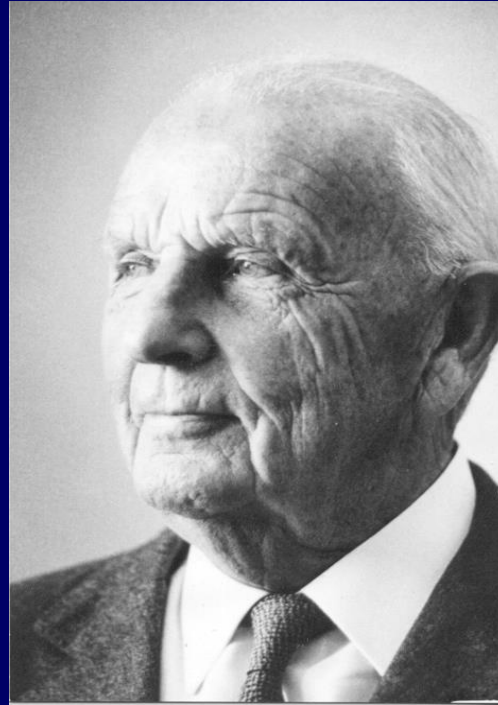


Agner Karup Erlang

[Erlang, A. K. (1909)] Principal Works of A. K. Erlang - The Theory of Probabilities and Telephone Conversations . First published in Nyt Tidsskrift for Matematik B, 20, 131-137.

# A Brief History

Later in the 1930s,  
extreme value theory was  
applied to model fatigue life of  
materials by W. Weibull and  
Gumbel [Kotz, S., Nadarajah, S. (2000)].



**Waloddi Weibull**  
1887-1979



**Gumbel, Emil Julius**  
(18.7.1891 -  
10.9.1966)

# A Brief History

In 1931, Kolmogorov, in his famous paper “Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung” (Analytical methods in probability theory) laid the foundations for the modern theory of Markov processes [Kolmogoroff, A. (1931)].

Kolmogoroff, A. (1931). Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung (in German). *Mathematische Annalen*, 104, 415-458. Springer-Verlag.



**Andrey Nikolaevich Kolmogorov**  
(25 April 1903 – 20 October 1987)

# A Brief History

---

In the 1940s quantitative analysis of reliability was applied to many operational and strategic problems in World War II [Blischke, W. R. & Murthy, D. N. P. (Ed.) (2003)] [Cox, D. R. (1989)].

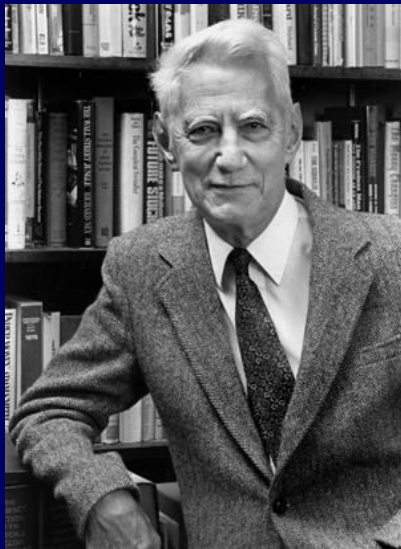
The first generation of electronic computers were quite undependable, thence many techniques were investigated for improving their reliability, such as error:

- control codes,
- replication of components,
- comparison monitoring and
- diagnostic routines.

# A Brief History

The most prominent researchers during that period were Shannon, Von Neumann and Moore, who proposed and developed theories for building reliable systems by using redundant and less reliable components.

These were the predecessors of the statistical and probabilistic techniques that form the foundation of modern dependability theory [Avizienis, A. (1997)].



C. E. Shannon



John von Neumann



Edward Forrester Moore

# A Brief History

---

In the 1950s, reliability became a subject of great engineering interest as a result of the:

- cold war efforts,
- failures of American and Soviet rockets, and
- failures of the first commercial jet aircraft, the British de Havilland comet [Barlow, R. E. & Proschan, F. (1967)][Barlow, R. E. (2002)].

# A Brief History

---

Epstein and Sobel's 1953 paper studying the exponential distribution was a landmark contribution.

Epstein, B. & Sobel, M. (1953). Life Testing. *Journal of the American Statistical Association*, 48(263), 486-502.



Milton Sobel



# A Brief History

In 1954, the Symposium on Reliability and Quality Control (it is now the IEEE Transactions on Reliability) was held for the first time in the United States.

In 1958, the First All-Union Conference on Reliability took place in Moscow [Gnedenko, B. V., Ushakov, I. A. (1995)] [Ushakov, I. (2007)].



Gnedenko Boris V.  
(1912-1995)

Gnedenko, B. V., Ushakov, I. A. (1995). Probabilistic Reliability Engineering. J. A. Falk (Ed.), Wiley-Interscience.

Ushakov, I. (2007). Is Reliability Theory Still Alive?. e-journal "Reliability: Theory & Applications", 1(2).

# A Brief History

---

In 1957 S. J. Einhorn and F. B. Thiess adopted Markov chains for modeling system intermittence [Einhorn, S. J. & Thiess, F. B. (1957)].

In 1960, P. M. Anselone employed Markov chains for evaluating availability of radar systems [Anselone, P. M. (1960)].

In 1961 Birnbaum, Esary and Saunders published a milestone paper introducing coherent structures [Birnbaum, Z. W., J. D. Esary and S. C. Saunders. (1961)].

# A Brief History

Fault Tree Analysis (FTA) was originally developed in 1962 at Bell Laboratories by H. A. Watson to evaluate the Minuteman I Intercontinental Ballistic Missile Launch Control System.

Afterwards, in 1962, Boeing and AVCO expanded use of FTA to the entire Minuteman II.



Minuteman I



Minuteman II

# A Brief History

---

In 1967, A. Avizienis integrated masking methods with practical techniques for error detection, fault diagnosis, and recovery into the concept of fault-tolerant systems [Avizienis, A., Laprie, J.-C., Randell, B. (2001)].

Fundamental Concepts of Dependability. LAAS-CNRS, Technical Report N01145.



A. Avizienis

# A Brief History

---

In late 1970s some works were proposed for mapping Petri nets to Markov chains [Molloy, M. K. (1981)][Natkin, S. 1980][Symons, F. J. W. 1978].

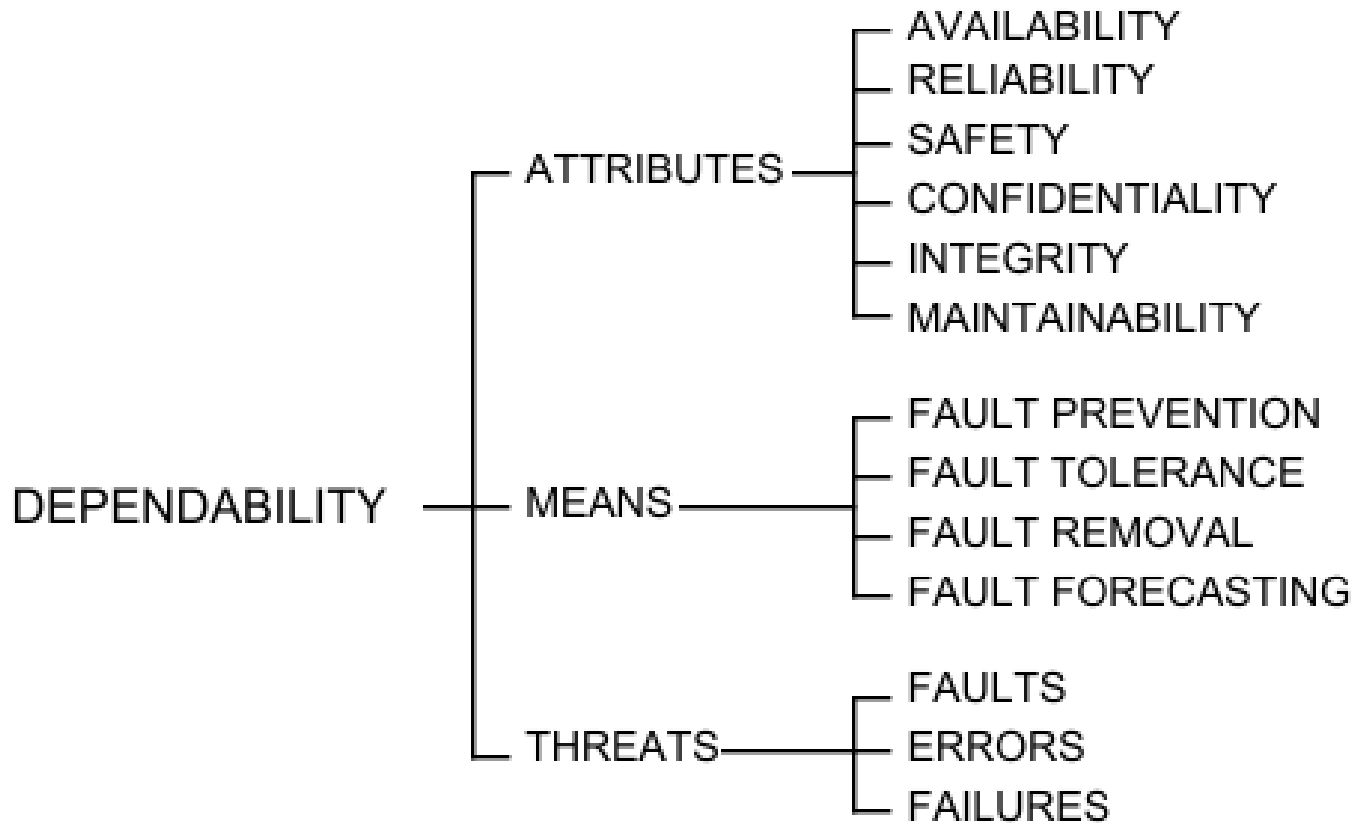
These models have been widely adopted as high-level Markov chain automatic generation models as well as for discrete event simulation.

Natkin was the first to apply what is now generally called Stochastic Petri nets to dependability evaluation of systems.



# **BASIC CONCEPTS**

# Basic Concepts



The dependability tree

Avizienis, A., Laprie, J.-C., Randell, B. (2001).  
Fundamental Concepts of Dependability. LAAS-CNRS,  
Technical Report N01145.

# Basic Concepts

---

Dependability of a system is the ability to deliver service that can justifiably be trusted.

A correct service is delivered when the service implements what is specified.

- A system failure is an event that occurs when the delivered service deviates from correct service.

A failure is thus a transition from correct service to incorrect service.

A transition from incorrect service to correct service is service restoration.



# Basic Concepts

- An error is that part of the system state that may cause a subsequent failure.

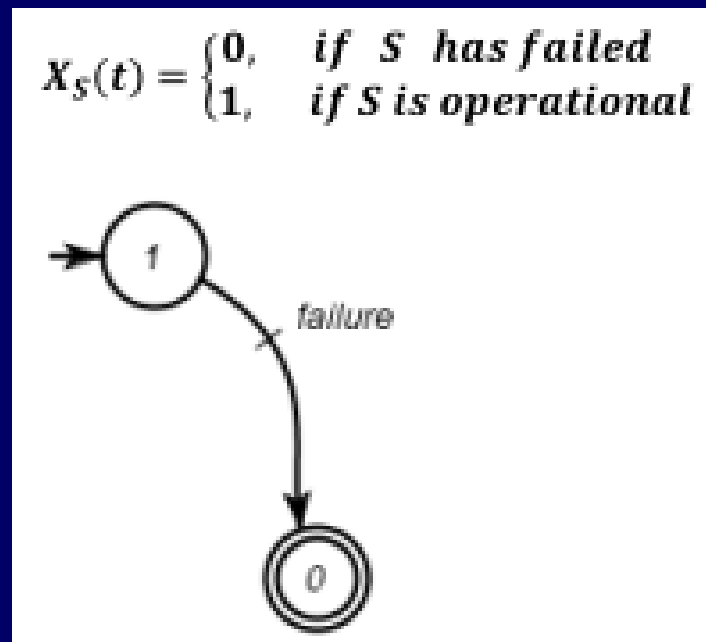
A failure occurs when an error reaches the system interface and alters the service.



# Basic Concepts

- Fault is the adjudged or hypothesized cause of an error.

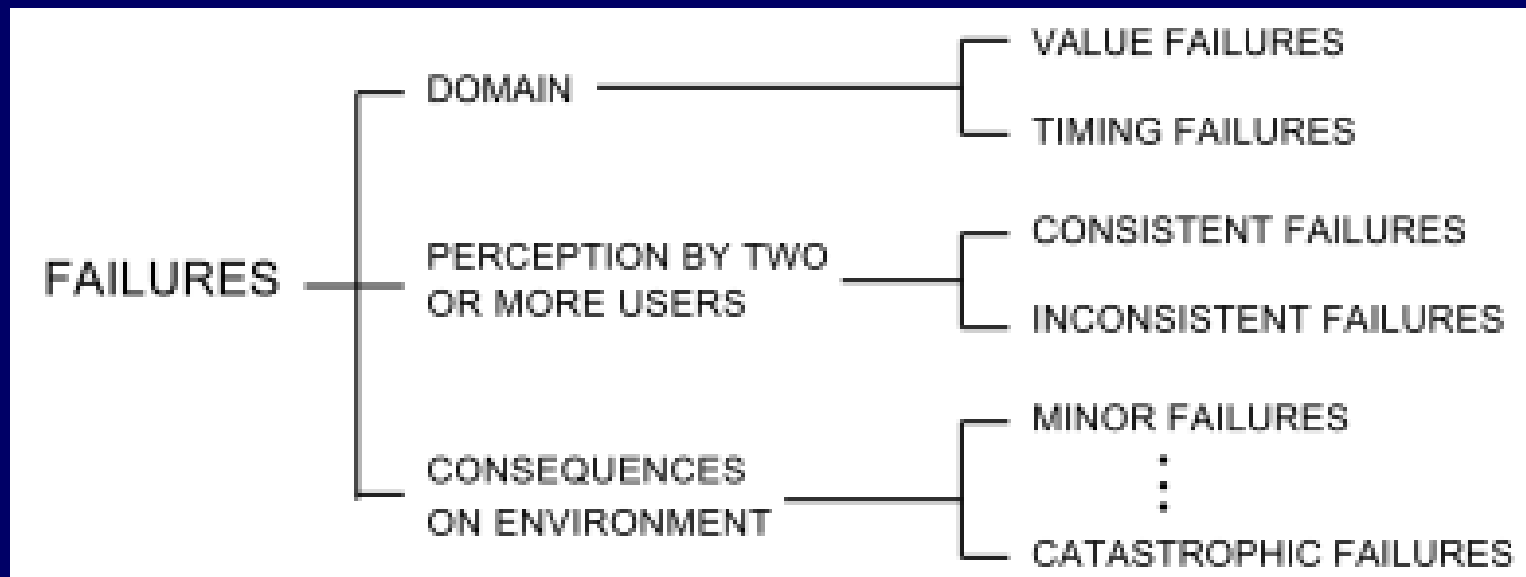
A fault is **active** when it produces an error; otherwise it is **dormant**.



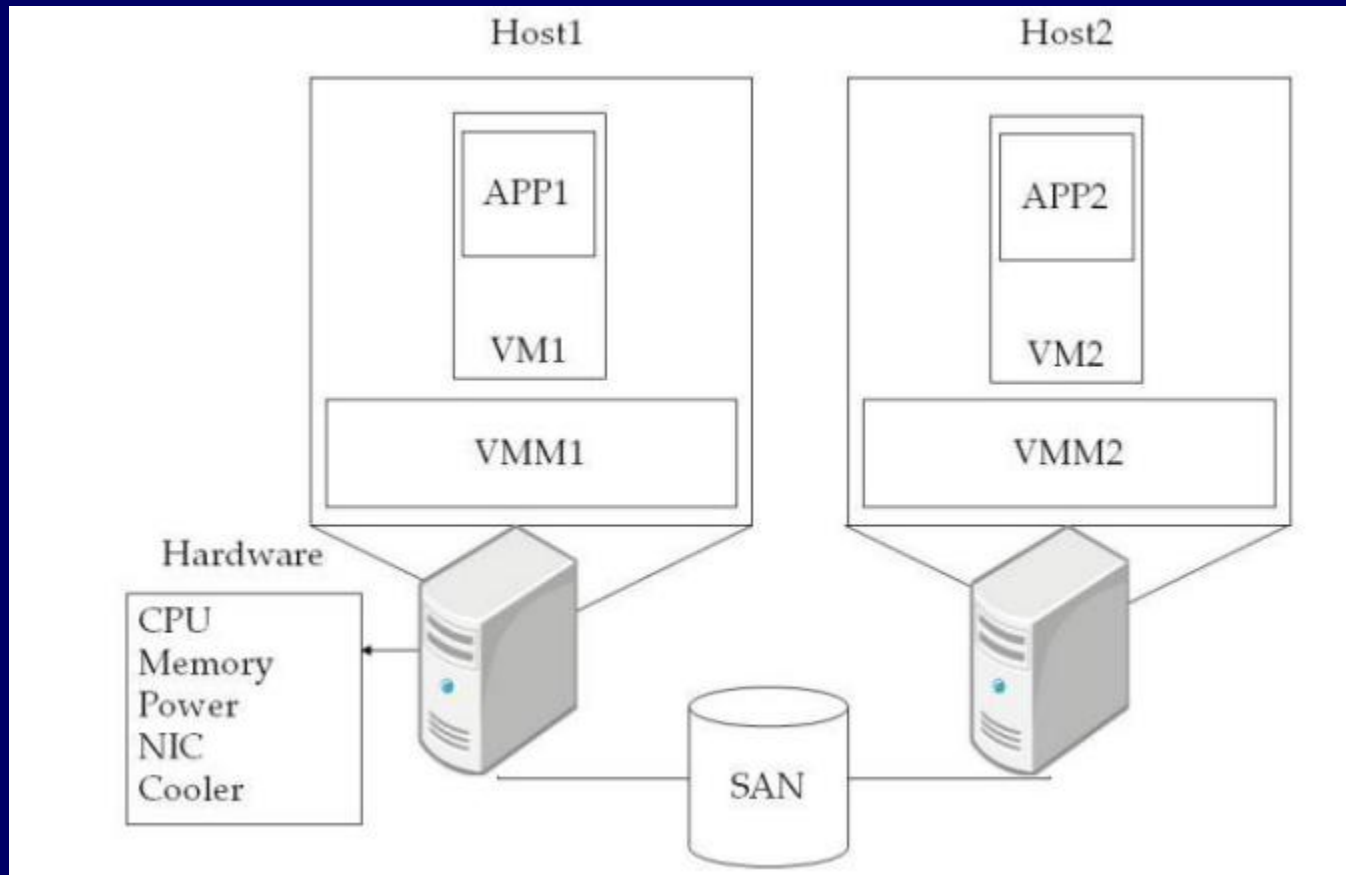
Consider an indicator random variable  $X(t)$  that represents the system state at time  $t$ .

# Basic Concepts

## ■ Failure Modes



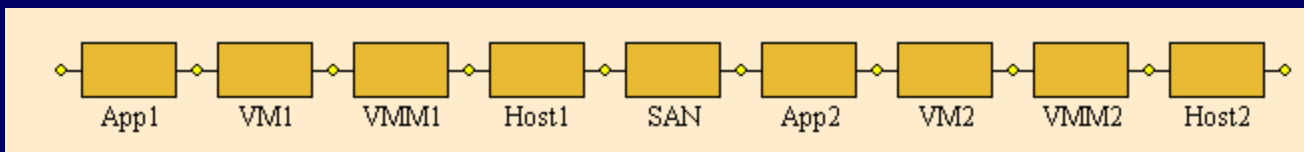
# A motivational example



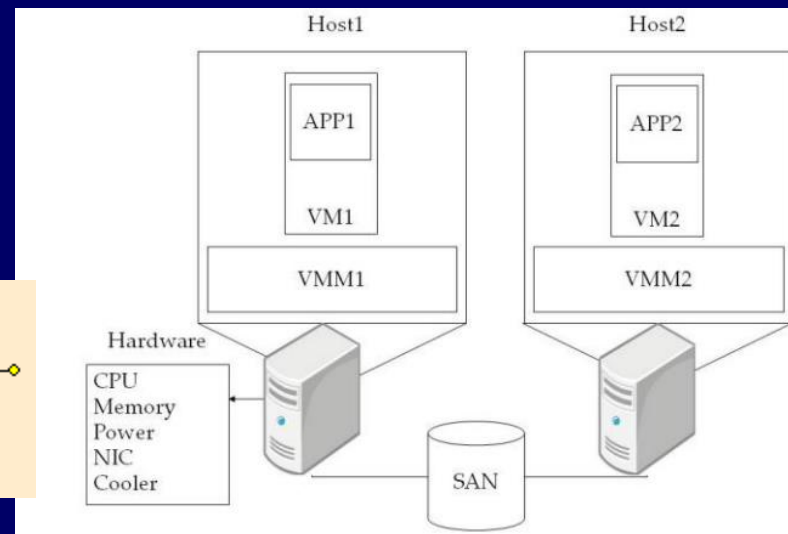
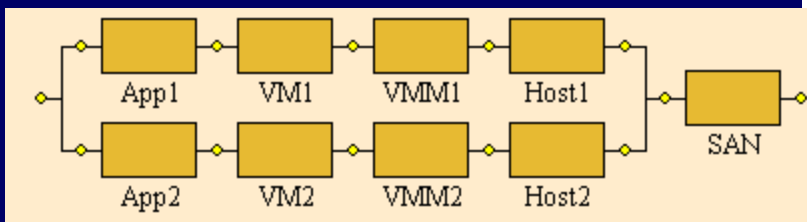
# A motivational example

- What is the respective RBD?

This?



Or this?



# A motivational example

---

- It is not clear.

Something is still missing!

- What is it?

The operational mode(s)

(success oriented networks: RBD and Relgraph)

or

The failure mode(s)

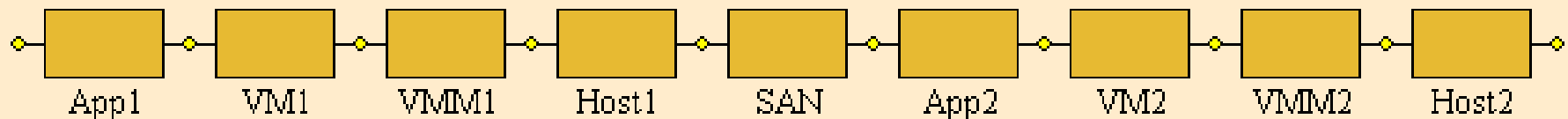
(failure oriented networks: FT)

# Operational Mode

is a condition that defines the system as operational.

- Operational Mode 1

$$OM_1 = App_1 \wedge VMM_1 \wedge VM_1 \wedge H_1 \wedge SAN \\ \wedge App_2 \wedge VMM_2 \wedge VM_2 \wedge H_2$$

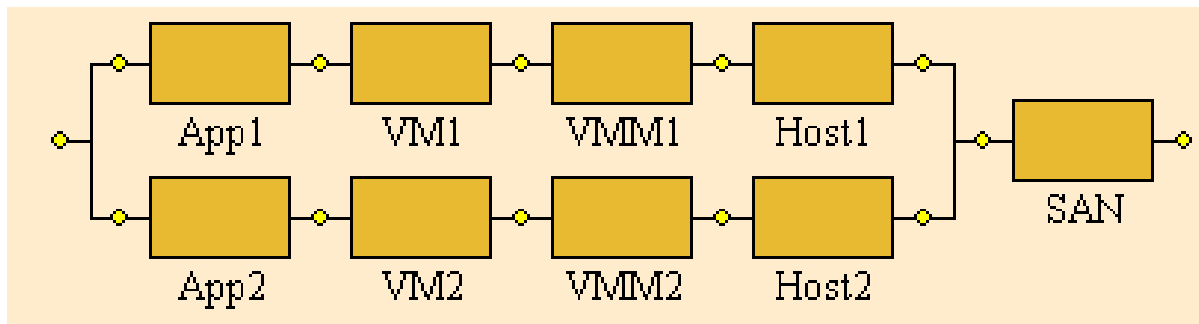


$$R(t) = 0.805735302, \quad t = 0.002 \text{ tu}$$

# Operational Mode

## Operational Mode 2

$$OM_2 = ((App_1 \wedge VMM_1 \wedge VM_1 \wedge H_1) \vee (App_2 \wedge VMM_2 \wedge VM_2 \wedge H_2)) \wedge SAN$$



$$R(t) = 0.975215145, \quad t = 0.002 \text{ } tu$$



# Basic Concepts

---

- Fault prevention: how to prevent the occurrence or introduction of faults;
- Fault tolerance: how to deliver correct service in the presence of faults;
- Fault removal: how to reduce the number or severity of faults;
- Fault forecasting: how to estimate the present number, the future incidence, and the likely consequences of faults.

# Basic Concepts

---

Fault prevention is attained by quality control techniques employed during the design and manufacturing of hardware and software, including structured programming, information hiding, modularization, and rigorous design.

Operational physical faults are prevented by shielding, radiation hardening, etc.

Interaction faults are prevented by training, rigorous procedures for maintenance, "foolproof" packages.

Malicious faults are prevented by firewalls and similar defenses.

# Basic Concepts

---

**Fault Tolerance** is intended to preserve the delivery of correct service in the presence of active faults.

- Active strategies

  - Phase:

    - 1) Error detection

    - 2) Recovery

- Passive strategies

  - Fault masking

# Basic Concepts

---

**Fault Removal** is performed both during the **development phase**, and during the **operational life** of a system.

Fault removal during the development phase of a system life-cycle consists of three steps: **verification, diagnosis, correction**.

**Checking the specification** is usually referred to as **validation**.

# Basic Concepts

---

**Fault Forecasting** is conducted by performing an evaluation of the system behavior with respect to fault occurrence or activation.

Classes:

**qualitative evaluation** identifies event combinations that would lead to system failures;

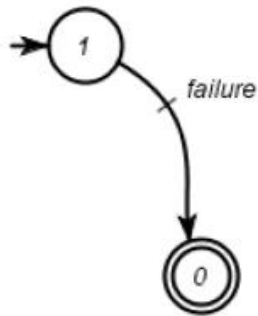
**probabilistic evaluation** evaluates the probabilities of attributes of dependability are satisfied.

The methods for qualitative and quantitative evaluation are either specific (e.g., failure mode and effect analysis for qualitative evaluation, or Markov chains and stochastic Petri nets for quantitative evaluation), or they can be used to perform both forms of evaluation (e.g., reliability block diagrams, fault-trees).

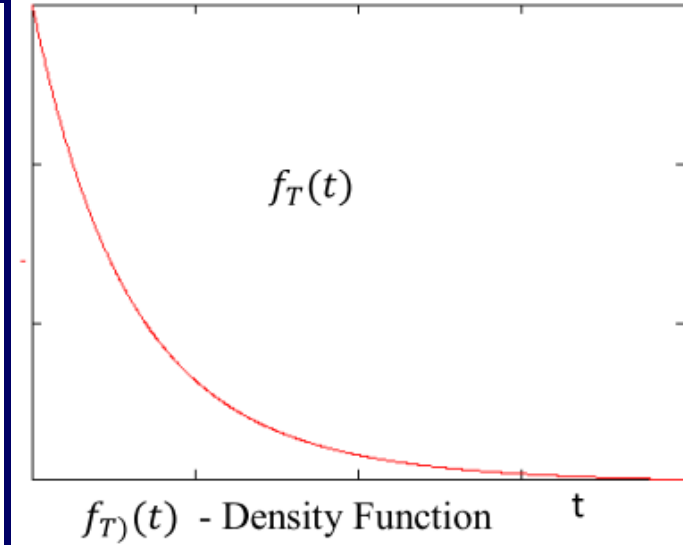
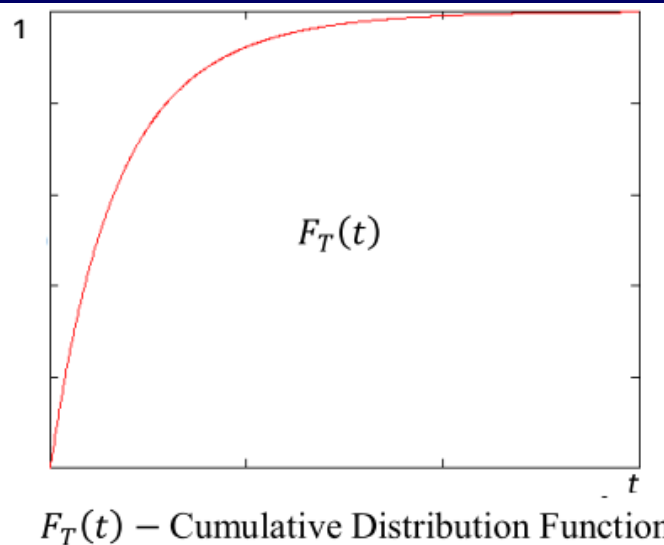
# Basic Concepts

- Time to Failure

$$X_S(t) = \begin{cases} 0, & \text{if } S \text{ has failed} \\ 1, & \text{if } S \text{ is operational} \end{cases}$$



States of  $X_S(t)$



Now, consider a random variable  $T$  as the time to reach the state  $X(t) = 0$ , given that the system started in state  $X(t) = 1$  at time  $t = 0$ . Therefore, the random variable  $T$  represents the **time to failure** of the system  $S$ ,  $F_T(t)$  its **cumulative distribution function**, and  $f_T(t)$  the respective **density function**, where:

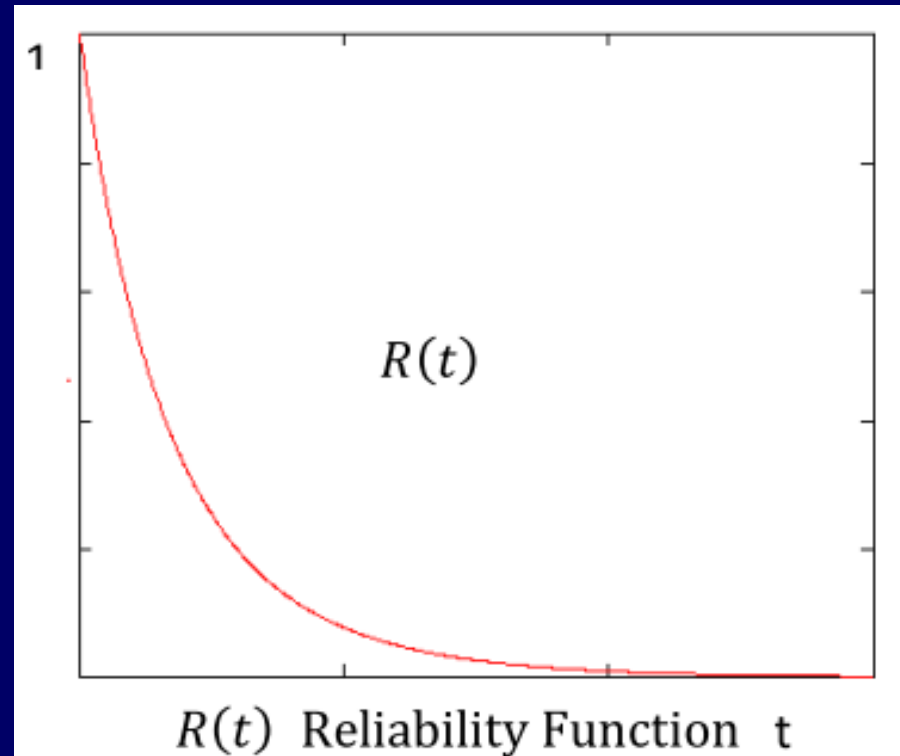
$$F_T(0) = 0 \quad \text{and} \quad \lim_{t \rightarrow \infty} F_T(t) = 1,$$

$$f_T(t) = \frac{dF_T}{dt},$$

$$\int_0^{\infty} f_T(t) \times dt = 1$$

# Basic Concepts

- Reliability



The probability that the system  $S$  does not fail up to time  $t$  (reliability) is

$$P\{T \geq t\} = R(t) = 1 - F_T(t),$$

$$R(0) = 1 \quad \text{and} \quad \lim_{t \rightarrow \infty} R(t) = 0.$$

# Basic Concepts

---

- Reliability

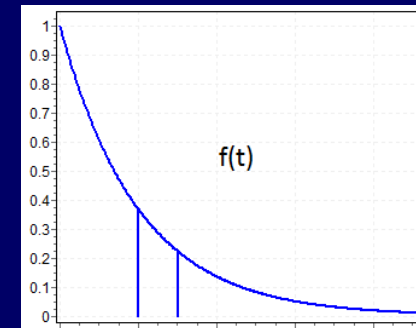
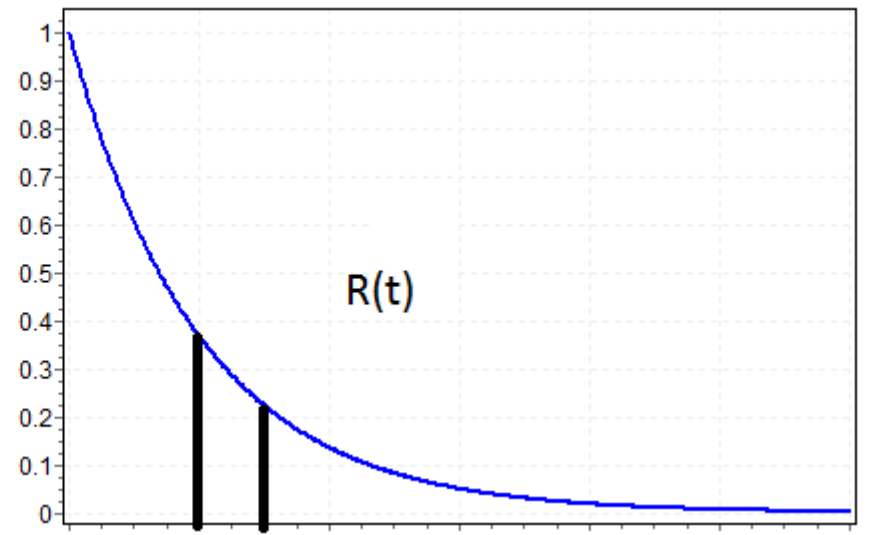
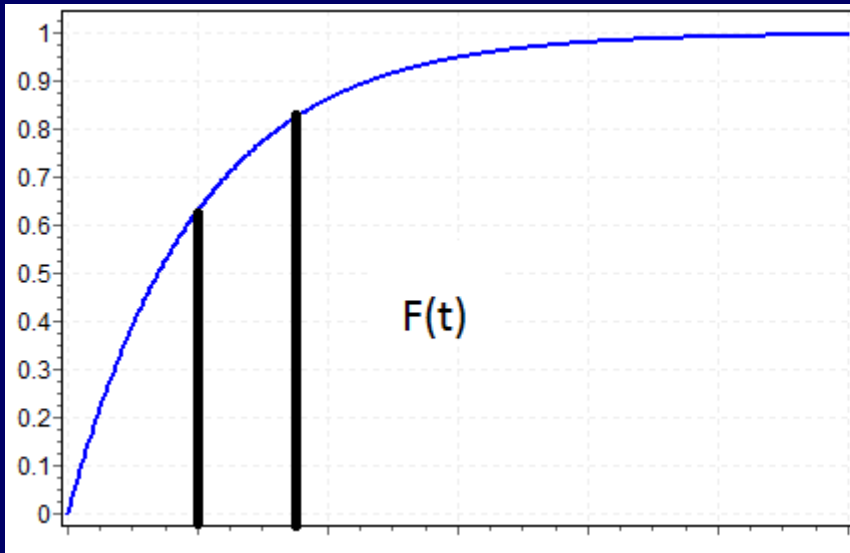
Reliability (Survivor function) - Complementary of the distribution

function:  $R(t) = 1 - F(t)$



# Basic Concepts

- Hazard function



# Basic Concepts

- Hazard function

The probability of the system S failing during the interval  $[t, t + \Delta t]$  if it has survived to the time  $t$  (conditional probability of failure) is

$$P\{t \leq T \leq t + \Delta t | T > t\} =$$

$$\frac{R(t) - R(t + \Delta t)}{R(t)}.$$

$P\{t \leq T \leq t + \Delta t | T > t\} / \Delta t$  is conditional probability of failure per time unit. When  $\Delta t \rightarrow 0$ , then

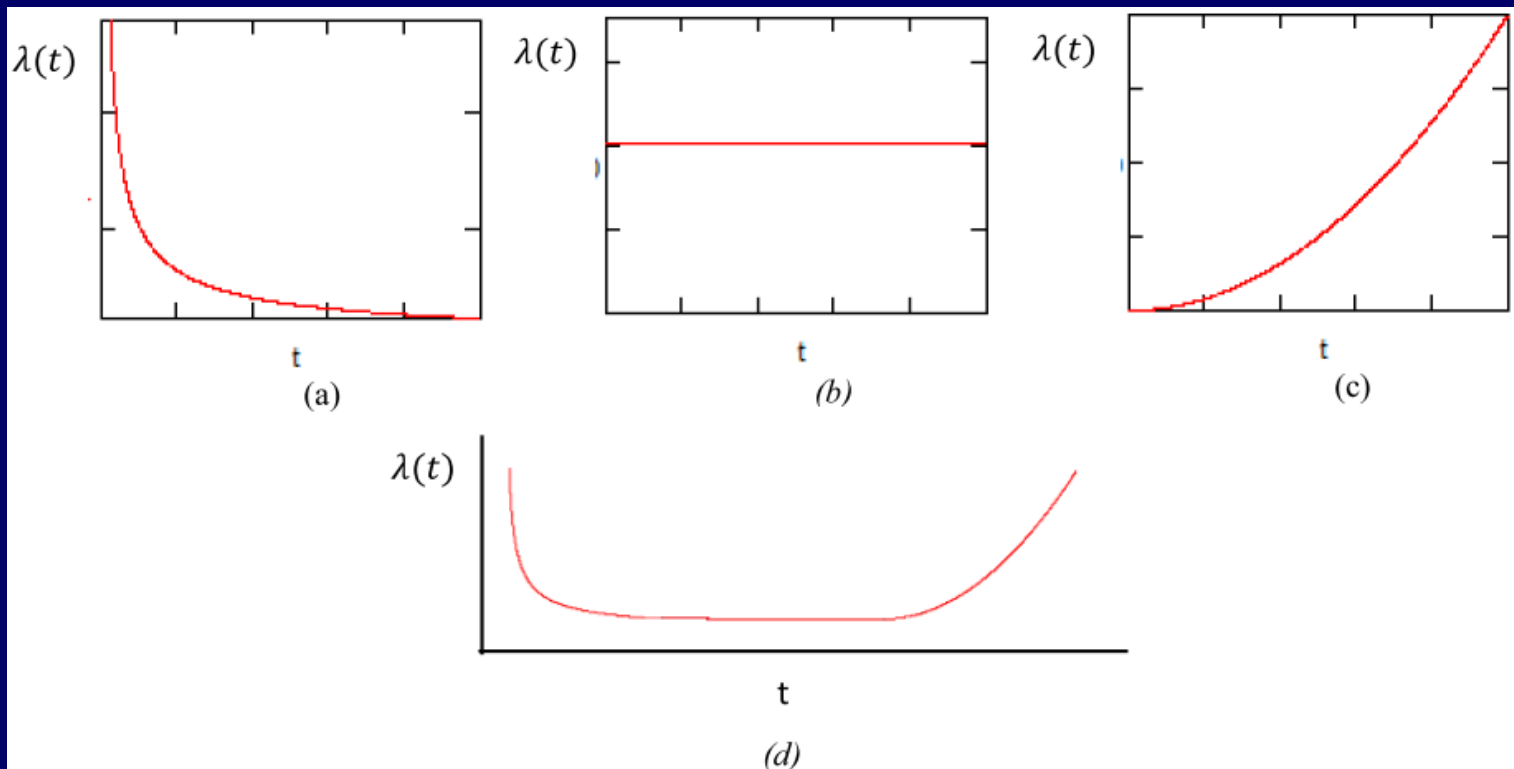
$$\lim_{\Delta t \rightarrow 0} \frac{R(t) - R(t + \Delta t)}{R(t) \times \Delta t} = \lim_{\Delta t \rightarrow 0} \frac{-[R(t + \Delta t) - R(t)]}{\Delta t} \times \frac{1}{R(t)} = -\frac{dR(t)}{dt} \times \frac{1}{R(t)} =$$

$$\frac{dF_T(t)}{dt} \times \frac{1}{R(t)} = \frac{f_T}{R(t)} = \lambda(t)$$

# Basic Concepts

- Hazard function

Hazard rates may be characterized as decreasing failure rate (DFR), constant failure rate (CFR) or increasing failure rate (IFR) according to  $\lambda(t)$ .



Hazard rate: (a) Decreasing, (b) Constant, (c) Increasing, (d) Bathtub curve

# Basic Concepts

- Cumulative Hazard function

Since

$$\lambda(t) = -\frac{dR(t)}{dt} \times \frac{1}{R(t)},$$

$$\lambda(t)dt = -\frac{dR(t)}{R(t)},$$

thus,

$$\int_0^t \lambda(t)dt = -\int_0^t \frac{dR(t)}{R(t)} =$$

$$-\int_0^t \lambda(t)dt = \ln R(t) =$$

$$R(t) = e^{-\int_0^t \lambda(t)dt} = e^{-H(t)}$$

# Basic Concepts

- Mean Time To Failure

$$MTTF = E[T] = \int_0^{\infty} t \times f_T(t) dt.$$

Since

$$f_T(t) = \frac{dF_T}{dt} = -\frac{dR(t)}{dt},$$

thus,

$$MTTF = E[T] = -\int_0^{\infty} \frac{dR(t)}{dt} \times t dt.$$

Let  $u = t$ ,  $dv = \frac{dR(t)}{dt} \times dt$ , and applying integration by parts ( $\int u dv = uv - \int v du$ ), then  $du = dt$ ,  $v = R(t)$ , hence:

# Basic Concepts

- Mean Time To Failure

$$\begin{aligned} MTTF &= - \int_0^{\infty} \frac{dR(t)}{dt} \times t dt = - \left[ t \times R(t) \Big|_0^{\infty} - \int_0^{\infty} R(t) \times dt \right] = \\ &= - \left[ 0 - \int_0^{\infty} R(t) \times dt \right] = \int_0^{\infty} R(t) \times dt, \end{aligned}$$

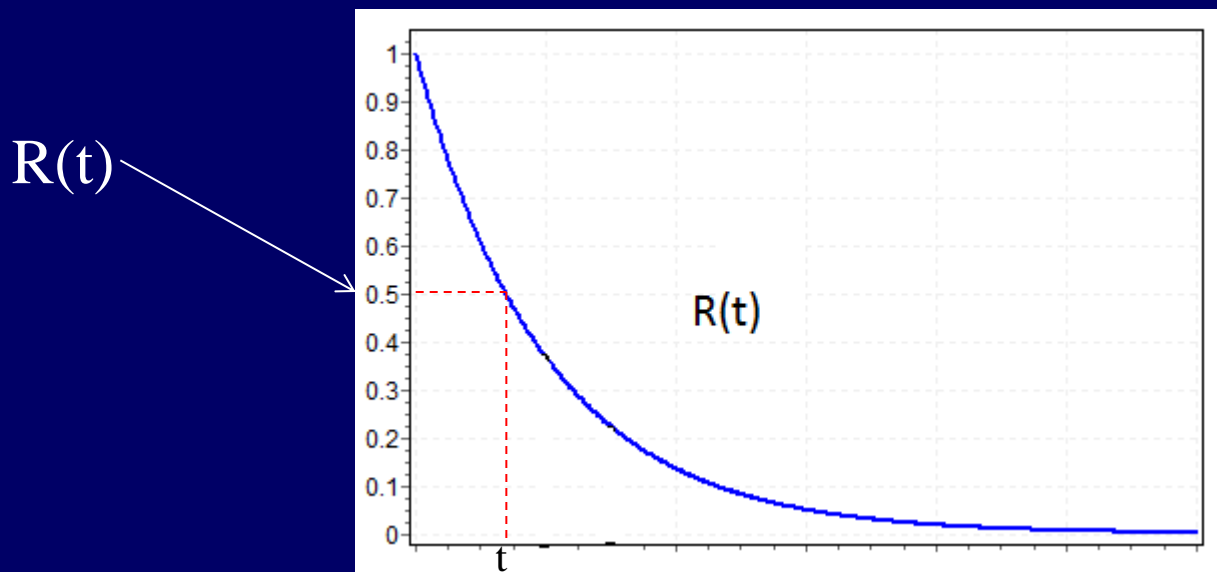
hence

$$MTTF = \int_0^{\infty} R(t) \times dt$$

# Basic Concepts

- Median Time To Failure

$$MedTTF = t, F_T = R(t) = 0.5$$

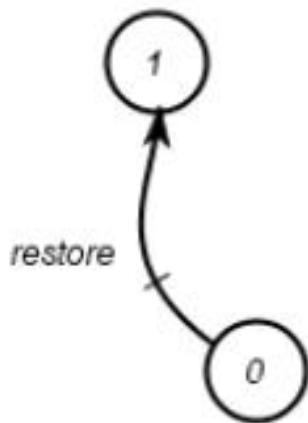


The median time to failure divides the time to fail distribution into two halves, where 50% of failures occur before  $MedTTF$  and the other 50% after.

# Basic Concepts

Consider a continuous time random variable  $X_S(t)$  that represents the system state.  $X_S(t)=0$  when  $S$  is failed,  $X_S(t)=1$  when  $S$  is operational

$$X_S(t) = \begin{cases} 0, & \text{if } S \text{ has failed} \\ 1, & \text{if } S \text{ is operational} \end{cases}$$



States of  $X_S(t)$

Now, consider the random variable  $D$  that represents the time to reach the state  $X_S(t)=1$ , given that the system started in state  $X_S(t)=0$  at time  $t=0$ .

Therefore, the random variable  $D$  represents the system **time to repair**,

$F_D(t)$  its **cumulative distribution function**, and  $f_D(t)$  the respective **density function**

$$F_D(0) = 0 \quad \text{and} \quad \lim_{t \rightarrow \infty} F_D(t) = 1,$$

$$f_D(t) = \frac{dF_D(t)}{dt},$$

$$f_D(t) \geq 0, \text{ and}$$

$$\int_0^{\infty} f_D(t) \times dt = 1$$



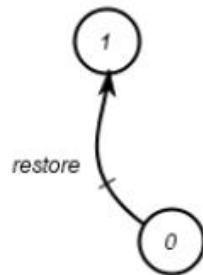
# Basic Concepts

- Maintainability

The **probability** that the system **S** will be repaired by **t** is defined as **maintainability**.

$$M(t) = P\{D \leq t\} = F_D(t) = \int_0^t f_D(t) \times dt$$

$$X_S(t) = \begin{cases} 0, & \text{if } S \text{ has failed} \\ 1, & \text{if } S \text{ is operational} \end{cases}$$



States of  $X_S(t)$

# Basic Concepts

- Mean Time To Repair

The **mean time to repair (MTTR)** is defined by:

$$MTTR = E[D] = \int_0^{\infty} t \times f_D(t) dt$$

An alternative often easier to compute *MTTR* is

$$MTTR = \int_0^{\infty} M(t) \times dt.$$

# Basic Concepts

## ■ Repairable Systems

Consider a repairable system  $S$  that is either operational (Up) or faulty (Down). Whenever the system fails, a set of activities are conducted in order to allow the restoring process.

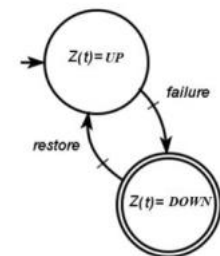
These activities might encompass administrative time, transportation time, logistic times etc.

When the maintenance team arrives to the system site, the actual repairing process may start.

Further, this time may also be divided into diagnosis time and actual repair time, checking time etc.

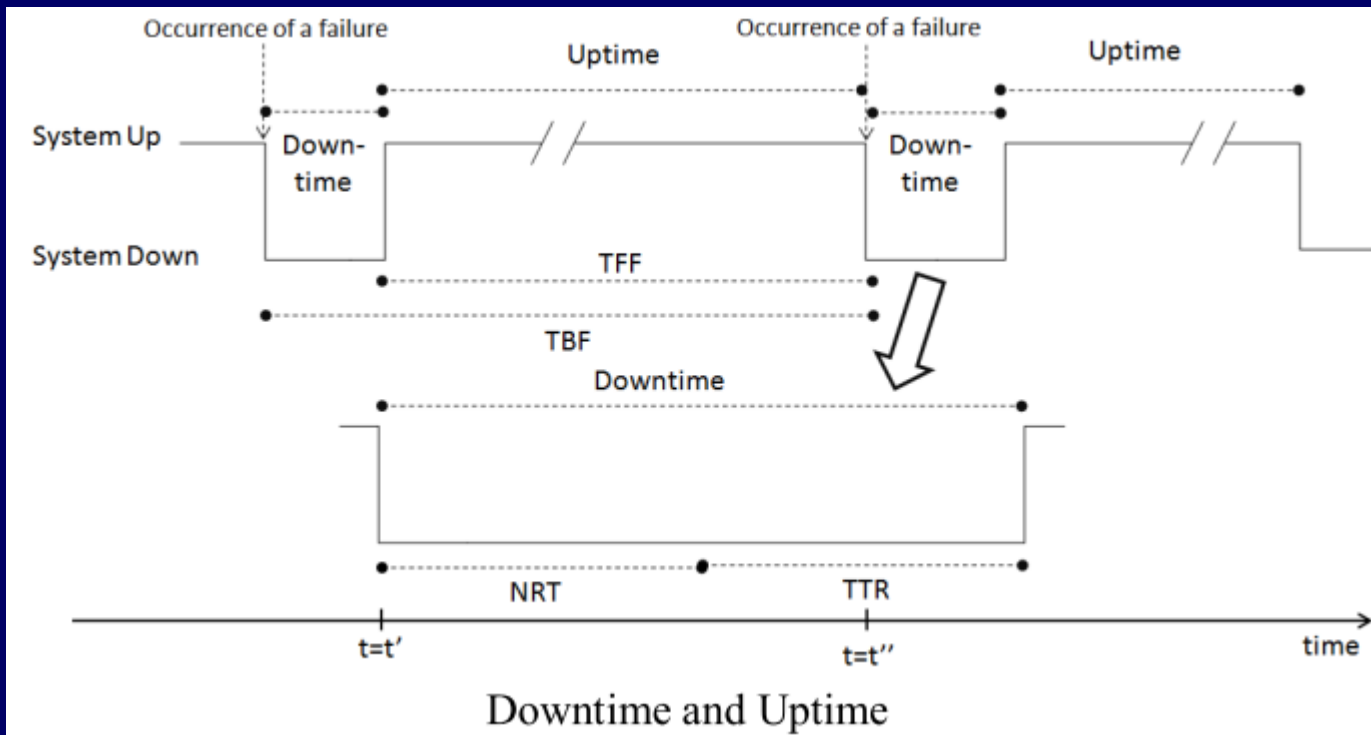
However, for sake of simplicity, we group these times such that the **downtime** equals the **time to restore** –  $TR$ , which is composed by **non-repair time** –  $NRT$  – (that groups transportation time, order times, deliver times, etc.) and **time to repair** –  $TTR$

$$\text{Downtime} = TR = NRT + TTR.$$



# Basic Concepts

- Downtime and Uptime



# Basic Concepts

- Availability

The simplest definition of **Availability** is expressed as the ratio of the expected system uptime to the expected system up and downtimes:

$$A = \frac{E[Uptime]}{E[Uptime] + E[Downtime]}$$

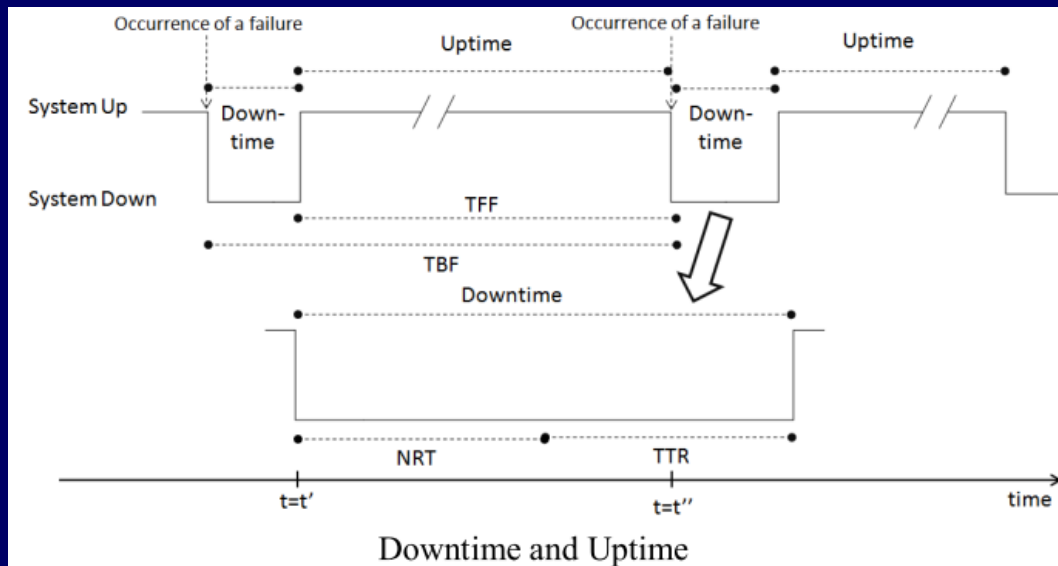
# Basic Concepts

## ■ Availability

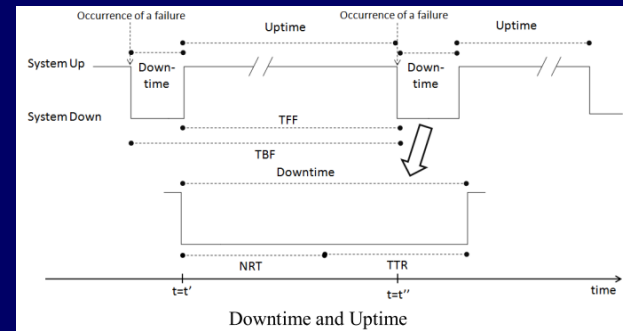
Consider that the system started operating at time  $t = t'$  and fails at  $t = t''$ , thus  $\Delta t = t'' - t' = \text{Uptime}$ .

Therefore, the system availability may also be expressed by:

$$A = \frac{MTTF}{MTTF + MTR}$$



# Basic Concepts



- Availability

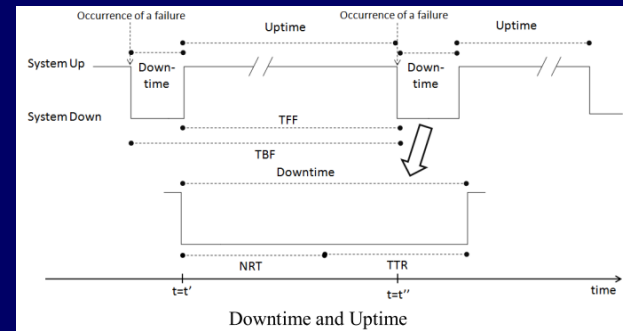
where ***MTR*** is the **mean time to restore**, defined by  $MTR = MNRT + MTTR$  (***MNRT*** – mean non-repair time, ***MTTR*** – mean time to repair), so:

$$A = \frac{MTTF}{MTTF + MNRT + MTTR}$$

If  $MNRT \cong 0$ ,

$$A = \frac{MTTF}{MTTF + MTTR}$$

# Basic Concepts



- Availability

As  $MTBF = MTTF + MTR = MTTF + MNRT + MTTR$ ,  
and if  $MNRT \cong 0$ , then  $MTBF = MTTF + MTTR$ .

Since  $MTTF \gg MTTR$ , thus  $MTBF \cong MTTF$ , therefore:

$$A = \frac{MTBF}{MTBF + MTTR}$$



# Basic Concepts

- Instantaneous Availability

The instantaneous availability is the probability that the system is operational at  $t$ , that is,

$$A(t) = P\{Z(t) = 1\} = E\{Z(t)\}, \quad t \geq 0.$$

If repairing is not possible, the instantaneous availability,  $A(t)$ , is equivalent to reliability,  $R(t)$ .

# Basic Concepts

- Steady State Availability

If the system approaches stationary states as the time increases, it is possible to quantify the steady state availability

$$A = \lim_{t \rightarrow \infty} A(t), \quad t \geq 0$$

# Probability Review

---

- Slides 32-120 (SPN1)

Já vimos este assunto.

# Exponential Distribution

---

- Arises commonly in reliability & queuing theory.
- A non-negative continuous random variable.
- It exhibits memoryless property (continuous counterpart of geometric distribution).
- Related to (discrete) Poisson distribution

# Exponential Distribution

---

- Often used to *model*
  - Interarrival times between two IP packets (or voice calls)
  - Service times at a file (web, compute, database) server
  - Time to failure, time to repair, time to reboot etc.
- The use of exponential distribution is an assumption that needs to be validated with experimental data; if the data does not support the assumption, then other distributions may be used

# Exponential Distribution

---

- For instance, Weibull distribution is often used to model times to failure;
- Lognormal distribution is often used to model repair time distributions
- Markov modulated Poisson process is often used to model arrival of IP packets (which has non-exponentially distributed inter-arrival times)

Remember these formulae

# Exponential Distribution: EXP( $\lambda$ )

- Mathematically (CDF and pdf are given as):

$$\text{CDF: } F(x) = \begin{cases} 1 - e^{-\lambda x}, & \text{if } 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases}$$

where  $\lambda$  is a parameter and the base of natural logarithm,  $e = 2.7182818284$

$$\text{pdf: } f(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

- Also

$$P(X > t) = \int_t^{\infty} f(x) dx = e^{-\lambda t}$$

$$P(a < X \leq b) = \int_a^b f(x) dx = F(b) - F(a) \\ = e^{-\lambda a} - e^{-\lambda b}$$

# Exponential Distribution: EXP( $\lambda$ )

$$R(t) = e^{-\lambda t}, \quad t \geq 0,$$

$$F(t) = 1 - e^{-\lambda t}, \quad t \geq 0,$$

$$h(t) = \lambda,$$

$$E[T] = MTTF = \frac{1}{\lambda},$$

$$Var[T] = \sigma^2 = \frac{1}{\lambda^2}.$$



# Exponential Distribution: EXP( $\lambda$ )

The memoryless property can be demonstrated with conditional reliability:

$$\begin{aligned} R(x | t) &= \Pr(T > x + t | T > t) = \frac{\Pr(T > x + t)}{\Pr(T > t)} \\ &= \frac{e^{-\lambda(t+x)}}{e^{-\lambda t}} = e^{-\lambda x} = R(x), \quad x \geq 0. \end{aligned}$$

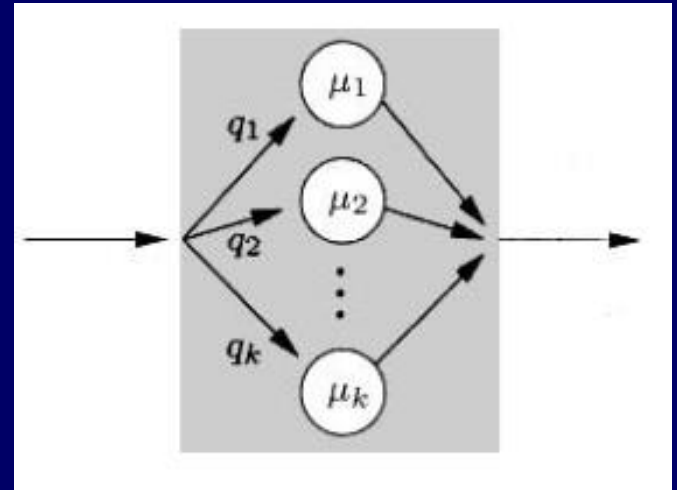
# Hyperexponential Distribution

$$F_X(x) = \sum_{j=1}^k q_j (1 - e^{-\mu_j x}), \quad x \geq 0.$$

$$\text{pdf: } f_X(x) = \sum_{j=1}^k q_j \mu_j e^{-\mu_j x}, \quad x > 0,$$

$$\text{mean: } \bar{X} = \sum_{j=1}^k \frac{q_j}{\mu_j} = \frac{1}{\mu}, \quad x > 0,$$

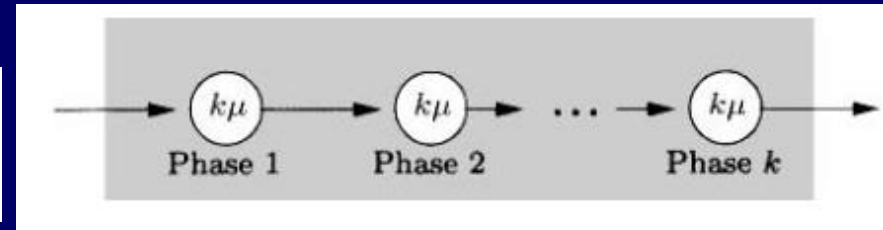
$$\text{variance: } \text{var}(X) = 2 \sum_{j=1}^k \frac{q_j}{\mu_j^2} - \frac{1}{\mu^2},$$



$$c_X = \sqrt{2\mu^2 \sum_{j=1}^k \frac{q_j}{\mu_j^2} - 1} \geq 1$$

# Erlang Distribution

$$F_X(x) = 1 - e^{-k\mu x} \cdot \sum_{j=0}^{k-1} \frac{(k\mu x)^j}{j!}, \quad x \geq 0, \quad k = 1, 2, \dots$$



pdf:  $f_X(x) = \frac{k\mu(k\mu x)^{k-1}}{(k-1)!} e^{-k\mu x}, \quad x > 0, \quad k = 1, 2, \dots,$

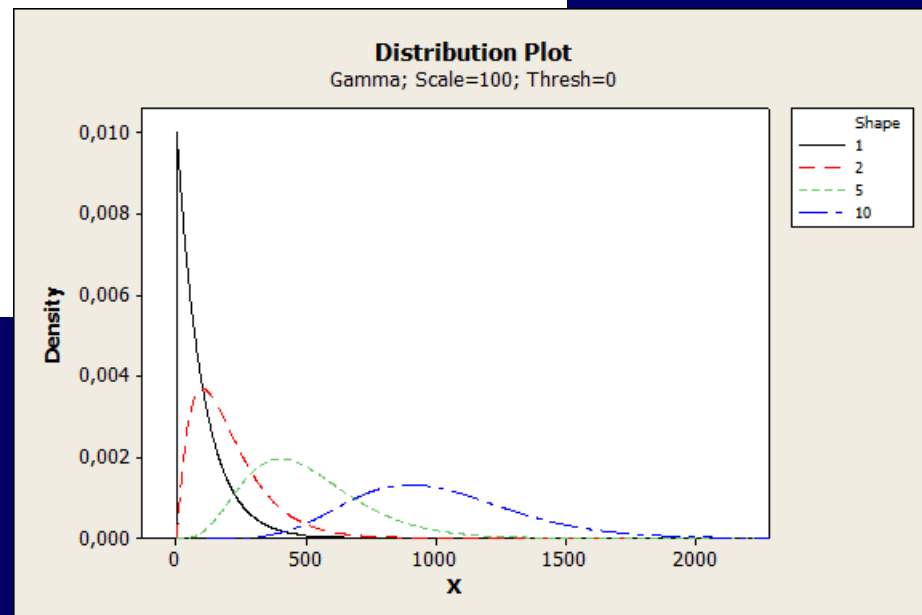
mean:  $\bar{X} = \frac{1}{\mu},$

variance:  $\text{var}(X) = \frac{1}{k\mu^2},$

coefficient of variation:  $c_X = \frac{1}{\sqrt{k}} \leq 1.$

$$k = \left\lceil \frac{1}{c_X^2} \right\rceil$$

$$\mu = \frac{1}{c_X^2 k \bar{X}}.$$



# Hypoexponential Distribution

pdf:  $f_X(x) = \sum_{i=1}^k a_i \mu_i e^{-\mu_i x}, \quad x > 0,$

with  $a_i = \prod_{j=1, j \neq i}^k \frac{\mu_j}{\mu_j - \mu_i}, \quad 1 \leq i \leq k,$

mean:  $\bar{X} = \sum_{i=1}^k \frac{1}{\mu_i},$

coefficient of variation:  $c_X = \left( 1 + 2 \frac{\sum_{i=1}^k \left( \frac{\mu_i \sum_{j=i+1}^k \mu_j}{\sum_{i=1}^k \mu_i^2} \right)}{\sum_{i=1}^k \mu_i^2} \right)^{-\frac{1}{2}}.$

# Weibull Distribution

$$F_X(x) = 1 - \exp(-(\lambda x)^\alpha), \quad x \geq 0$$

$$f_X(x) = \alpha \lambda (\lambda x)^{\alpha-1} \exp(-(\lambda x)^\alpha), \quad \lambda > 0,$$

shape parameter  $\alpha$

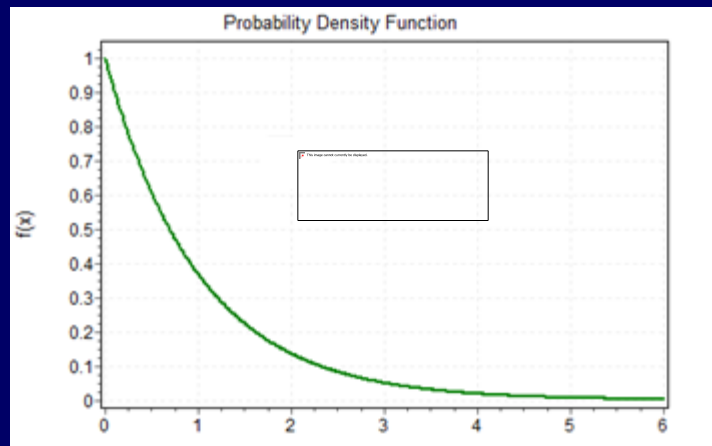
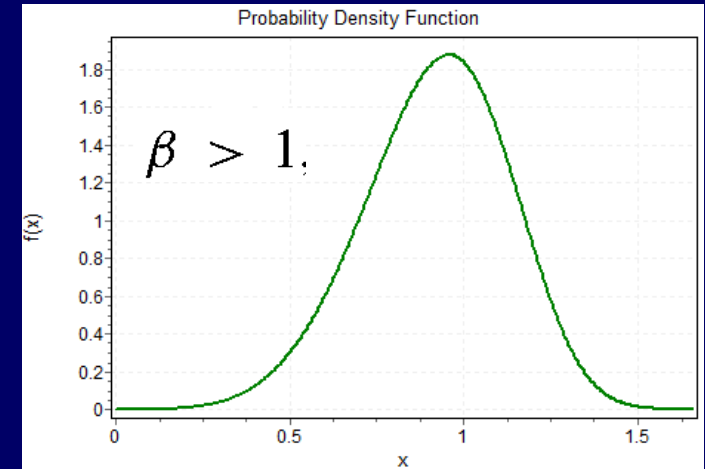
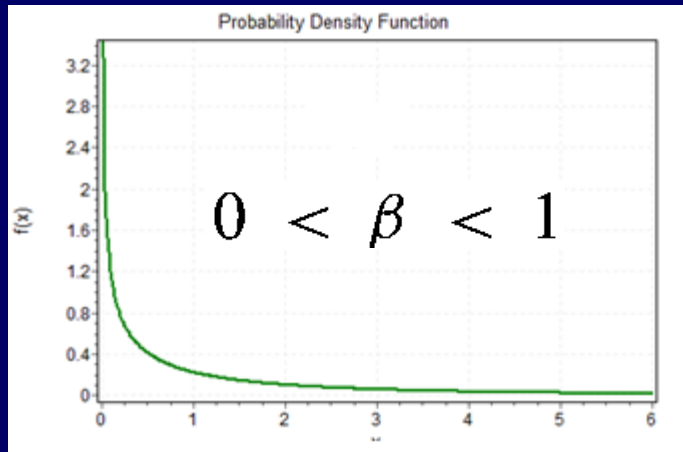
scale parameter  $\lambda > 0$ .

$\alpha < 0$  means infant mortality and  $\alpha > 0$  means wear out

$$\bar{X} = \frac{1}{\lambda} \Gamma\left(1 + \frac{1}{\alpha}\right),$$
$$c_X^2 = \frac{\Gamma(1 + 2/\alpha)}{\{\Gamma(1 + 1/\alpha)\}^2} - 1$$

Weibull distribution is often used to model times to failure

# Weibull Distribution



# Lognormal Distribution

$$F_X(x) = \Phi\left(\frac{\ln(x) - \lambda}{\alpha}\right), \quad x > 0$$

$$f_X(x) = \frac{1}{\alpha x \sqrt{2\pi}} \exp(-\{\ln(x) - \lambda\}^2 / 2\alpha^2), \quad x > 0$$

$$\bar{X} = \exp(\lambda + \alpha^2/2)$$

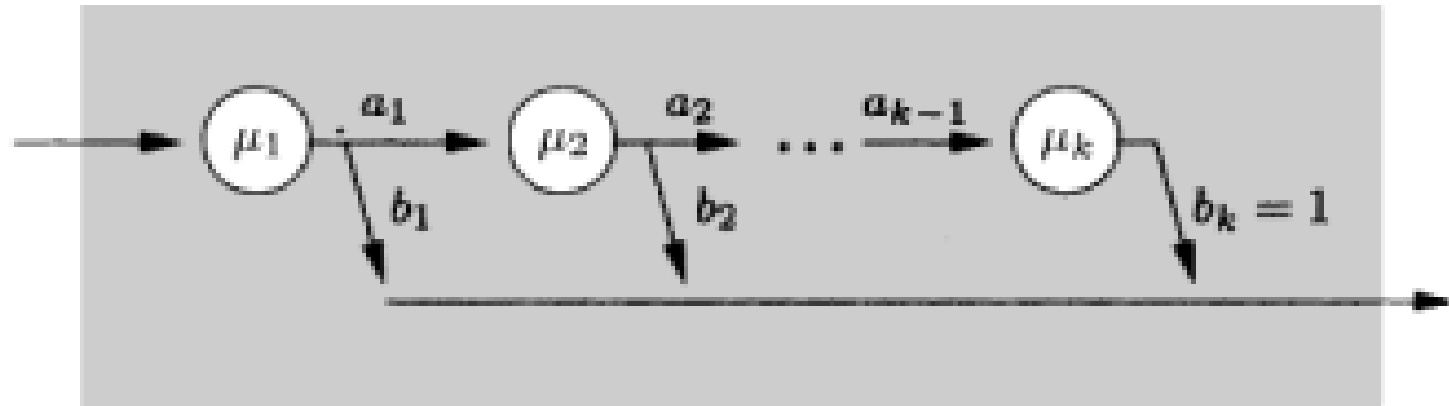
$$c_X^2 = \exp(\alpha^2) - 1$$

$$\alpha = \sqrt{\ln(c_X^2 + 1)}, \quad \lambda = \ln \bar{X} - \frac{\alpha^2}{2}$$

Lognormal distribution is often used to model repair time distributions

The importance of this distribution arises from the fact that the product of  $n$  mutually independent random variables has a lognormal distribution in the limit  $n \rightarrow \infty$ .

# Cox Distribution



The model consists of  $k$  phases in series with exponentially distributed times and rates  $\mu_1, \mu_2, \dots, \mu_k$ . After phase  $j$ , another phase  $j + 1$  follows with probability  $a_j$  and with probability  $b_j = 1 - a_j$  the total time span is completed.

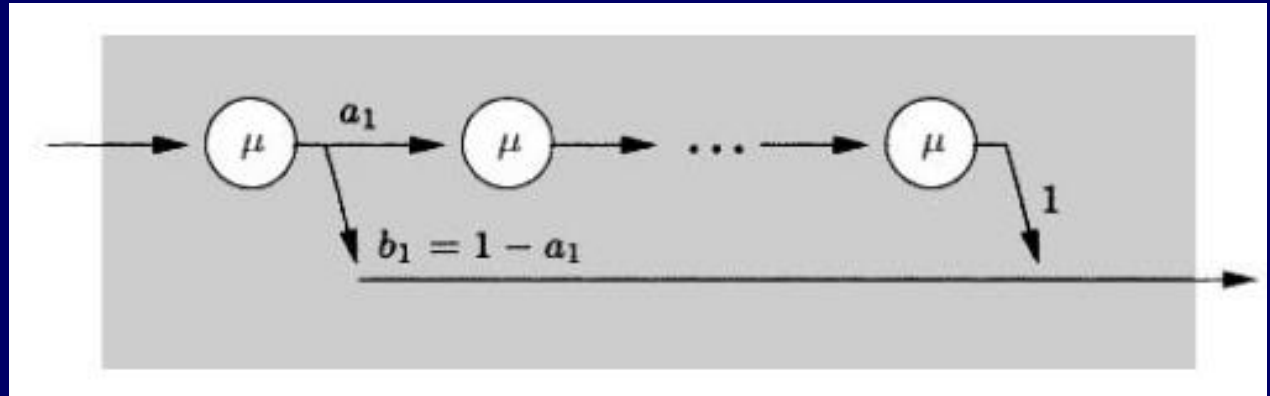


# Cox Distribution

Case 1:  $c_X \leq 1$

$$\mu_j = \mu \quad j = 1, \dots, k,$$

$$a_j = 1 \quad j = 2, \dots, k-1$$



$$\bar{X} = \frac{b_1 + k(1 - b_1)}{\mu},$$

$$\text{var}(X) = \frac{k + b_1(k - 1)(b_1(1 - k) + k - 2)}{\mu^2},$$

$$c_X^2 = \frac{k + b_1(k - 1)(b_1(1 - k) + k - 2)}{[b_1 + k(1 - b_1)]^2}.$$

$$k = \left\lceil \frac{1}{c_X^2} \right\rceil$$

$$b_1 = \frac{2kc_X^2 + (k - 2) - \sqrt{k^2 + 4 - 4kc_X^2}}{2(c_X^2 + 1)(k - 1)},$$

$$\mu = \frac{k - b_1 \cdot (k - 1)}{\bar{X}}.$$

# Cox Distribution

Case 2:  $c_X > 1$

$$\bar{X} = \frac{1}{\mu_1} + \frac{a}{\mu_2},$$

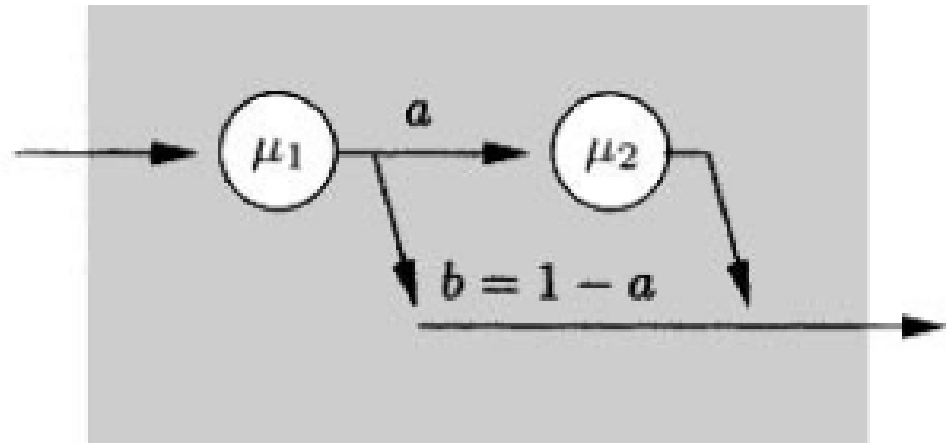
$$\text{var}(X) = \frac{\mu_2^2 + a\mu_1^2(2-a)}{\mu_1^2 \cdot \mu_2^2},$$

$$c_X^2 = \frac{\mu_2^2 + a\mu_1^2(2-a)}{(\mu_2 + a\mu_1)^2}.$$

$$\mu_1 = \frac{2}{\bar{X}}$$

$$a = \frac{1}{2c_X^2}$$

$$\mu_2 = \frac{1}{\bar{X}c_X^2}$$



# A Very Brief Introduction to Reliability Data Analysis

---

The aim is the selection and the specification of suitable reliability (and maintainability) models based on failure (and repair) data.

- Non-parametric approaches
- Parametric approaches

# A Very Brief Introduction to Reliability Data Analysis

The observation of failures (or repairs) times can be represented by:

Failure	$1^{st}$	$2^{nd}$	...	$n$
Time	$t_1$	$t_2$		$t_n$

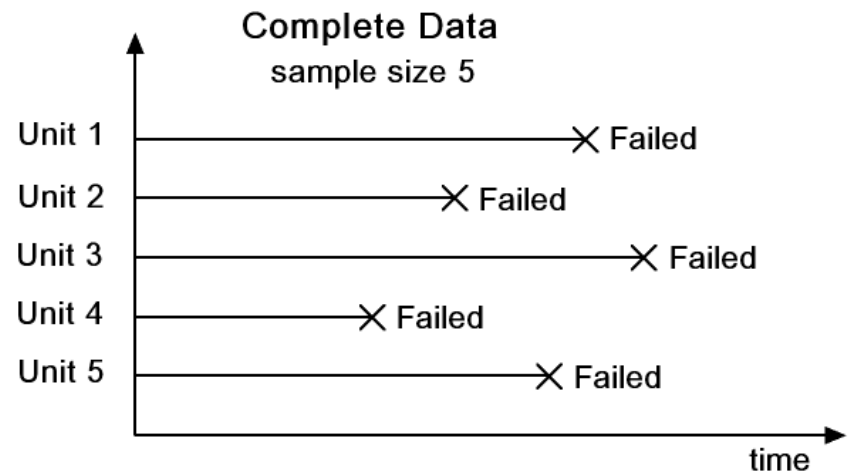
The functions  $f_T(t)$ ,  $F_T(t)$ ,  $R(t)$  ( $M(t)$ ) and  $h(t)$  and  $H(t)$  represent the failure time (repair time) of the population.

# A Very Brief Introduction to Reliability Data Analysis

## A taxonomy of data

Failure data may be classified as:

- Operational × Test-generated failures
- Grouped × Ungrouped data
- Large samples × Small samples
- Complete × Censored data



# A Very Brief Introduction to Reliability Data Analysis

## A taxonomy of data

Failure data may be classified as:

- Operational × Test-generated failures
- Grouped × Ungrouped data
- Large samples × Small samples
- Complete × Censored data

Failure times are usually either field data or failures observed from reliability testing.

Often failure field data are grouped into time intervals in which the exact failure times are not preserved.

For large sample sizes, grouping data into time intervals may be preferred.

Testing may result in small sample sizes.

Failure data obtained from testing are likely to be more precise and appropriate.

However, field data usually provide larger data samples and reflect the operating environment conditions.

# A Very Brief Introduction to Reliability Data Analysis

## A taxonomy of data

Failure data may be classified as:

- Operational × Test-generated failures
- Grouped × Ungrouped data
- Large samples × Small samples
- Complete × Censored data

Censoring occurs when data are incomplete when units are removed from the analysis. The censoring occurs because:

- units may have been removed before their failures  
or
- because the test finishes before the respective failures occur.

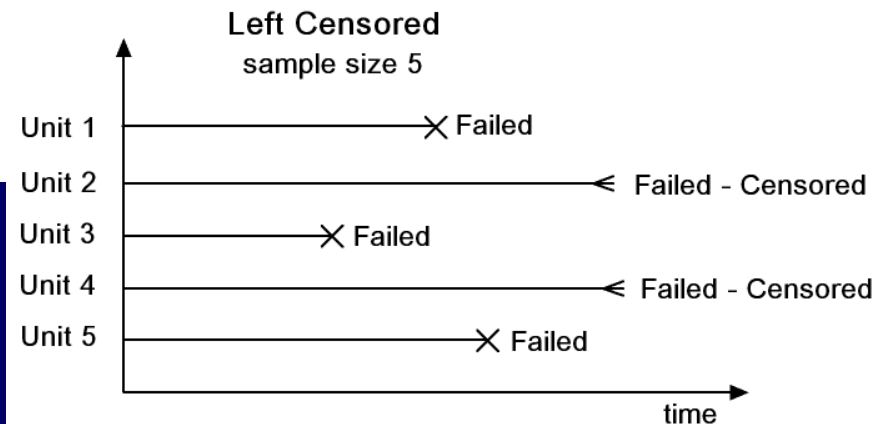
# A Very Brief Introduction to Reliability Data Analysis

- Singly censored data: all units have the same test time.
- Multiply censored data: test time or operating time differ from censored units.
- Left censored: failure time occurs before a specified time.
- Right censored: failure time occurs after a specified time.
  - Type I - right censored: the testing stops at  $T$  time units.
  - Type II - right censored: the testing stops when  $r$  out of  $n$  failures occur.

## A taxonomy of data

Failure data may be classified as:

- Operational × Test-generated failures
- Grouped × Ungrouped data
- Large samples × Small samples
- Complete × Censored data





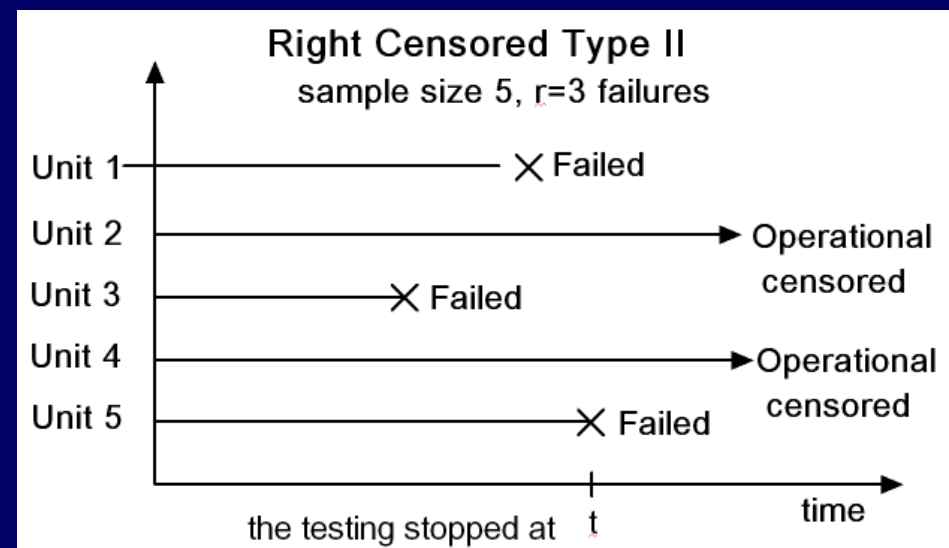
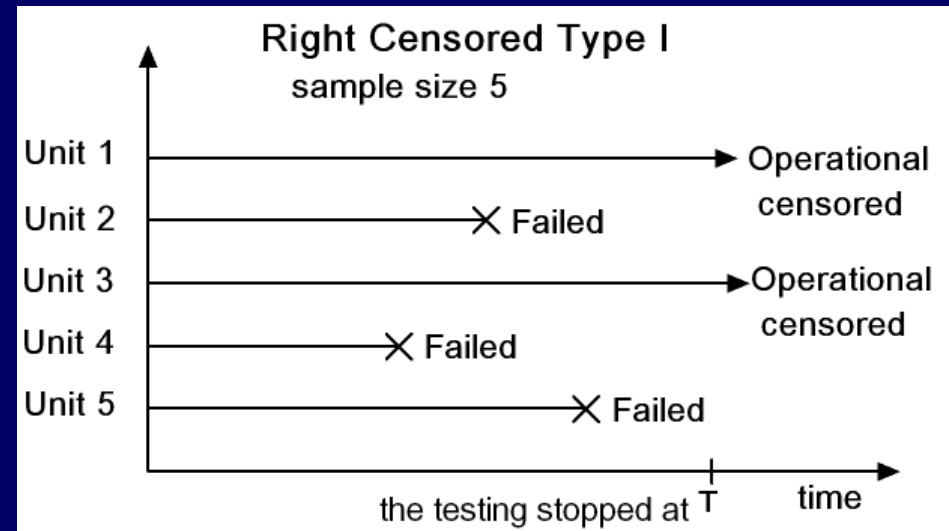
# A Very Brief Introduction to Reliability Data Analysis

## A taxonomy of data

Failure data may be classified as:

- Operational × Test-generated failures
- Grouped × Ungrouped data
- Large samples × Small samples
- Complete × Censored data

- Singly censored data: all units have the same test time.
- Multiply censored data: test time or operating time differ from censored units.
- Left censored: failure time occurs before a specified time.
- Right censored: failure time occurs after a specified time.
  - Type I - right censored: the testing stops at  $T$  time units.
  - Type II - right censored: the testing stops when  $r$  out of  $n$  failures occur.



# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

CI MTF

### Ungrouped Complete Data

Consider  $t_1, t_2, \dots, t_n$ , where  $t_i \leq t_{i+1}$  are  $n$  ordered failure times.

$$\hat{R}(t_i) = \frac{n - i}{n} = 1 - \frac{i}{n}$$

$$\hat{F}(t_i) = 1 - \hat{R}(t_i) = \frac{i}{n}$$

$$\hat{F}(t_i) = 1 - \hat{R}(t_i) = \frac{i}{n + 1}$$

$$\hat{F}(t_i) = 1 - \hat{R}(t_i) = \frac{i - 0.3}{n + 0.4}$$

$$\hat{f}(t_i) =$$

$$\hat{\lambda}(t_i) = \frac{\hat{f}(t_i)}{\hat{R}(t_i)}$$

$\bar{M}$

Confidence interval  
bootstrap.

Ungrouped Complete Data	
n =	10
i	Failure times
0	0
1	15.4
2	18.9
3	20.1
4	24.5
5	29.3
6	33.9
7	48.2
8	54.7
9	72
10	86.1

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

CI MTF

### Grouped Complete Data

Failures that have been placed into time intervals, their original values are lost.

Consider  $k$  time intervals where  $t_1, t_2, \dots, t_k$  are the time instants representing the ends of each time interval, such that  $t_i \leq t_{i+1}$ .

Let  $n_1, n_2, \dots, n_k$  be the number of units that survived at respective ordered time  $t_1, t_2, \dots, t_k$ , and  $n$  the number of units at risk at the beginning of the test.

$$\hat{R}(t_i) = \frac{n_i}{n}, \quad i = 1, 2, \dots, k$$

$$\hat{F}(t_i) = 1 - \hat{R}(t_i)$$

$$\begin{aligned} \hat{f}(t_i) &= -\frac{\hat{R}(t_{i+1}) - \hat{R}(t_i)}{t_{i+1} - t_i} \\ &= \frac{n_i - n_{i+1}}{(t_{i+1} - t_i) \times n} \end{aligned}$$

$$\begin{aligned} \hat{\lambda}(t_i) &= \frac{\hat{f}(t_i)}{\hat{R}(t_i)} = \frac{n_i - n_{i+1}}{(t_{i+1} - t_i) \times n_i} \\ & \quad t_i < t < t_{i+1} \end{aligned}$$

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

CI MTF

The *MTTF* is estimated considering the midpoint of each interval and fraction of units that have failed in each interval.

$$\bar{t}_i = \frac{t_i + t_{i+1}}{2}$$

$$\widehat{MTTF} = \sum_{i=0}^{k-1} \bar{t}_i \frac{n_i - n_{i+1}}{n}$$

$$t_0 = 0, n_0 = n$$

Confidence interval for the *MTTF*: adopt bootstrap.

Time Interval	Number failing	Number surviving
0	0	70
5	3	67
10	7	60
15	8	52
20	9	43
25	13	30
30	18	12
35	12	0

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

### Ungrouped Censored Data

For singly censored on the right,  $R(t)$ ,  $f(t)$ , and  $\lambda(t)$  may be iteratively estimated from the equation adopted for **Ungrouped Complete Data**.

We know that:

$$\hat{F}(t_i) = 1 - \hat{R}(t_i) = \frac{i}{n+1}$$

So:

$$\hat{R}(t_i) = 1 - \frac{i}{n+1} = \frac{n+1-i}{n+1}$$

Therefore:

$$\hat{R}(t_{i-1}) = \frac{n+1-(i-1)}{n+1} = \frac{n+2-i}{n+1}$$

Now, consider:

$$\hat{R}(t_i) = \hat{R}(t_{i-1}) \times$$

*P(Unit will not fail between  $t_i$*

*and  $t_{i-1}$ , given it has*

*survived  $t_{i-1}$ )*

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Two events may occur at  $t_i$  (since  $t_i$  is there, otherwise it is not there): a failure or a censoring. So:

$$\delta_i = \begin{cases} 1 & \text{if failure occurs at } t_i \\ 0 & \text{if censoring occurs at } t_i \end{cases}$$

Then:

$$\hat{R}(t_i) = \left( \frac{n+1-i}{n+2-i} \right)^{\delta_i} \times \hat{R}(t_{i-1})$$

$$\hat{F}_T(t_i) = 1 - \hat{R}(t_i)$$

$$\hat{F}(t_i) = \left( \frac{i}{i-1} \right)^{\delta} \times \hat{F}(t_{i-1})$$

$$\hat{f}(t_i) = \frac{1}{(n+1) \times (t_i - t_{i-1})}$$

$$\hat{\lambda}(t_i) = \frac{1}{(n+1-i) \times (t_i - t_{i-1})}$$

Hence:

	n =	10		
$\hat{R}$	i	Failure times	$\delta$	R(t)
	0	0		1
$\hat{R}$	1	150	1	0.909091
	2	340	0	0.909091
Sc	3	560	1	0.808081
P	4	800	1	0.707071
	5	1130	0	0.707071
	6	1720	1	0.589226
	7	2470	0	0.589226
	8	4210	0	0.589226
	9	5230	1	0.392817
	10	6890	1	0.196409

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

### Kaplan-Meier Method

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

So

$$P(A \cap B) = P(A|B) \times P(B)$$

If  $A$  and  $B$  are independent, then:

$$P(A|B) = P(A), \text{ so:}$$

$$P(A \cap B) = P(A) \times P(B)$$

Therefore

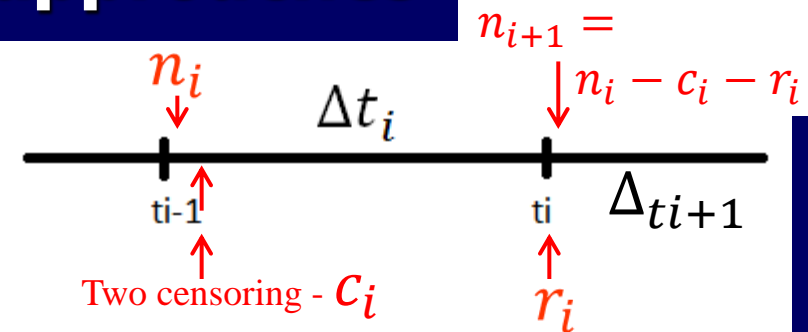
$$\hat{R}(t_i) = \hat{R}(t_{i-1}) \times \hat{R}(\Delta t_i)$$

$$\text{where } t_i = t_{i-1} + \Delta t_i$$

If at  $t_i$  we have  $r_i$  failures, then:

$$\hat{R}(\Delta t_i) = 1 - \frac{r_i}{n_i}$$

where  $n_i$  is the number of available units at the instant  $t_i - \Delta t_i$  without considering the censoring, that is, shortly after  $t_{i-1}$ . You should bear in mind that the interval  $(t_i - \Delta t_i, t_i]$  is open at the left hand side.



The probability of a subject surviving to any point in time  $\mathbf{T} = (\mathbf{t} + \Delta \mathbf{t})$  is the product of the cumulative survival probability up to time  $\mathbf{t}$  and the probability of surviving interval  $\Delta \mathbf{t}$ .

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Mathematica

### Kaplan-Meier Method

For the sake of calculating  $n_i$ , it is assumed that censoring occurs shortly after the failures at  $t_{i-1}$ .

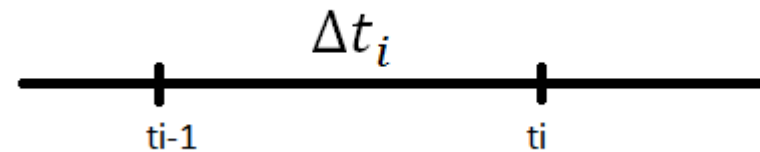
Therefore:  $\hat{R}(t_2) = \hat{R}(t_1) \times \hat{R}(\Delta t_2)$

$$\hat{R}(t_2) = \left(1 - \frac{r_1}{n_1}\right) \times \left(1 - \frac{r_2}{n_2}\right)$$

Generalizing:

$$R(t) = \prod_{t \leq t_i} \left(1 - \frac{r_i}{n_i}\right)$$

$$i \in (1, m), i \in \mathbb{Z}$$





# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Mathematica

### Kaplan-Meier method summary

#### Kaplan-Meier method for ungrouped complete data

As the data set is complete (no censoring), then  $c_i = 0, \forall \Delta t_i$ .

And since the data set is ungrouped (exact time to failure)  $r_i = 1, \forall \Delta t_i$ .

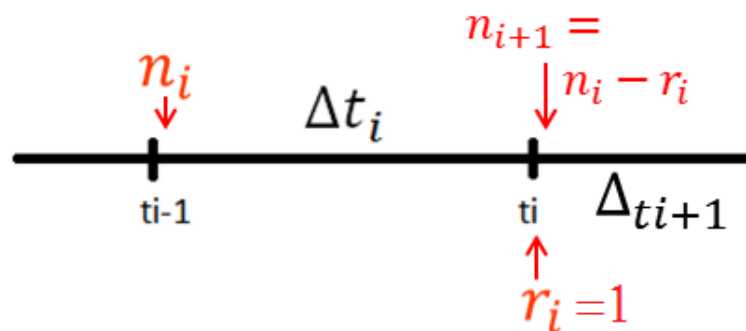
Hence:

$$n_{i+1} = n_i - 1$$

Therefore:

$$\hat{R}(t) = \prod_{t \leq t_i} \left(1 - \frac{1}{n_i}\right)$$

$$i \in (1, m), i \in \mathbb{Z}$$



# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

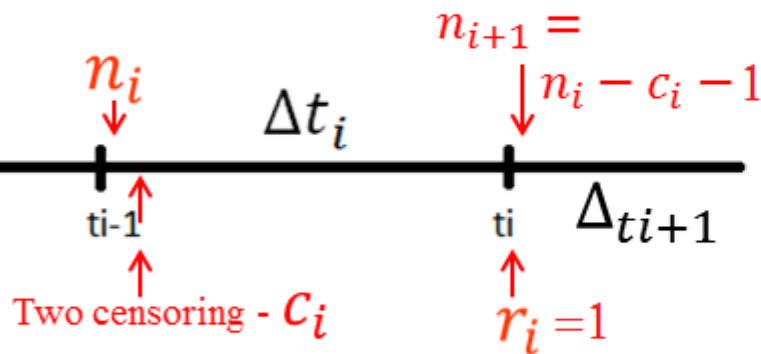
Excel

Mathematica

### Kaplan-Meier method summary

#### Kaplan-Meier method

for ungrouped censored data



Hence:

$$n_{i+1} = n_i - c_i - 1$$

Therefore:

$$R(t) = \prod_{t \leq t_i} \left(1 - \frac{1}{n_i}\right)$$

$$i \in (1, m), i \in \mathbb{Z}$$

Since the data set is ungrouped (exact time to failure)  $r_i = 1, \forall \Delta t_i$ .

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

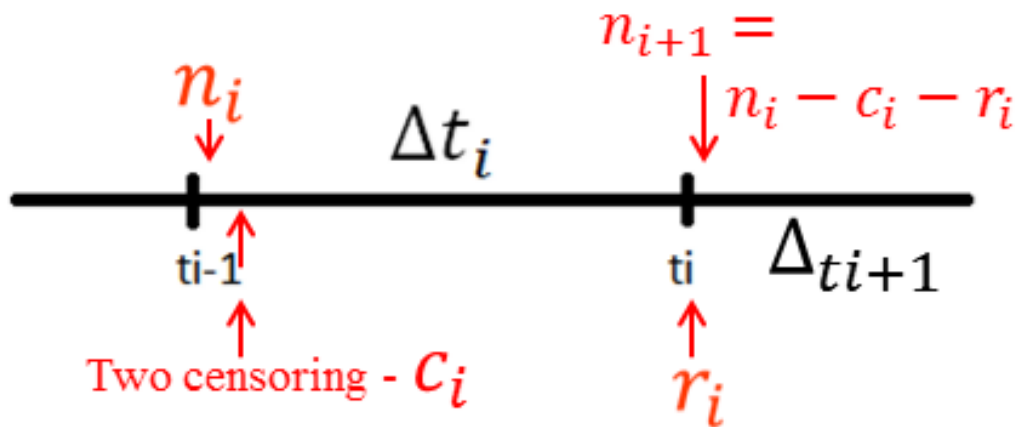
Excel

Mathematica

### Kaplan-Meier method summary

### Kaplan-Meier method

for grouped censored data



$$\hat{R}(t) = \prod_{t \leq t_i} \left( 1 - \frac{r_i}{n_i} \right)$$

$$i \in (1, m), i \in \mathbb{Z}$$

$$n_{i+1} = n_i - c_i - r_i$$

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Mathematica

### Kaplan-Meier Method

A sample of 21 highly stressed automotive components is put on a life test with the following results shown in Table .

# of units	3	1+	1	1+	1	1+	1+	1	1	1+	1+	1	1	1	1+	2+	1	1+
Time to Failure or Suspension (Months)	9	9	11	12	13	13	15	17	21	22	24	26	28	30	32	35	39	41

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Mathematica

### Kaplan-Meier Method

We can use the Kaplan-Meier estimator to determine reliability estimates for each failure time. The following table can be constructed .

Time to Failure or Suspension (Months)	# of Failures ( $r_i$ )	# of Suspensions	Available units ( $n_i$ )	$1 - \frac{r_i}{n_i}$	$\prod \left(1 - \frac{r_i}{n_i}\right)$
9	3	1	21	0.857	0.857
11	1		17	0.941	0.807
12		1	16	1	0.807
13	1	1	15	0.933	0.753
15		1	13	1	0.753
17	1		12	0.917	0.690
21	1		11	0.909	0.627
22		1	10	1	0.627
24		1	9	1	0.627
26	1		8	0.875	0.549
28	1		7	0.857	0.471
30	1		6	0.833	0.392
32		1	5	1	0.392
35		2	4	1	0.392
39	1		2	0.5	0.196
41		1	1	1	0.196
Total	11	10			

Table Analysis of example 2 data. (m=16)

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

Excel

Mathematica

### Kaplan-Meier Method

The reliability estimates for the failure times are shown in Table .

Time to Failure	Reliability Estimate
9	0.857
11	0.807
13	0.753
17	0.690
21	0.627
26	0.549
28	0.471
30	0.392
39	0.196

Table. Reliability estimates.

# A Very Brief Introduction to Reliability Data Analysis

## Non-parametric approaches

---

- Other methods:
  - Actuarial method
  - Rank method
  - ...

# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

### General process:

- Identifying a theoretical distribution
  - Build graphs and compute statistics, analyze the empirical failure rate, and consider the properties of theoretical distributions
- Estimating the distribution parameters
  - Point estimation
    - Graphical methods
    - Least square method
    - Method of moments
    - Maximum Likelihood Estimation method
  - Confidence interval
- Performing the goodness-of-fit test
  - KS, AD,  $\chi^2$ ...



# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

Mathematica

### Point estimation

#### Graphical method

A distribution is transformed into a standard distribution by means of linear transformation. On the graph paper with y axis so calibrated,  $x$  and  $y$  are linearly related with positive slope, where  $y$  represents a cdf  $F(x)$  with some scale and location parameters.

#### Method of Least Squares

The method of least squares fits a curve (or straight line) to a series of data points, by minimizing the sum of squared deviations of the fitted curve and the actual data points.

#### Method of Matching Moments

The theoretical moments of the distribution are equated with the sample moments.

#### Method of Maximum Likelihood

The core of this method is selecting as estimate of the distribution parameter a value for which the observed sample is most “likely” to occur.

# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

### Method of Maximum Likelihood

Assume that  $X$  denotes the time to failure of device. The time to failure is exponentially distributed with failure rate  $\lambda$ .

$$f(x) = \lambda e^{-\lambda x} \quad \lambda > 0, x > 0$$

We intend to estimate  $\lambda$  from random sample  $X_1, X_2, \dots, X_n$ , where  $x = (x_1, x_2, \dots, x_n)$  is the vector representing the observed values of the sample.

$$\hat{\lambda} = \theta(x)$$

The joint pdf of  $X_1, X_2, \dots, X_n$  is given by

$$L(x, \lambda) = \lambda^n e^{-\lambda \sum_{j=1}^n x_j}$$

$L(x, \lambda)$  is called the likelihood function, which is the function of the unknown parameter  $\lambda$  and the real data  $x$ .

The parameter value that maximizes the likelihood function is called the maximum likelihood estimator. The MLE can be interpreted as the parameter value that is most likely to explain the dataset.

The parameter value that maximizes the log-likelihood function will maximize the likelihood function.

$$\ln(L(x, \lambda)) = \ln \left( \lambda^n e^{-\lambda \sum_{j=1}^n x_j} \right)$$

$$\ln(L(x, \lambda)) = \ln \lambda^n + \ln e^{-\lambda \sum_{j=1}^n x_j}$$

$$\ln(L(x, \lambda)) = n \ln \lambda - \lambda \sum_{j=1}^n x_j$$

# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

Excel

### Method of Maximum Likelihood

The function  $\ln(L(x, \lambda))$  can be maximized by deriving it with respect to  $\lambda$ , setting the resulting expression to zero, and solving the equation for  $\lambda$ .

Therefore:

$$\frac{\partial \ln(L(x, \lambda))}{\partial \lambda} = \frac{n}{\lambda} - \sum_{j=1}^n x_j = 0$$

So:

$$\hat{\lambda} = \frac{n}{\sum_{j=1}^n x_j}$$

Now assume for a certain system that we observed 60 failures during  $T = 50,000$  hours. Hence:

$$\hat{\lambda} = \frac{60}{50000} = 0.0012 \text{ failures/hour}$$

## Reliability Data

### Exercises

Excel

Mathematica

Consider that we have observed 60 units of a specific type until the respective failures. The failure times were registered and are depicted in the spreadsheet.

Assuming the time to failure is exponentially distributed, compute the confidence interval for  $\lambda$ .

Consider a reliability test starts at 0 and that all (n) failures are reported as failures. The test finishes when all fail or after  $r$  failures occur (right censoring type II). The confidence for  $\hat{\lambda}$  and  $\widehat{MTTF}$  can be computed

$$(\lambda_l \quad \lambda_u) = \left( \frac{\chi_{2n, 1-\alpha/2}^2}{2S_{n:r}}, \quad \frac{\chi_{2n, \alpha/2}^2}{2S_n} \right)$$

$$\begin{aligned} & (MTTF_l \quad MTTF_u) \\ &= \left( \frac{2S_{n:r}}{\chi_{2n, 1-\alpha/2}^2}, \quad \frac{2S_{n:r}}{\chi_{2n, 1-\alpha/2}^2} \right) \end{aligned}$$

Now, also consider an accelerated test in which 60 units have been placed. The test was finished when 10 failures occurred.

The observed failure times are registered in the spreadsheet. Assuming the time to failure is exponentially distributed, compute the confidence interval for  $\lambda$ .

# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

Excel

Mathematica

### Confidence Interval Exponential Distribution

If right censoring type I is considered, the method still provides a useful approximation.

The same process can be applied to estimate the confidence interval for MTTR.

# A Very Brief Introduction to Reliability Data Analysis

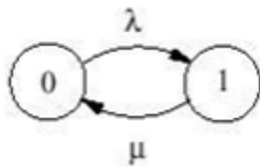
## Parametric approaches

Excel

Consider now that the units have been repaired. The respective time to repairs are also registered in the spreadsheet.

Compute confidence interval for the availability.

The CTMC representing the system:



$$A = \frac{1}{1 + \frac{\lambda}{\mu}} \quad \text{where } \rho = \frac{\lambda}{\mu}.$$

re: 
$$\hat{A} = \frac{1}{1 + \hat{\rho}} = \frac{\hat{\lambda}}{\hat{\mu}}$$

confidence interval for  $\rho$  is  $(\rho_l, \rho_u)$ ,

$$\rho_l = \frac{\hat{\rho}}{f_{2n, 2n; \alpha/2}} \quad \rho_u = \frac{\hat{\rho}}{f_{2n, 2n; 1-\alpha/2}}$$

the confidence interval for  $A$  is

$(A_l, A_u)$ , where:

$$A_l = \frac{1}{1 + \rho_u} \quad A_u = \frac{1}{1 + \rho_l}$$

# A Very Brief Introduction to Reliability Data Analysis

## Parametric approaches

---

### Confidence Interval

You may also adopt:

- Adopt Bootstrap or Semi-parametric Bootstrap
- If possible, you may also use t-student distribution or
- Central Limit Theorem

---

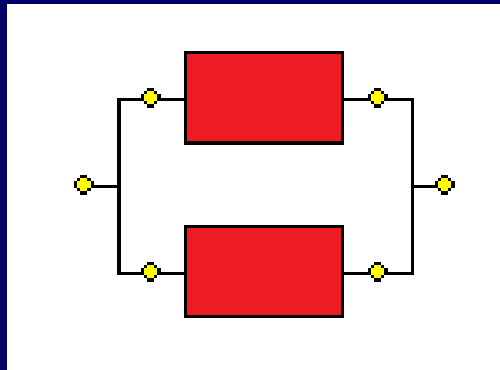
# **REDUNDANCY MECHANISMS**



# Redundancy Mechanisms

---

## ■ Parallel Redundancy



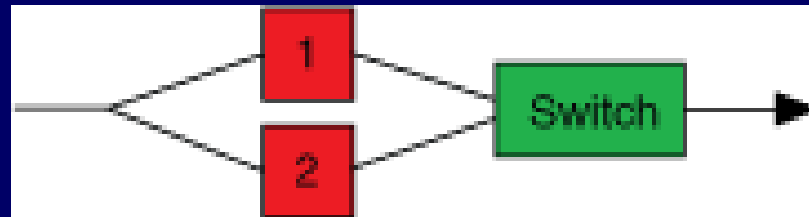
Parallel Redundancy refers to the approach of having multiple units running in parallel. All units are highly synchronized and receive the same input information at the same time.

But because all the units are powered up and actively engaged, the system is at risk of encountering failures in many units.

# Redundancy Mechanisms

---

## ■ Parallel Redundancy



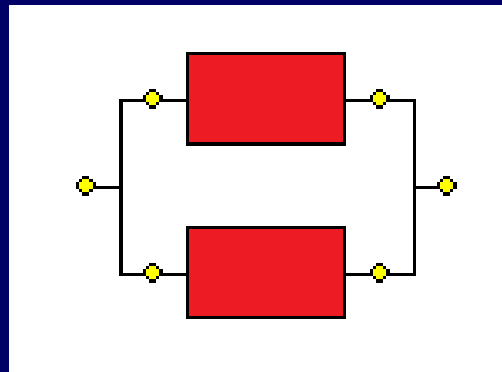
Deciding which unit is correct can be challenging if you only have two units. Sometimes you just have to choose which one you are going to trust the most and it can get complicated.

If you have more than two units the problem is simpler, usually the majority wins or the two that agree win.

# Redundancy Mechanisms

---

## ■ Parallel Redundancy (Active-Active) – load sharing

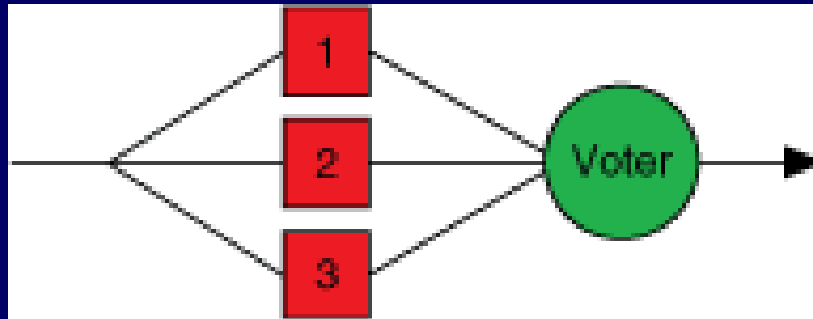


Active-Active refers to the approach of having multiply units sharing the load.

As the units are powered up and actively engaged, the system is at risk of encountering failures in many units.

# Redundancy Mechanisms

## ■ Triple Modular Redundancy (TMR)



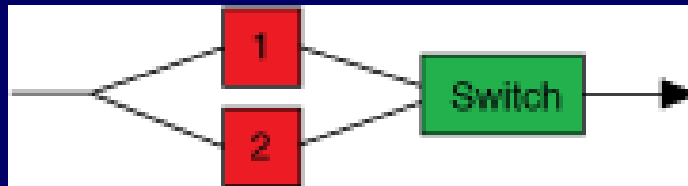
Deciding which unit is correct can be challenging if you only have two units. Sometimes you just have to choose which one you are going to trust the most and it can get complicated.

If you have more than two units the problem is simpler, usually the majority wins or the two that agree win.

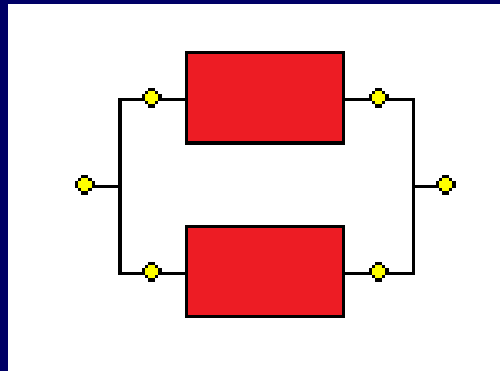
A generalization is named NMR

# Redundancy Mechanisms

- **Hot Standby** In hot standby, the secondary unit is powered up.



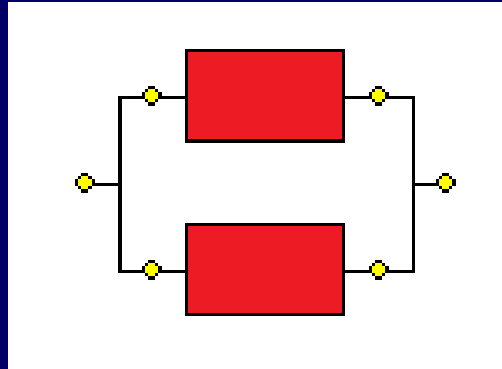
If you use the secondary unit as the watchdog and/or voter to decide when to switch over, you can eliminate the need for a third party to this job.



This design does not preserve the reliability of the standby unit. However, it shortens the downtime, which in turn increases the availability of the system.

# Redundancy Mechanisms

## ■ Hot Standby



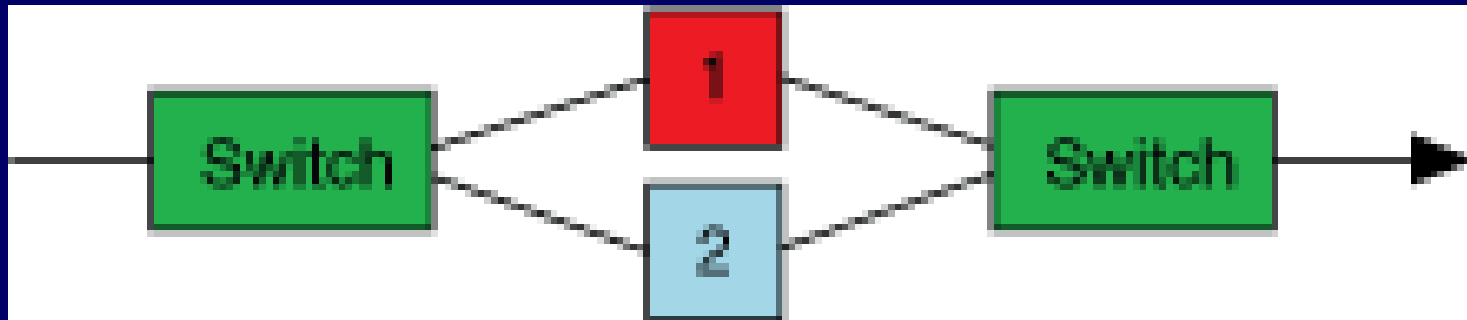
Some flavors of *Hot Standby* are similar to *Parallel Redundancy*. These naming conventions are commonly interchanged.

For us, Hot Standby and Parallel Redundancy (active-active) are the same mechanism!

But, attention!

# Redundancy Mechanisms

## ■ Cold Standby



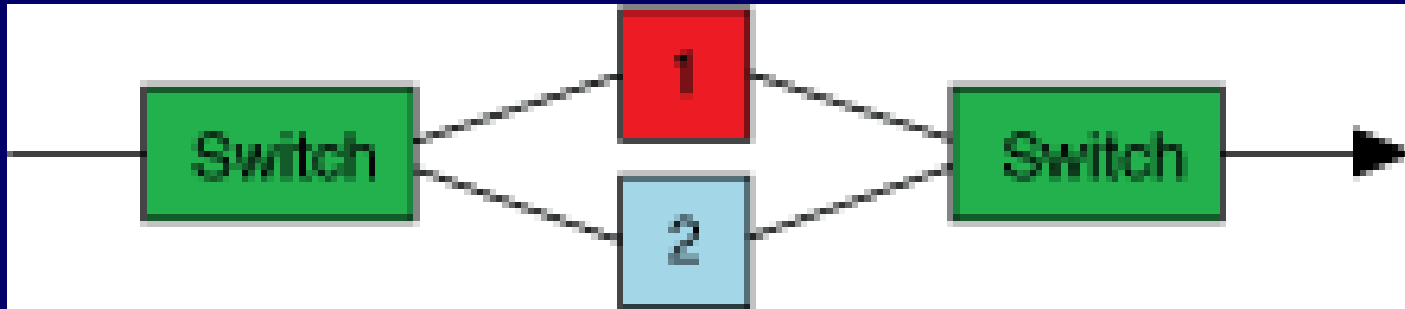
In cold standby, the secondary unit is powered off, thus preserving the reliability of the unit.

The drawback of this design is that standby unit have to power up, since it is initially powered off.

Perfect switching AND non-prefect switching

# Redundancy Mechanisms

## ■ Warm Standby



In warm standby, the secondary unit is powered up, but not receiving the workload.

It is common to assume that in such a state the standby component has higher reliability than when receiving the workload (properly working).

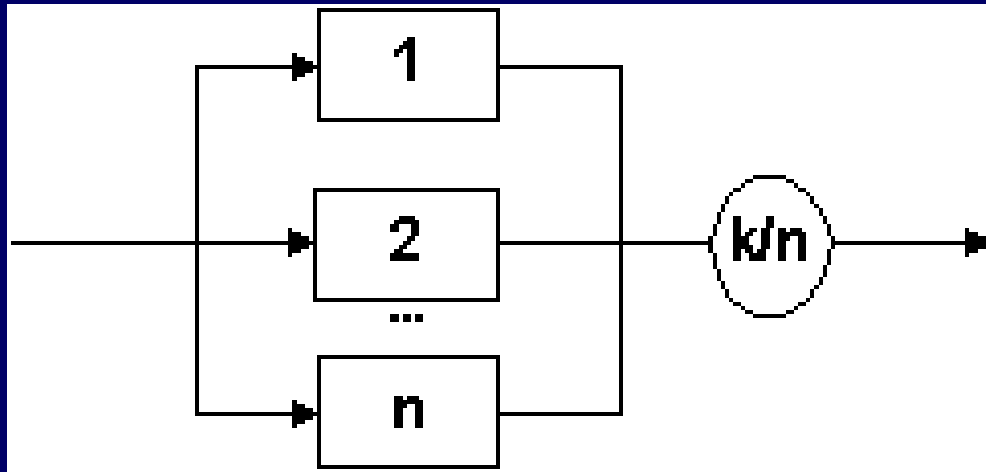
When the main component fails, the standby device promptly assumes the task.

Its switching time is shorter than the cold standby's switching time .



# Redundancy Mechanisms

## ■ K out of N



Consider a system composed of  $n$  identical and independent components that is operational if at least  $k$  out of its  $n$  components are working properly.

This sort of redundancy is named *k out of n*

# Redundancy Mechanisms

---

- RAID (redundant array of independent disks)

Many types of RAID have been developed and more will probably come out in the future.

The technology is driven by the variety of methods available for connecting multiple disks as well as various coding techniques, alternative read-and-write strategies, and the flexibility in organization to “tune” the architecture of the system.

# Redundancy Mechanisms

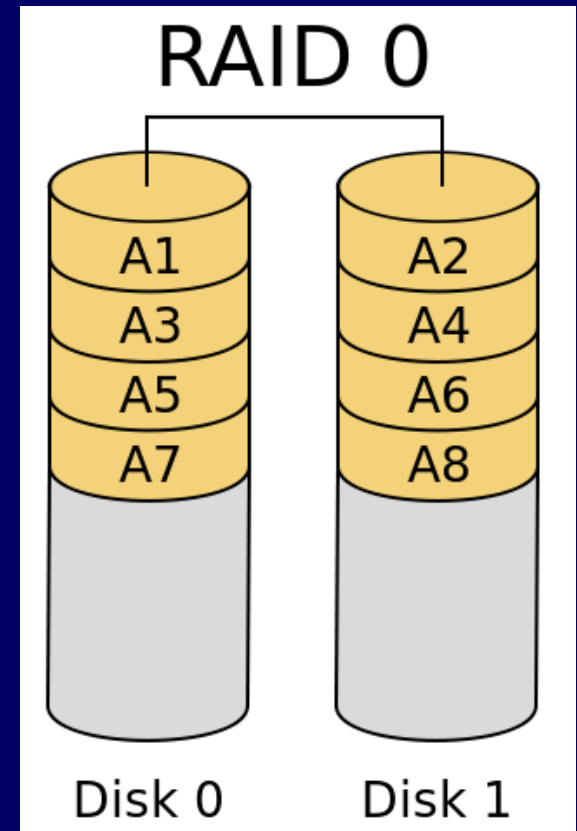
## ■ RAID 0

involves striping, which is the distribution of data across multiple disk drives in equally sized chunks.

For example, a 150 KB file can be striped, or chunked, across ten 15 KB chunks.

The RAID set of striped disks appears as a single, logical disk to the operating system.

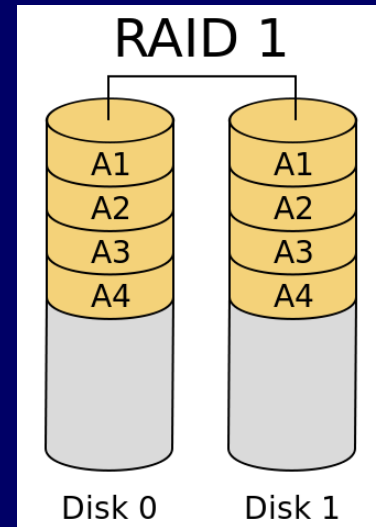
RAID-0 does not provide any data redundancy.



# Redundancy Mechanisms

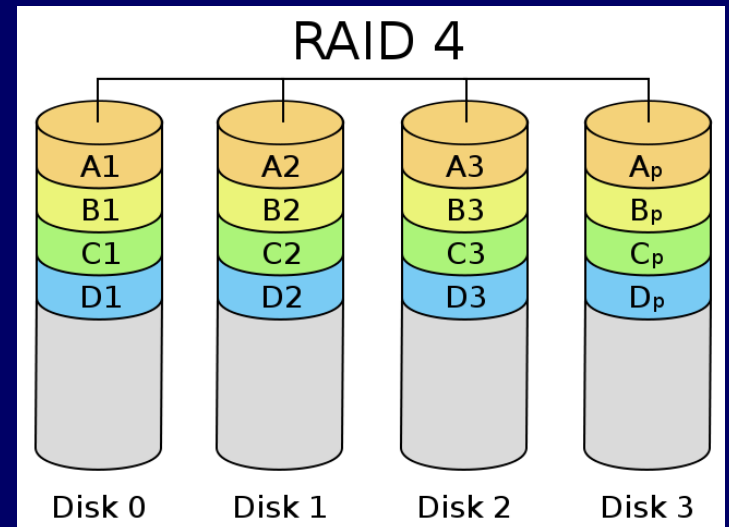
## ■ RAID 1

uses mirroring, or shadowing: all data written on a given disk is duplicated on another disk.



## ■ RAID 4

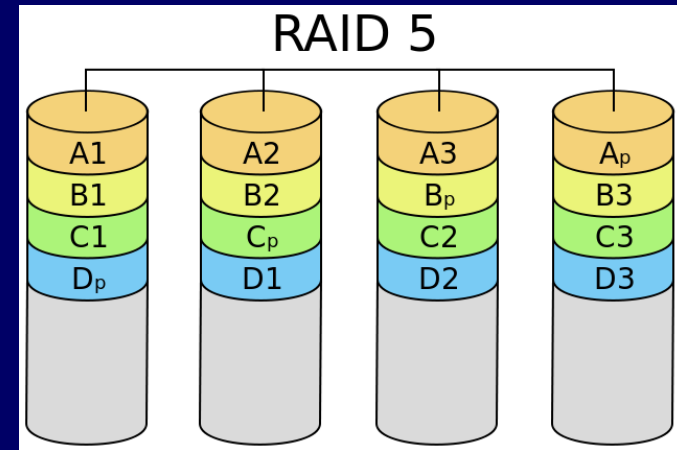
uses block-level striping with a dedicated parity disk.



# Redundancy Mechanisms

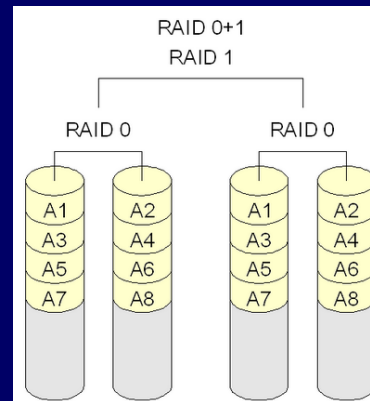
## ■ RAID 5

is similar to RAID 4 except that the parity data is striped across all HDDs instead of written on a dedicated HDD.



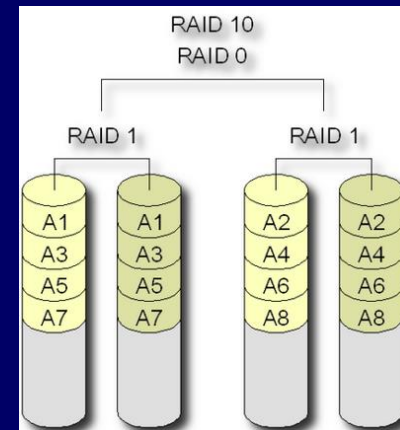
## ■ RAID 0+1

striped sets in a mirrored set.



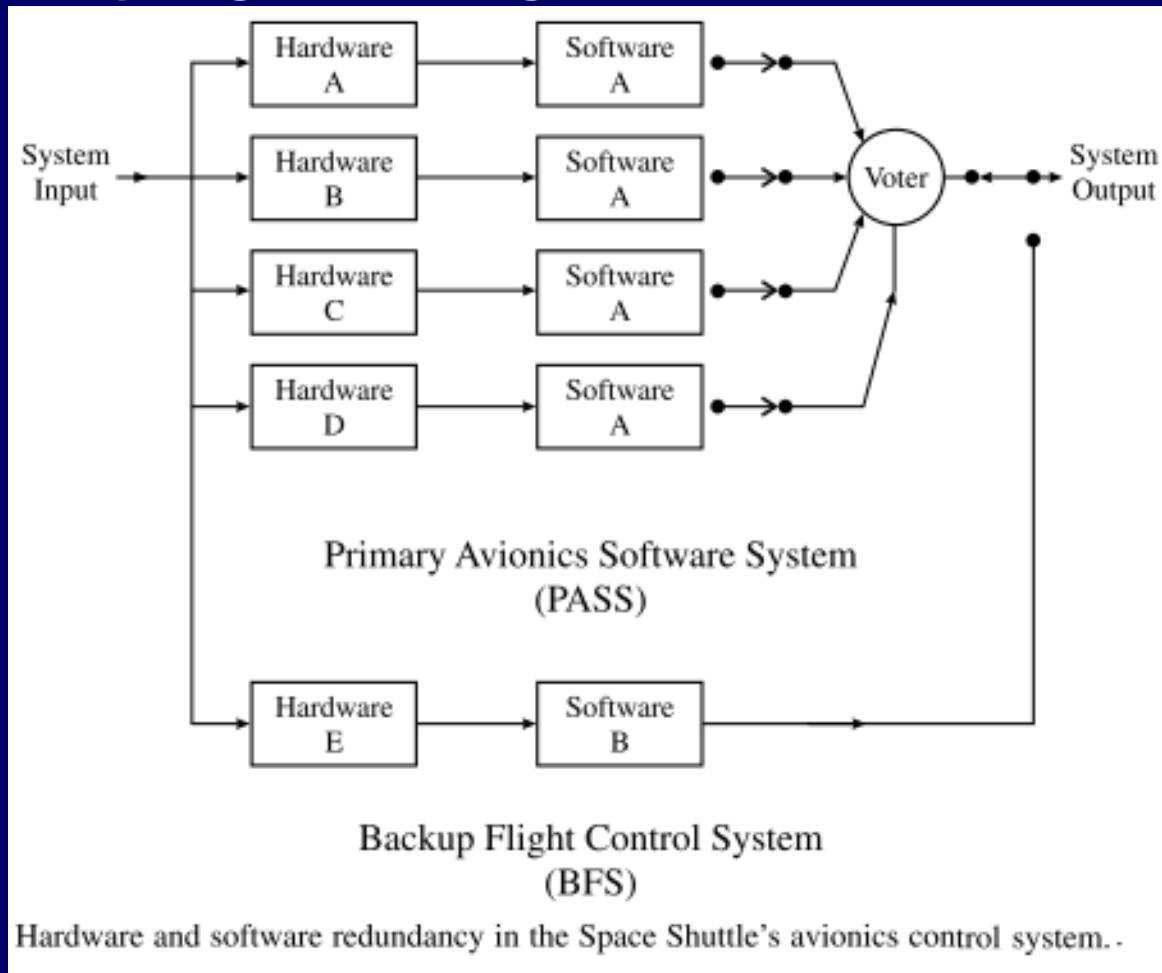
## ■ RAID 1+0 (RAID 10)

mirrored sets in a striped set.



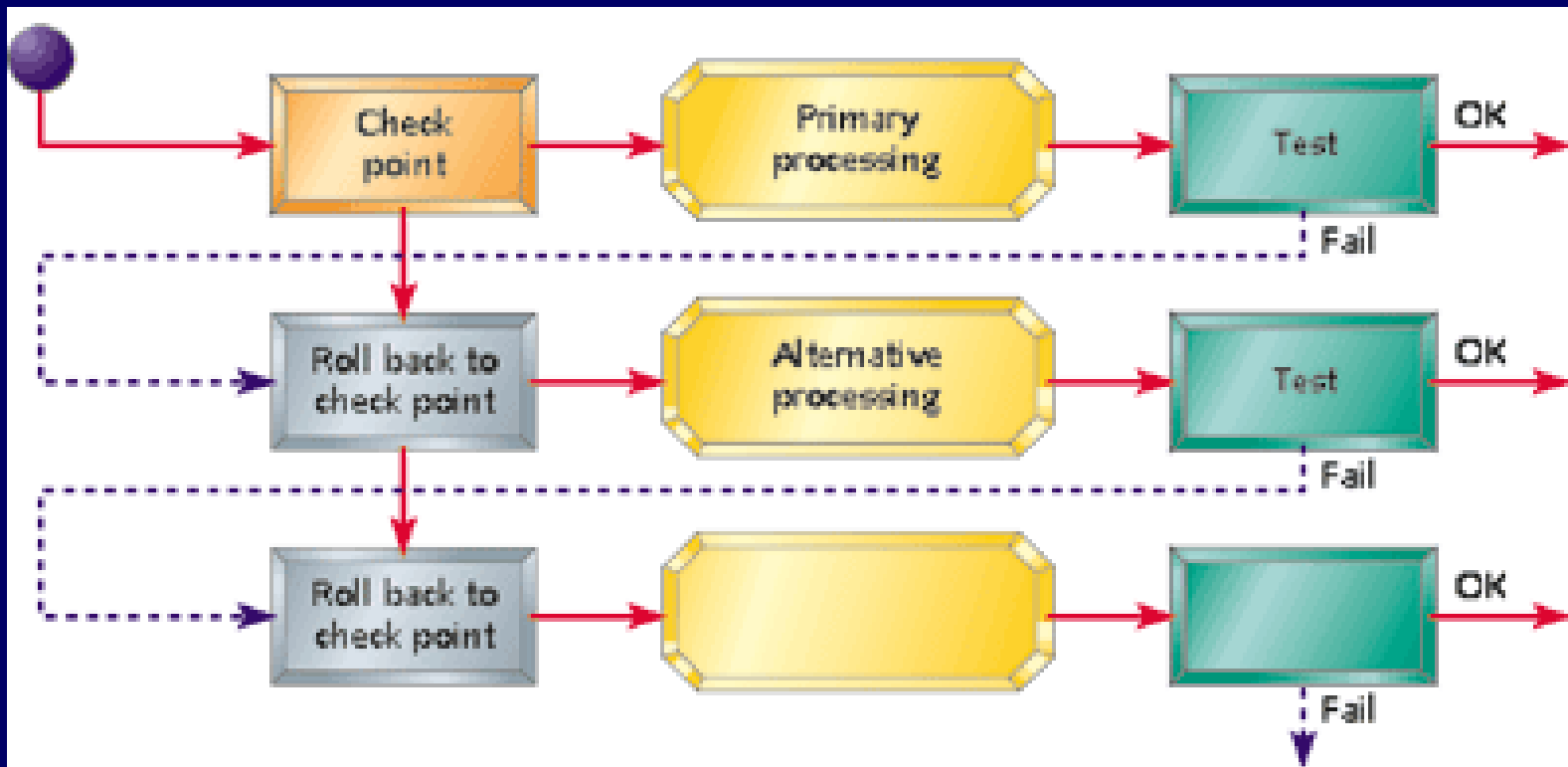
# Redundancy Mechanisms

## ■ N-version programming



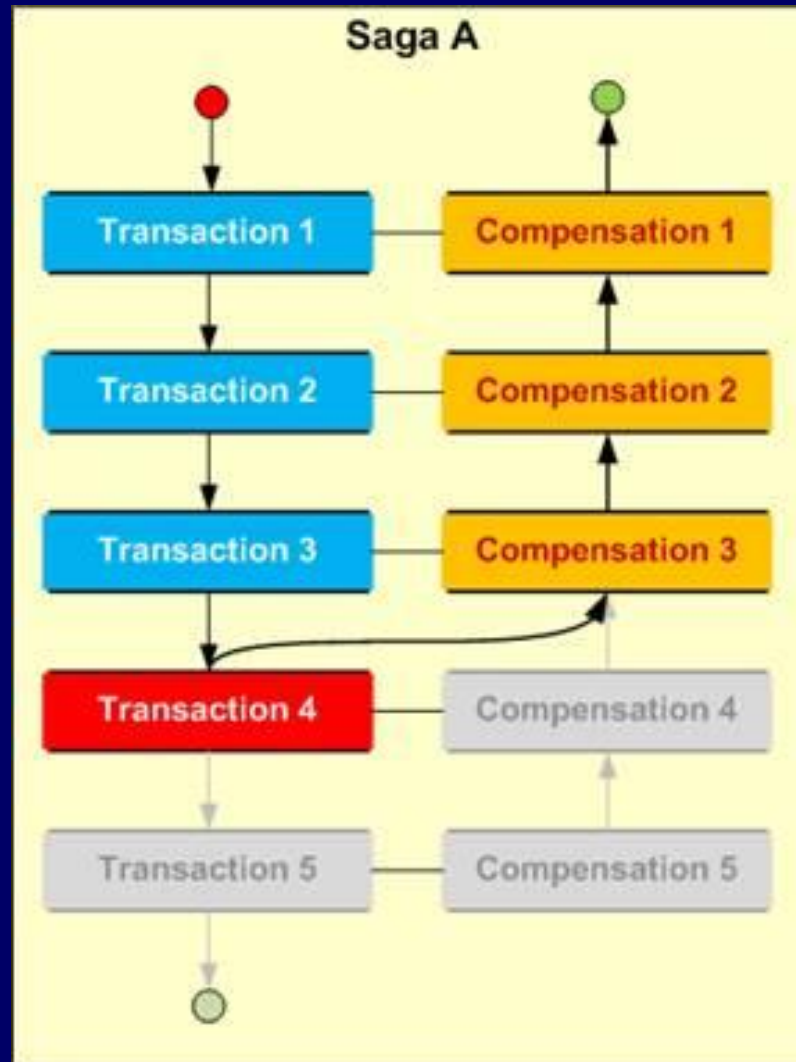
# Redundancy Mechanisms

## ■ Checkpoints and recovering



# Redundancy Mechanisms

## ■ Backward Recovery





# Redundancy Mechanisms

---

## ■ Reboot

The simplest - but weakest - recovery technique.

From the implementation standpoint is to reboot or restart the system.

## ■ Journaling - To employ these techniques requires that:

1. a copy of the original database, disk, and filename be stored,
2. all transactions that affect the data must be stored during execution, and
3. the process be backed up to the beginning and the computation be retried.

Clearly, items (2) and (3) require a lot of storage; in practice, journaling can only be executed for a given time period, after which the inputs and the process must be erased and a new journaling time period created.



# **COHERENT SYSTEM**

# Coherent System

## ■ Structure Function

### Operations

- $\{+, -, \times, \div\}$  – arithmetic operations

Consider a system  $S$  composed by a set of components,  $C = \{c_i | 1 \leq i \leq n\}$ , where the state of the system  $S$  and its components could be either operational or failed. Let the discrete random variable  $x_i$  indicate the state of component  $i$ , thus:

$$x_i = \begin{cases} 0 & \text{if the component } i \text{ has failed} \\ 1 & \text{if the component } i \text{ is operational} \end{cases}$$

The vector  $\mathbf{x} = (x_1, x_2, \dots, x_i, \dots, x_n)^1$  represents the state of each component of the system, and it is named state vector. The system state may be represented by a discrete random variable  $\phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_i, \dots, x_n)$ , such that

$$\phi(\mathbf{x}) = \begin{cases} 0 & \text{if the system has failed} \\ 1 & \text{if the system is operational} \end{cases}$$

$\phi(\mathbf{x})$  is called the structure function of the system.

If one is interested in representing the system state at a specific time  $t$ , the components' state variables should be interpreted as a random variables at time  $t$ . Hence,  $\phi(\mathbf{x}(t))$ , where  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_i(t), \dots, x_n(t))$ .

# Coherent System

## ■ Structure Function

For any component  $c_i$ ,

$$\phi(\mathbf{x}) = x_i \phi(1_i, \mathbf{x}) + (1 - x_i) \phi(0_i, \mathbf{x}),$$

where  $\phi(1_i, \mathbf{x}) = \phi(x_1, x_2, \dots, 1_i, \dots, x_n)$  and  $\phi(0_i, \mathbf{x}) = \phi(x_1, x_2, \dots, 0_i, \dots, x_n)$ .

The first term ( $x_i \phi(1_i, \mathbf{x})$ ) represents a state where the component  $c_i$  is operational and the state of the other components are random variables ( $\phi(x_1, x_2, \dots, 1_i, \dots, x_n)$ ). The second term ( $(1 - x_i) \phi(0_i, \mathbf{x})$ ), on the other hand, states the condition where the component  $c_i$  has failed and the state of the other components are random variables ( $\phi(x_1, x_2, \dots, 0_i, \dots, x_n)$ ).

Equation is known as factoring of the structure function and very useful for studying complex system structures, since through its repeated application, one can eventually reach a subsystem whose structure function is simple to deal with (1).

# Coherent System

---

## ■ Irrelevant Component

A component of a system is irrelevant to the dependability of the system if the state of the system is not affected by the state of the component.

$c_i$  is irrelevant to the structure function if  $\phi(1_i, \mathbf{x}) = \phi(0_i, \mathbf{x})$ .

# Coherent System

A system with structure function  $\phi(\mathbf{x})$  is said to be **coherent** if and only if  $\phi(\mathbf{x})$  is non-decreasing in each  $x_i$  and every component  $c_i$  is relevant.

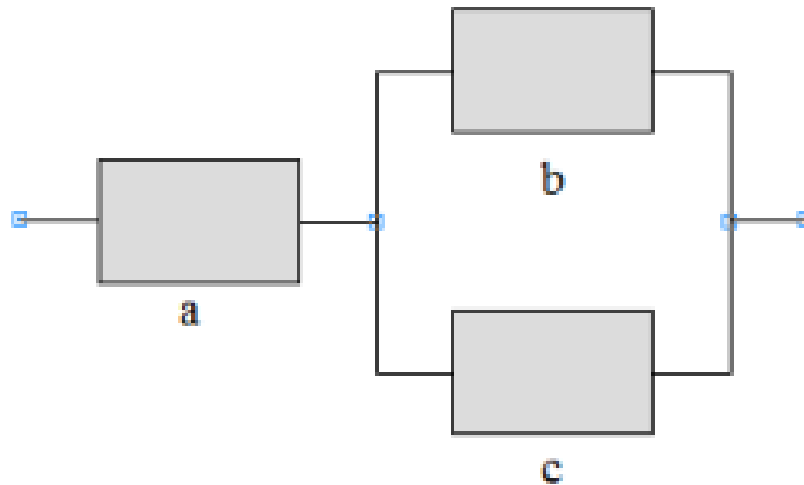
A function  $\phi(\mathbf{x})$  is non-decreasing if for every two state vectors  $\mathbf{x}$  and  $\mathbf{y}$ , such that  $\mathbf{x} < \mathbf{y}$ , then  $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$ .

Another aspect of coherence that should also be highlighted is that replacing a failed component in working system does not make the system fail. But, it does not also mean that a failed system will work if a failed component is substituted by an operational component.

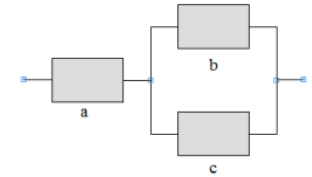
# Coherent System

## ■ Example - Structure Function

Consider a coherent system  $(C, \phi)$  composed of three blocks,  $C = \{a, b, c\}$



# Coherent System



## ■ Example - Structure Function

factoring on component  $a$ , we have:

$$\phi(x_a, x_b, x_c) = x_a \phi(1_a, x_b, x_c) + (1 - x_a) \phi(0_a, x_b, x_c) = x_a \phi(1_a, x_b, x_c),$$

since  $\phi(0_a, x_b, x_c) = 0$ .

Now factoring  $\phi(1_a, x_b, x_c)$  on component  $b$ ,

$$\phi(1_a, x_b, x_c) = x_b \phi(1_a, 1_b, x_c) + (1 - x_b) \phi(1_a, 0_b, x_c).$$

As  $\phi(1_a, 1_b, x_c) = 1$ , thus:

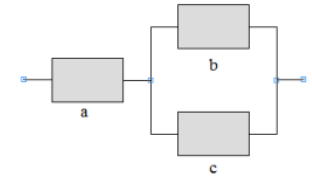
$$\phi(1_a, x_b, x_c) = x_b + (1 - x_b) \phi(1_a, 0_b, x_c).$$

Therefore:

$$\phi(x_a, x_b, x_c) = x_a \phi(1_a, x_b, x_c) = x_a \times [x_b + (1 - x_b) \phi(1_a, 0_b, x_c)].$$



# Coherent System



## ■ Example - Structure Function

Fact  $\phi(1_a, 0_b, x_c)$  on component  $c$  to get:

$$\phi(1_a, 0_b, x_c) = x_c \phi(1_a, 0_b, 1_c) + (1 - x_c) \phi(1_a, 0_b, 0_c).$$

Since  $\phi(1_a, 0_b, 1_c) = 1$  and  $\phi(1_a, 0_b, 0_c) = 0$ , thus:

$$\phi(1_a, 0_b, x_c) = x_c.$$

So

$$\begin{aligned} \phi(x_a, x_b, x_c) &= x_a \times [x_b + (1 - x_b) \phi(1_a, 0_b, x_c)] = \\ &= x_a \times [x_b + (1 - x_b) x_c] = \end{aligned}$$

$$\phi(x_a, x_b, x_c) = x_a x_b + x_a x_c (1 - x_b) =$$

$$\phi(x_a, x_b, x_c) = x_a [1 - (1 - x_b)(1 - x_c)].$$

# Coherent System

## ■ Logical Function

$$s_i = \begin{cases} F & \text{if the component } i \text{ has failed} \\ T & \text{if the component } i \text{ is operational} \end{cases}$$

$$\varphi(\mathbf{bs}) = \begin{cases} F & \text{if the system has failed} \\ T & \text{if the system is operational} \end{cases}$$

Operations

- $\{\wedge, \vee, \neg\}$  – logic operations

$\mathbf{bs} = (s_1, s_2, \dots, s_i, \dots, s_n)$  represents the Boolean state of each component of the system.

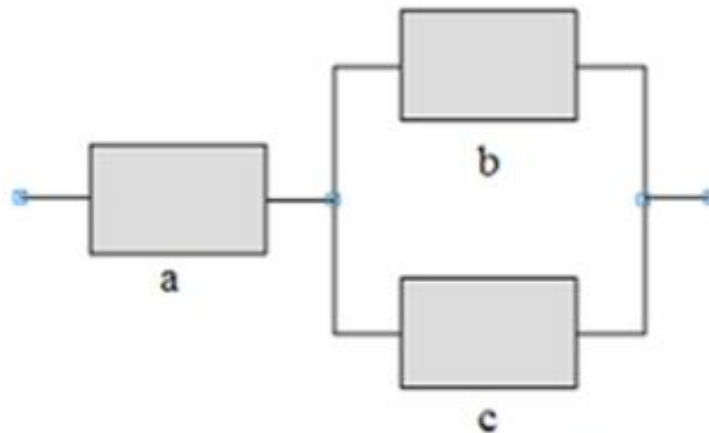
The system state could be either operational or failed.

The operational system state is represented by  $\varphi(\mathbf{bs})$ , whereas  $\overline{\varphi(\mathbf{bs})}$  denotes a faulty system.

# Coherent System

## ■ Example – Logical Function

Example: Consider a system  $(C, \phi)$  composed of three blocks,  $C = \{a, b, c\}$



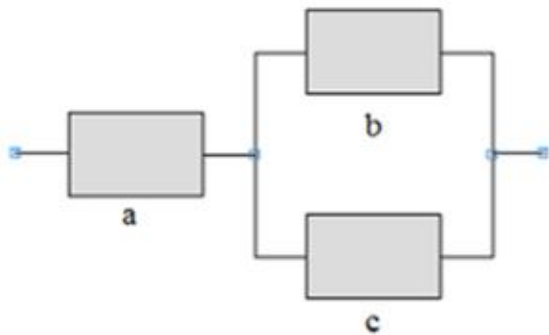
$$\varphi(s_a, s_b, s_c) = s_a \wedge (s_b \vee s_c) = s_a \wedge (\overline{\overline{s_b} \wedge \overline{s_c}})$$

# Coherent System

## ■ Example – Converting a Logical Function into a Structure Function

Using the notation described,  $s_i$  is equivalent to  $x_i$ ,  $\bar{s}_i$  represents  $1 - x_i$ ,  $\varphi(\mathbf{bs})$  is the counterpart of  $\phi(\mathbf{x}) = 1$ ,  $\overline{\varphi(\mathbf{bs})}$  depicts  $\phi(\mathbf{x}) = 0$ ,  $\wedge$  represents  $\times$ , and  $\vee$  is the respective counterpart of  $+$ .

Consider a system  $(C, \phi)$  composed of three blocks,  $C = \{a, b, c\}$



$$\varphi(s_a, s_b, s_c) = s_a \wedge (\overline{s_b \wedge s_c}).$$

$$\phi(\mathbf{x}) = x_a \times [1 - (1 - x_b) \times (1 - x_c)]$$



# MODELING

# Modeling Techniques

---

## ■ Classification

- State-space based models
  - CTMC, SPN, SPA
- Combinatorial models
  - RBD, FT, RG

---

# Combinatorial models

# Reliability Block Diagram

---

- RBD is success oriented diagram.
- Each component of the system is represented as a block
- RBDs are networks of functional blocks connected such that they affect the functioning of the system
- Failures of individual components are assumed to be independent for easy solution.
- System behavior is represented by connecting the blocks
  - Blocks that are all required are connected in series
  - Blocks among which only one is required are connected in parallel
  - When at least  $k$  out of  $n$  are required, use  $k$ -of- $n$  structure



# Reliability Block Diagram

---

- A RBD is not a block schematic diagram of a system, although they might be isomorphic in some particular cases.
- Although RBD was initially proposed as a model for calculating reliability, it has been used for computing availability, maintainability etc.

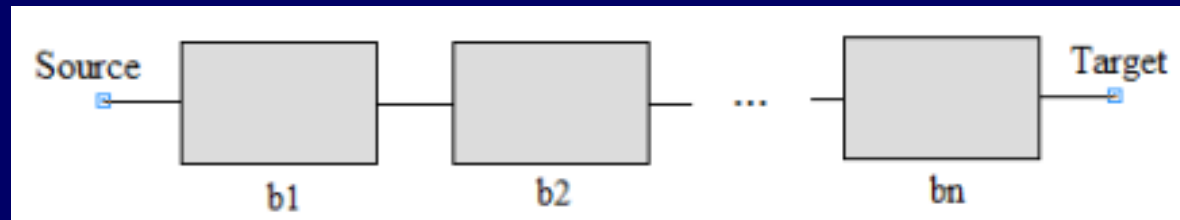
# Reliability Block Diagram

---

- Series

# Reliability Block Diagram

## ■ Series



$$P\{\phi(\mathbf{x}) = 1\} = P\{\phi(x_1, x_2, \dots, x_i, \dots, x_n) = 1\} = \prod_{i=1}^n P\{x_i = 1\} = \prod_{i=1}^n p_i = 1.$$

Therefore, the system reliability is

$$R_S(t) = P\{\phi(\mathbf{x}, t) = 1\} = \prod_{i=1}^n P\{x_i(t) = 1\} = \prod_{i=1}^n R_i(t),$$

where  $R_i(t)$  is the reliability of block  $b_i$ .

Likewise, the system instantaneous availability is

$$A_S(t) = P\{\phi(\mathbf{x}, t) = 1\} = \prod_{i=1}^n P\{x_i(t) = 1\} = \prod_{i=1}^n A_i(t),$$

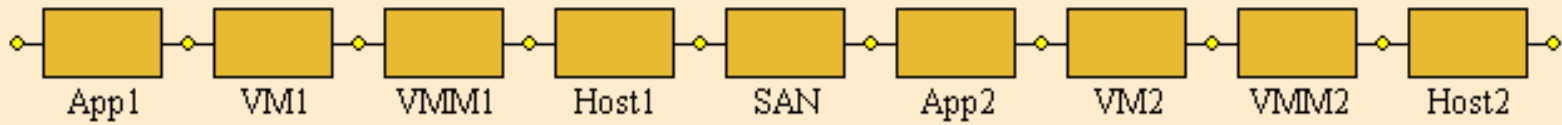
where  $A_i(t)$  is the instantaneous availability of block  $b_i$ .

The steady state availability is

$$A_S = P\{\phi(\mathbf{x}) = 1\} = \prod_{i=1}^n P\{x_i = 1\} = \prod_{i=1}^n A_i,$$

where  $A_i$  is steady state availability of block  $b_i$ .

# Computing the Reliability



$$\begin{aligned} R(t) &= e^{-\lambda_{app1}t} \times e^{-\lambda_{VM1}t} \\ &\quad \times e^{-\lambda_{VMM1}t} \times e^{-\lambda_{H1}t} \\ &\quad \times e^{-\lambda_{SAN}t} \times \\ &e^{-\lambda_{app2}t} \times e^{-\lambda_{VM2}t} \times e^{-\lambda_{VMM2}t} \\ &\quad \times e^{-\lambda_{H2}t} = \\ &e^{-(\lambda_{app1} + \lambda_{VM1} + \lambda_{VMM1} + \lambda_{H1} + \lambda_{SAN} + \lambda_{app2} + \lambda_{VM2} + \lambda_{VMM2} + \lambda_{H2})t} \end{aligned}$$

$$R(t) = 0.805735302, \quad t = 0.002 \text{ tu}$$

# Reliability Block Diagram

## ■ Series

Series system of  $n$  independent components, where the  $i$  component has lifetime exponentially distributed with rate  $\lambda_i$

Thus lifetime of the system is exponentially distributed with parameter  $\sum_{i=1}^n \lambda_i$

and system MTTF =  $1 / \sum_{i=1}^n \lambda_i$

# Reliability Block Diagram

## ■ Series

R.v.  $X$ : series system life time

R.v.  $X_i$ :  $i^{\text{th}}$  comp's life time (arbitrary distribution)

$$0 \leq E[X] \leq \min\{E[X_i]\}$$

Case of *weakest link*

$$X = \min\{X_1, X_2, \dots, X_n\}$$

$$R_X(t) = \prod_{i=1}^n R_{X_i}(t) \leq \min_i \{R_{X_i}(t)\}, \quad (0 \leq R_{X_i}(t) \leq 1)$$

$$\begin{aligned} E[X] &= \int_0^{\infty} R_X(t) dt \leq \min_i \left\{ \int_0^{\infty} R_{X_i}(t) dt \right\} \\ &= \min_i \{E[X_i]\} \end{aligned}$$

# Reliability Block Diagram

## ■ Example:

Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are  $\lambda_1 = 0.00001$  failures per hour,  $\lambda_2 = 0.00002$  failures per hour,  $\lambda_3 = 0.00003$  failures per hour, and  $\lambda_4 = 0.00004$  failures per hour, respectively. The sw system cannot work when any one of the web services is down.

- a) Calculate the total sw system failure rate.
- b) Calculate MTTF of sw system.
- c) Calculate the  $R(t)$  at 730h

# Reliability Block Diagram

## ■ Example:

Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are  $\lambda_1 = 0.00001$  failures per hour,  $\lambda_2 = 0.00002$  failures per hour,  $\lambda_3 = 0.00003$  failures per hour, and  $\lambda_4 = 0.00004$  failures per hour, respectively. The sw system cannot work when any one of the web services is down.

- a) Calculate the total sw system failure rate.
- b) Calculate MTTF of sw system.
- c) Calculate the  $R(t)$  at 730h



# Reliability Block Diagram

## ■ Example:

The **sw system cannot work** when **any one of the web services is down**.



The **sw system only works** when **all web services work**.

$ws_1 \stackrel{\text{def}}{=} \text{web services 1 working}$

$ws_2 \stackrel{\text{def}}{=} \text{web services 2 working}$

$ws_3 \stackrel{\text{def}}{=} \text{web services 3 working}$

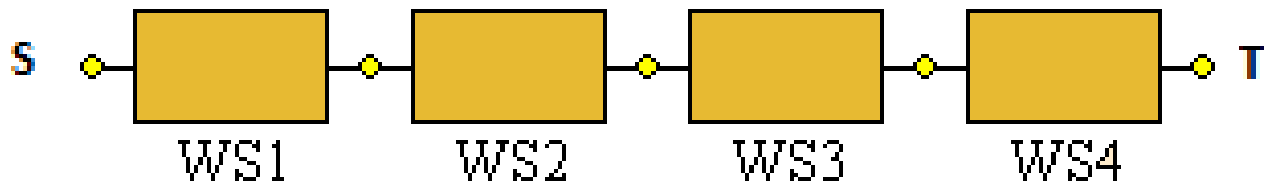
$ws_4 \stackrel{\text{def}}{=} \text{web services 4 working}$

$$\varphi(ws_1, ws_2, ws_3, ws_4) = ws_1 \wedge ws_2 \wedge ws_3 \wedge ws_4$$

# Reliability Block Diagram

## ■ Example:

$$\varphi(wS_1, wS_2, wS_3, wS_4) = wS_1 \wedge wS_2 \wedge wS_3 \wedge wS_4$$



– a)  $\sum_{i=1}^n \lambda_i$

$$\begin{aligned} \lambda_s &= 0.00001 + 0.00002 + 0.00003 + 0.00004 \\ &= 0.0001 \text{ failures per hour} \end{aligned}$$

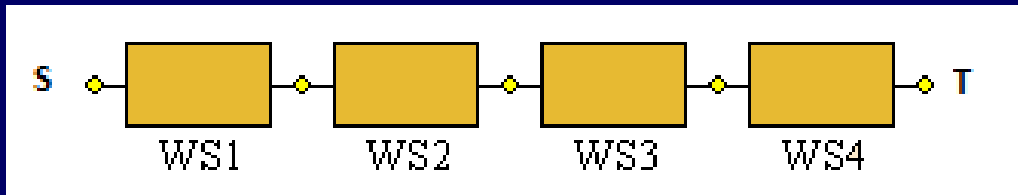
– b)  $MTTF = 1 / \sum_{i=1}^n \lambda_i$

$$MTTF_s = \frac{1}{0.0001} = 10,000 \text{ h}$$

# Reliability Block Diagram

## ■ Example:

– c)



$$\phi(x_1, x_2, x_3) = x_1 x_2 x_3 x_4$$

$$P\{\phi(x_1, x_2, x_3) = 1\} = E\{\phi(x_1, x_2, x_3)\} = E\{x_1 x_2 x_3 x_4\}$$

If the components are independent, then:

$$P\{\phi(x_1, x_2, x_3) = 1\} = E\{x_1\} E\{x_2\} E\{x_3\} E\{x_4\} =$$

As

$$P\{\phi(x_1, x_2, x_3) = 1\} = R(t), \text{ then}$$

$$P\{\phi(x_1, x_2, x_3) = 1\} = R(t) = r_1(t) r_2(t) r_3(t) r_4(t)$$

And, since  $r_i(t) = e^{-\lambda_i t}$ , therefore:

$$R(t) = e^{-\lambda_1 t} \times e^{-\lambda_2 t} \times e^{-\lambda_3 t} \times e^{-\lambda_4 t} = e^{-(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)t}$$

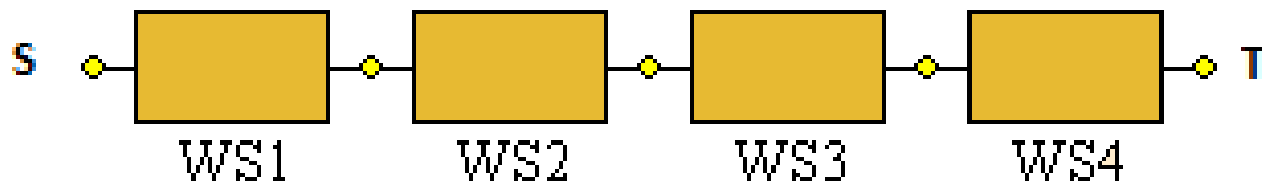
$$R(730h) = e^{-(0.00001 + 0.00002 + 0.00003 + 0.00004) \times 730} = 0.929600830$$

# Reliability Block Diagram

## ■ Problem:

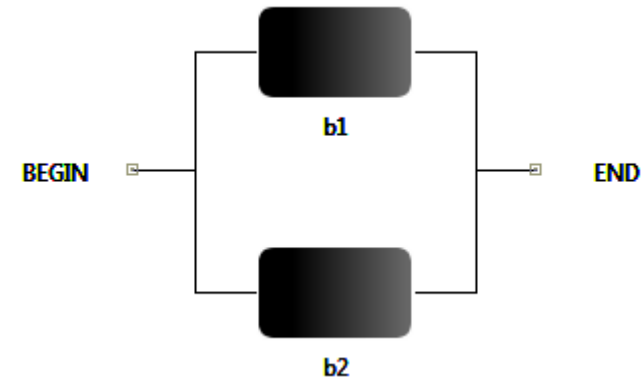
Now, considering the previous example, suppose that the repairing time of each web service is exponentially distributed with average 2h.

- Compute the steady state availability.
- Compute the downtime in minutes in one year period.



# Reliability Block Diagram

## ■ Parallel



# Reliability Block Diagram

## ■ Parallel

$$P\{\phi(\mathbf{x}) = 1\} = P\{\phi(x_1, x_2, \dots, x_i, \dots, x_n) = 1\} = 1 - \prod_{i=1}^n P\{x_i = 0\} = 1 - \prod_{i=1}^n (1 - P\{x_i = 1\}) =$$
$$P\{\phi(\mathbf{x}) = 1\} = 1 - \prod_{i=1}^n (1 - p_i).$$

Thus  $P\{\phi(\mathbf{x}) = 1\} = 1 - (1 - p_i)^n$ .

The system reliability is then:

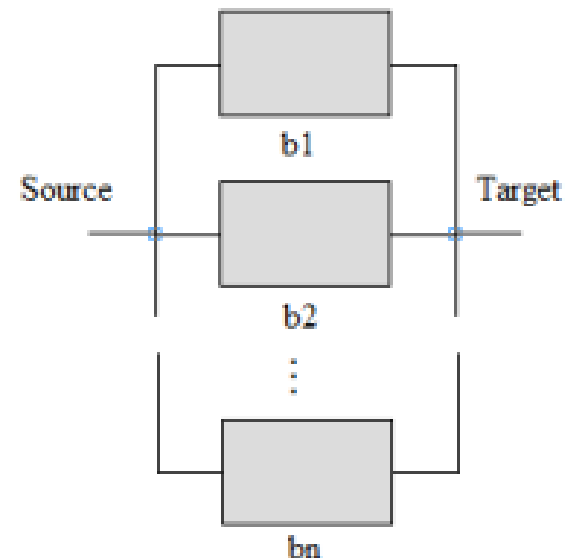
$$R_p(t) = 1 - \prod_{i=1}^n P\{x_i(t) = 0\} = 1 - \prod_{i=1}^n (1 - P\{x_i(t) = 1\})$$

$$R_p(t) = 1 - \prod_{i=1}^n Q_i(t) = 1 - \prod_{i=1}^n (1 - R_i(t)) ,$$

such that,

$$Q_i(t) = P\{x_i(t) = 0\} = 1 - P\{x_i(t) = 1\} = 1 - R_i(t),$$

where  $R_i(t)$  and  $Q_i(t)$  are the reliability and the unreliability of block  $b_i$ , respectively.



# Reliability Block Diagram

## ■ Parallel

Similarly, the system instantaneous availability is

$$A_p(t) = P\{\phi(\mathbf{x}, t) = 1\} = 1 - \prod_{i=1}^n P\{x_i(t) = 0\} = 1 - \prod_{i=1}^n 1 - A_i(t),$$

$$A_p(t) = P\{\phi(\mathbf{x}, t) = 1\} = 1 - \prod_{i=1}^n UA_i(t) = 1 - \prod_{i=1}^n 1 - A_i(t),$$

such that,  $UA_i(t) = P\{x_i(t) = 0\} = 1 - P\{x_i(t) = 1\} = 1 - A_i(t)$ ,

where  $A_i(t)$  and  $UA_i(t)$  are the instantaneous availability and unavailability of block  $b_i$ , respectively.

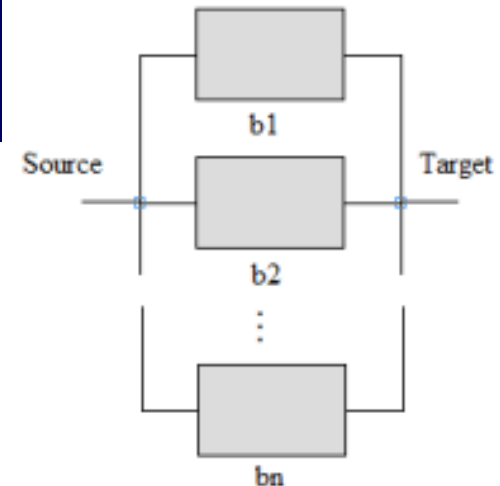
The steady state availability is

$$A_p = P\{\phi(\mathbf{x}) = 1\} = 1 - \prod_{i=1}^n UA_i = 1 - \prod_{i=1}^n 1 - A_i,$$

where  $A_i$  and  $UA_i$  are the steady availability and unavailability of block  $b_i$ , respectively.

Due to the importance of the parallel structure, the following simplifying notation is adopted:

$$P\{\phi(\mathbf{x}) = 1\} = 1 - \prod_{i=1}^n (1 - P\{x_i = 1\}) = \prod_{i=1}^n P\{x_i = 1\} = \prod_{i=1}^n p_i = 1 - (1 - p_i)^n.$$



# Reliability Block Diagram

Examples

## ■ Parallel

For a parallel system with  $n$  independent and identical components with rate  $\lambda$

$$R_{ps}(t) = 1 - (1 - e^{-\lambda t})^n$$

and system

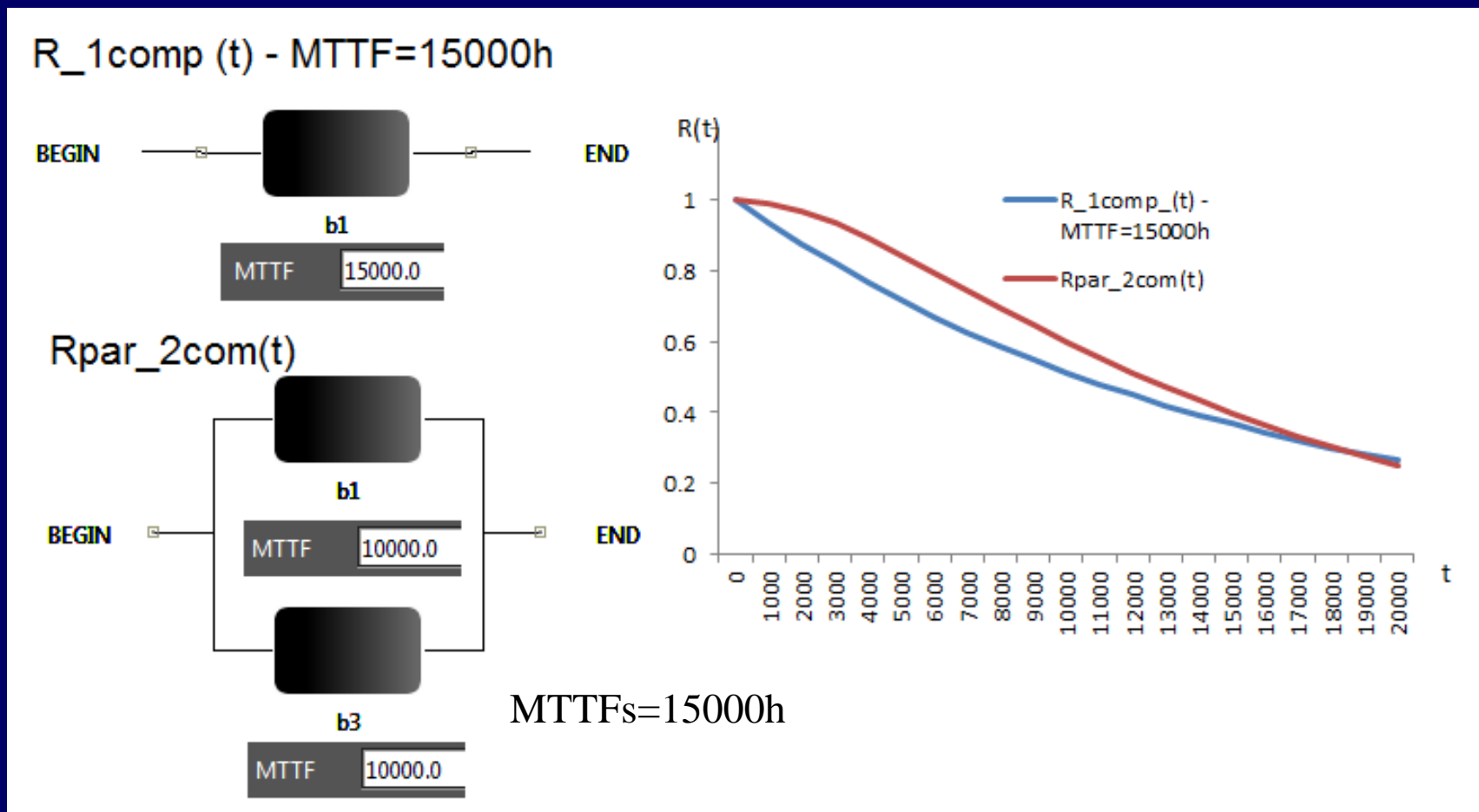
$$MTTF = \int_0^{\infty} R(t) \times dt = \int_0^{\infty} [1 - (1 - e^{-\lambda t})^n] dt = \frac{1}{\lambda} \sum_{i=1}^n \frac{1}{i}$$



# Reliability Block Diagram

Comp

## ■ Example



# Reliability Block Diagram

## ■ Example

The **system works** when **at least one server works**.

$s_1 \stackrel{\text{def}}{=} \text{server 1 working}$

$s_2 \stackrel{\text{def}}{=} \text{server 2 working}$

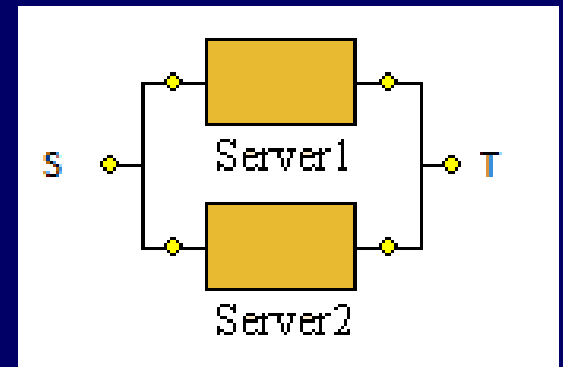
$$\varphi(s_1, s_2) = s_1 \vee s_2 \Leftrightarrow \overline{\varphi(s_1, s_2)} = \bar{s}_1 \wedge \bar{s}_2$$

We know that

$$P\{\phi(\mathbf{x}) = 1\} = 1 - (1 - p_1)(1 - p_2)$$

As

$P\{\phi(\mathbf{x}) = 1\}$  can be  $R(t)$ ,  $A(t)$ ,  $A$



# Reliability Block Diagram

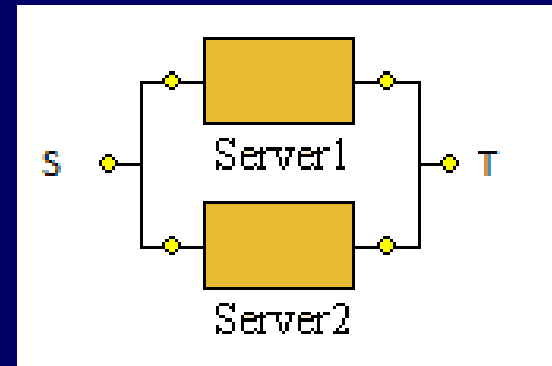
## ■ Example

We know that

$$P\{\phi(\mathbf{x}) = 1\} = 1 - (1 - p_1)(1 - p_2)$$

As

$P\{\phi(\mathbf{x}) = 1\}$  can be  $R(t)$ ,  $A(t)$ ,  $A$



– a)

$$\begin{aligned} R(t) &= 1 - (1 - R_1(t))(1 - R_2(t)) \\ &= R_1(t) + R_2(t) - R_1(t)R_2(t) \\ &= e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{-(\lambda_1 + \lambda_2)t} \end{aligned}$$

# Reliability Block Diagram

Examples

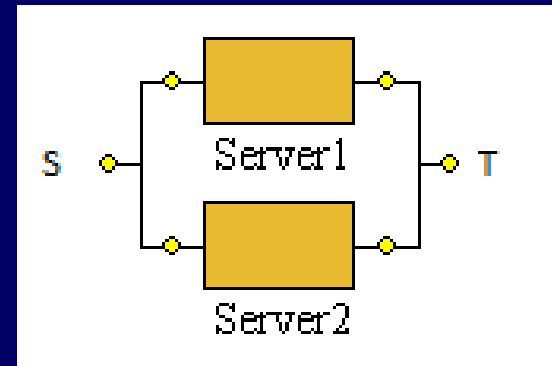
## ■ Example

We know that

$$P\{\phi(\mathbf{x}) = 1\} = 1 - (1 - p_1)(1 - p_2)$$

As

$P\{\phi(\mathbf{x}) = 1\}$  can be  $R(t)$ ,  $A(t)$ ,  $A$



– b)

$$\begin{aligned} MTTF_p &= \int_0^{\infty} R(t) dt = \int_0^{\infty} (e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{-(\lambda_1 + \lambda_2)t}) dt \\ &= \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \frac{1}{\lambda_1 + \lambda_2} \end{aligned}$$

– c)

$$R(730h) = 0.9997906870$$

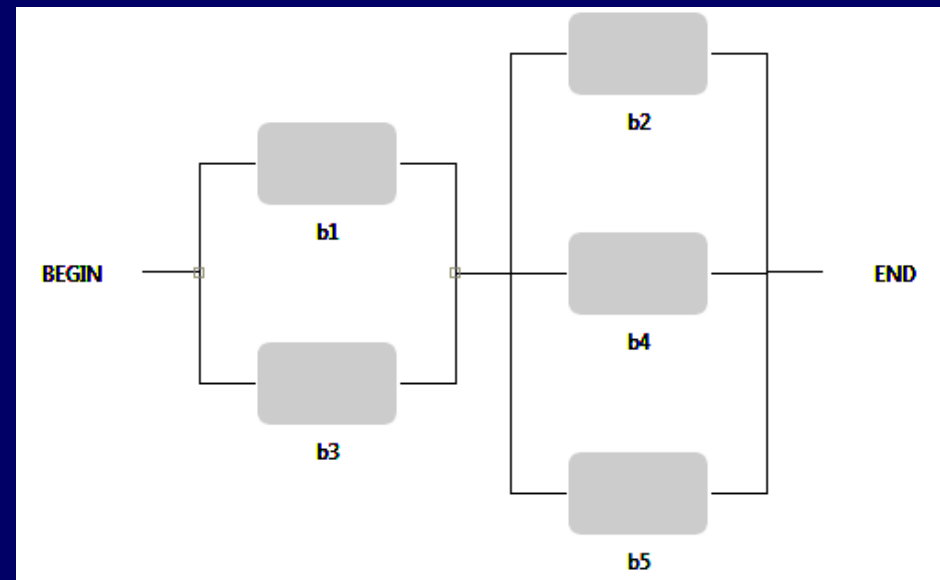
$$MTTF = 105\,000h$$

# Reliability Block Diagram

## ■ Series-Parallel System

- Series-parallel system:  $n$  stages in series, stage  $i$  with  $n_i$  parallel components.
- For  $i=1, \dots, n$ ,  $R_{ij} = R_j$ ,  $n_i \geq j \geq 1$
- Reliability of series-parallel system is given by

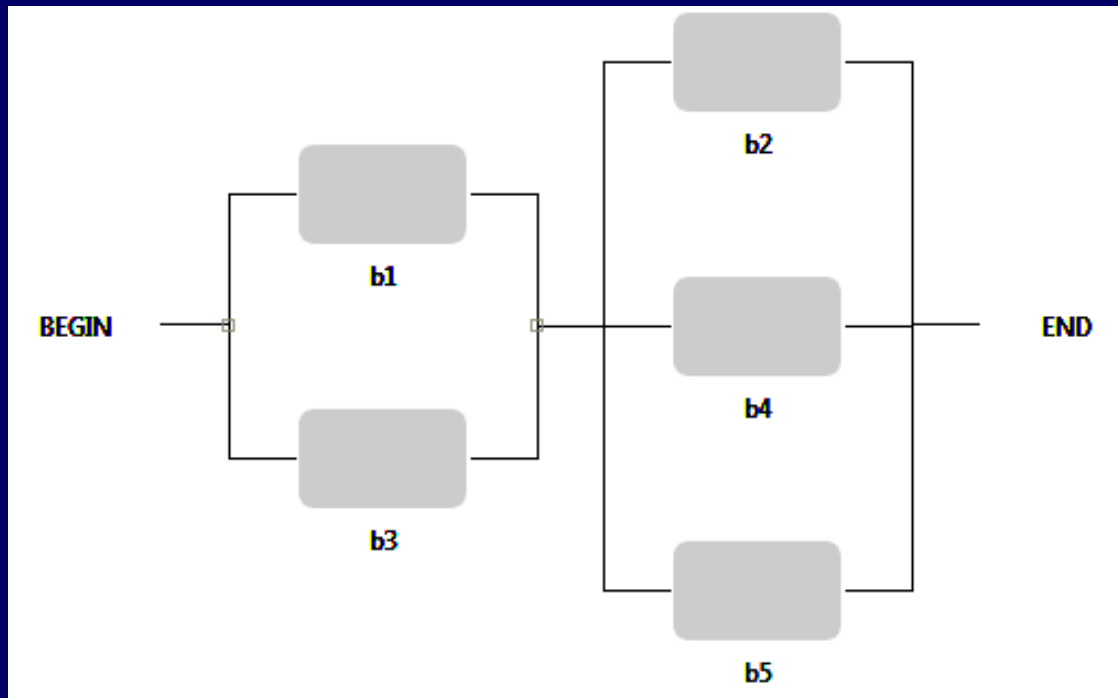
$$R_{sp} = \prod_{i=1}^n [1 - (1 - R_i)^{n_i}]$$



# Reliability Block Diagram

## ■ Series-Parallel System

Example:

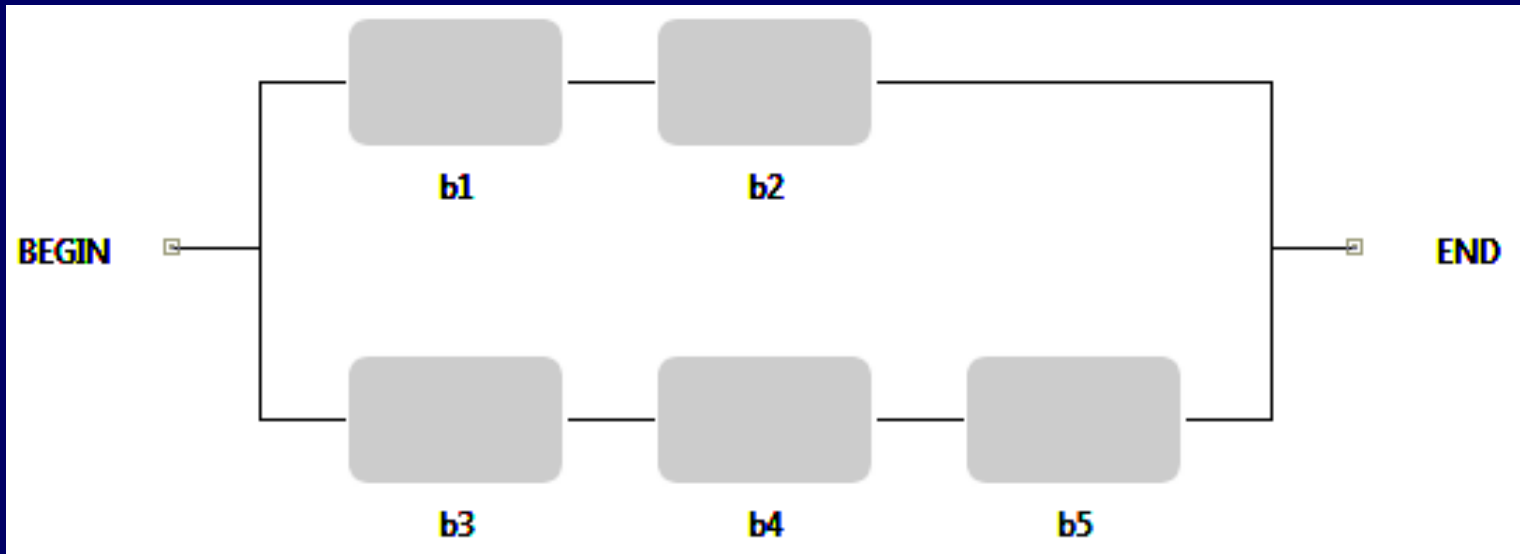


$$P = (1 - (1 - p_1)(1 - p_3)) \times (1 - (1 - p_2)(1 - p_4)(1 - p_5))$$

# Reliability Block Diagram

## ■ Series-Parallel System

Example:

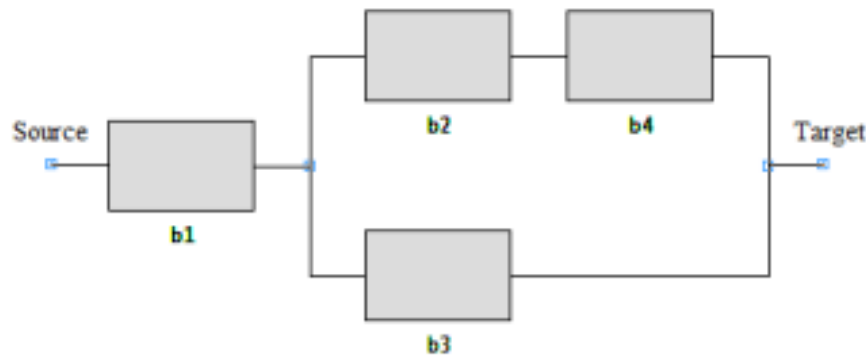


$$P = (1 - (1 - p_1 p_2))(1 - p_3 p_4 p_5)$$

# Reliability Block Diagram

## ■ Example:

Consider a system  $S_1$  represented by four blocks ( $b_1, b_2, b_3, b_4$ ) where each block has  $r_1, r_2, r_3$  and  $r_4$  as their respective reliabilities.



RBD of System  $S_1$

The system reliability of the system  $S_1$  is

$$R_{S_1} = r_1 \times [1 - (1 - r_2 \times r_4) \times (1 - r_3)].$$



# Reliability Block Diagram

## ■ Problem

Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are  $\lambda_1 = 0.00001$  failures per hour,  $\lambda_2 = 0.00002$  failures per hour,  $\lambda_3 = 0.00003$  failures per hour, and  $\lambda_4 = 0.00004$  failures per hour, respectively. The sw system provides the proper service if the web services 1 or 3 are up and the web services 2 or 4 are up.

- a) Calculate MTTF of sw system.
- b) Calculate the  $R(t)$  at 730h

# Reliability Block Diagram

---

## ■ Problem

Now, considering the previous example, suppose that the repairing time of each web service is exponentially distributed with average 2h.

- a) Compute the steady state availability.
- b) Compute the downtime in hours in one year period.

# Reliability Block Diagram

## ■ K out of N

Sequence of Bernoulli trials:  $n$  independent repetitions.

- $n$  consecutive executions of an **if-then-else** statement

$S_n$ : sample space of  $n$  Bernoulli trials

$$S_1 = \{0, 1\}$$

$$S_2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$$

$$S_n = \{2^n \text{ } n\text{-tuples of 0s and 1s}\}$$

# Reliability Block Diagram

## ■ K out of N

Consider  $s \in S_n$ , such that,  $s = (\underbrace{1, 1, \dots, 1}_k, \underbrace{0, 0, \dots, 0}_{n-k})$

$$s = A_1 \cap A_2 \cap \dots \cap A_k \cap \bar{A}_{k+1} \cap \dots \cap \bar{A}_n$$

$$\begin{aligned} P(s) &= P(A_1)P(A_2)\dots P(A_k)P(\bar{A}_{k+1})\dots P(\bar{A}_n) \\ &= p^k q^{n-k} \end{aligned}$$

$P(s)$ : Prob. of sequence of  $k$  successes followed by  $(n-k)$  failures. What about any sequence of  $k$  successes out of  $n$  trials?

# Reliability Block Diagram

## ■ K out of N

$k$  1's can be arranged in  $\binom{n}{k}$  different ways,

$$\begin{aligned} p(k) &= P(\text{Exactly } k \text{ successes and } n - k \text{ failures}) \\ &= \binom{n}{k} p^k (1 - p)^{n-k} \end{aligned}$$

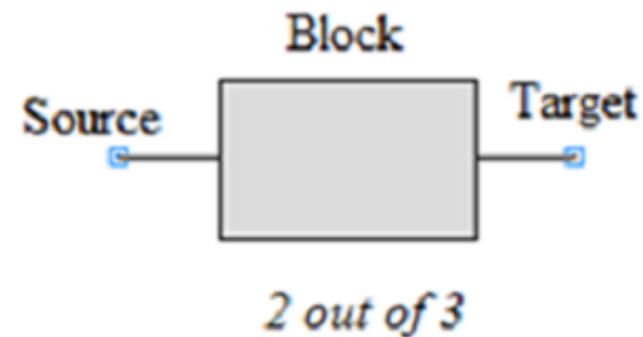
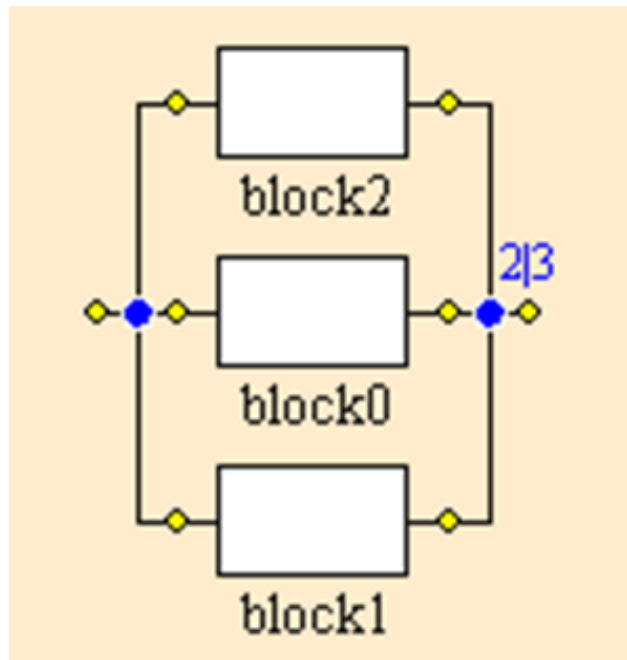
$k=n$ , reduces to Series system  $p(n) = p^n$

$k=1$ , reduces to Parallel system  $p(1) = 1 - (1 - p)^n$

# Reliability Block Diagram

## Example: 2 out of 3 system

$n$  statistically identical components; also statistically independent

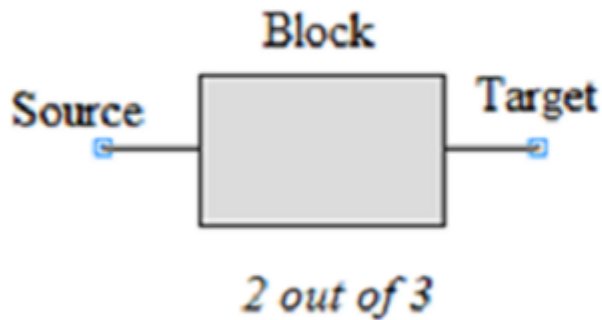


# Reliability Block Diagram

## Example: 2 out of 3 system

$n$  statistically identical components; also statistically independent

$$\sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}$$



If  $n = 3$  and  $k = 2$ , then

$$\sum_{i=2}^3 \binom{3}{i} p^i (1-p)^{3-i} =$$

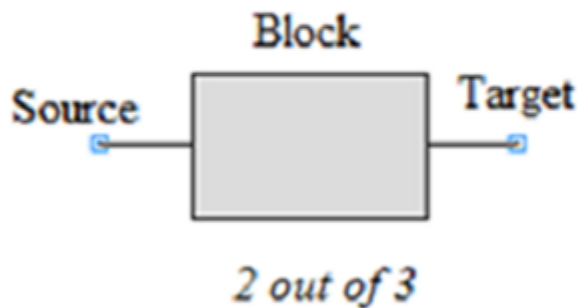
$$\binom{3}{2} p^2 (1-p)^{3-2} + \binom{3}{3} p^3 (1-p)^{3-3} =$$

$$3p^2(1-p) + p^3 = 3p^2 - 2p^3.$$

# Reliability Block Diagram

## Example: 2 out of 3 system

$n$  statistically identical components; also statistically independent



$$R_S(t) = \sum_{x=k}^n \binom{n}{x} e^{-\lambda t x} (1 - e^{-\lambda t})^{n-x}$$

$$\text{MTTF} = \sum_{x=k}^n \binom{n}{x} \int_0^{\infty} e^{-\lambda t x} (1 - e^{-\lambda t})^{n-x} dt$$

$$\text{MTTF} = \frac{1}{\lambda} \sum_{x=k}^n \frac{1}{x}$$



# Reliability Block Diagram

## ■ 2 out of 3

Assume independence and that the reliability of a single component is:  $R_{Simplex}(t) = e^{-\lambda t}$

we get:  $R_{2oo3}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$

$$\begin{aligned} E[X] &= \int_0^{\infty} R_{2oo3}(t) dt = \int_0^{\infty} 3e^{-2\lambda t} dt - \int_0^{\infty} 2e^{-3\lambda t} dt \\ &= \frac{5}{6\lambda} = MTTF_{2oo3} \end{aligned}$$

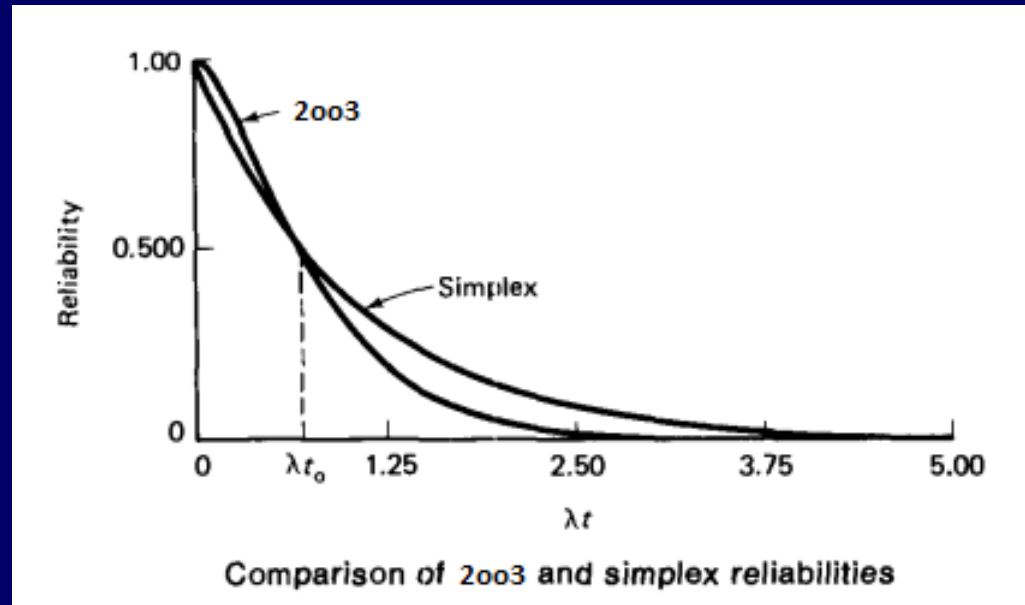
Comparing with expected life of a single

component:  $MTTF_{2oo3} = \frac{5}{6\lambda} < \frac{1}{\lambda} = MTTF_{Simplex}$

# Reliability Block Diagram

Examples

2 out of 3

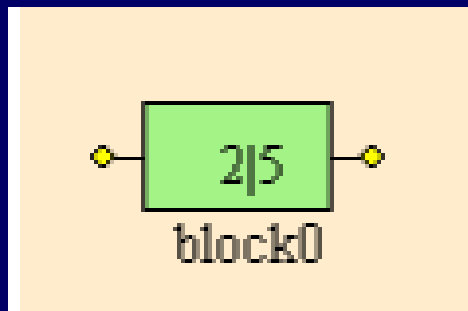


Thus 2oo3 actually reduces (by 16%) the MTTF over the simplex system.

Although 2oo3 has lower MTTF than does Simplex, it has higher reliability than Simplex for “short” missions, defined by mission time  $t < (\ln 2)/\lambda$ .

# Reliability Block Diagram

## Example: 2 out of 5



$\lambda = 0.1 \stackrel{\text{def}}{=} a \text{ component failure rate}$

$\mu = 0.9 \stackrel{\text{def}}{=} a \text{ component repair rate}$

$A = \frac{\mu}{(\mu + \lambda)} \stackrel{\text{def}}{=} a \text{ Component Availability}$

$$A_B = \sum_{i=k}^n \binom{n}{i} A^i (1-A)^{(n-i)}$$

$$A_B = \sum_{k=2}^5 \binom{5}{k} A^k (1-A)^{5-k}$$

$$= 0.0081 + 0.0729 + 0.32805 + 0.59049 = 0.99954$$

Block Availability =

\*\*\*\*\* Outputs asked for the model: 2005 \*\*\*\*\*

Steady-State Availability

SS\_Avail: 9.99540000e-001

# Reliability Block Diagram

## ■ Example

For a system with 6 HDDs in a RAID-0 disk set, if the reliability of each HDD at  $t=3$  years is 0.9, the reliability of the RAID set is

$$R_{\text{RAID set}}(t) = \prod_{i=1}^6 R_{\text{HDD}}(t)$$

$$R_{\text{RAID set}}(3 \text{ years}) = \prod_{i=1}^6 R_{\text{HDD}}(3 \text{ years}) =$$

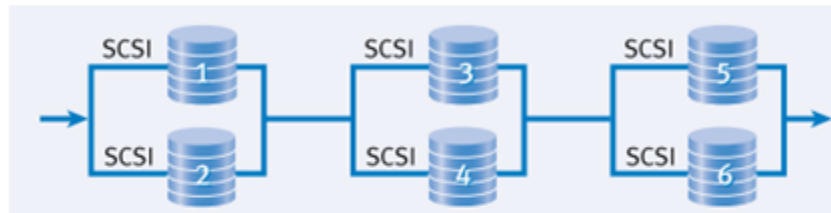
$$R_{\text{RAID set}}(3 \text{ years}) = \prod_{i=1}^6 0.9 = 0.531441$$



# Reliability Block Diagram

## ■ Example

Consider the reliability of each HDD at  $t=3$  years as 0.9. For a storage system with 6 HDDs configured as RAID-1 array, what is the storage system reliability at  $t=3$  years?



*# of RAID sets*

$$R_{\text{RAID-1 set}}(t) = \prod_{i=1}^{\# \text{ of RAID sets}} R_{\text{RAID-1}}^i(t)$$

$$R_{\text{RAID-1}}(t) = 1 - (1 - R_{\text{HDD}}(t))^2 = 0.99$$

$$R_{\text{RAID-1 set}}(t) = \prod_{i=1}^3 (R_{\text{RAID-1}}^i(t))$$

$$R_{\text{RAID-1 set}}(t) = 0.99 \times 0.99 \times 0.99 = 0.97029899$$

# Reliability Block Diagram

RAID-5 can tolerate one HDD failure in an array of  $n$  HDDs. For example, if the parity HDD fails, the remaining data HDDs are not affected, but redundancy is lost. If a data HDD fails, the RAID controller uses the remaining data HDDs and the parity HDD to recalculate the missing data on the fly. System performance slightly degrades until the failed HDD is replaced; however, no data is lost.

All data in the RAID set will be lost if another HDD fails before the failed HDD is restored.

The mathematical relationship that evaluates the reliability of  $n$  HDDs in a RAID-5 configuration is

$$R_{\text{RAID-5 set}}(t) = \sum_{j=n-1}^n \binom{n}{j} R_{\text{HDD}}^j(t) \times \left(1 - R_{\text{HDD}}^j(t)\right)^{n-j}$$

# Reliability Block Diagram

---

- Example



# Reliability Block Diagram

## ■ Importance Indices

### Reliability Importance

The *reliability importance*, or Birnbaum importance (B-importance), of component  $i$  is defined as

$$I_i^B = \frac{\partial R_s(\mathbf{p})}{\partial p_i} \quad 0 \leq p_i \leq 1$$

$p_i$  is the reliability of component  $i$ ,  $\mathbf{p}$  is the vector of component reliabilities, and  $R_s$  is the reliability of the system.

$$I_i^B = R_s(1_i, \mathbf{p}^i) - R_s(0_i, \mathbf{p}^i),$$

where  $\mathbf{p}^i$  represents the component reliability vector with the  $i$ th component removed.

$$I_i^B = E(\phi(1_i, \mathbf{x}^i) - \phi(0_i, \mathbf{x}^i)) = \Pr(\phi(1_i, \mathbf{x}^i) - \phi(0_i, \mathbf{x}^i) = 1)$$

where  $\phi$  is the structural function of the system.



# Reliability Block Diagram

## ■ Importance Indices

Normalized Reliability Importance

$$I_{n_i}^B = \frac{I_i^B}{I_x}$$

where  $I_{n_i}^B$  is normalized reliability importance and

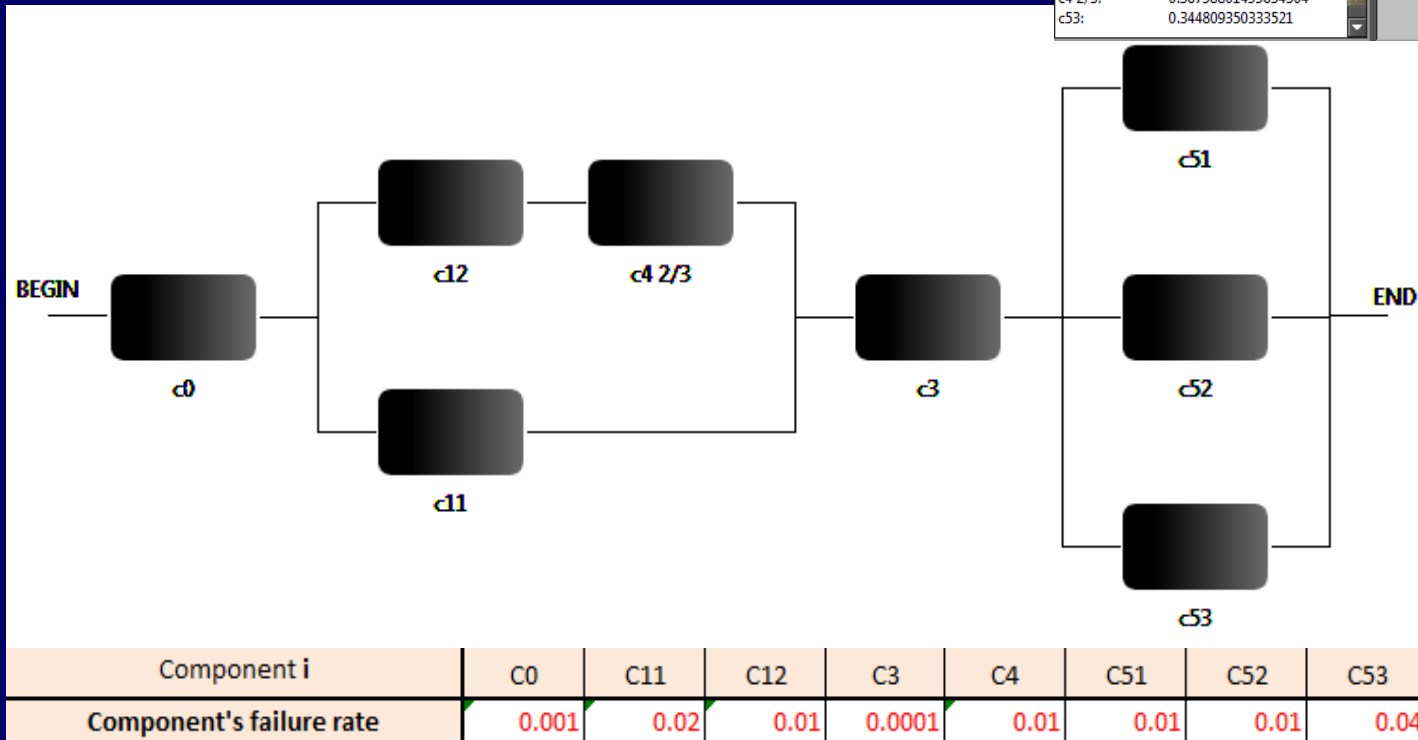
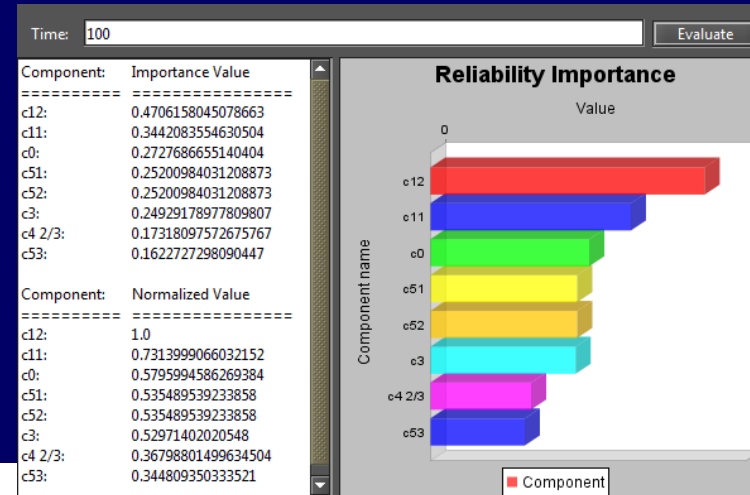
$$I_x = \max_{\forall i} \{I_i^B\}.$$

# Reliability Block Diagram

## Importance Indices

Excel

Mercury  
RI\_RBD



# Reliability Block Diagram

## ■ Importance Indices

Availability Importance

$$I_i^A = A_s(1_i, \mathbf{p}^i) - A_s(0_i, \mathbf{p}^i)$$

Normalized Availability Importance

$$I_{ni}^A = \frac{I_i^A}{I_x}$$

where  $I_{ni}^A$  is normalized availability importance and

$$I_x = \max_{\forall i} \{I_i^A\}.$$

# Reliability Block Diagram

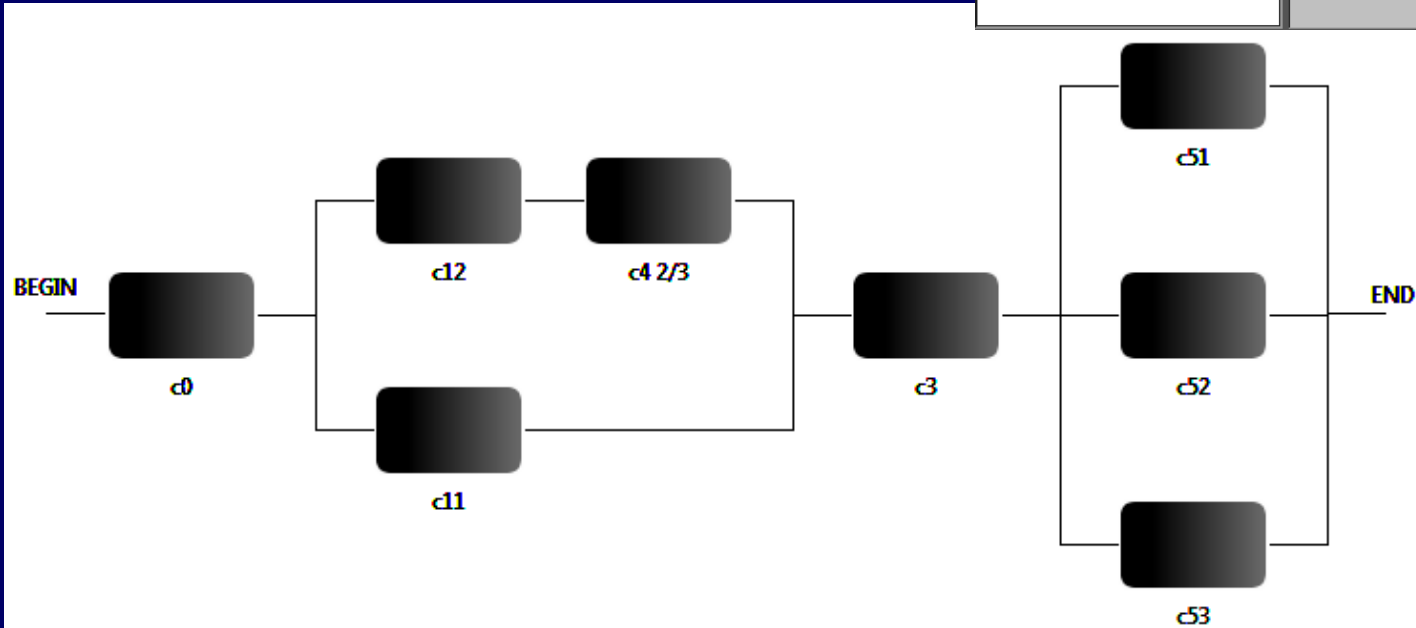
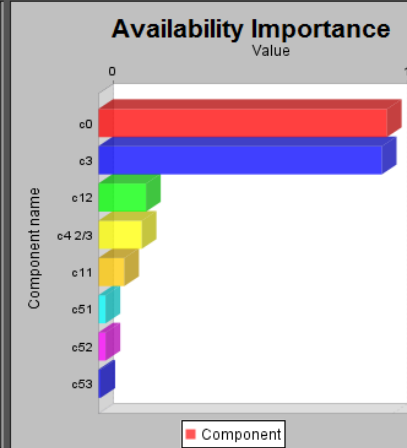
## Importance Indices

Mercury  
RI\_RBD

Component:	Importance Value
c0:	0.9805600622799777
c3:	0.9632500611809194
c12:	0.16268554669369795
c4 2/3:	0.1478977168822564
c11:	0.08874922218453907
c51:	0.025028798433878063
c52:	0.025028798433878063
c53:	0.00796370859259754

Component:	Normalized Value
c0:	1.0
c3:	0.9823529411764708
c12:	0.16591084315164226
c4 2/3:	0.15082983956981455
c11:	0.09050870578818113
c51:	0.025525002900568503
c52:	0.025525002900568503
c53:	0.008121591831999043



Component i	C0	C11	C12	C3	C4	C51	C52	C53
Component's failure rate	0.001	0.02	0.01	0.0001	0.0001	0.01	0.01	0.04
Component's repair rate	0.05	0.1	0.1	0.05	0.05	0.1	0.1	0.1

# Reliability Block Diagram

## ■ Importance Indices

Reliability and Cost Importance

$$I_i^{BC} = I_i^B \times \left( 1 - \frac{C_i}{C_{Sys}} \right)$$

Availability and Cost Importance

$$I_i^{AC} = I_i^A \times \left( 1 - \frac{C_i}{C_{Sys}} \right)$$

where  $C_i$  is the cost of component  $i$ , and  $C_{Sys}$  is the system cost.

# Reliability Block Diagram

## ■ Importance Indices

### Normalized Reliability Cost Importance

$$I_{n_i}^{BC} = \frac{I_i^{BC}}{I_x^{BC}}$$

$$I_x^{BC} = \max_{\forall i} \{I_i^{BC}\}$$

### Normalized Availability Cost Importance

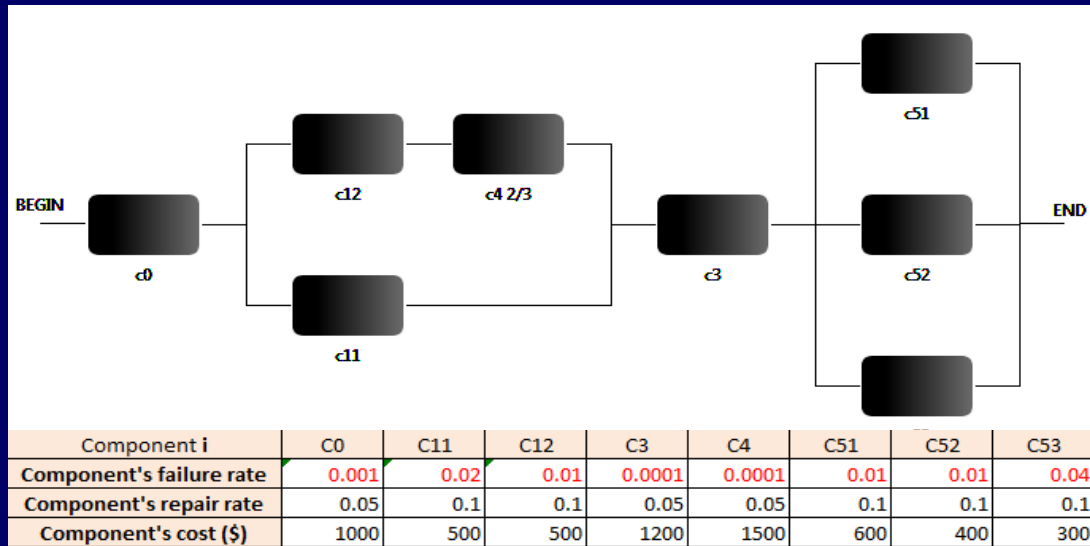
$$I_{n_i}^{AC} = \frac{I_i^{AC}}{I_x^{AC}}$$

$$I_x^{AC} = \max_{\forall i} \{I_i^{AC}\}$$

# Reliability Block Diagram

## Importance Indices

Mercury  
RI\_RBD



Component i	C0	C11	C12	C3	C4	C51	C52	C53
Component's failure rate	0.001	0.02	0.01	0.0001	0.0001	0.01	0.01	0.04
Component's repair rate	0.05	0.1	0.1	0.05	0.05	0.1	0.1	0.1
Component's cost (\$)	1000	500	500	1200	1500	600	400	300

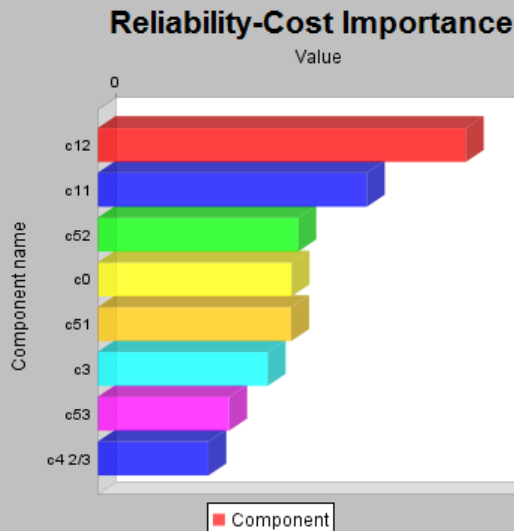
Time: 100

Evaluate

Component:	Importance Value
c12:	0.4313978207988774
c11:	0.3155243258411295
c52:	0.23520918429128282
c0:	0.22730722126170033
c51:	0.22680885628087985
c3:	0.19943343182247847
c53:	0.15415909331859246
c4 2/3:	0.12988573179506824

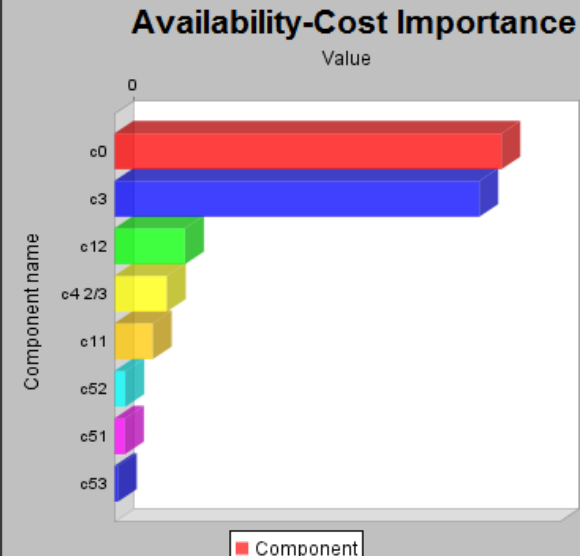
Component:	Normalized Value
c12:	1.0
c11:	0.7313999066032152
c52:	0.5452257126744737
c0:	0.5269085987517622
c51:	0.5257533657932425
c3:	0.462295872179328
c53:	0.35734787216383085
c4 2/3:	0.30108110317882775



Component:	Importance Value
c0:	0.8171333852333148
c3:	0.7706048489447356
c12:	0.14912841780255645
c4 2/3:	0.1109232876616923
c11:	0.0813534536691608
c52:	0.023360211871619525
c51:	0.022525918590490256
c53:	0.007565523162967663

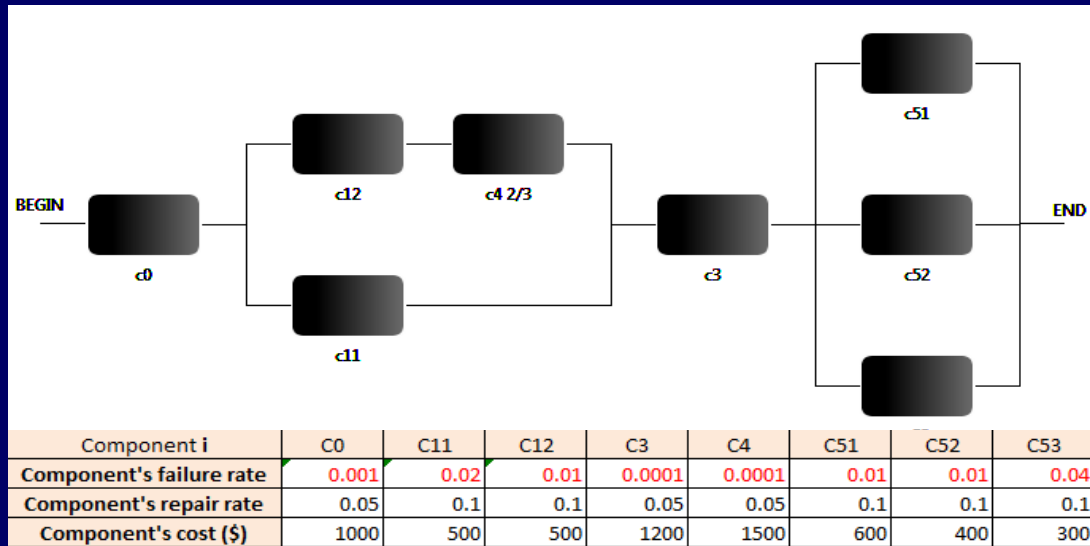
Component:	Normalized Value
c0:	1.0
c3:	0.943058823529412
c12:	0.18250192746680646
c4 2/3:	0.1357468556128331
c11:	0.09955957636699922
c52:	0.028588003248636723
c51:	0.027567003132613982
c53:	0.00925861468847891



# Reliability Block Diagram

## Importance Indices

Mercury  
RI\_RBD



Time: 100

Evaluate

Time: 100

Evaluate

Component: Importance Value

```

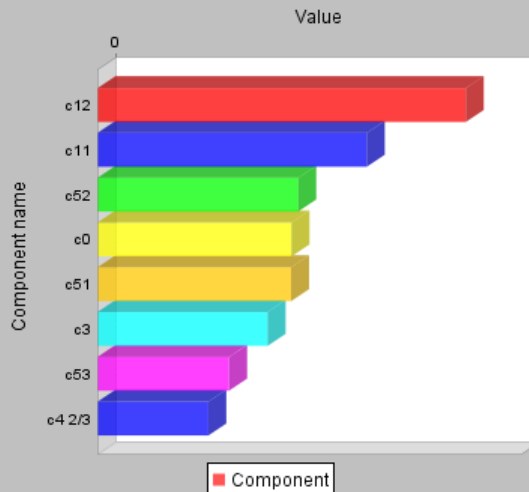
=====
c12: 0.4313978207988774
c11: 0.3155243258411295
c52: 0.23520918429128282
c0: 0.22730722126170033
c51: 0.22680885628087985
c3: 0.19943343182247847
c53: 0.15415909331859246
c4 2/3: 0.12988573179506824
    
```

Component: Normalized Value

```

=====
c12: 1.0
c11: 0.7313999066032152
c52: 0.5452257126744737
c0: 0.5269085987517622
c51: 0.5257533657932425
c3: 0.462295872179328
c53: 0.35734787216383085
c4 2/3: 0.30108110317882775
    
```

### Reliability-Cost Importance



Component: Importance Value

```

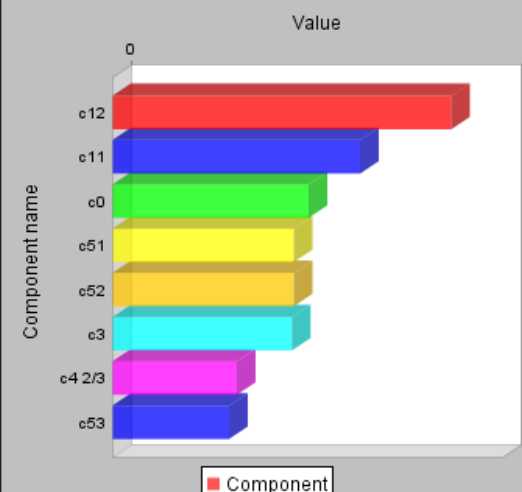
=====
c12: 0.4706158045078663
c11: 0.3442083554630504
c0: 0.2727686655140404
c51: 0.25200984031208873
c52: 0.25200984031208873
c3: 0.24929178977809807
c4 2/3: 0.17318097572675767
c53: 0.1622727298090447
    
```

Component: Normalized Value

```

=====
c12: 1.0
c11: 0.7313999066032152
c0: 0.5795994586269384
c51: 0.535489539233858
c52: 0.535489539233858
c3: 0.52971402020548
c4 2/3: 0.36798801499634504
c53: 0.344809350333521
    
```

### Reliability Importance

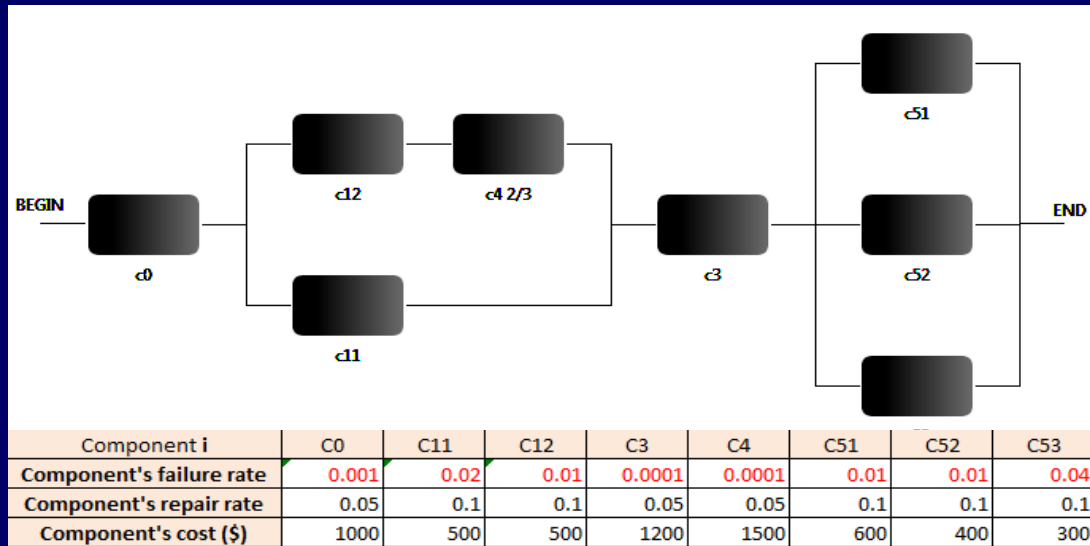




# Reliability Block Diagram

## ■ Importance Indices

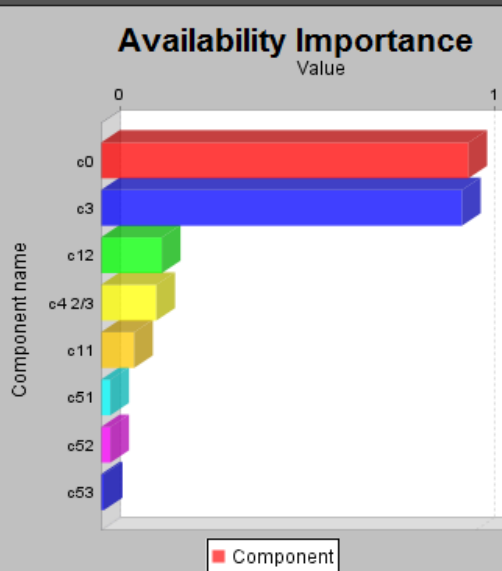
Mercury  
RI\_RBD



Component:	Importance Value
c0:	0.9805600622799777
c3:	0.9632560611809194
c12:	0.16268554669369795
c4 2/3:	0.1478977168822564
c11:	0.08874922218453907
c51:	0.025028798433878063
c52:	0.025028798433878063
c53:	0.00796370859259754

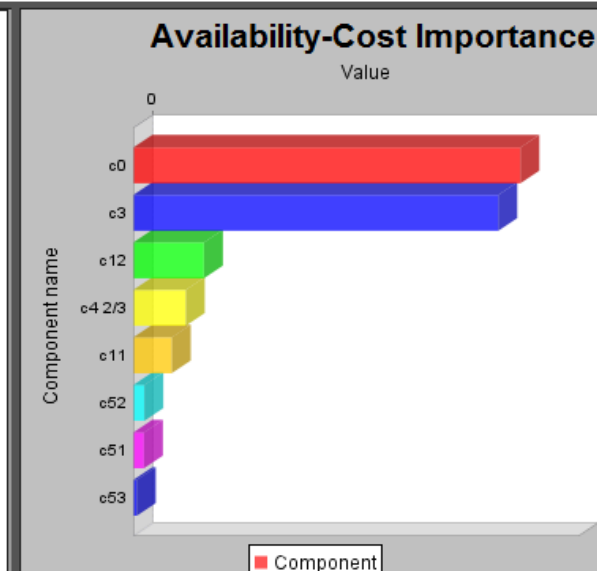
Component:	Normalized Value
c0:	1.0
c3:	0.9823529411764708
c12:	0.16591084315164226
c4 2/3:	0.15082983956981455
c11:	0.09050870578818113
c51:	0.025525002900568503
c52:	0.025525002900568503
c53:	0.008121591831999043



Component:	Importance Value
c0:	0.8171333852333148
c3:	0.7706048489447356
c12:	0.14912841780255645
c4 2/3:	0.1109232876616923
c11:	0.0813534536691608
c52:	0.023360211871619525
c51:	0.022525918590490256
c53:	0.007565523162967663

Component:	Normalized Value
c0:	1.0
c3:	0.943058823529412
c12:	0.18250192746680646
c4 2/3:	0.1357468556128331
c11:	0.09955957636699922
c52:	0.028588003248636723
c51:	0.027567003132613982
c53:	0.00925861468847891



# Fault Tree

---

- FT is failure oriented diagram.
- The system failure is represented by the TOP event.
- The TOP event is caused by lower level events (faults, component's failures etc).
- The term event is somewhat misleading, since it actually represents a state reached by event occurrences.
- The combination of events is described by logic gates.
- The most common FT elements are the TOP event, AND and OR gates, and basic events.
- The events that are not represented by combination of other events are named basic events.

# Fault Tree




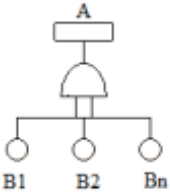
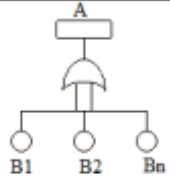
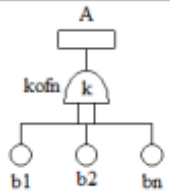
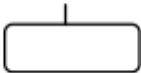
---

- Failures of individual components are assumed to be independent for easy solution.
- In FTs, the system state may be described by a Boolean function that is evaluated as true whenever the system fails.
- The system state may also be represented by a structure function, which, opposite to RBDs, represents the system failure.
- If the system has more than one undesirable state, a Boolean function (or a structure function) should be defined for representing each failure mode.
- Many extensions have been proposed which adopt other gates such as XOR, transfer and priority gates.

# Fault Tree

## Basic Symbols

Basic Symbols and their description

Symbol	Description
	TOP event represents the system failure.
	Basic event is an event that may cause a system failure.
	Basic repeated event.
	AND gate generates an event (A) if All event $B_i$ have occurred.
	OR gate generates an event (A) if at least one event $B_i$ have occurred.
	KOFN gate generates an event (A) if at least K events $B_i$ out of N have occurred.
	The comment rectangle.

# Fault Tree

## ■ Structure Function

Consider a system  $S$  composed of a set of components,  $C = \{c_i | 1 \leq i \leq n\}$ . Let the discrete random variable  $y_i(t)$  indicate the state of component  $i$ , thus:

$$y_i(t) = \begin{cases} 1 & \text{if the component } i \text{ is faulty at time } t \\ 0 & \text{if the component } i \text{ is operational at time } t \end{cases}$$

The vector  $\mathbf{y}(t) = (y_1(t), y_2(t), \dots, y_i(t), \dots, y_n(t))$  represents the state of each component of the system, and it is named state vector. The system state may be represented by a discrete random variable  $\psi(\mathbf{x}(t)) = \phi(y_1(t), y_2(t), \dots, y_i(t), \dots, y_n(t))$ , such that

$$\psi(\mathbf{y}(t)) = \begin{cases} 0 & \text{if the system is operational at time } t \\ 1 & \text{if the system is faulty at time } t \end{cases}$$

$\psi(\mathbf{y}(t))$  is named the Fault Tree structure function of the system.

# Fault Tree

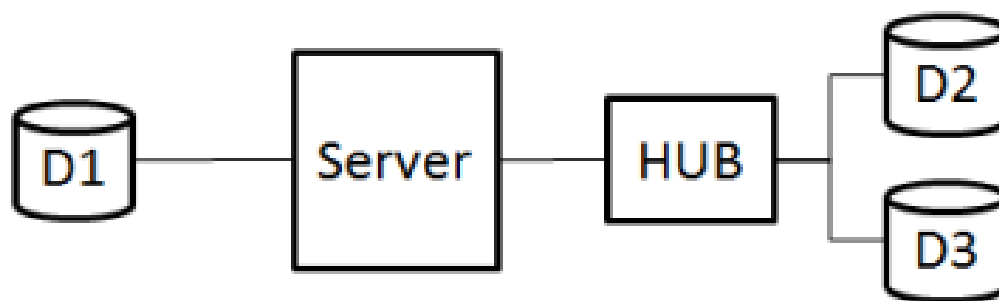
## ■ Logical Function

**FT Logic Function**  $\Psi$  denotes the counterpart that represents the FT structure function ( $\psi$ ) According to the notation previously introduced,  $s_i$  (a Boolean variable) is equivalent to  $x_i$  and  $\bar{s}_i$  represents  $1 - x_i$ . The  $\Psi(\mathbf{bs})$  (Logical function that describes conditions that cause a system failure) is the counterpart of  $\psi(\mathbf{y}(t)) = 1$  (FT structural function – represents system failures),  $\overline{\Psi(\mathbf{bs})}$  depicts of  $\psi(\mathbf{y}(t)) = 0$ ,  $\wedge$  represents  $\times$ , and  $\vee$  is the respective counterpart of  $+$ .

# Fault Tree

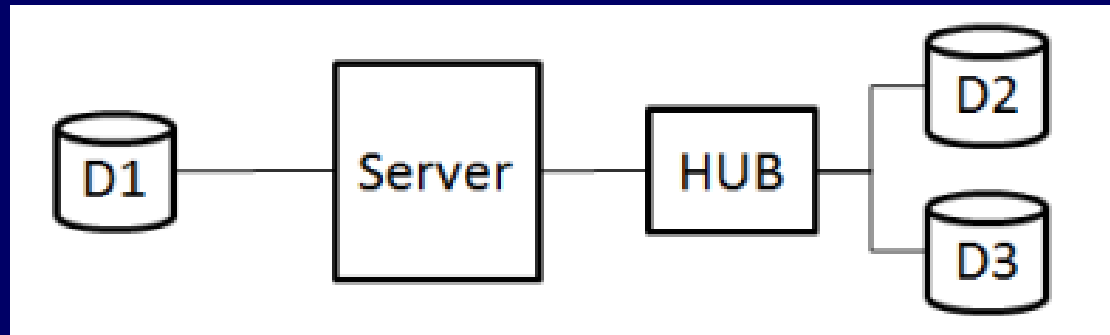
## ■ Example

Consider a system in which software applications read, write and modify the content of the storage device  $D_1$  (source). The system periodically replicates the production data (generated by the software application) of one storage device ( $D_1$ ) in two storage replicas (targets) so as to allow recovering data in the event of data loss or data corruption. The system is composed of three storage devices ( $D_1, D_2, D_3$ ), one server and hub that connects the disks  $D_2$  and  $D_3$  to the server



# Fault Tree

## ■ Example

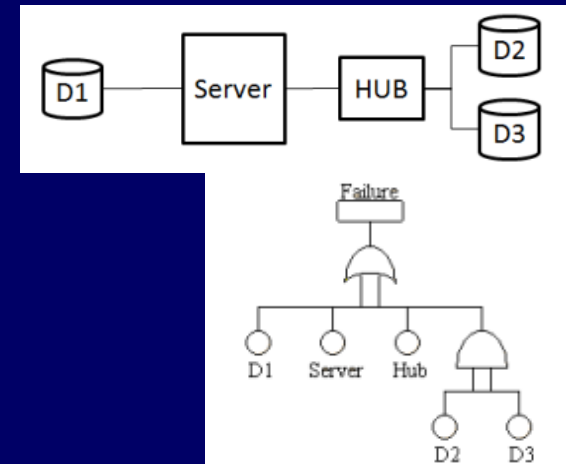


The system is considered to have failed if the hardware infrastructure does not allow the software applications to read, write or modify data on  $D_1$ , and if no data replica is available,

Hence, if  $D_1$  or the Server  
or the Hub,  
or both replica storages ( $D_2, D_3$ ) have failed.



# Fault Tree



## ■ Example

$$\Psi(\mathbf{bs}) = s_0 \vee s_1 \vee s_2 \vee (s_3 \wedge s_4),$$

$$\overline{\overline{s_0 \vee s_1 \vee s_2 \vee (s_3 \wedge s_4)}} =$$
$$\overline{s_0} \wedge \overline{s_1} \wedge \overline{s_2} \wedge \overline{(s_3 \wedge s_4)} =$$

The respective FT structure function may be expressed as

$$\psi(\mathbf{y}(t)) = [1 - (1 - y_0(t)) \times (1 - y_1(t)) \times (1 - y_2(t)) \times (1 - y_3(t) \times y_4(t))].$$

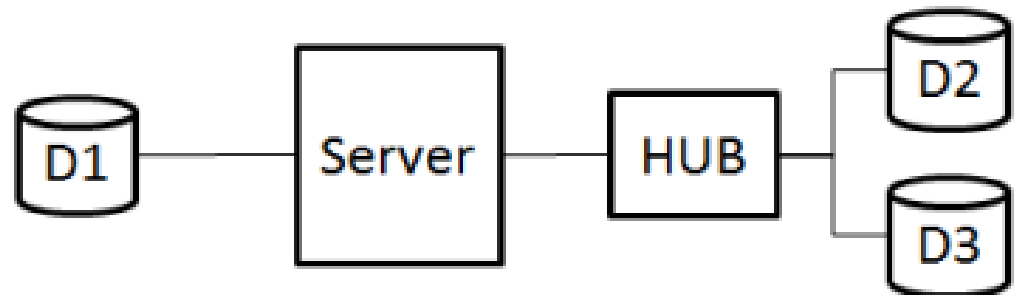
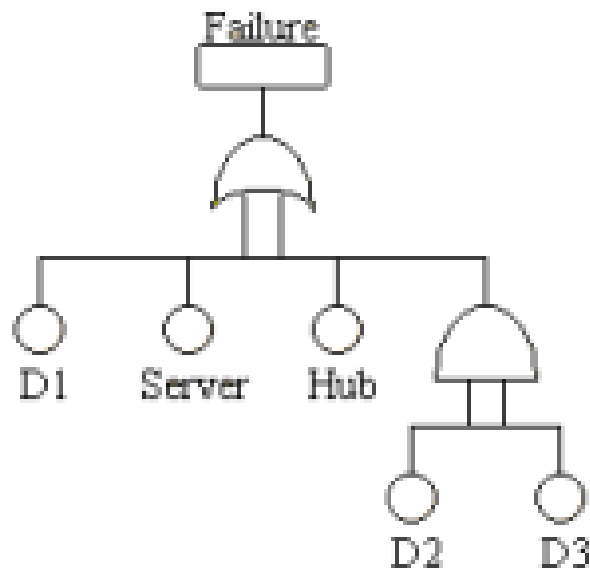
if  $y_0(t) = 1$  or  $y_1(t) = 1$  or  $y_2(t) = 1$  or  $y_3(t) = y_4(t) = 1$ , then  $\psi(\mathbf{y}(t)) = 1$ , which denotes a system failure.

# Fault Tree

## ■ Problem

Consider that the constant failure rates are  $\lambda_s = 0.00002$ ,  $\lambda_H = 0.00001$ ,  $\lambda_{D1} = 0.00008$ ,  $\lambda_{D2} = 0.00009$ , and  $\lambda_{D3} = 0.00007$ , respectively.

- Calculate the  $R(t)$  at 730h
- Calculate MTTF of system.



# Fault Tree

## ■ Problem

Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are  $\lambda_1 = 0.00001$  failures per hour,  $\lambda_2 = 0.00002$  failures per hour,  $\lambda_3 = 0.00003$  failures per hour, and  $\lambda_4 = 0.00004$  failures per hour, respectively. The sw system provides the proper service if the web services 1 or 3 are up and the web services 2 or 4 are up.

- a) Calculate MTTF of sw system.
- b) Calculate the  $R(t)$  at 730h

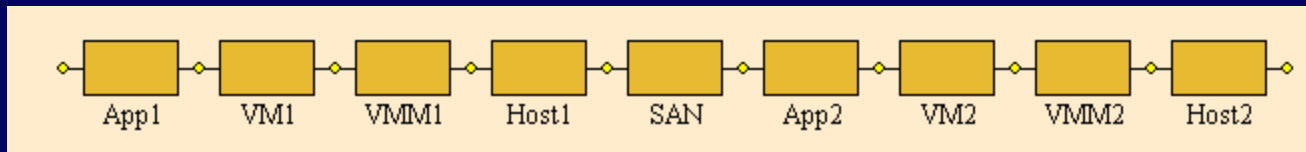


# **ANALYSIS METHODS**

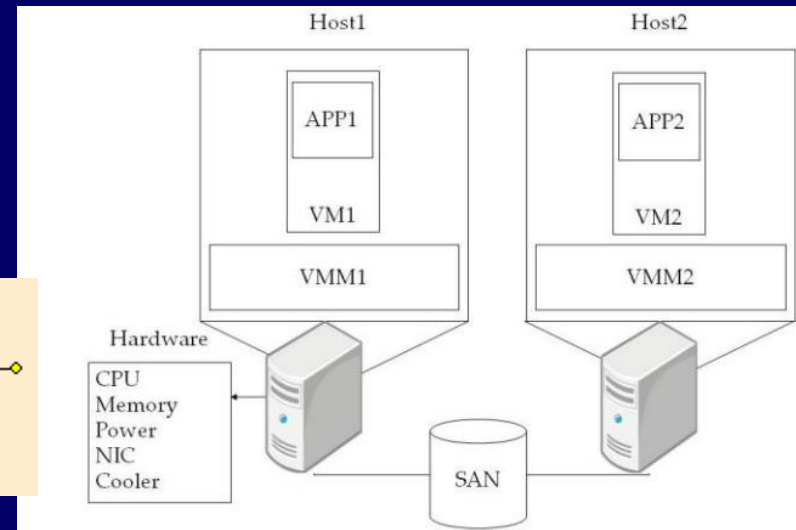
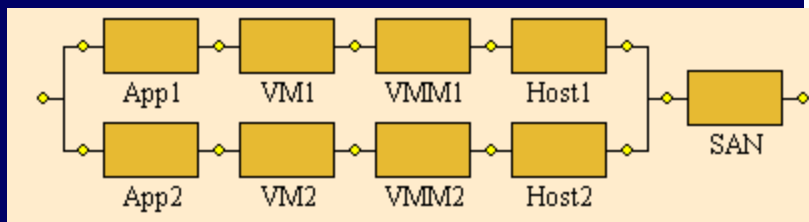
# Computing the Reliability

- What is the respective RBD?

This?



Or this?



# Analysis by Space Enumeration

## ■ The method by an example

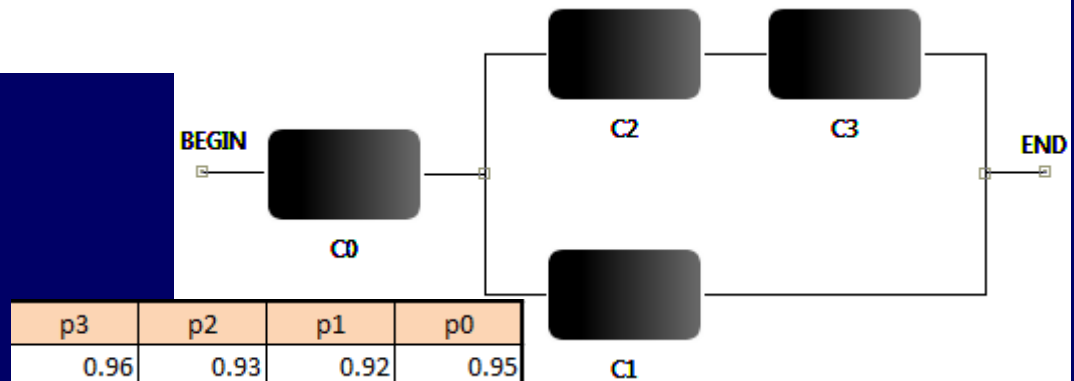
State-space enumeration method proceeds by determining the whole set of state vectors, checking for each one if the system is operational or not.

The whole set of state vectors represents all the combinations where each of the  $m$  component can be good or bad, resulting in  $2^m$  combinations.

Each of these combinations is considered as an event  $E_i$ . These events are all mutually exclusive (disjoint) and the reliability expression is simply the probability of the union of the subset of events that contain a path between the source and the target nodes.

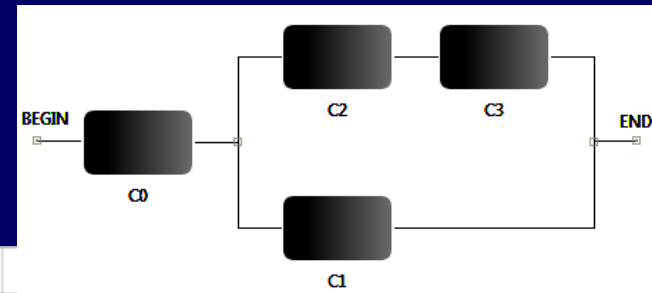
$$R_{s,t}(S) = \Pr(E_1 \cup E_2 \cup \dots \cup E_m) = \Pr(E_1) + \Pr(E_2) + \dots + P(E_m)$$

$$\text{where } E_i \cap E_j = \emptyset \forall i, \forall j, i \neq j$$



# Analysis by Space Enumeration

## ■ The method by an example



C3	C2	C1	C0	$\phi$						
0	0	0	0	0	0.04	0.07	0.08	0.05	0.00000	
0	0	0	1	0	0.04	0.07	0.08	0.95	0.00000	
0	0	1	0	0	0.04	0.07	0.92	0.05	0.00000	
0	0	1	1	0	0.04	0.07	0.92	0.95	0.00245	
0	1	0	0	0	0.04	0.93	0.08	0.05	0.00000	
0	1	0	1	0	0.04	0.93	0.08	0.95	0.00000	
0	1	1	0	0	0.04	0.93	0.92	0.05	0.00000	
0	1	1	1	0	0.04	0.93	0.92	0.95	0.03251	
1	0	0	0	0	0.96	0.07	0.08	0.05	0.00000	
1	0	0	1	0	0.96	0.07	0.08	0.95	0.00000	
1	0	1	0	0	0.96	0.07	0.92	0.05	0.00000	
1	0	1	1	0	0.96	0.07	0.92	0.95	0.05873	
1	1	0	0	0	0.96	0.93	0.08	0.05	0.00000	
1	1	0	1	0	0.96	0.93	0.08	0.95	0.06785	
1	1	1	0	0	0.96	0.93	0.92	0.05	0.00000	
1	1	1	1	0	0.96	0.93	0.92	0.95	0.78031	
										0.94185
p3	p2	p1	p0							Ps
0.96	0.93	0.92	0.95							

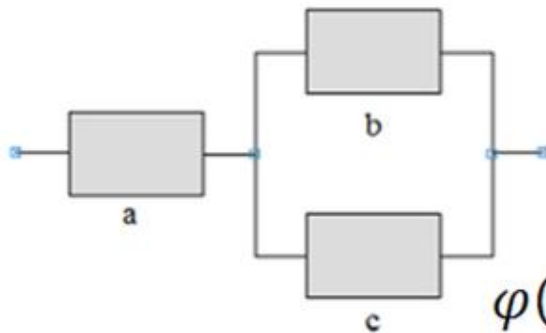
Excel

Mercury  
Ex.: Rel\_Enumeration

# Analysis by Expected Value of the Structure Function

## ■ The method by an example

Consider a system  $(C, \phi)$  composed of three blocks,  $C = \{a, b, c\}$



$$\varphi(s_a, s_b, s_c) = s_a \wedge (s_b \vee s_c) = s_a \wedge (\overline{s_b} \wedge \overline{s_c})$$

$$\phi(\mathbf{x}) = x_a \times [1 - (1 - x_b) \times (1 - x_c)]$$

$$R_S = P\{\phi(\mathbf{x}) = 1\} = E[\phi(\mathbf{x})] = E[x_a \times [1 - (1 - x_b) \times (1 - x_c)]] =$$

$$R_S = P\{\phi(\mathbf{x}) = 1\} = E[x_a] \times E[1 - (1 - x_b) \times (1 - x_c)] =$$

$$R_S = P\{\phi(\mathbf{x}) = 1\} = E[x_a] \times [1 - E[(1 - x_b)] \times E[(1 - x_c)]] =$$

$$R_S = P\{\phi(\mathbf{x}) = 1\} = E[x_a] \times [1 - (1 - E[x_b]) \times (1 - E[x_c])] =$$

$$R_S = P\{\phi(\mathbf{x}) = 1\} = p_a \times [1 - (1 - p_b) \times (1 - p_c)] = p_a \times [1 - q_b \times q_c]$$



# Analysis by Expected Value of the Structure Function

## ■ Summary of the Process

As  $x_i$  is a binary variable, thus  $x_i^k = x_i$  for any  $i$  and  $k$ ; hence  $\phi(\mathbf{x})$  is a polynomial function in which each variable  $x_i$  has degree 1.

Summarizing, the main steps for computing the system failure probability, by adopting this method are:

- i) obtain the system structure function.
- ii) remove the powers of each variable  $x_i$ ; and
- iii) replace each variable  $x_i$  by the respective  $p_i$ .

# Analysis by Expected Value of the Structure Function

## ■ Example

Consider a *2 out of 3* system represented by the RBD in figure . The logical function of the RBD presented in figure is

$$\varphi(\mathbf{bs}) = (s_1 \wedge s_2) \vee (s_1 \wedge s_3) \vee (s_2 \vee s_3)$$

Therefore

$$\varphi(\mathbf{bs}) = \overline{\overline{(s_1 \wedge s_2) \vee (s_1 \wedge s_3) \vee (s_2 \vee s_3)}}$$

$$\varphi(\mathbf{bs}) = \overline{\overline{(s_1 \wedge s_2)} \wedge \overline{\overline{(s_1 \wedge s_3)}} \wedge \overline{\overline{(s_2 \wedge s_3)}}}$$

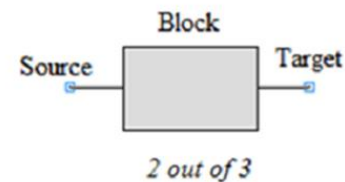
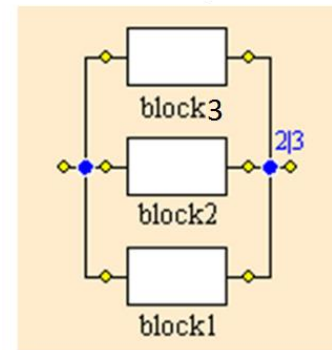
$\Leftrightarrow$

$$\phi(\mathbf{x}) = 1 - (1 - x_1x_2)(1 - x_1x_3)(1 - x_2x_3).$$

Considering that  $x_i$  is binary variable, thus  $x_i^k = x_i$  for any  $i$  and  $k$ , hence, after simplification

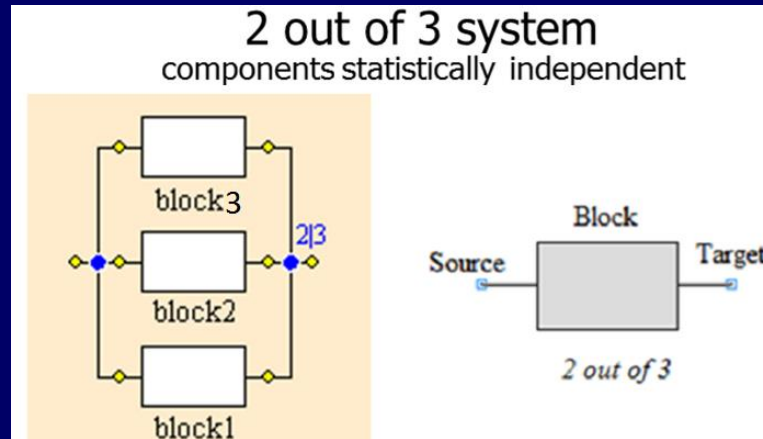
$$\phi(\mathbf{x}) = x_1x_2 + x_1x_3 + x_2x_3 - 2x_1x_2x_3.$$

2 out of 3 system  
components statistically independent



# Analysis by Expected Value of the Structure Function

## ■ Example



Since  $\phi(\mathbf{x})$  is Bernoulli random variable, its expected value is equal to  $P\{\phi(\mathbf{x}) = 1\}$ , that is,  $E[\phi(\mathbf{x})] = P\{\phi(\mathbf{x}) = 1\}$ , thus

$$\begin{aligned} P\{\phi(\mathbf{x}) = 1\} &= E[\phi(\mathbf{x})] = E[x_1x_2 + x_1x_3 + x_2x_3 - 2x_1x_2x_3] = \\ &E[x_1x_2] + E[x_1x_3] + E[x_2x_3] - 2 \times E[x_1x_2x_3] = \\ &E[x_1]E[x_2] + E[x_1]E[x_3] + E[x_2]E[x_3] - 2 \times E[x_1]E[x_2]E[x_3]. \end{aligned}$$

Therefore

$$P\{\phi(\mathbf{x}) = 1\} = p_1p_2 + p_1p_3 + p_2p_3 - 2 \times p_1p_2p_3.$$

As  $p_1 = p_2 = p_3 = p$

$$P\{\phi(\mathbf{x}) = 1\} = 3p^2 - 2p^3$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Method

This method is based on the conditional probability of the system according to the states of certain components. Consider the system structure function as depicted in

$$\phi(\mathbf{x}) = x_i \phi(1_i, \mathbf{x}) + (1 - x_i) \phi(0_i, \mathbf{x})$$

and identify the pivot component  $i$ ,

then

$$P\{\phi(\mathbf{x}) = 1\} = E[x_i \phi(1_i, \mathbf{x}) + (1 - x_i) \phi(0_i, \mathbf{x})] = E[x_i \phi(1_i, \mathbf{x})] + E[(1 - x_i) \phi(0_i, \mathbf{x})]$$

If  $x_i$  is independent, then:

$$E[x_i] \times E[\phi(1_i, \mathbf{x})] + E[(1 - x_i)] \times E[\phi(0_i, \mathbf{x})].$$

As  $x_i$  is a Bernoulli random variable, thus:

$$P\{\phi(\mathbf{x}) = 1\} = p_i \times E[\phi(1_i, \mathbf{x})] + (1 - p_i) \times E[\phi(0_i, \mathbf{x})].$$

Since  $E[\phi(1_i, \mathbf{x})] = P\{\phi(1_i, \mathbf{x}) = 1\}$  and  $E[\phi(0_i, \mathbf{x})] = P\{\phi(0_i, \mathbf{x}) = 1\}$ ,

then:

$$P\{\phi(\mathbf{x}) = 1\} = p_i \times P\{\phi(1_i, \mathbf{x}) = 1\} + (1 - p_i) \times P\{\phi(0_i, \mathbf{x}) = 1\}.$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example

Consider the system composed of three components,  $a$ ,  $b$  and  $c$ , depicted in the figure where  $\phi(x_a, x_b, x_c)$  denotes the system structure function.

As  $P\{\phi(\mathbf{x}) = 1\} = E[x_i \phi(1_i, \mathbf{x}) + (1 - x_i) \phi(0_i, \mathbf{x})]$ , then:

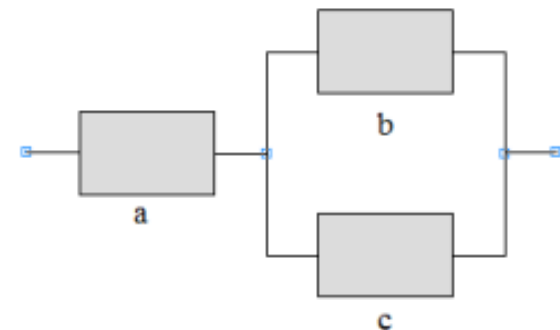
$$\begin{aligned} P\{\phi(x_a, x_b, x_c) = 1\} &= p_a \times E[\phi(1_a, x_b, x_c)] \\ &+ \\ &(1 - p_a) \times E[\phi(0_a, x_b, x_c)] \end{aligned}$$

But as  $E[\phi(0_a, x_b, x_c)] = 0$ , so:

$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a \times E[\phi(1_a, x_b, x_c)].$$

Since

$$E[\phi(1_a, x_b, x_c)] = P\{\phi(1_a, x_b, x_c) = 1\},$$



# Pivotal Decomposition, Factoring or Conditioning

## Example

Now factoring on component  $b$ ,

$$P\{\phi(1_a, x_b, x_c) = 1\} = p_b \times E[\phi(1_a, 1_b, x_c)] + (1 - p_b) \times E[\phi(1_a, 0_b, x_c)],$$

then

$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a \times [p_b \times E[\phi(1_a, 1_b, x_c)] + (1 - p_b) \times E[\phi(1_a, 0_b, x_c)]].$$

As  $E[\phi(1_a, 1_b, x_c)] = 1$ , thus:

$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a [p_b + (1 - p_b) \times E[\phi(1_a, 0_b, x_c)]].$$

Now, as we know that

$$E[\phi(1_a, 0_b, x_c)] = P\{\phi(1_a, 0_b, x_c) = 1\}, \text{ and}$$

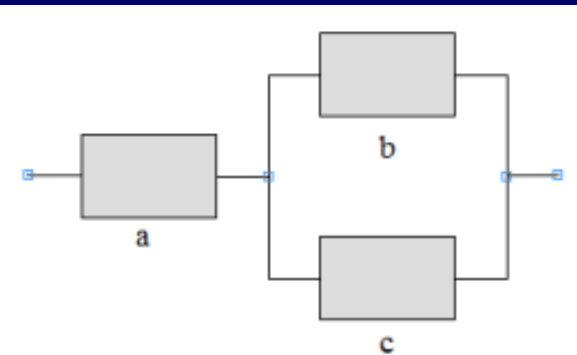
$$P\{\phi(1_a, 0_b, x_c) = 1\} = E[x_c \phi(1_a, 0_b, 1_c) + (1 - x_c) \phi(1_a, 0_b, 0_c)],$$

then

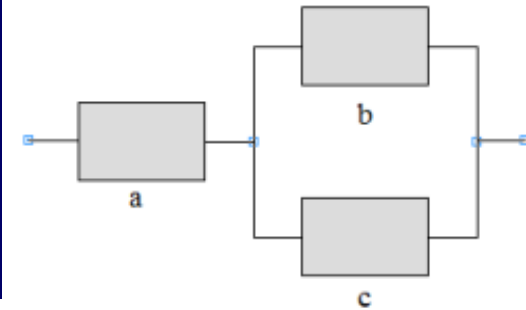
$$E[\phi(1_a, 0_b, x_c)] = E[x_c] E[\phi(1_a, 0_b, 1_c)] + E[(1 - x_c)] E[\phi(1_a, 0_b, 0_c)],$$

thus

$$E[\phi(1_a, 0_b, x_c)] = p_c \times E[\phi(1_a, 0_b, 1_c)] + (1 - p_c) \times E[\phi(1_a, 0_b, 0_c)].$$



# Pivotal Decomposition, Factoring or Conditioning



## ■ Example

As  $E[\phi(1_a, 0_b, 1_c)] = P\{\phi(1_a, 0_b, 1_c) = 1\} = 1$   
and  $E[\phi(1_a, 0_b, 0_c)] = P\{\phi(1_a, 0_b, 0_c) = 1\} = 0$ ,  
then

$$E[\phi(1_a, 0_b, x_c)] = p_c.$$

Therefore:

$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a [p_b + (1 - p_b) \times p_c] =$$

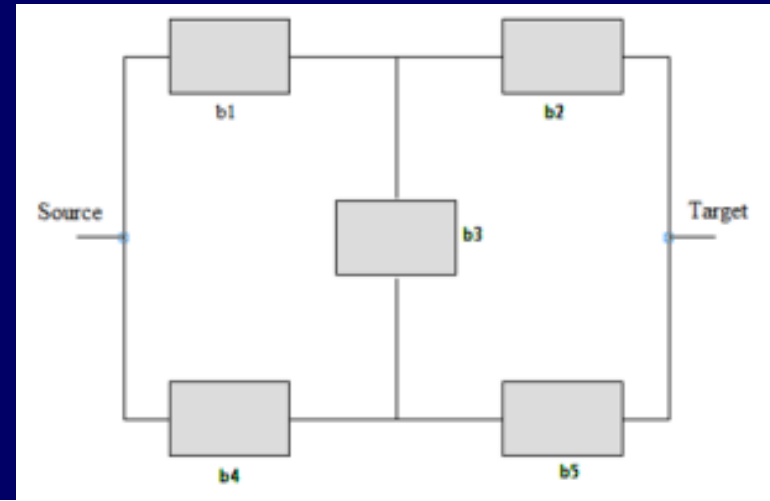
$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a p_b + p_a p_c (1 - p_b),$$

which is

$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a [1 - (1 - p_b)(1 - p_c)].$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure



$$\phi(\mathbf{x}) = x_i \phi(1_i, \mathbf{x}) + (1 - x_i) \phi(0_i, \mathbf{x})$$

Factoring on  $b_3$

$$\phi(\mathbf{x}) = x_3 \phi(1_3, \mathbf{x}) + (1 - x_3) \phi(0_3, \mathbf{x})$$

$$P\{\phi(\mathbf{x}) = 1\} = E[x_3 \phi(1_3, \mathbf{x}) + (1 - x_3) \phi(0_3, \mathbf{x})] =$$

$$P\{\phi(\mathbf{x}) = 1\} = E[x_3 \phi(1_3, \mathbf{x})] + E[(1 - x_3) \phi(0_3, \mathbf{x})] =$$

By independency

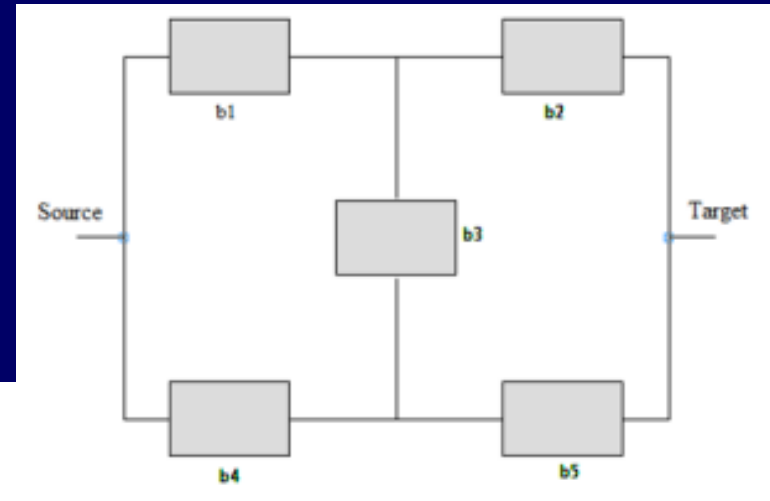
$$P\{\phi(\mathbf{x}) = 1\} = E[x_3] E[\phi(1_3, \mathbf{x})] + E[(1 - x_3)] E[\phi(0_3, \mathbf{x})] =$$

$$P\{\phi(\mathbf{x}) = 1\} = p_3 E[\phi(1_3, \mathbf{x})] + (1 - p_3) E[\phi(0_3, \mathbf{x})] =$$



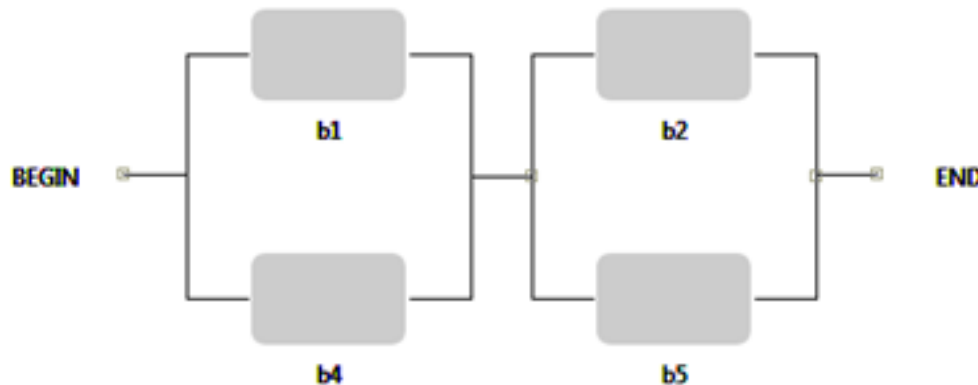
# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure



If  $x_3 = 1 \Rightarrow p_3 = 1$ , then:

Configuration 1:



$$P\{\phi(\mathbf{x}) = 1\} = E[\phi(1_3, \mathbf{x})] = P\{\phi(1_3, \mathbf{x}) = 1\}$$

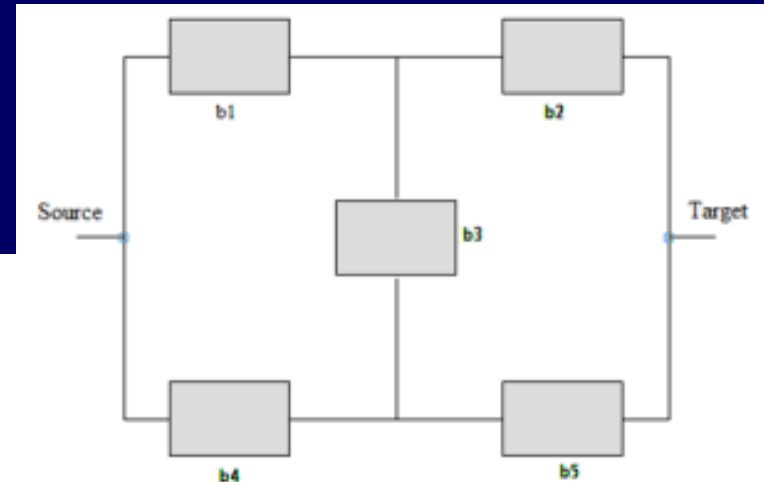
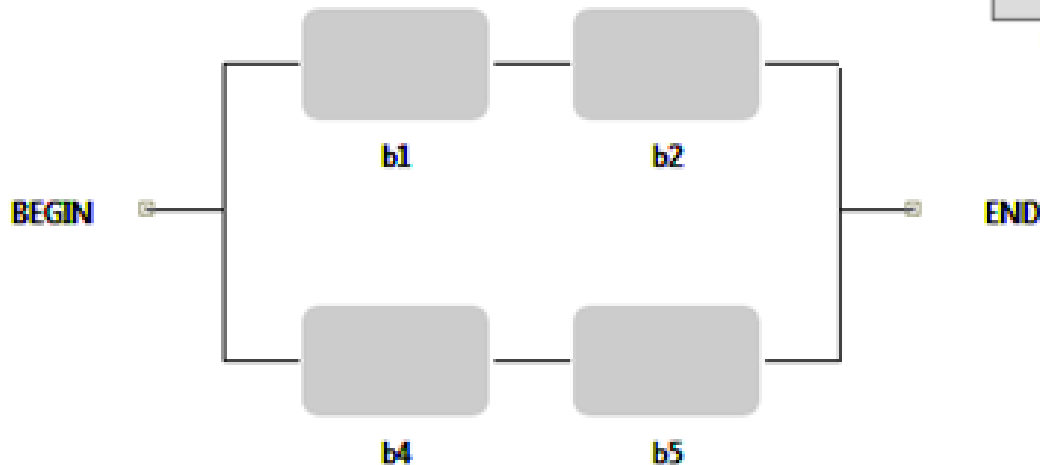
$$P\{\phi(1_3, \mathbf{x}) = 1\} = (1 - (1 - p_1)(1 - p_4)) \times (1 - (1 - p_2)(1 - p_5))$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure

If  $x_3 = 0 \Rightarrow p_3 = 0$ , then:

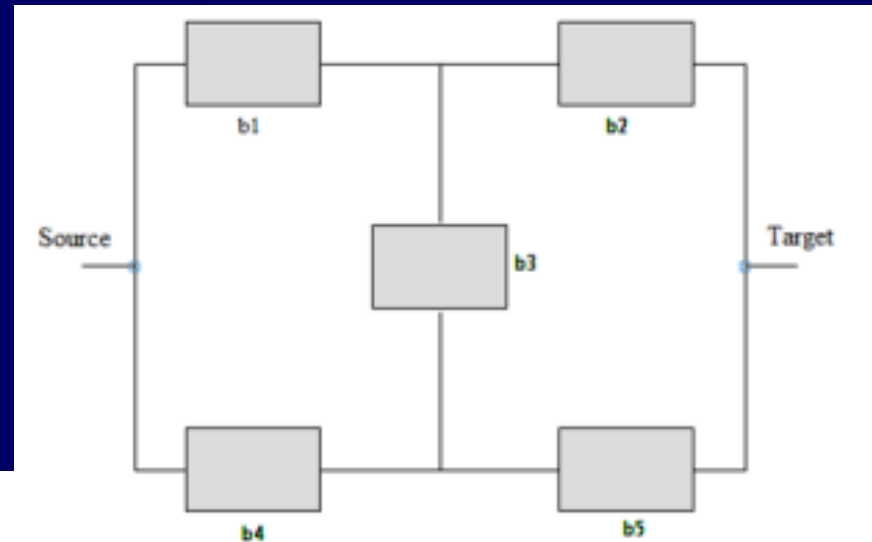
Configuration 2:



$$P\{\phi(\mathbf{x}) = 1\} = E[\phi(0_3, \mathbf{x})] = P\{\phi(0_3, \mathbf{x}) = 1\}$$
$$P\{\phi(0_3, \mathbf{x}) = 1\} = (1 - (1 - p_1 p_2)(1 - p_4 p_5))$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure



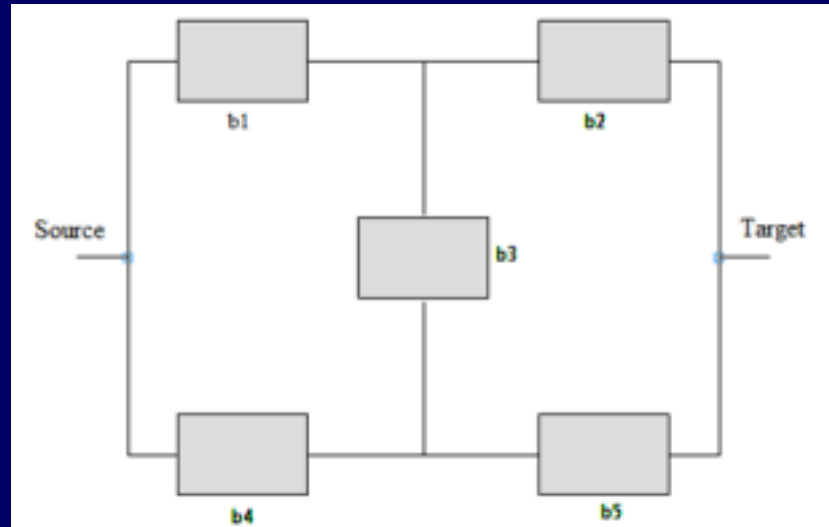
Therefore:

$$P\{\phi(\mathbf{x}) = 1\} = p_3 \times P\{\phi(1_3, \mathbf{x}) = 1\} + (1 - p_3) \times P\{\phi(0_3, \mathbf{x}) = 1\}$$

$$P\{\phi(\mathbf{x}) = 1\} = p_3 \times \left( \left( (1 - (1 - p_1)(1 - p_4)) \times (1 - (1 - p_2)(1 - p_5)) \right) \right) + (1 - p_3) \left( \left( (1 - (1 - p_1)(1 - p_4)) \times (1 - (1 - p_2)(1 - p_5)) \right) \right)$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure



$$\begin{aligned}
 R_{bridge}(t) = & e^{-(\lambda_2 + \lambda_3 + \lambda_4)t} - e^{-(\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5)t} + e^{-(\lambda_1 + \lambda_3 + \lambda_5)t} \\
 & + 2e^{-\sum_i \lambda_i t} - e^{-(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5)t} - e^{-(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)t} - e^{-(\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5)t} \\
 & + e^{-(\lambda_1 + \lambda_2)t} + e^{-(\lambda_4 + \lambda_5)t} - e^{-(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_5)t}
 \end{aligned}$$

# Pivotal Decomposition, Factoring or Conditioning

## ■ Example – Bridge Structure

$$MTTF = \int_0^{\infty} R_{bridge}(t) dt$$

$$MTTF = \frac{1}{\lambda_1 + \lambda_2} + \frac{1}{\lambda_4 + \lambda_5} + \frac{1}{\lambda_2 + \lambda_3 + \lambda_4} + \frac{2}{\sum_{i=1}^5 \lambda_i}$$

$$- \frac{1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} - \frac{1}{\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5} - \frac{1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5}$$

$$- \frac{1}{\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5} - \frac{1}{\lambda_1 + \lambda_3 + \lambda_5} - \frac{1}{\lambda_1 + \lambda_2 + \lambda_4 + \lambda_5}$$

# Reductions

---

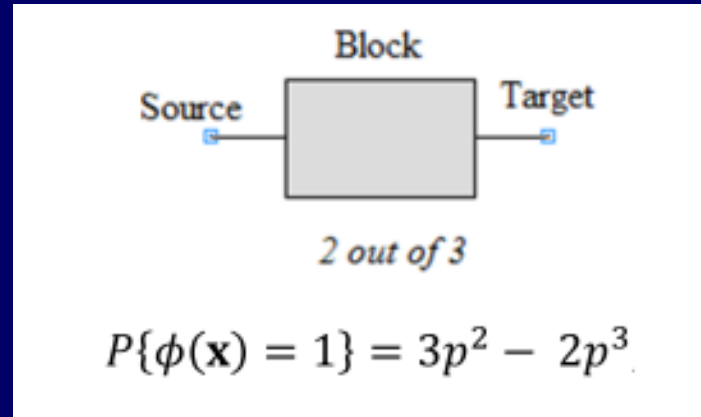
The dependability evaluation of complex system structures might be conducted iteratively by indentifying series, parallel, *k out of n* and *bridge* subsystems, evaluating each of those subsystems, and then reducing each subsystem to one respective equivalent block.

This process may be iteratively applied to the resultant structures until a single block results.

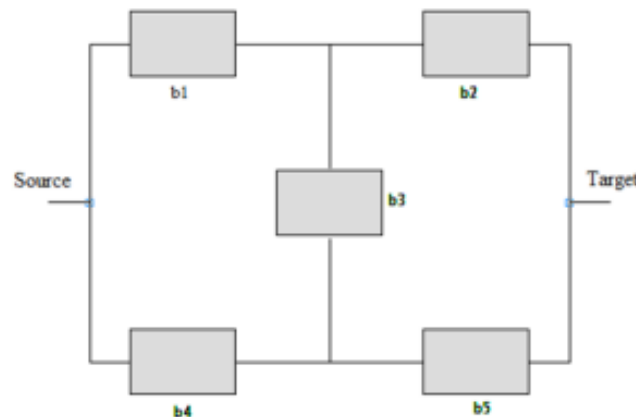


# Reductions

## ■ 2 out of 3 reduction



## ■ Bridge reduction



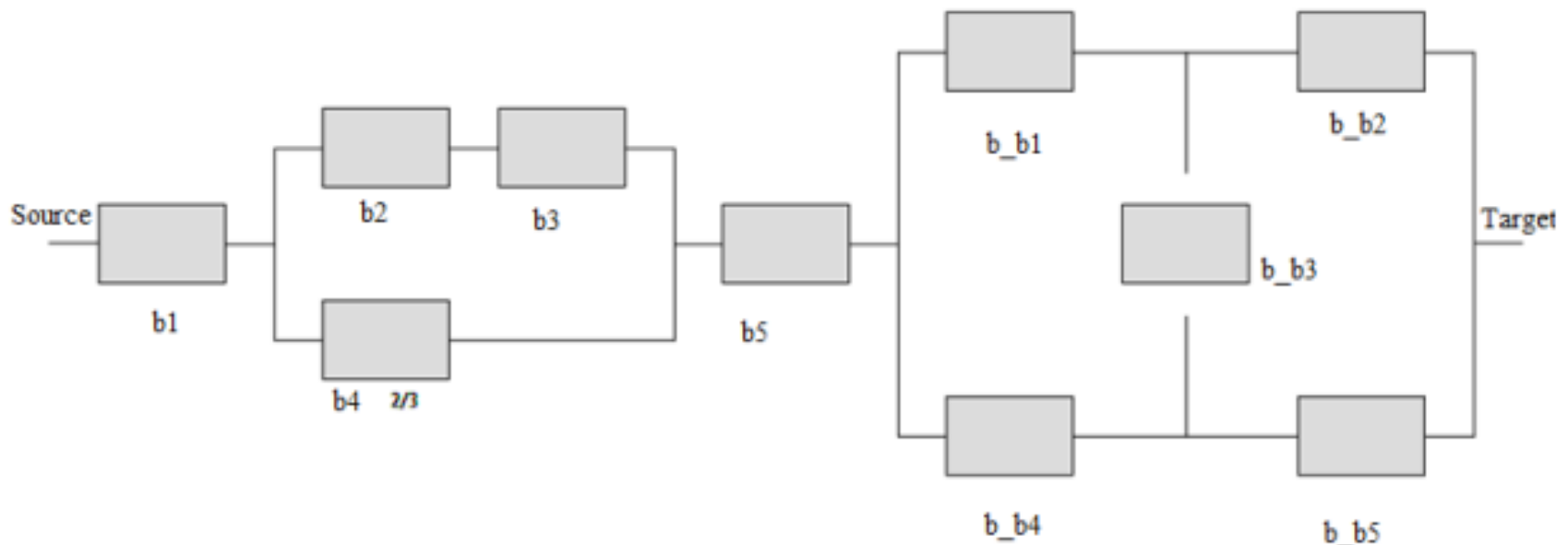
$$P\{\phi(\mathbf{x}) = 1\} = p_3 \times P\{\phi(1_3, \mathbf{x}) = 1\} + (1 - p_3) \times P\{\phi(0_3, \mathbf{x}) = 1\}$$



# Reductions

## ■ Example

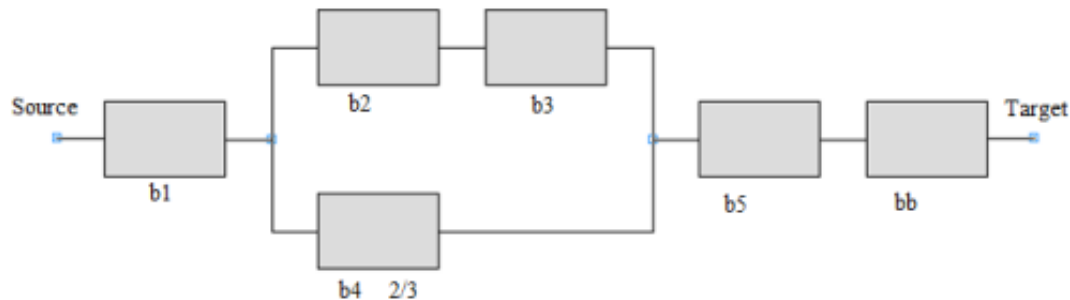
Consider a system composed of four basic blocks ( $b_1, b_2, b_3, b_5$ ), one *2 out of 3* and one bridge structure. The three components of the *2 out of 3* block are equivalent, that is, the failure probability of each component is the same ( $p_4$ ). The failure probabilities of components  $b_1, b_2, b_3, b_5$  and the failure probability of the bridge structure are  $p_{b1}, p_{b2}, p_{b3}, p_{b4}$  and  $p_{b5}$ , respectively.



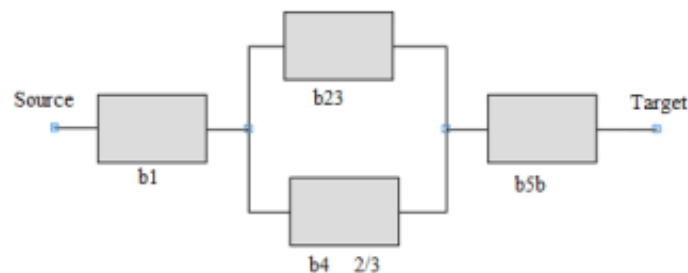
# Reductions

## ■ Example

The *2 out of 3* structure can be represented one equivalent block whose reliability is  $3p^2 - 2p^3$ . The bridge structure can be transformed into one component,  $b_b$ , whose failure probability is  $p_{bb} = (1 - (1 - p_{b1}p_{b2})(1 - p_{b4}p_{b5})(1 - p_{b1}p_{b3}p_{b5})(1 - p_{b2}p_{b3}p_{b4}))$ .



After that, two series reductions may be applied, one reducing blocks  $b_2$  and  $b_3$  into block  $b_{23}$ ; and a second that combines blocks  $b_5$  and  $b_b$  and reduces it to the block  $b_{5b}$ . The reliability of block  $b_{23}$  is  $p_{23} = p_2 \times p_3$ , and the block reliability of block  $b_{5b}$  is  $p_{5b} = p_5 \times [(1 - (1 - p_{b1}p_{b2})(1 - p_{b4}p_{b5})(1 - p_{b1}p_{b3}p_{b5})(1 - p_{b2}p_{b3}p_{b4}))]$ .



# Reductions

## ■ Example

Now a parallel reduction may be applied to merge blocks  $b_{23}$  and  $b_4$ .

The block  $b_{234}$  represents the block  $b_{23}$  and  $b_4$  composition, whose reliability is  $p_{234} = 1 - (1 - p_2 \times p_3) \times (1 - 3p^2 - 2p^3)$ .



Finally, a final series reduction may be applied to RBD and one block RBD is generated, whose reliability is

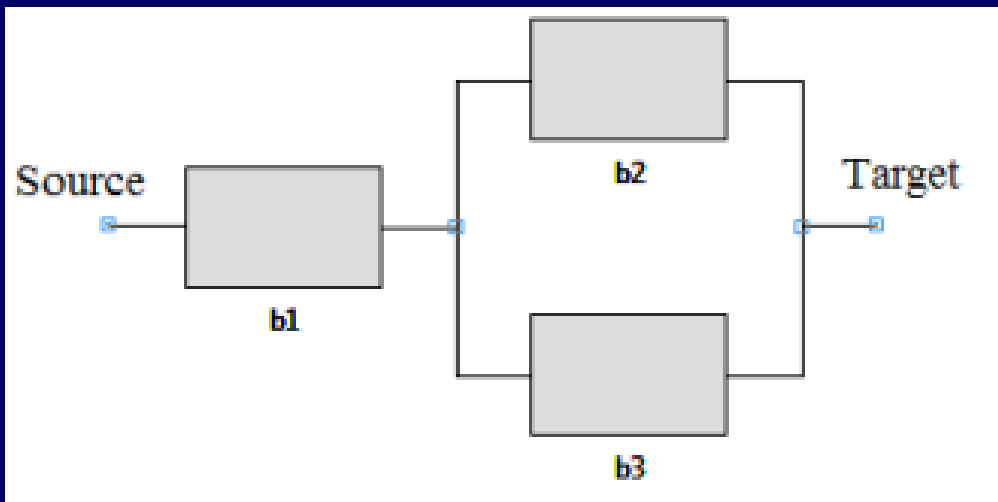
$$p_{12345b} = p_1 \times [1 - (1 - p_2 \times p_3) \times (1 - 3p^2 - 2p^3)] \times [p_5 \times [(1 - (1 - p_{b1}p_{b2}))(1 - p_{b4}p_{b5})(1 - p_{b1}p_{b3}p_{b5})(1 - p_{b2}p_{b3}p_{b4})]]].$$



# Computation Based on Minimal Paths and Minimal Cuts

## ■ Path and Minimal Path

Consider a system  $S$  with  $n$  components and its structure function  $\phi(\mathbf{x})$ , where  $SCS = \{c_1, c_2, \dots, c_n\}$  is the set of components. A state vector  $\mathbf{x}$  is named a **path vector** if  $\phi(\mathbf{x}) = 1$ , and the respective set of operational components is defined as **path set**. More formally, the respective path set of a state vector is defined by  $PS(\mathbf{x}) = \{c_i | \phi(\mathbf{x}) = 1, x_i = 1, c_i \in SCS\}$ . A path vector  $\mathbf{x}$  is called **minimal path vector** if  $\phi(\mathbf{x}) = 1$ , for any  $\mathbf{y} < \mathbf{x}$ , and the respective path set is named **minimal path set**, that is  $MPS(\mathbf{x}) = \{c_i | c_i \in PS(\mathbf{x}), \phi(\mathbf{x}) = 1 \forall \mathbf{y} < \mathbf{x}\}$ .



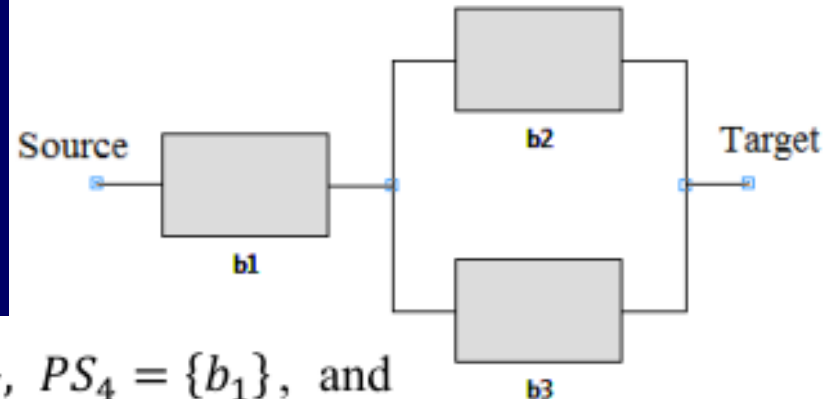
$PS_1$  is a minimal path set  
 $PS_2$  is a minimal path set  
 $PS_3$  is not minimal

$PS_1 = \{b_1, b_2\}$ ,  $PS_2 = \{b_1, b_3\}$  and  $PS_3 = \{b_1, b_2, b_3\}$  are path sets

# Computation Based on Minimal Paths and Minimal Cuts

## ■ Cut and Minimal Cut

A state vector  $\mathbf{x}$  is named a **cut vector** if  $\phi(\mathbf{x}) = 0$ , and the respective set of faulty components is defined as **cut set**. Therefore,  $CS(\mathbf{x}) = \{c_i | \phi(\mathbf{x}) = 0, x_i = 0, c_i \in SCS\}$ . A cut vector  $\mathbf{x}$  is called **minimal cut vector** if  $\phi(\mathbf{x}) = 1$ , for any  $\mathbf{y} > \mathbf{x}$ , and the respective path set is named **minimal cut set**, that is  $MCS(\mathbf{x}) = \{c_i | c_i \in CS(\mathbf{x}), \phi(\mathbf{x}) = 1 \forall \mathbf{y} > \mathbf{x}\}$ .



$PS_1 = \{b_1, b_2\}$ ,  $PS_2 = \{b_1, b_3\}$ ,  $PS_3 = \{b_1, b_2, b_3\}$ ,  $PS_4 = \{b_1\}$ , and  
 $PS_5 = \{b_2, b_3\}$

$PS_4$  is a minimal cut set,

$PS_5$  is a minimal cut set,

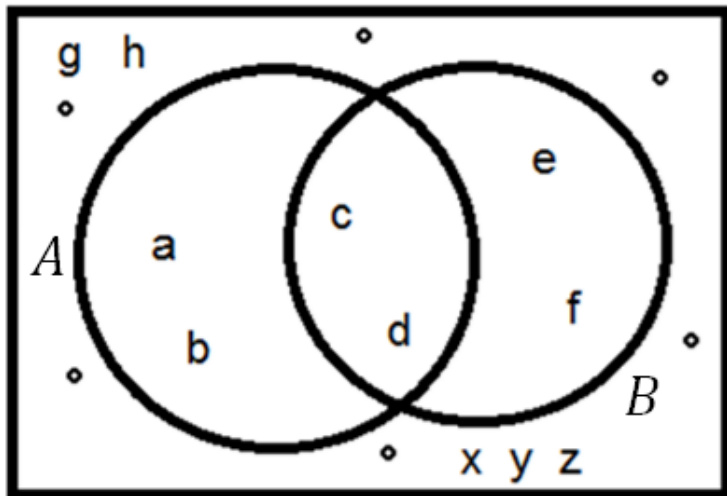
The same is not true for  $PS_1$ ,  $PS_2$ , and  $PS_3$ .

# Sum-of-Disjoint-Products (SDP) method

$$A = \{a, b, c, d\}$$

$$B = \{c, d, e, f\}$$

$$\Omega = \{a, b, c, d, e, f, g, \dots, z\}$$



$$A \cup B = A \cup (A^c \cap B)$$

$$A^c = \{e, f, g, h, \dots, z\}$$

$$A^c \cap B = \{e, f, g, h, \dots, z\} \cap \{c, d, e, f\} =$$

$$A^c \cap B = \{e, f\}$$

$$A \cup B = \{a, b, c, d\} \cup \{e, f\} = \{a, b, c, d, e, f\}$$

$\Leftrightarrow$

$$A \cup B = \{a, b, c, d, e, f\} = \{a, b, c, d\} \cup \{c, d, e, f\}$$

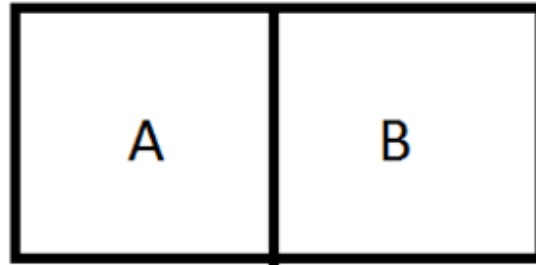
Now, consider  $P(A \cup B) = P(A \cup (A^c \cap B))$

As  $A \cap (A^c \cap B) = \emptyset$ ,

since  $A$  and  $(A^c \cap B)$  are disjoint, then

$$P(A \cup B) = P(A) + P(A^c \cap B)$$

# Sum-of-Disjoint-Products (SDP) method



**Disjoint Terms: Addition Law** The addition law of probabilities is the underlying justification for the SDP method. If two or more events have no elements in common, the probability that at least one of the events will occur is the sum of the probabilities of the individual events. If two events  $A$  and  $B$  have elements in common, the union of these two events,  $A \cup B$ , may be expressed as the union of event  $A$  with event  $B$ , where  $A^c$  denotes the complement of  $A$ . Then we have the following equation for evaluation of the probability of  $A \cup B$ :

$$\Pr(A \cup B) = \Pr(A) + \Pr(A^c B).$$

# Sum-of-Disjoint-Products (SDP) method

---

Similarly with three events  $A$ ,  $B$ , and  $C$ , we have

$$\Pr(A \cup B \cup C) = \Pr(A) + \Pr(A^c B) + \Pr(A^c B^c C).$$

With  $n$  events  $A_1, A_2, \dots, A_n$ , we have

$$\Pr(A_1) + \Pr(A_1^c A_2) + \Pr(A_1^c A_2^c A_3) + \dots + \Pr(A_1^c \dots A_{n-1}^c A_n)$$



# Sum-of-Disjoint-Products (SDP) method

Considering a system composed of three independent components  $b_1, b_2$  and  $b_3$ , where the components failure probabilities are  $p_1, p_2$  and  $p_3$ , respectively.

The respective RBD logical function is:

$$\varphi(s_1, s_2, s_3) = s_1 \wedge (s_2 \vee s_3)$$

Then define all minimal paths:

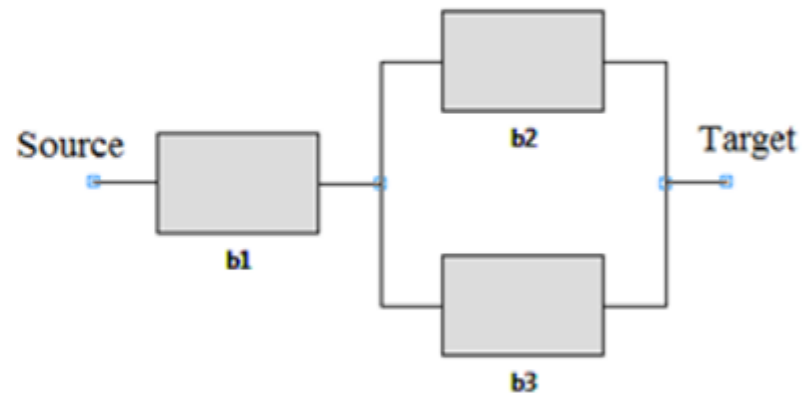
$$\varphi(s_1, s_2, s_3) = (s_1 \wedge s_2) \vee (s_1 \wedge s_3)$$

The minimal paths are:

$$PS_1 = \{b_1, b_2\} \text{ and } PS_2 = \{b_1, b_3\}.$$

(and  $PS_4 = \{b_1\}$ , and  $PS_5 = \{b_2, b_3\}$  are minimal cut sets)

$$\varphi(s_1, s_2, s_3) \Leftrightarrow \phi(x_1, x_2, x_3) = 1$$



# Sum-of-Disjoint-Products (SDP) method

The respective RBD logical function is:

$$\varphi(s_1, s_2, s_3) = s_1 \wedge (s_2 \vee s_3)$$

Then define all minimal paths:

$$\varphi(s_1, s_2, s_3) = (s_1 \wedge s_2) \vee (s_1 \wedge s_3)$$

The minimal paths are:

$$PS_1 = \{b_1, b_2\} \text{ and } PS_2 = \{b_1, b_3\}.$$

(and  $PS_4 = \{b_1\}$ , and  $PS_5 = \{b_2, b_3\}$

are minimal cut sets)

$$\varphi(s_1, s_2, s_3) \Leftrightarrow \phi(x_1, x_2, x_3) = 1$$

Therefore:

$$P(\varphi(s_1, s_2, s_3)) = P(\phi(x_1, x_2, x_3) = 1).$$

Then, applying the SDP formula:

$$P(A \cup B) = P(A) + P(A^c \cap B)$$

$$P(PS_1 \cup PS_2) = P(PS_1) + P(PS_1^c \cap PS_2).$$

Every component within the minimal path  $PS_1$  must properly work for  $PS_1$  being responsible for  $\phi(x_1, x_2, x_3) = 1$

# Sum-of-Disjoint-Products (SDP) method

So,

$$PS_1 \stackrel{eq}{\Leftrightarrow} s_1 \wedge s_2 \text{ and}$$

$$P(PS_1) = P(s_1 \wedge s_2).$$

As  $PS_1 \stackrel{eq}{\Leftrightarrow} s_1 \wedge s_2$ , thus:

$$PS_1^c \stackrel{eq}{\Leftrightarrow} \overline{s_1 \wedge s_2}$$

Since  $PS_2 \stackrel{eq}{\Leftrightarrow} s_1 \wedge s_3$ , thus:

$$PS_1^c \cap PS_2 \stackrel{eq}{\Leftrightarrow} \overline{s_1 \wedge s_2} \wedge s_1 \wedge s_3$$

Therefore:

$$P(PS_1^c \cap PS_2) = P(\overline{s_1 \wedge s_2} \wedge s_1 \wedge s_3)$$

$$PS_1^c \cap PS_2 \stackrel{eq}{\Leftrightarrow} \overline{s_1 \wedge s_2} \wedge s_1 \wedge s_3$$

So,

$$P(PS_1^c \cap PS_2) = P(\overline{s_1 \wedge s_2} \wedge s_1 \wedge s_3)$$

Then:

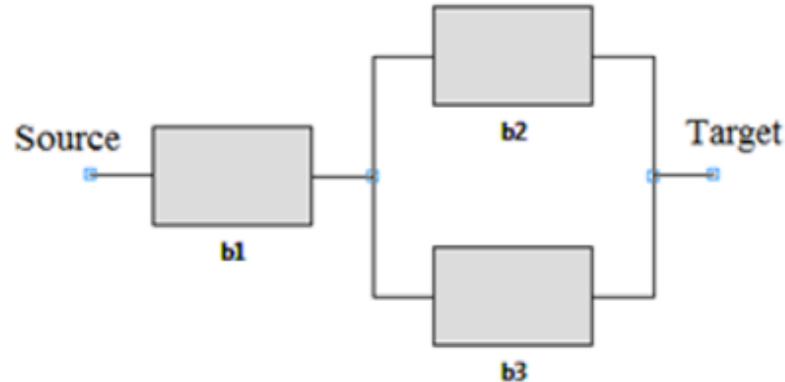
$$P(\varphi(s_1, s_2, s_3)) = P(s_1 \wedge s_2) + P(\overline{s_1 \wedge s_2} \wedge s_1 \wedge s_3) =$$

$$P(s_1 \wedge s_2) + P((\overline{s_1} \vee \overline{s_2}) \wedge s_1 \wedge s_3) =$$

$$P(s_1 \wedge s_2) + P((\overline{s_1} \wedge s_1 \wedge s_3) \vee (\overline{s_2} \wedge s_1 \wedge s_3)) =$$

$$P(s_1 \wedge s_2) + P(\overline{s_2} \wedge s_1 \wedge s_3) =$$

# Sum-of-Disjoint-Products (SDP) method



Now, consider  $P(x_1) = P(x_2) = P(x_3) = 0.9$

$$P(\phi(x_1, x_2, x_3) = 1) = 0.9 \times 0.9 + (1 - 0.9) \times 0.9 \times 0.9 =$$

$$P(\phi(x_1, x_2, x_3) = 1) = 0.891$$

It is worth noting that:

$$P(\phi(x_1, x_2, x_3) = 1) =$$

$$P(x_1) \times (1 - (1 - P(x_2))) \times (1 - P(x_2)) =$$

$$0.9 \times (1 - (1 - 0.9)) \times (1 - 0.9) = 0.891$$

---

# **State-space based models**

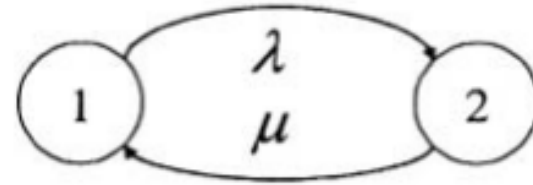
# CTMC

## Single Component System Availability Model

Consider a system with one component or when the system is considered as a black-box. This systems may have a normal functioning (1) state and a failed state (2).

System

If the TTF and TTR are exponentially distributed with rate  $\lambda$  and  $\mu$ , respectively, the CTMC that represents the system availability model is



A simple 2-state CTMC

$$\pi_1(0) = 1$$

$$\pi_1(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

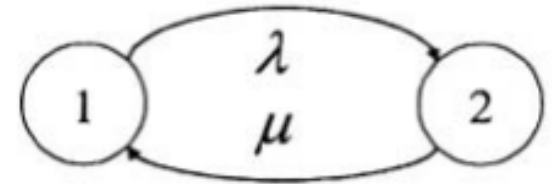
$$\pi_2(t) = \frac{\lambda}{\lambda + \mu} + \frac{\mu}{\lambda + \mu} e^{-(\lambda + \mu)t}$$

$$\pi_1(t) + \pi_2(t) = 1$$

$$A(t) = \pi_1(t)$$

Instantaneous availability

# CTMC



A simple 2-state CTMC

Single Component System Availability Model

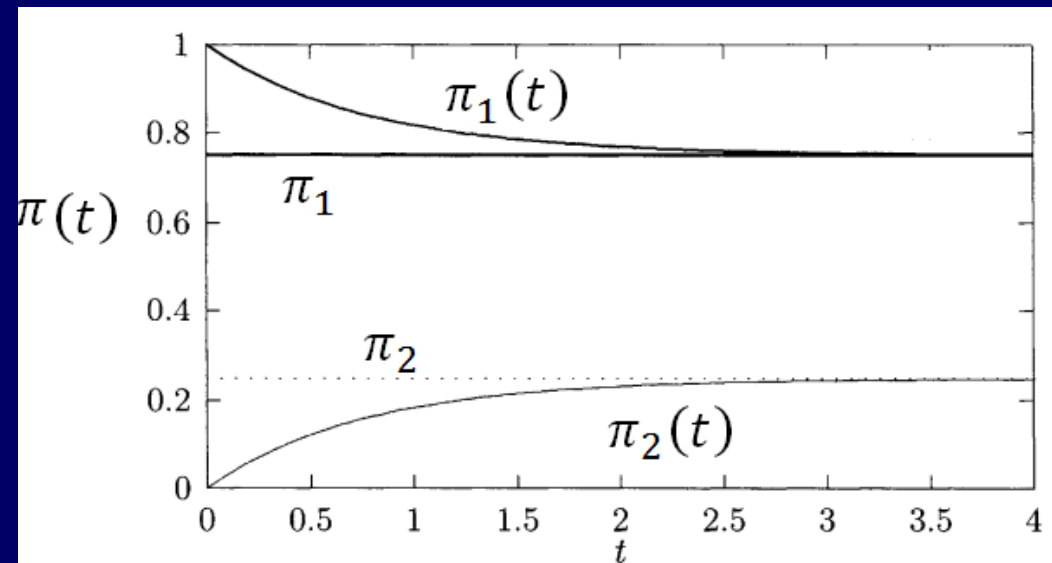
$$\pi_1(t) = \pi_1 = \frac{\mu}{\lambda + \mu}, t \rightarrow \infty$$

$$\pi_2(t) = \pi_2 = \frac{\lambda}{\lambda + \mu}, t \rightarrow \infty$$

$A = \pi_1$   
Steady state availability

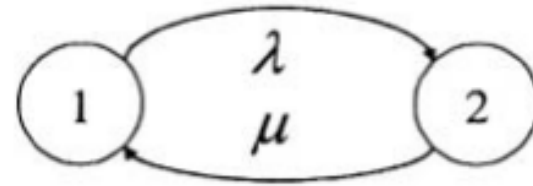
$$Q = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

Figure shows the transient and steady-state behavior of the 2-state CTMC for  $3\lambda = \mu = 1$ .



# CTMC

## Single Component System Availability Model



A simple 2-state CTMC

$$\pi_1(t) = \pi_1 = \frac{\mu}{\lambda + \mu}, t \rightarrow \infty$$

$$\pi_2(t) = \pi_2 = \frac{\lambda}{\lambda + \mu}, t \rightarrow \infty$$

$$A = \pi_1$$

Steady state availability

$$DT = (1 - A) \times T$$

$T$  – time period

Downtime

$$DT = (1 - A) \times 8760h$$

hours in a year

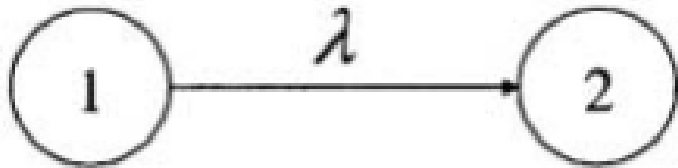
$$DT = (1 - A) \times 525,600 \text{ min}$$

minutes in a year



# CTMC

## Single Component System Reliability Model



$$\pi_1(0) = 1$$

$$\pi_1(t) = e^{-\lambda t}$$

$$\pi_1(t) + \pi_2(t) = 1$$

$$R(t) = \pi_1(t)$$

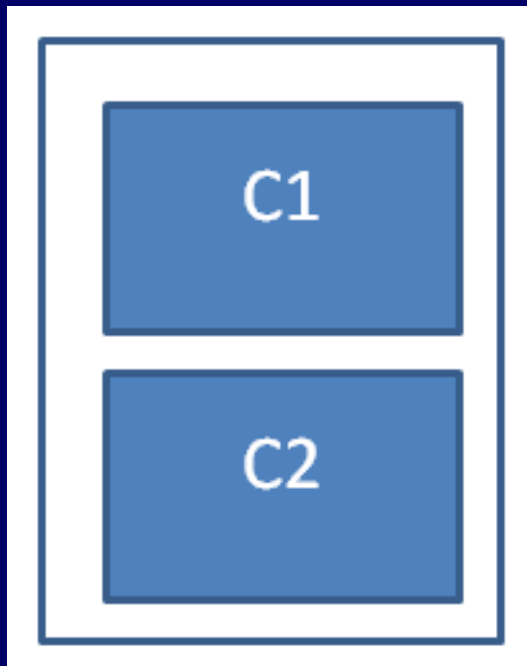
Reliability

$$R(t) = \pi_1(t) = 0, t \rightarrow \infty$$

$$MTTF = \int_0^{\infty} R(t)dt = \int_0^{\infty} e^{-\lambda t} dt = \frac{1}{\lambda}$$

# CTMC

Two Component System - Hot Standby  
Availability Model



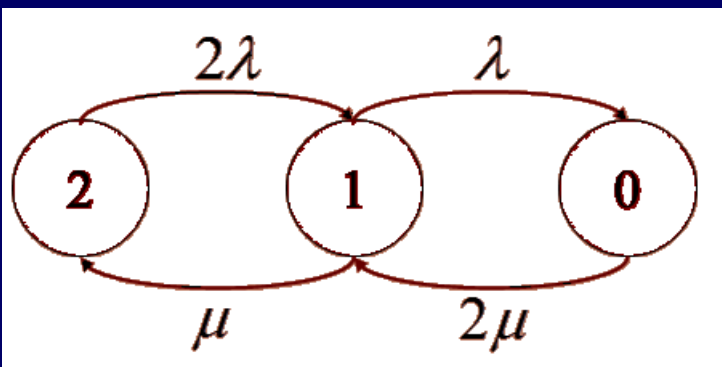
Two-component parallel redundant system with the same repair rate  $\mu$  and the same failure rate for both components is ( $\lambda$ ).

When both the components fail, the system fails.

# CTMC

2 Machines each with  
two repair facilities

Two Component System - Hot Standby  
Availability Model



Non-shared (independent) repair

$A(t) = \pi_2(t) + \pi_1(t)$   
Instantaneous availability

$$A = \pi_2 + \pi_1 = \frac{\mu(2\lambda + \mu)}{(\lambda + \mu)^2}$$

Steady state availability

$$DT = (1 - A) \times T$$

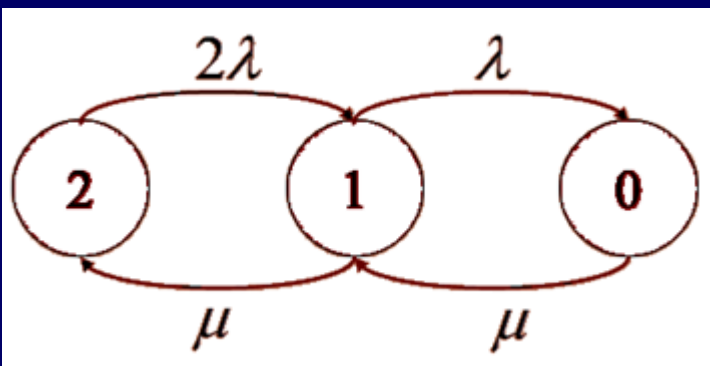
$T$  – time period

Downtime

# CTMC

2 Machines each with  
only one repair facility

Two Component System - Hot Standby  
Availability Model



Shared repair

$$A(t) = \pi_2(t) + \pi_1(t)$$

Instantaneous availability

$$A = \pi_2 + \pi_1 = \frac{\mu(2\lambda + \mu)}{2\lambda^2 + 2\lambda\mu + \mu^2}$$

Steady state availability

$$DT = (1 - A) \times T$$

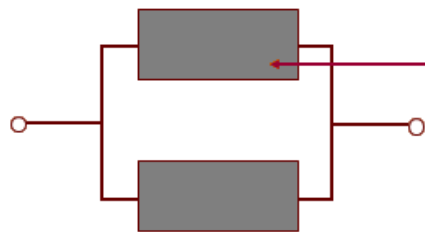
$T$  – time period

Downtime

# CTMC

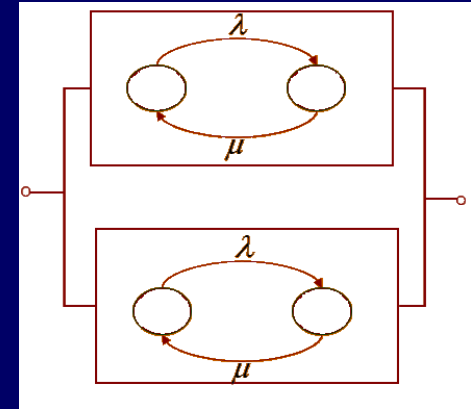
## Two Component System - Hot Standby Availability Model

Non-shared case can be modeled & solved using a RBD or a FTREE but shared case needs the use of Markov chains.



$$A = \frac{\mu}{\lambda + \mu}$$

$$A_{sys} = 1 - \left(1 - \frac{\mu}{\lambda + \mu}\right)^2$$



$$A_{ss1} = \frac{\mu}{\lambda + \mu}$$

$$UA_{ss1} = \frac{\lambda}{\lambda + \mu}$$

$$A_{ss2} = \frac{\mu}{\lambda + \mu}$$

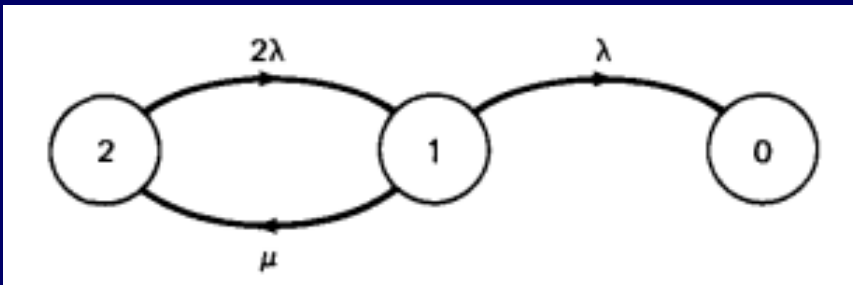
$$UA_{ss2} = \frac{\lambda}{\lambda + \mu}$$

$$A_{sys} = 1 - \left(1 - \frac{\mu}{\lambda + \mu}\right)^2 = 1 - \left(\frac{\lambda}{\lambda + \mu}\right)^2$$

$$A_{sys} = \frac{\mu(2\lambda + \mu)}{(\lambda + \mu)^2}$$

# CTMC

## Two Component System - Hot Standby Reliability Model



Some authors erroneously claim that reliability models do not admit repair.

$$R(t) = 1 - \pi_0(t)$$

$$MTTF = \int_0^{\infty} R(t) dt = \frac{3}{2\lambda} + \frac{\mu}{2\lambda^2}$$

# CTMC

M Machines each with  
only one repair facility

## ■ Example – Availability model

M similar machines independent repair facility.  
Hot Standby

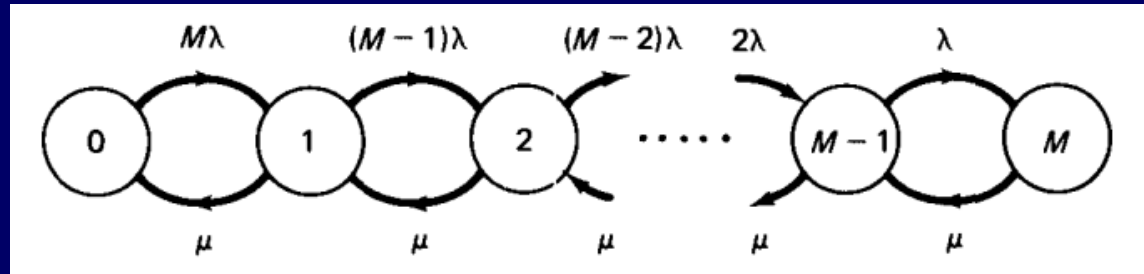
Failure rate of each machine is  $\lambda$

Repair rate is  $\mu$

Generalization of the two-component system  
Model with shared repair facility

$$A_{sys} = 1 - \frac{\rho^M M!}{\sum_{k=0}^M \rho^k \frac{M!}{(M-k)!}}$$

$$\rho = \frac{\lambda}{\mu}$$



# CTMC

M Machines each with only M repair facilities

## ■ Example – Availability model

M similar machines independent repair facility.

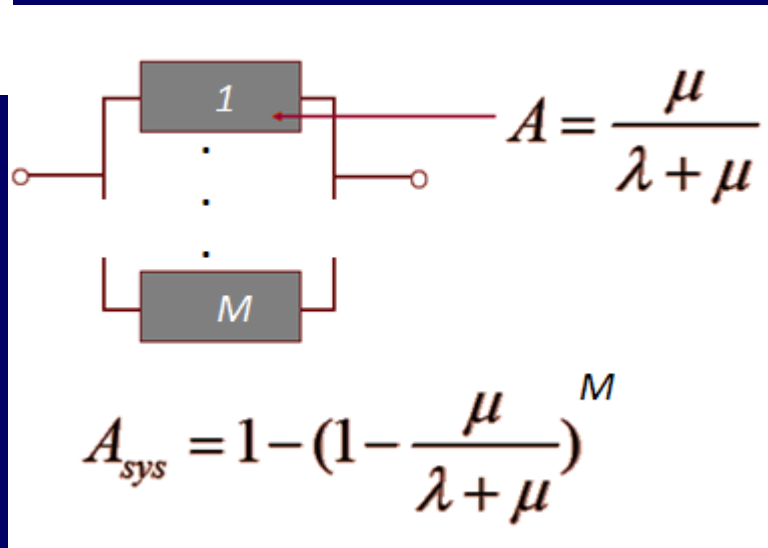
Hot Standby

Failure rate of each machine is  $\lambda$

Repair rate is  $\mu$

Generalization of the two-component system model with independent repair facility

$$A_{sys} = 1 - \left(\frac{\rho}{1 + \rho}\right)^M$$
$$\rho = \frac{\lambda}{\mu}$$



System availability is then computed using a combinatorial approach



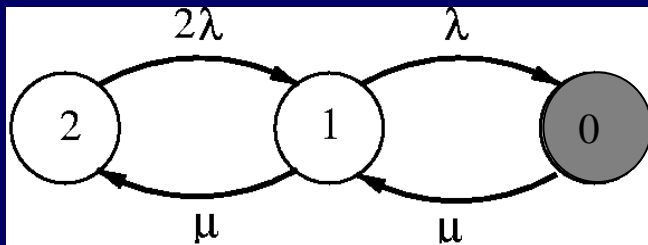
# CTMC

## Hot Standby

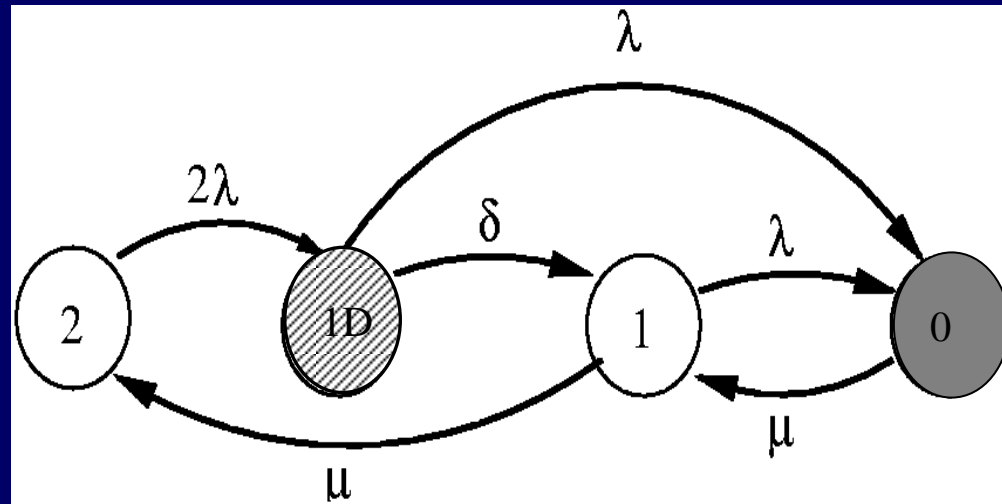
2-equal component availability model without perfect switching (with finite detection delay)

## Hot Standby

2-equal component availability model with perfect switching



$$A_{ss} = 1 - \pi_0$$

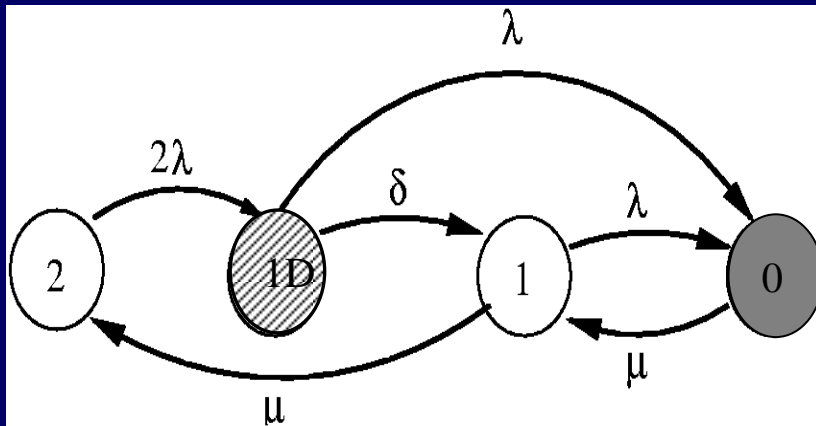


$$A = \pi_2 + \pi_1 + r\pi_{1D}$$

# CTMC

## Hot Standby

2-equal component availability model  
without perfect switching (with finite  
detection delay)



$$A = \pi_2 + \pi_1 + r\pi_{1D}$$

We can model this by assigning a reward rate  $e^{-\delta \times t_{th}}$  to the state 1D, 1 to the state 0 and 0 to the remaining states

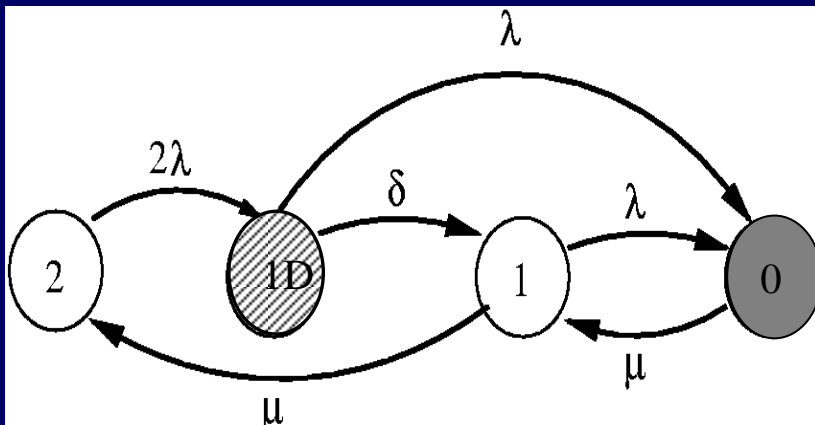
Then Unavailability is given by

$$U(\delta, t_{th}) = \pi_0 + e^{-\delta t_{th}} \pi_{1D}$$

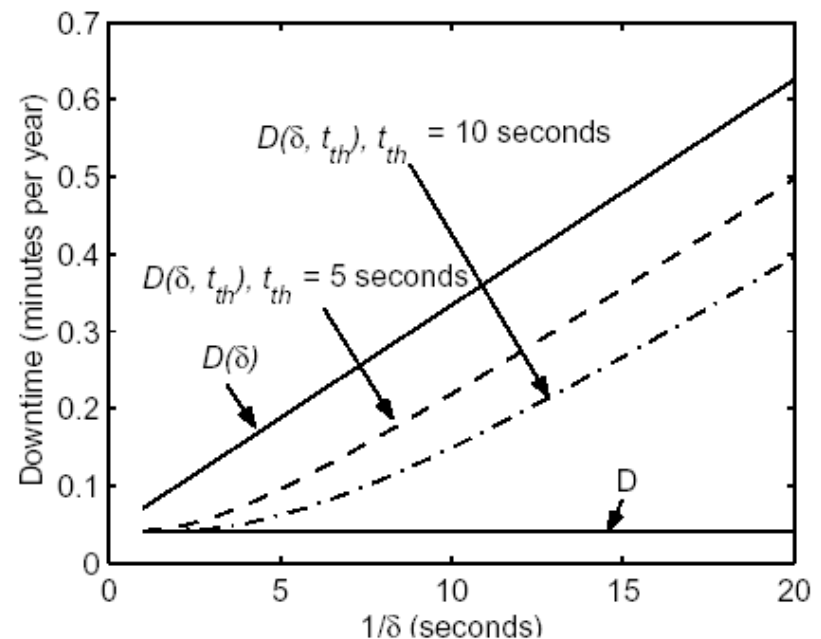
# CTMC

## Example

Plot of downtime  $D(\delta)$ ,  $D(\delta, t_{th})$ , and  $D$  (for 3 state model without state 1D) as functions of  $1/\delta$  (in seconds) for  $1/\lambda = 10,000$  h and  $1/\mu = 2$  h.



$$U(\delta, t_{th}) = \pi_0 + e^{-\delta t_{th}} \pi_{1D}$$



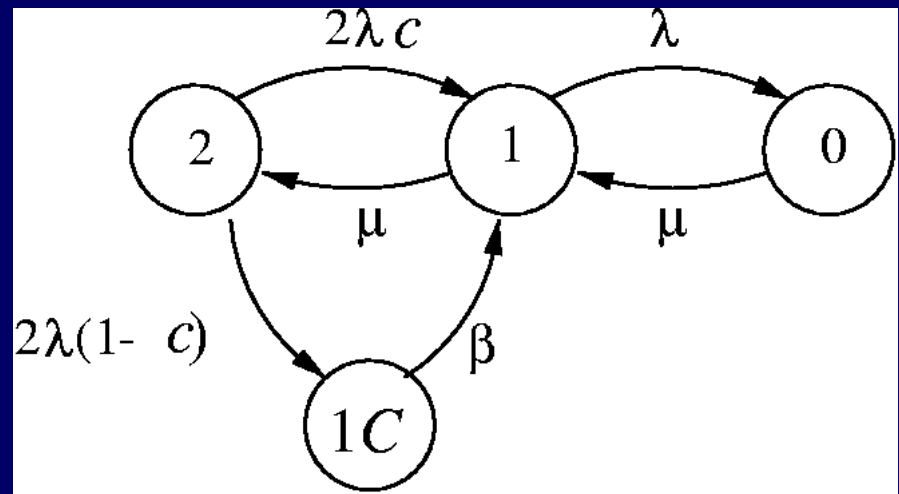
# CTMC

## Hot Standby

2-equal component without perfect switching with imperfect coverage availability model

Coverage factor =  $c$  (conditional probability that the fault is correctly handled)

$1C$  state is a reboot (down) state.



$$U(\beta, c) = \pi_0 + \pi_{1C} = \frac{\lambda\beta + \mu^2(1-c)}{\mu\beta E} \quad (E = \rho\pi_0^{-1})$$

$$D(\beta, c) = U(\beta, c) \times 8760 \times 60 \quad (\text{down time in min/year})$$

# CTMC

ColdStandbyCTMC\_MC

## Cold Standby

$\lambda$ : 0.001

$\mu$ : 0.1

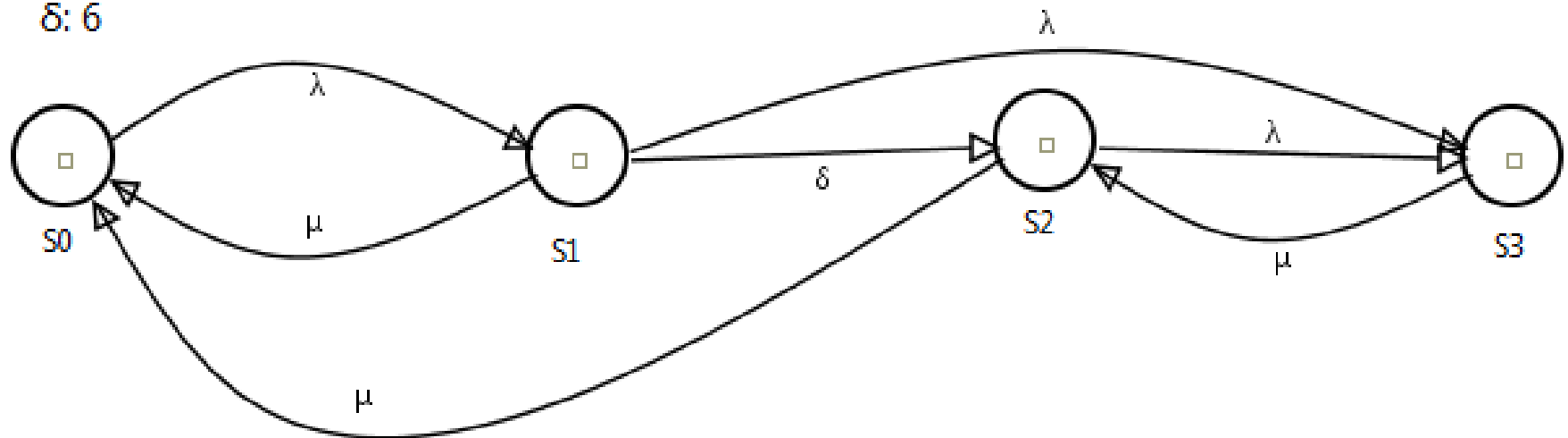
$\delta$ : 6

A:  $(P\{S0\}+P\{S2\})$

DTyh:  $8760*(P\{S1\}+P\{S3\})$

A: 0.9900375256

DTyh: 87.27127561.

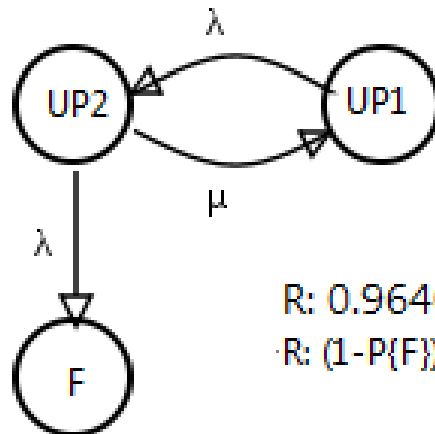


# CTMC

Rel\_CTMCColdStandby\_PS\_IC

## Cold Standby Reliability Model with Perfect Switching

$\lambda: 0.001$   
 $\mu: 0.1$



R: 0.9640809

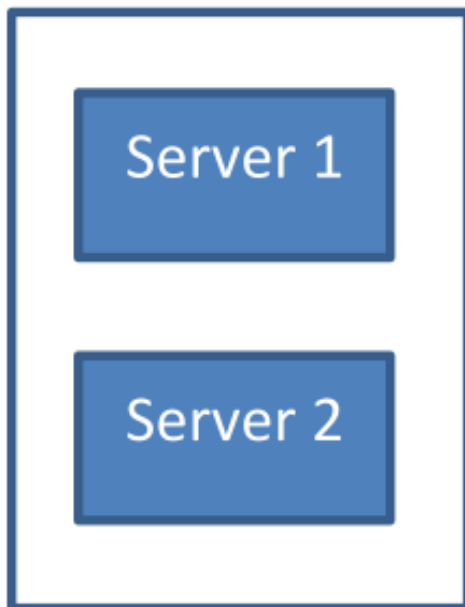
R:  $(1 - P\{F\})$

MTTA: 101999.9999849

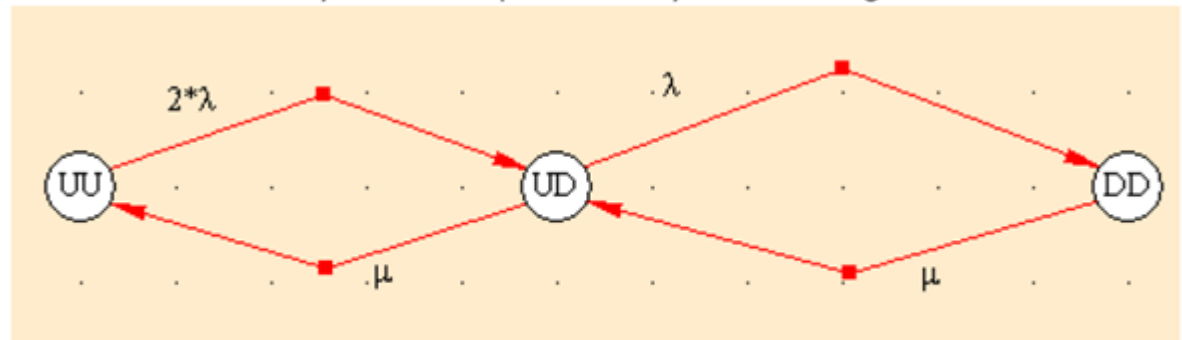
# CTMC

## Active-Active Redundancy

Consider a system with two parallel servers. The system is considered to be operational if at least one of the servers is operational.



An availability model is represented by the following CTMC:



$$A = \pi(UU) + \pi(UD) = \frac{\mu(2\lambda + \mu)}{2\lambda^2 + 2\lambda\mu + \mu^2}$$

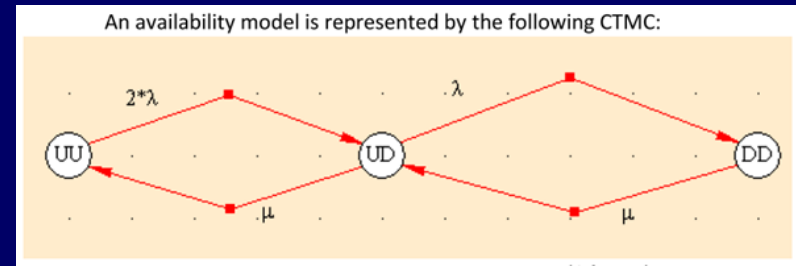
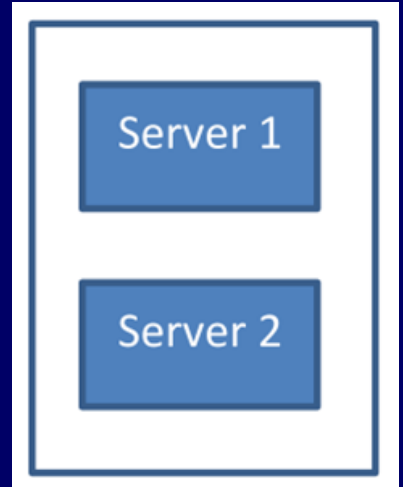
# CTMC

## Active-Active Redundancy

### Capacity oriented availability

Now, if the users are interested not as much whether the system is operational or not, but rather in the service capacity the system may deliver. Considering the depicted architecture, it is assumed that if the two servers are operational, the system may deliver its full service capacity. If only one server is operational, the system may deliver only half of its service capacity. And when none of the servers is operational, the system may not deliver the service. Therefore Capacity Oriented Availability (COA) is:

$$COA = (2 \times \pi(UU) + \pi(DU)) / 2$$



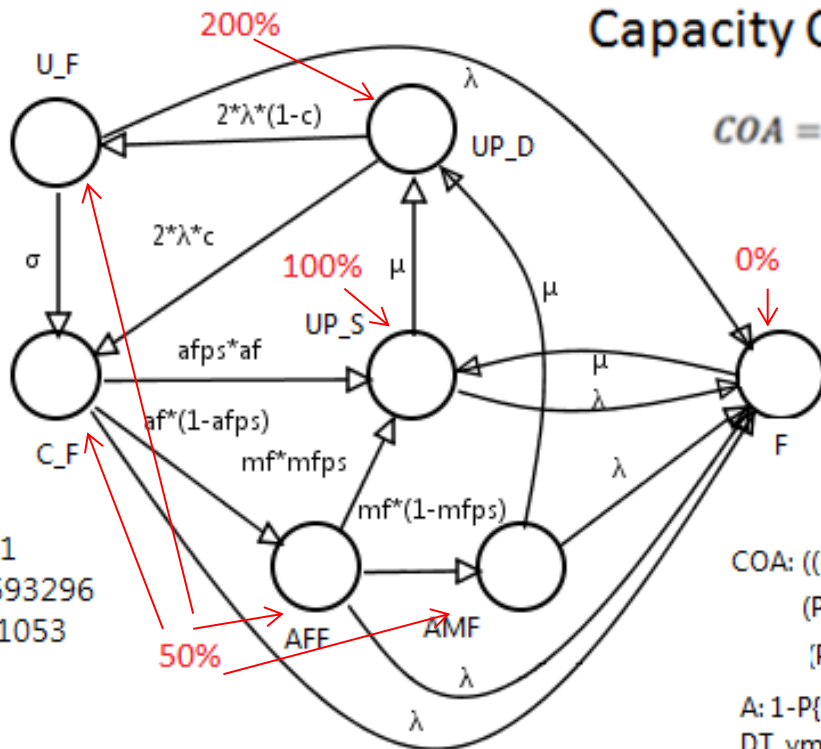
$$COA = \frac{\mu(\lambda + \mu)}{2\lambda^2 + 2\lambda\mu + \mu^2}$$



# CTMC

Active-Active system with imperfect fault coverage,  
automatic and manual failover mechanism

$\lambda$ : 1/10000  
 $\mu$ : 1/24  
 $\sigma$ : 0.25  
 af: 8  
 mf: 2  
 c: 0.99  
 afps: 0.95  
 mfps: 0.98  
 NS: 2



Capacity Oriented Availability

$$COA = \frac{\sum_i^n \pi_i \times r_i}{NS}$$

$$r_i := \begin{cases} n_i & \text{if } i = UP\_D, UP\_S \\ 0.5 \times n_i & \text{if } i = U\_F, C\_F, AFF, AMF \\ 0 & \text{otherwise} \end{cases}$$

where  $n_i \in \mathbb{N}$  is the number of active servers at state  $i$ .

$NS$  is the total number of servers.

$$COA: ((P\{UP\_D\} \times R\{UP\_D\}) + (P\{U\_F\} \times R\{U\_F\}) + (P\{C\_F\} \times R\{C\_F\}) + (P\{UP\_S\} \times R\{UP\_S\}) + (P\{AFF\} \times R\{AFF\}) + (P\{AMF\} \times R\{AMF\}) + (P\{F\} \times R\{F\})) / (NS)$$

$$A: 1 - P\{F\}$$

$$DT\_ym: ((P\{F\} \times 8760) \times 60)$$

A: 0.9999952351

DT\_ym: 6.0793693296

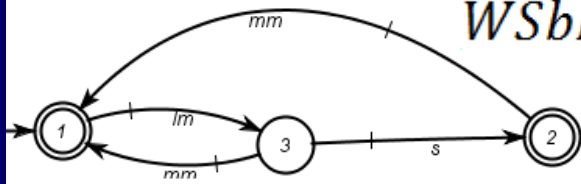
COA: 0.9975661053

# CTMC

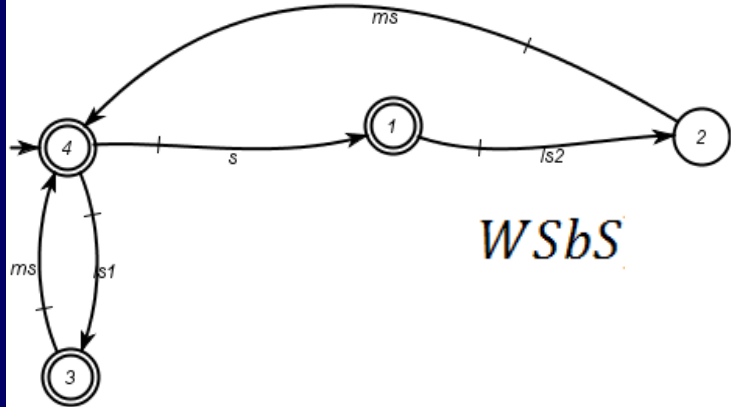
Warmstandby

Warm Standby

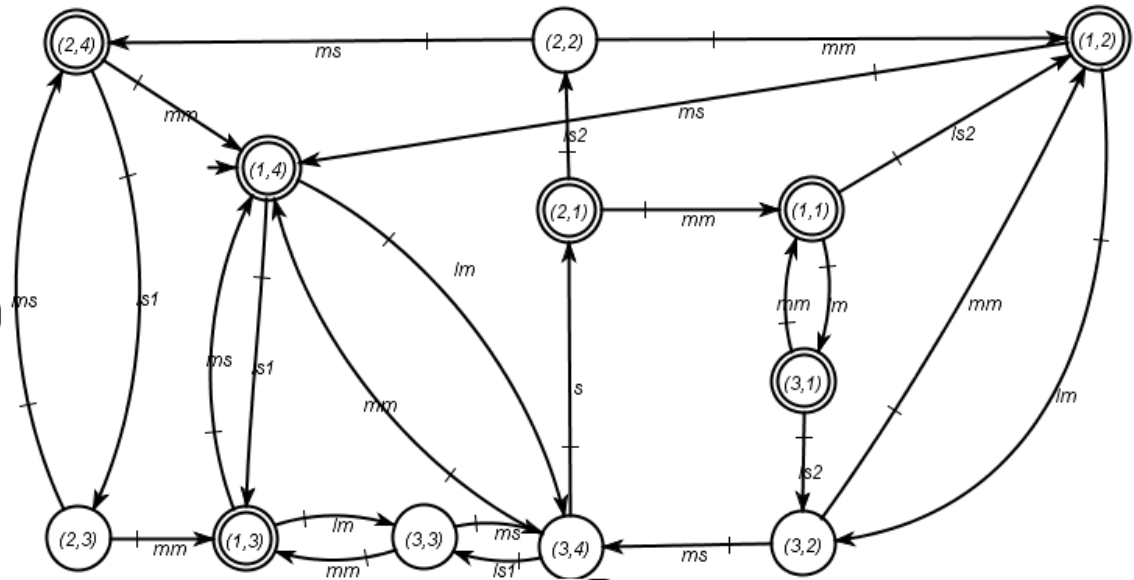
*WSbM*



*WSbS*



$WSB = (WSbM \parallel WSbS)$



# CTMC

## ■ Example – Availability model

### An equivalent 2-state availability model

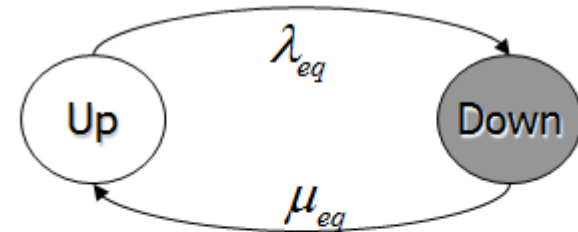
It is interesting to consider an equivalent 2-state availability model that has the same steady state availability as the given multi-state availability model.

To represent system availability in the simple form of equivalent 2-state system, we need to properly define equivalent failure rate  $\lambda_{eq}$  and equivalent repair rate  $\mu_{eq}$ , such that

$$A = \frac{MTTF_{eq}}{MTTF_{eq} + MTTR_{eq}} = \frac{\mu_{eq}}{\lambda_{eq} + \mu_{eq}}$$

Failure rate of each machine is  $\lambda$

Repair rate is  $\mu$



$$\lambda_{eq} = \frac{\lambda \pi_{M-1}}{\pi_0 + \pi_1 + \pi_2 + \dots + \pi_{M-1}}$$

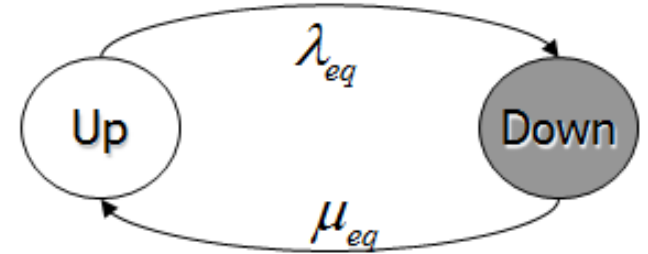
$$\mu_{eq} = \mu$$

# CTMC

## ■ Example – Availability model

An equivalent 2-state availability model

Let  $U$  be the set of up states,  $D$  the set of down states,  $R$  the set of all transitions from  $U$  to  $D$ ,  $G$  the set of all transition from  $D$  to  $U$ ,  $t_{ij}$  the transition from state  $i$  to  $j$



$$\lambda_{eq} = \frac{\lambda \pi_{M-1}}{\pi_0 + \pi_1 + \pi_2 + \dots + \pi_{M-1}}$$

$$\mu_{eq} = \mu$$

$$\lambda_{eq} = \sum_{t_{ij} \in R} P(\text{system in state } i \mid \text{system is up}) \times q_{ij} = \frac{\sum_{t_{ij} \in R} \pi_i \times q_{ij}}{\sum_{k \in U} \pi_k}$$

$$\mu_{eq} = \sum_{t_{ij} \in G} P(\text{system in state } i \mid \text{system is down}) \times q_{ij} = \frac{\sum_{t_{ij} \in G} \pi_i \times q_{ij}}{1 - \sum_{k \in U} \pi_k}$$

# CTMC

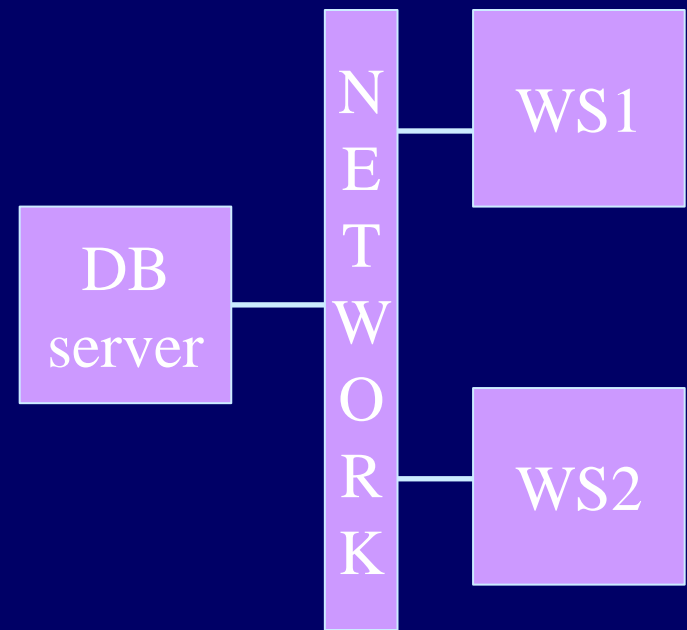
## ■ Example

Consider a system consisting of two web-servers, one database server and a network infrastructure. The system is operational as long as one web-server and the database server are operational. It is assumed that a network infrastructure is fault-free. The database server repairing has priority over the web-servers' repairing activities. The failure rates of the web-servers and of the database server are constant ( $\lambda_{ws}$ ,  $\lambda_{db}$  respectively), and the respective time to repair are exponentially distributed with rate  $\mu_{ws}$  and  $\mu_{db}$ .

$$\lambda_{ws} = 1.14 \times 10^{-4} \text{ failures per hour}$$

$$\lambda_{db} = 2.28 \times 10^{-4} \text{ failures per hour}$$

$$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2} \text{ repairings per hour}$$

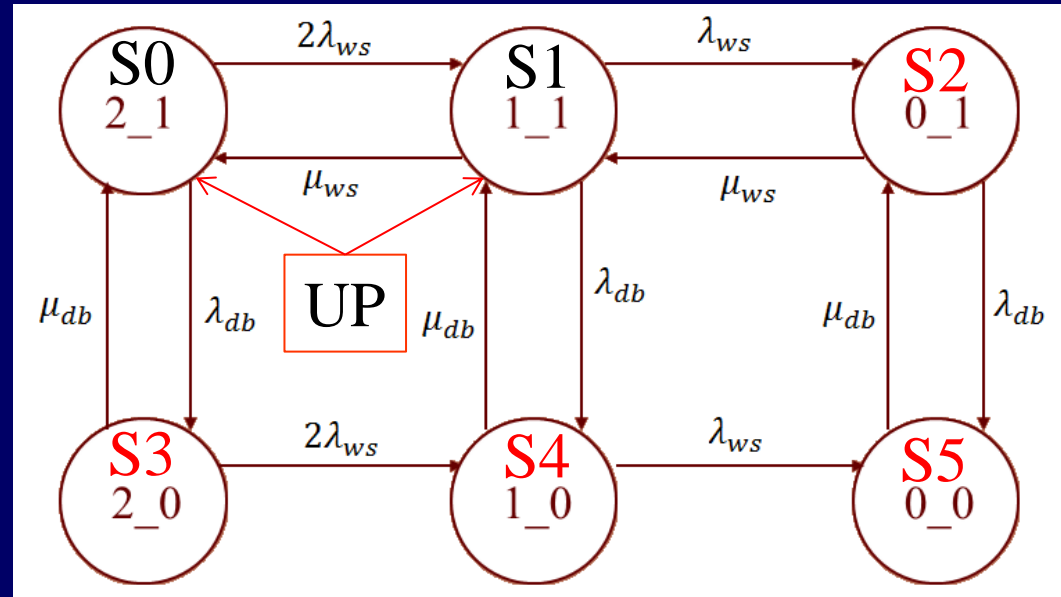
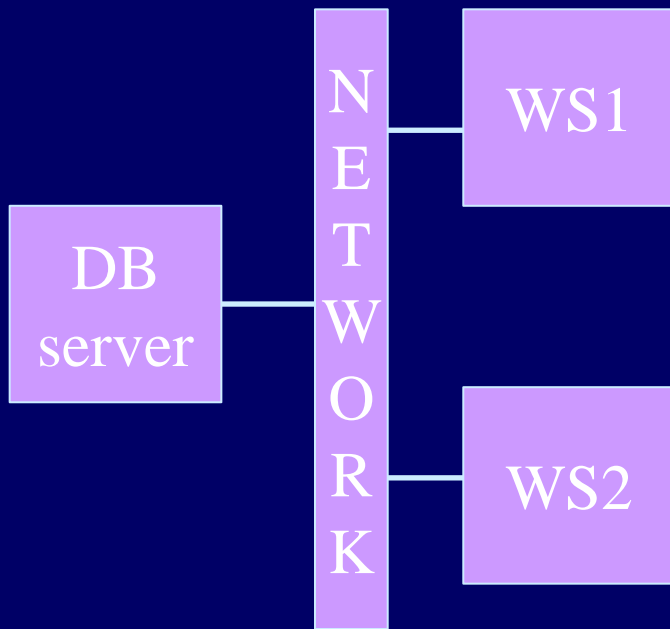


# CTMC

ws1 ws2 dbs

Excel

## Example – Availability model



$S_i$	$\pi_i$	Up/Down
S0	0.98910959199	U
S1	0.00543748939	U
S2	0.00001502574	D
S3	0.00537867258	D
S4	0.00005897750	D
S5	0.00000024281	D
A	0.99454708138	

$\lambda_{ws} = 1.14 \times 10^{-4}$  failures per hour

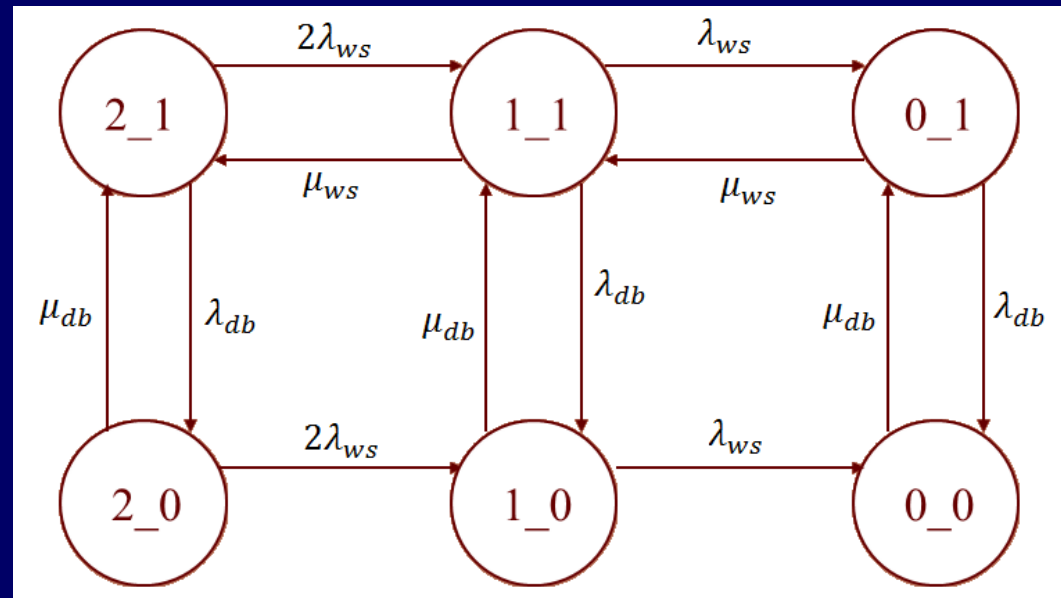
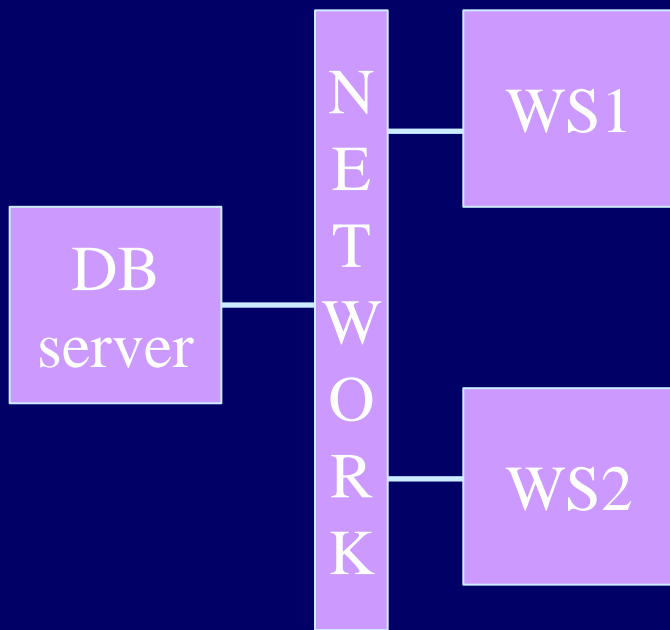
$\lambda_{db} = 2.28 \times 10^{-4}$  failures per hour

$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2}$  repairings per hour

# CTMC

Excel

## ■ Example – Availability model



$$A = \pi_{2_1} + \pi_{1_1} = 0.994547080$$

$$\text{Downtime} = (1 - A) \times T = 2866.05467 \text{ minutes}$$

$$T = 8760h \times 60min = 525,600 \text{ minutes in one year.}$$

$$\lambda_{ws} = 1.14 \times 10^{-4} \text{ failures per hour}$$

$$\lambda_{db} = 2.28 \times 10^{-4} \text{ failures per hour}$$

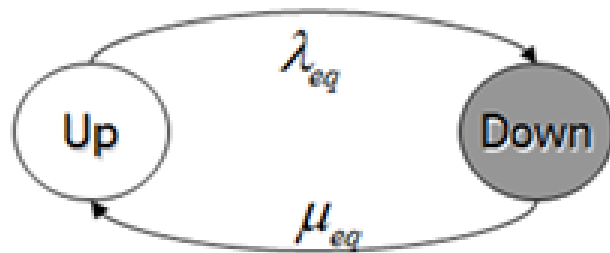
$$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2} \text{ repairings per hour}$$

# CTMC

Excel

## ■ Example – Availability model

The equivalent two-state model

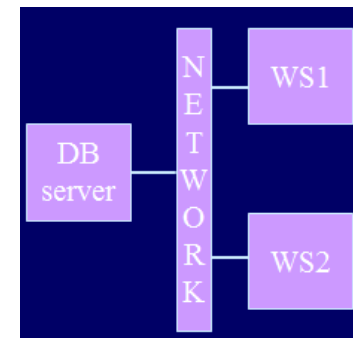


$$\lambda_{eq} = \sum_{t_j \in R} P(\text{system in state } i \mid \text{system is up}) \times q_{ij} = \frac{\sum_{t_j \in R} \pi_i \times q_{ij}}{\sum_{k \in U} \pi_k}$$

$$\mu_{eq} = \sum_{t_j \in G} P(\text{system in state } i \mid \text{system is down}) \times q_{ij} = \frac{\sum_{t_j \in G} \pi_i \times q_{ij}}{1 - \sum_{k \in U} \pi_k}$$

$\lambda_{eq}$	0.000228623
$\mu_{eq}$	0.041698143

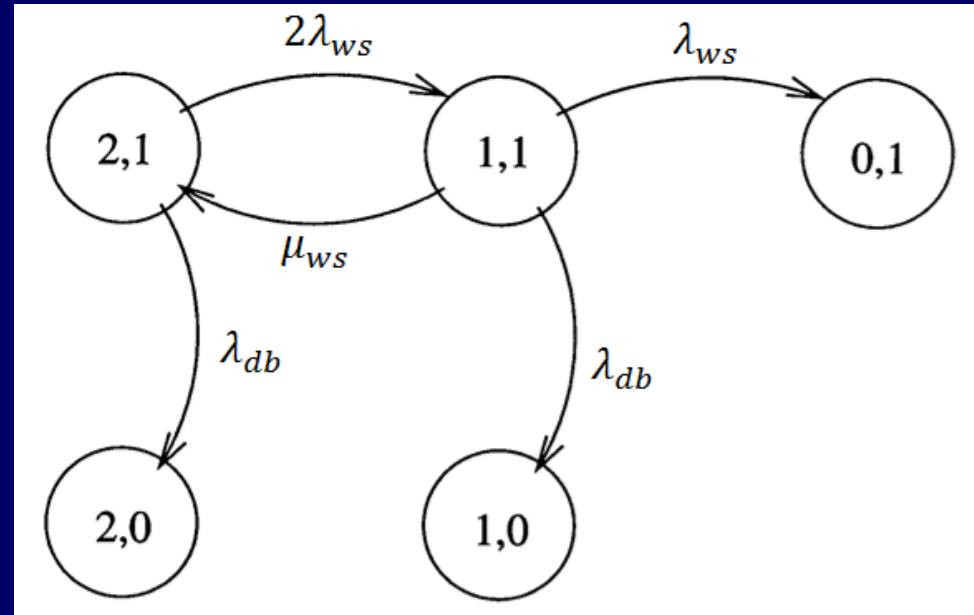
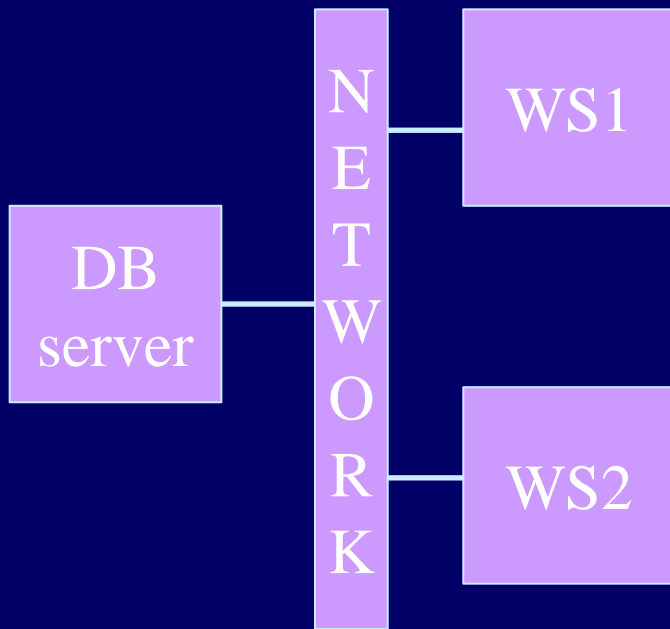
$S_i$	$\pi_i$
UP	0.99454708138
Down	0.00545291862
A rm	0.99454708138





# CTMC

## ■ Example – Reliability model



States (0,1), (1,0) and (2,0) are absorbing states and (2,1) and (1,1) are transient states.

Absorbing states can be combined into a single one.

$$\lambda_{ws} = 1.14 \times 10^{-4} \text{ failures per hour}$$

$$\lambda_{db} = 2.28 \times 10^{-4} \text{ failures per hour}$$

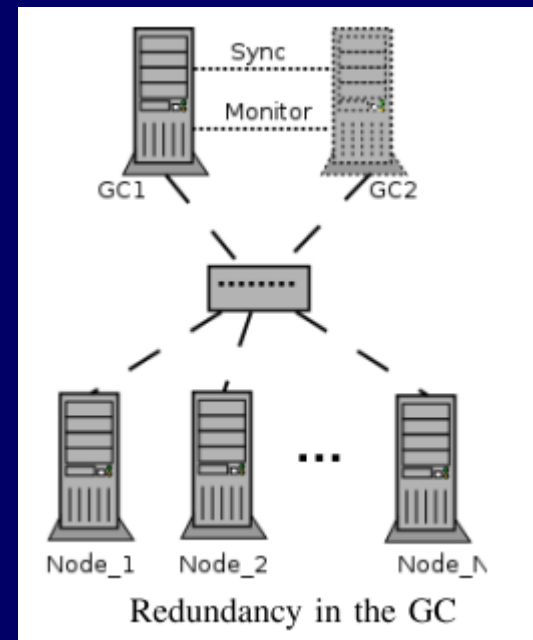
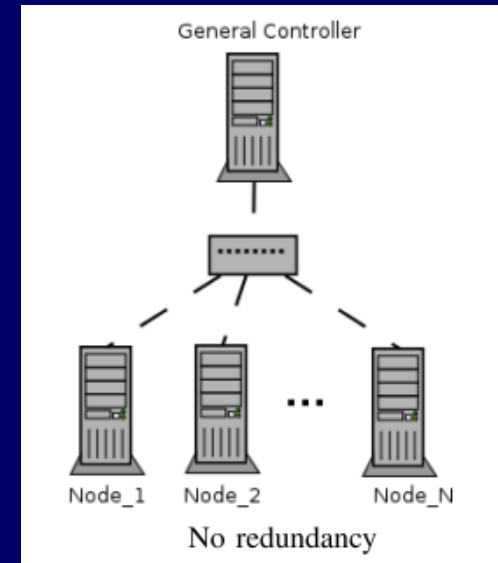
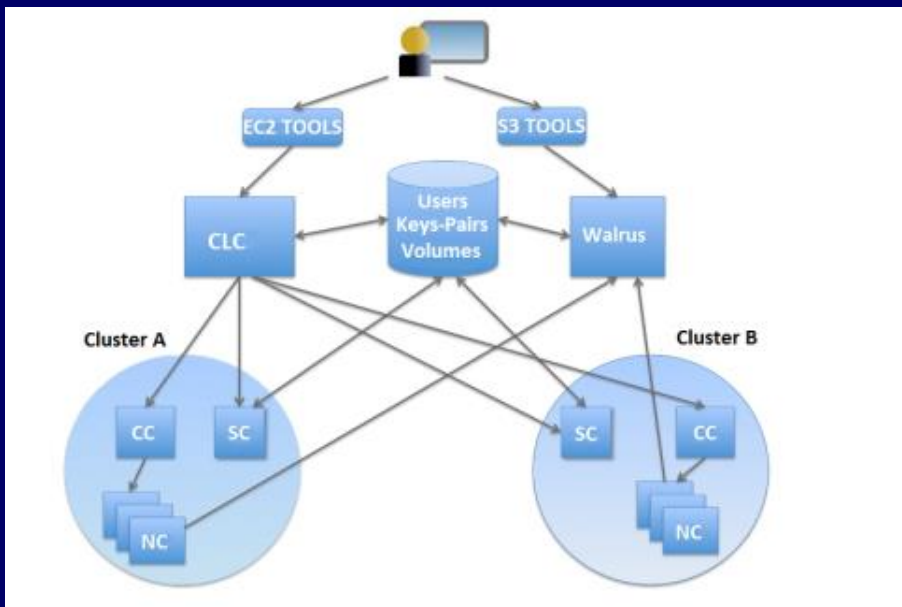
$$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2} \text{ repairings per hour}$$

$$R(t) = \pi_{2,1}(t) + \pi_{1,1}(t)$$

# CTMC

## ■ Example - Availability model

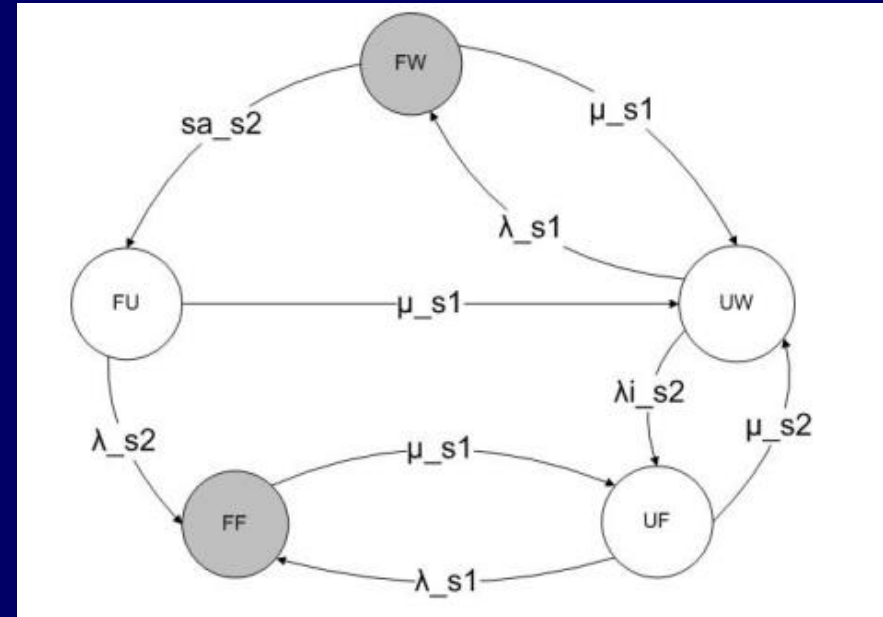
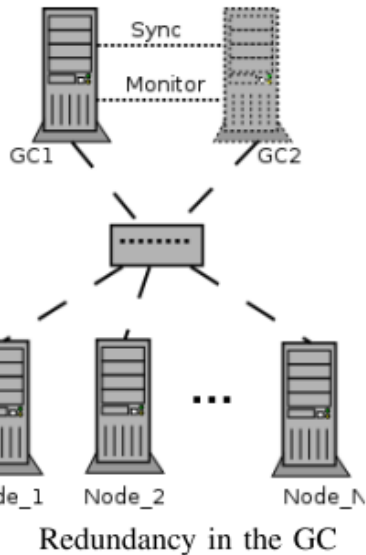
EUCALYPTUS is composed by five high-level components: Cloud Controller, Cluster Controller, Node Controller, Storage Controller, and Walrus. The Cloud Controller (CLC) is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage).



# CTMC

## ■ Example - Availability model

EUCALYPTUS is composed by five high-level components: Cloud Controller, Cluster Controller, Node Controller, Storage Controller, and Walrus. The Cloud Controller (CLC) is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage).



Parameter	Description	Value
$\lambda_{s1} = \lambda_{s2} = 1/\lambda$	Mean time for host failure	1/180.721
$\lambda_{i_s2} = 1/\lambda_i$	Mean time for inactive host failure	1/216.865
$\mu_{s1} = \mu_{s2} = 1/\mu$	Mean time for host repair	1/0.9667
$sa_{s2} = 1/sa$	Mean time to system activate	1/0.005

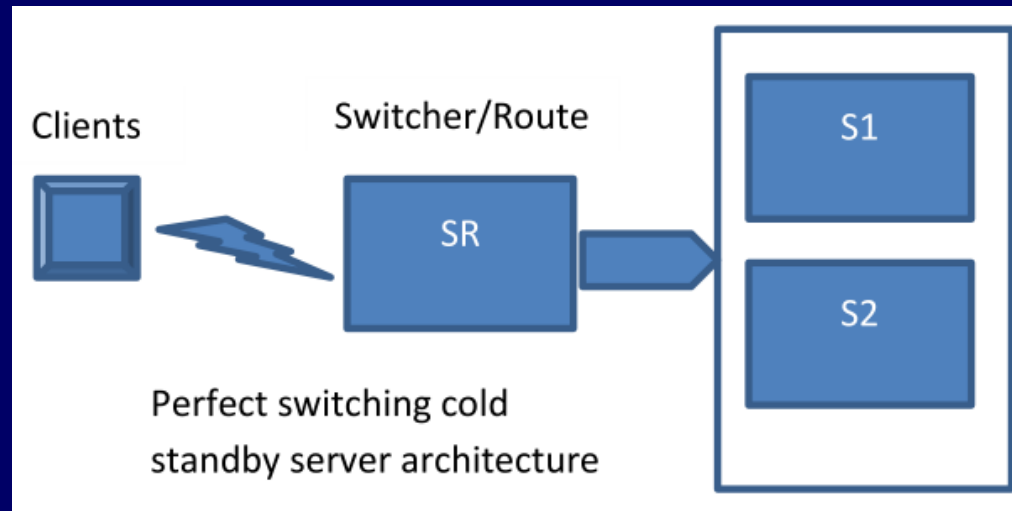
$$A_{GC} = \frac{\mu(\lambda_i(\mu + sa) + \mu^2 + sa(\lambda + \mu))}{\lambda_i(\lambda + \mu)(\mu + sa) + \mu^2(\lambda + \mu) + sa(\lambda^2 + \lambda\mu + \mu^2)}$$

# CTMC

## ■ Example – Reliability model

**System composed by Two Subsystem:**

**One Switch/Router and Server Cluster**



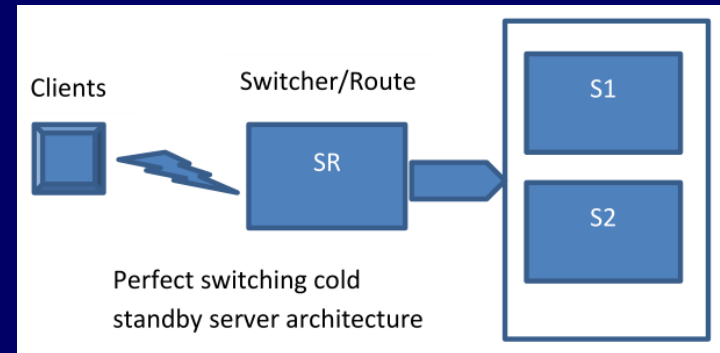
The system is composed by a Switcher/Router and Serve subsystem. The system fails if the Switcher/Router fails OR if the Serve subsystem fails. The Server subsystem is composed by two servers, S1 and S2. S1 is the main server and S2 is the spare server. They are configured in Cold Standby, that is, S2 starts as soon as S1 fails. The start-up time of S2 is zero. This is named perfect switching.

# CTMC

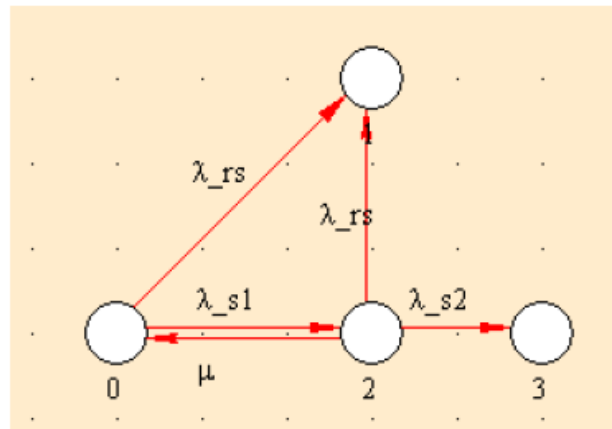
## ■ Example – Reliability model

System composed by Two Subsystem:

One Switch/Router and Server Cluster



The CTMC reliability model



Absorbing states can be combined into a single one

Variable	Value
lambda_rs	1/2000
lambda_s1	1/15000
mu	1/24
lambda_s2	1/15000

The unity of these rates is  $h^{-1}$ .

System Unreliability:

$$UR(4000h) = 0.181615244$$

System Reliability:

$$R(4000h) = 0.818384756$$

$\lambda_{rs}$  is failure rate of the Switcher/Router.

$\lambda_{s1}$  is failure rate of the Server 1.

$\lambda_{s2}$  is failure rate of the Server 2.

$\mu$  is the repair rate assigned to Server 1 repair activity.

# CTMC

## ■ Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

Two main strategies:

Condition-based (inspection-based)  
PM considered here

Time-Based PM

We model TTF by Hypoexponential  $\text{HYPO}(\lambda_1, \lambda_2)$  distribution.

Time to trigger inspection is assumed to be  $\text{EXP}(\lambda_{in})$ ,

Time to carry out inspection is  $\text{EXP}(\mu_{in})$ ,

Time to repair is  $\text{EXP}(\mu)$ ,

Time to carry out PM is  $\text{EXP}(y\mu)$ .

# CTMC

## ■ Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

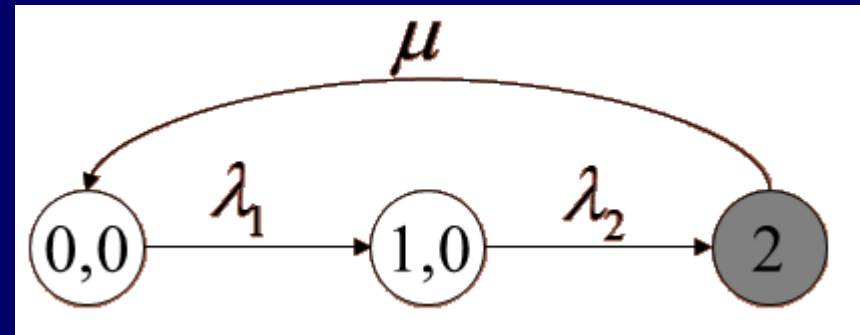
CTMC with corrective maintenance only

Time to failure is HYPO( $\lambda_1, \lambda_2$ );

(0,0) & (1,0) are up states;

2 is a down state

Time to corrective maintenance is EXP( $\mu$ )



# CTMC

## ■ Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

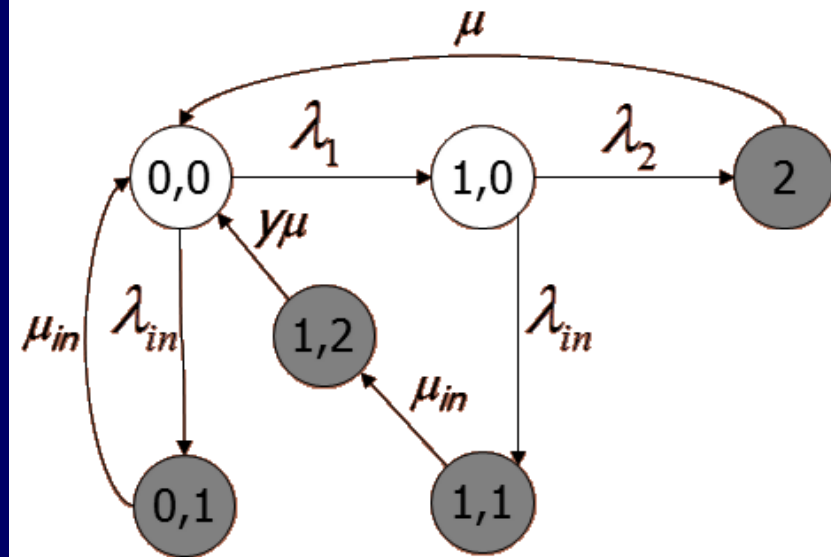
CTMC with preventive maintenance

Inspection triggered after  $\text{EXP}(\lambda_{in})$  intervals

Time to carry out inspection is  $\text{EXP}(\mu_{in})$

Time to carry out PM is  $\text{EXP}(\gamma\mu)$

PM carried out if inspection finds the system to be in degraded state (1,0)



$$A = \pi_{0,0} + \pi_{1,0}$$

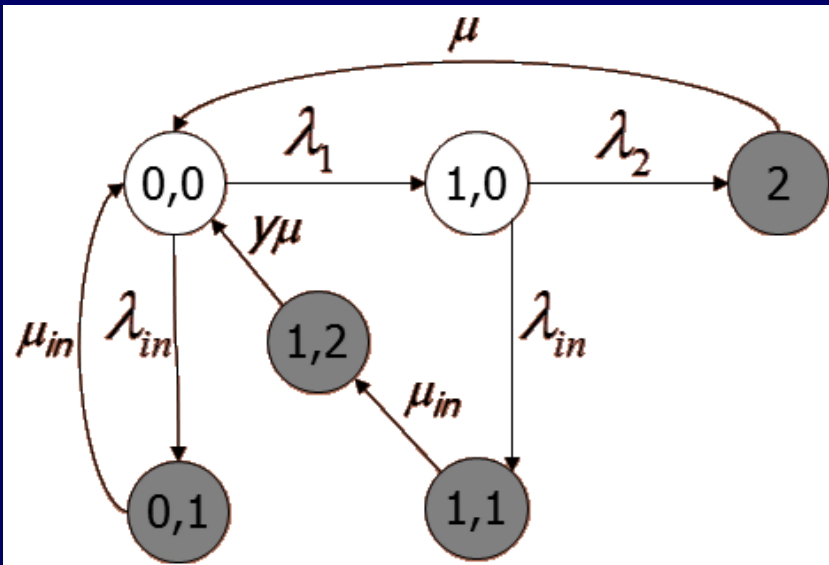


# CTMC

## ■ Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

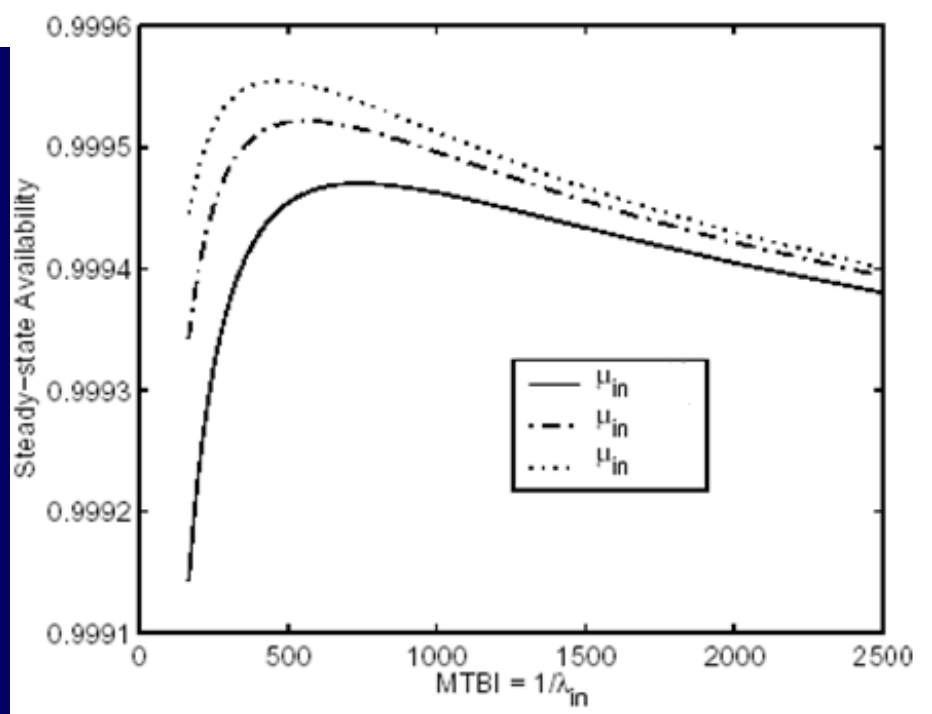
$\lambda_1 = 0.001 \text{ h}^{-1}$   
 $\lambda_2 = 0.001 \text{ h}^{-1}$   
 $\mu_{in} = 10 \text{ h}^{-1}$   
 $\mu = 0.1 \text{ h}^{-1}$   
 $y = 5$   
 $\lambda = 0.0005 \text{ h}^{-1}$



$$A = \pi_{0,0} + \pi_{1,0}$$

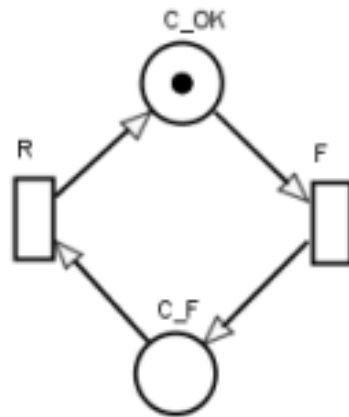
MTBI – mean time between inspections

Availability as function of  $MTBI = 1/\lambda_{in}$



# SPN

## Single Component System Availability Model



Transition	Time	Time	Type of Service
F	MTTF	$\lambda$	single Server
R	MTTR	$\mu$	single Server

The instantaneous availability :

$$A(t) = P\{(m(C\_OK) = 1)(t)\}$$

$$= \sum_{\forall M_i \in RS} r_i \times \pi_i(t) = \frac{\lambda e^{-t(\lambda+\mu)} + \mu}{\lambda + \mu}$$

Downtime in period  $T$  :

$$DT = T \times P\{(m(C\_F) = 1)\} = T \times \left(1 - \frac{\lambda}{\lambda + \mu}\right)$$

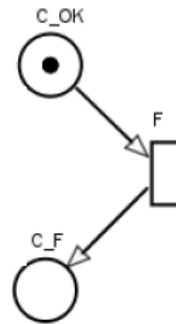
The stationary availability :

$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i = \frac{\lambda}{\lambda + \mu}$$

$$r_i = \begin{cases} 1 & \text{se } m_i(C\_OK) = 1 \\ 0 & \text{se } m_i(C\_OK) = 0 \end{cases}$$

# SPN

## Single Component System Reliability Model



Transition	Time	Time	Type of Service
F	MTTF	$\lambda$	single Server

Although the reliability of the basic component is analytically defined by  $R(t) = e^{-t\lambda}$ , it is possible to calculate the respective value through numerical transient analysis, once the transition  $R$  is removed. The reliability can be calculated by:

$$R(t) = P\{(m(C\_OK) = 1)(t)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i(t),$$

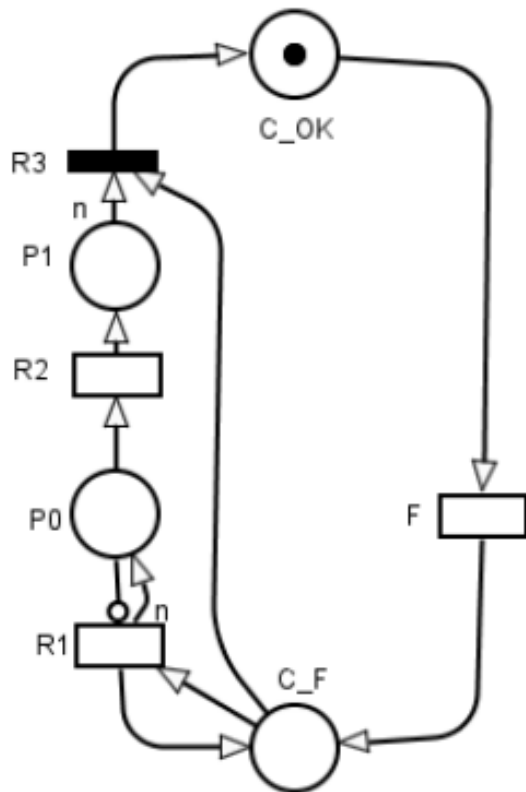
where

$$r_i = \begin{cases} 1 & \text{se } m_i(C\_OK) = 1 \\ 0 & \text{se } m_i(C\_OK) = 0 \end{cases}$$

# SPN

## Basic Model with Erlang Distributed Repair Time

Availability Model



$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in ERS} r_i \times \pi_i,$$

$$r_i = \begin{cases} 1 & \text{se } m_i(C\_OK) = 1 \\ 0 & \text{se } m_i(C\_OK) = 0 \end{cases}$$

$$E[T_E] = \bar{X} \quad e \quad DP[T_E] = SD$$

$$n = \left( \frac{\bar{X}}{DP} \right)^2$$

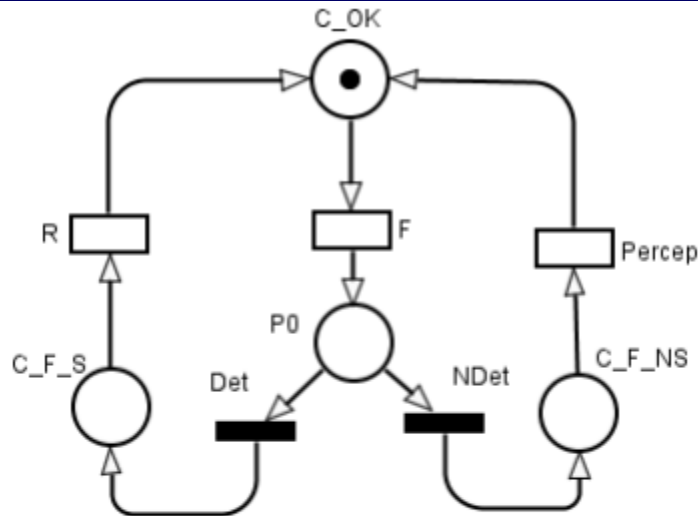
$$\lambda = \frac{n}{\bar{X}}$$

Transition	Type	Time or weight	Rate	Type of service
F	E	MTTF	$\lambda = \frac{1}{MTTF}$	single Server
R1	E	MTTR/n	$\mu = \frac{n}{MTTR}$	single Server
R2	E	MTTR/n	$\mu = \frac{n}{MTTR}$	single Server
R3	I	W=1		

Basic Model with the Erlang Distributed Repair Time

# SPN

## Basic Model with imperfect coverage availability model



$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i$$

$$r_i = \begin{cases} 1 & \text{se } m_i(C\_OK) = 1 \\ 0 & \text{se } m_i(C\_OK) = 0 \end{cases}$$

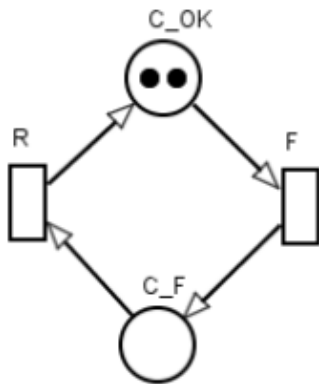
Transition	Type	Time or Wieht	Rate	Type of Service
<i>F</i>	<i>E</i>	<i>MTTF</i>	$\lambda$	<i>single server</i>
<i>Det</i>	<i>I</i>	$W_{Det}$		
<i>Ndet</i>	<i>I</i>	$W_{NDet}$		
<i>Percep</i>	<i>E</i>	<i>MTTP</i>	$\beta$	<i>single server</i>
<i>R</i>	<i>E</i>	<i>MTTR</i>	$\mu$	<i>single server</i>

Failure Coverage Basic Model

# SPN

## Hot Standby Model

Availability Model



$$A = P\{(m(C_{OK}) = 2) \vee (m(C_{OK}) = 1)\}$$

$$= \sum_{\forall m_i \in RS} r_i \times \pi_i = 1 - \frac{2\lambda^2}{2\lambda^2 + 2\lambda\mu + \mu^2}$$

$$r_i = \begin{cases} 1 & \text{se } (m(C_{OK}) = 2) \vee (m(C_{OK}) = 1) \\ 0 & \text{se } m_i(C_{OK}) = 0. \end{cases}$$

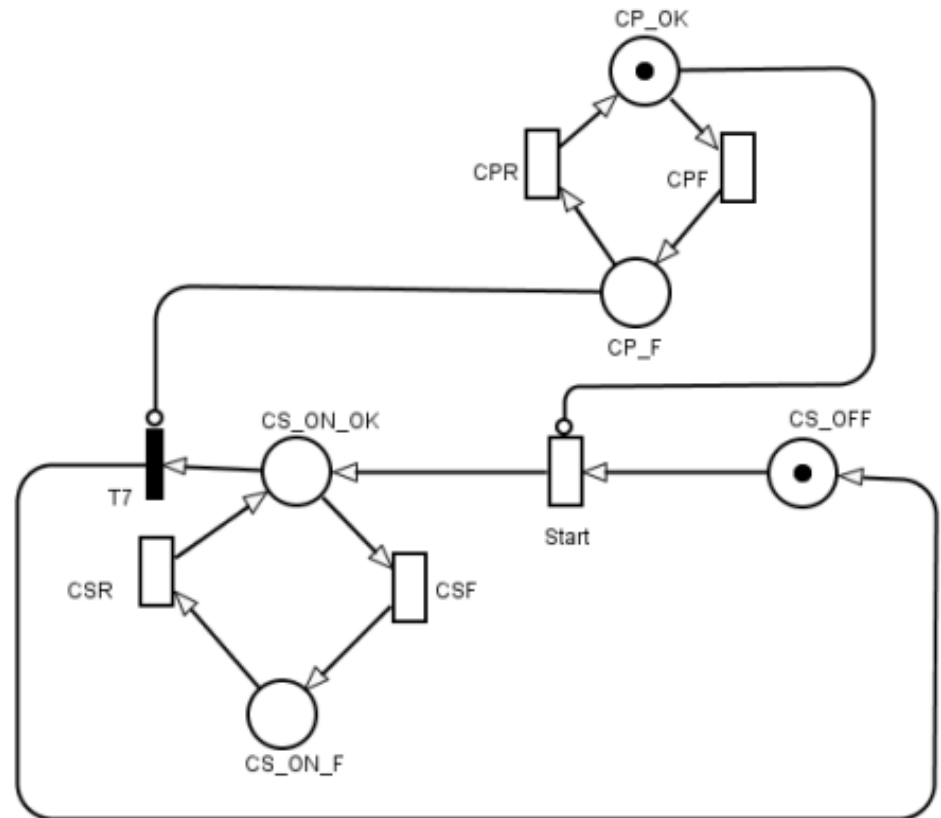
Transition	Type	Rate	Type of Service
<i>F</i>	<i>MTTF</i>	$\lambda$	<i>infinity Server</i>
<i>R</i>	<i>MTTR</i>	$\mu$	<i>single Server</i>

# SPN

## Cold Standby

### Availability Model

Transition	Type	Time or Weight	Rate	Type of Service
CPF	E	MTTF_CP	$\lambda$	single server
CPR	E	MTTR_CP	$\mu$	single server
CSF	E	MTTF_CS	$\alpha$	single server
CSR	E	MTTR_CS	$\beta$	single server
Start	E	TTS	$\mu$	single server
T7	I	W=1		



The stationary availability of the component is calculated by the expression:

$$A = P\{((m(CP\_OK) = 1) \vee (m(CS\_ON\_OK) = 1))\} = \sum_{\forall M_i \in RS} r_i \times \pi_i$$

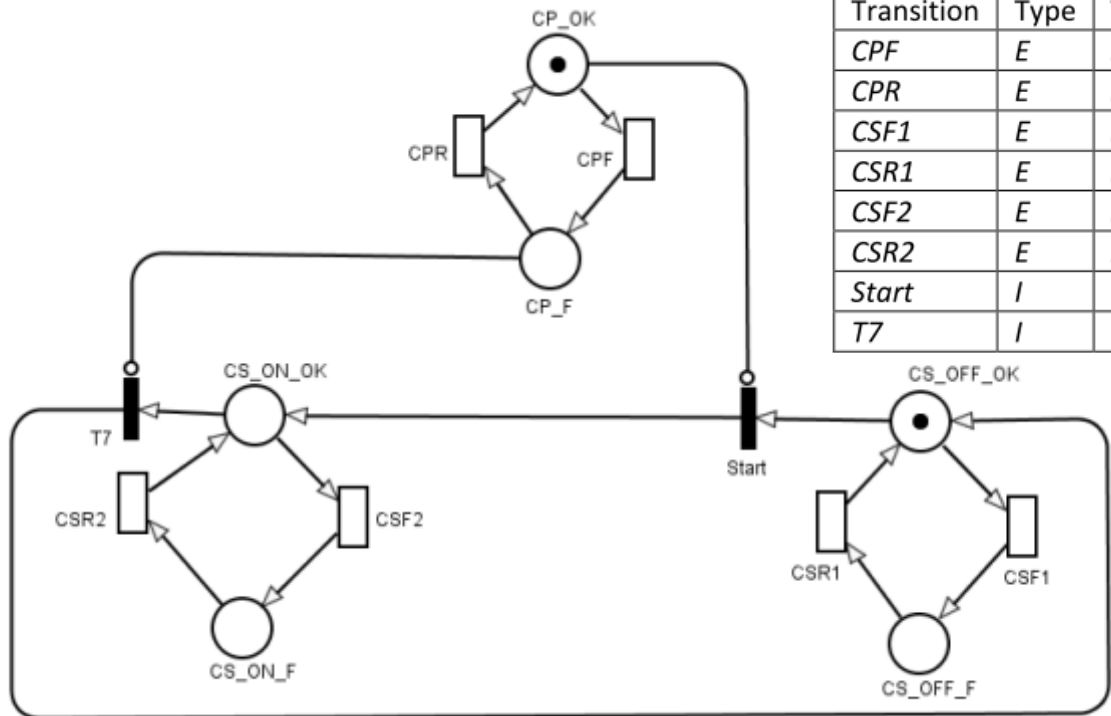
where  $r_i$  is a function that

$$r_i = \begin{cases} 1 & \text{se } (m(CP\_OK) = 1) \vee (m(CS\_ON\_OK) = 1) \\ 0 & \text{se } (m(CP\_OK) = 0) \wedge (m(CS\_ON\_OK) = 0) \end{cases}$$

# SPN

## Warm Standby Availability Model

The *Warm Standby* model is similar to the *Cold Standby* model. However, in a system with *Warm Standby* redundancy, the reserve component remains energized (but inoperative), so that, when the main component fails, the reserve component takes over operations without the delay that occurs in a *Cold Standby* system.



Transition	Type	Time or Weight	Rate	Type of Service	Priority
CPF	E	MTTF_CP	$\lambda$	single server	
CPR	E	MTTR_CP	$\mu$	single server	
CSF1	E	MTTF1_CS	$\alpha$	single server	
CSR1	E	MTTR1_CS	$\beta$	single server	
CSF2	E	MTTF2_CS	$\alpha$	single server	
CSR2	E	MTTR2_CS	$\beta$	single server	
Start	I	W=1			1
T7	I	W=1			1

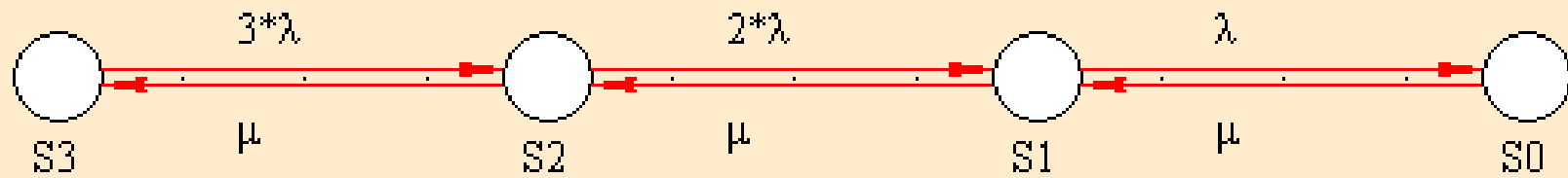


# SPN

## 2 out of 3 with shared repair

Availability Model

The CTMC model:



$$\lambda = \frac{1}{8760} h^{-1}$$

$$\mu = \frac{1}{24} h^{-1}$$

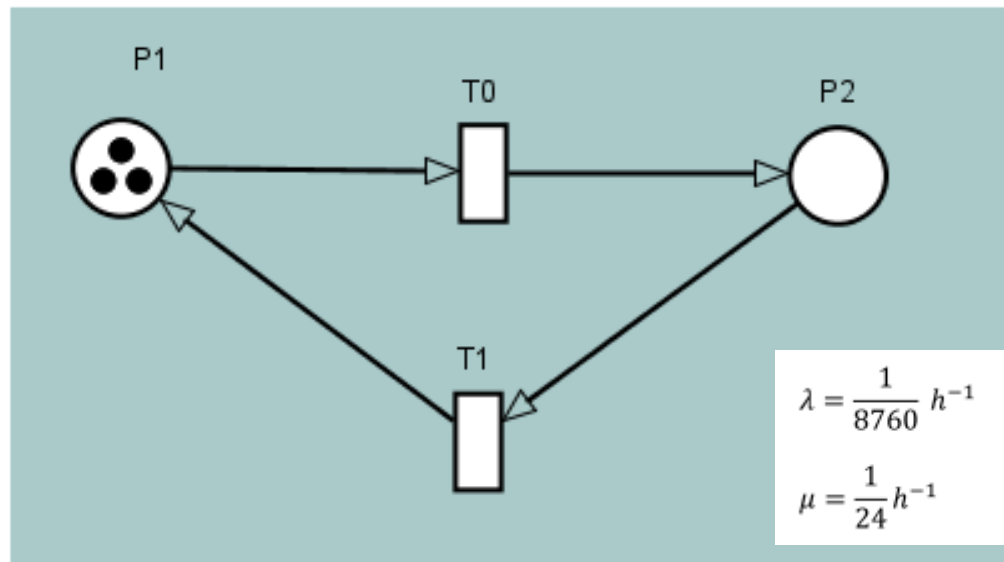
$$\text{Availability} = \pi(S_3) + \pi(S_2) = 9.99955210e-001$$

# SPN

## 2 out of 3 with shared repair

Availability Model

The equivalent SPN model:

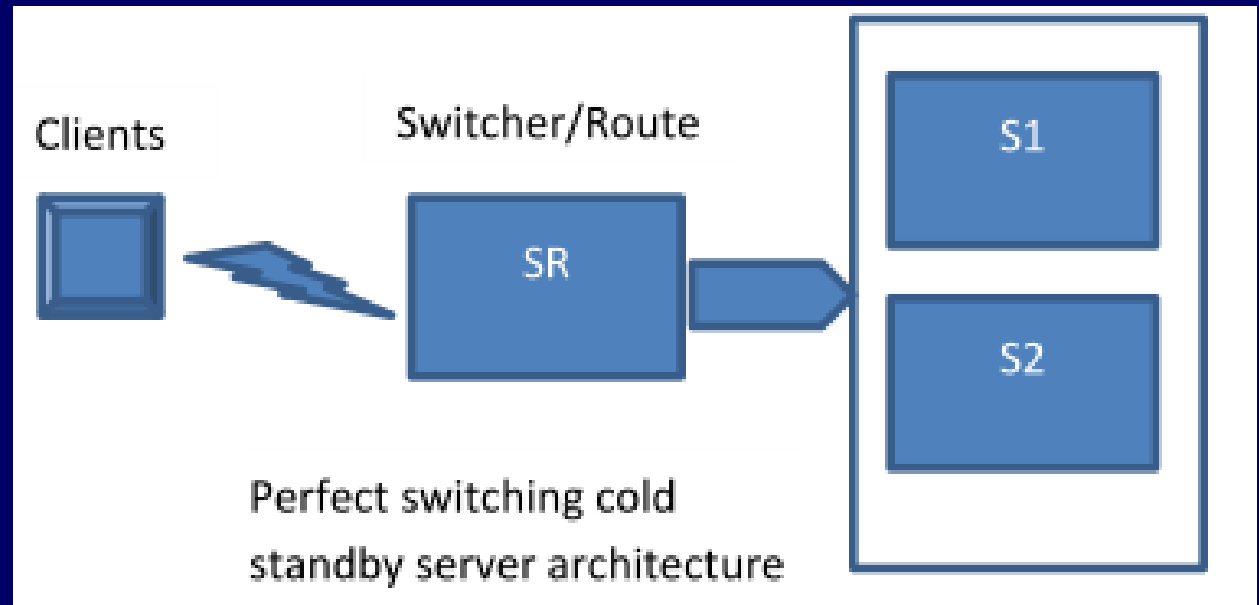


The result obtained through TimeNET:

The Availability =  $P\{\#P1 \geq 2\} = 0.9999552$ .

# SPN

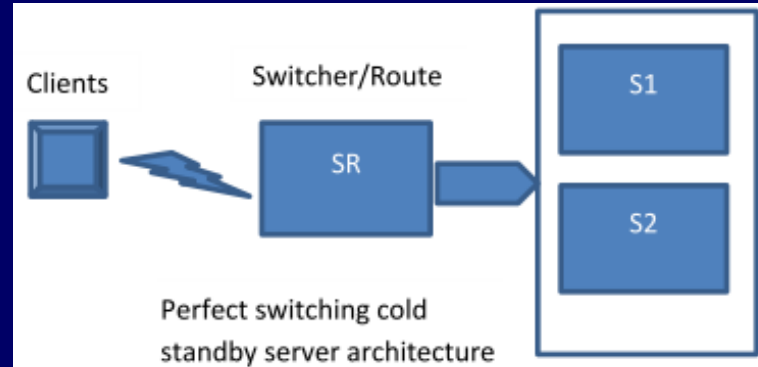
## Example



The system is composed by a Switcher/Router and Serve subsystem. The system fails if the Switcher/Router fails OR if the Serve subsystem fails. The Server subsystem is composed by two servers, S1 and S2. S1 is the main server and S2 is the spare server. They are configured in Cold Standby, that is, S2 starts as soon as S1 fails. The start-up time of S2 is zero.

# SPN

## Example



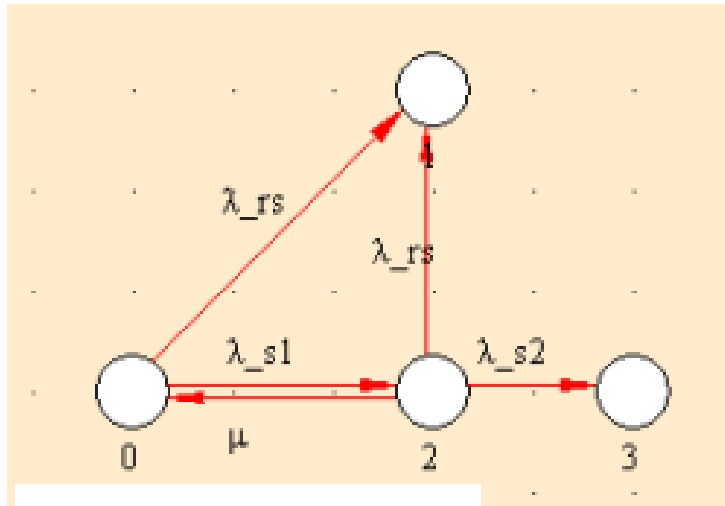
### CTMC reliability model

$\lambda_{rs}$  is failure rate of the Switcher/Router.

$\lambda_{s1}$  is failure rate of the Server 1.

$\lambda_{s2}$  is failure rate of the Server 2.

$\mu$  is the repair rate assigned to Server 1 repair activity.



Variable	Value
lambda_rs	1/2000
lambda_s1	1/15000
mu	1/24
lambda_s2	1/15000

The unity of these rates is  $h^{-1}$ .

System Unreliability:

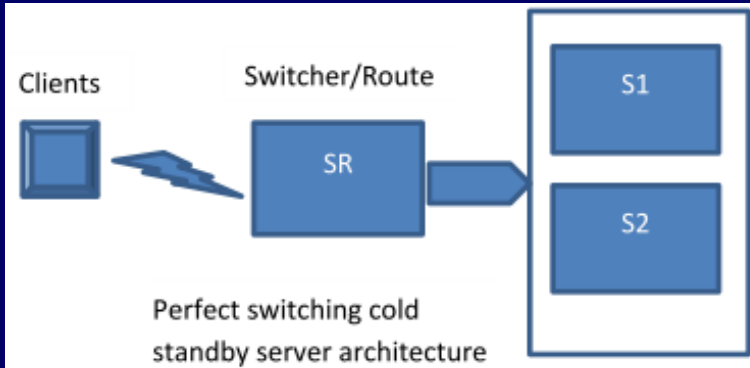
$$UR(4000h) = 0.181615244$$

System Reliability:

$$R(4000h) = 0.818384756$$

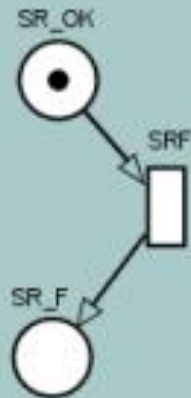
# SPN

## Example



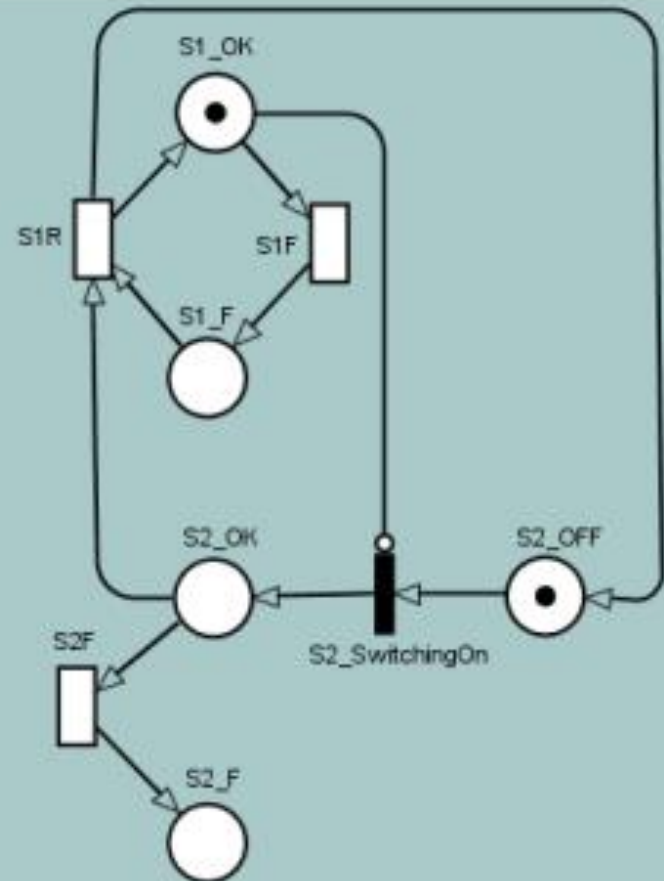
### SPN reliability model

System Unreliability:  
 $UR(4000h) = 0.18161528133$   
 System Reliability:  
 $R(4000h) = 0.81838471867$



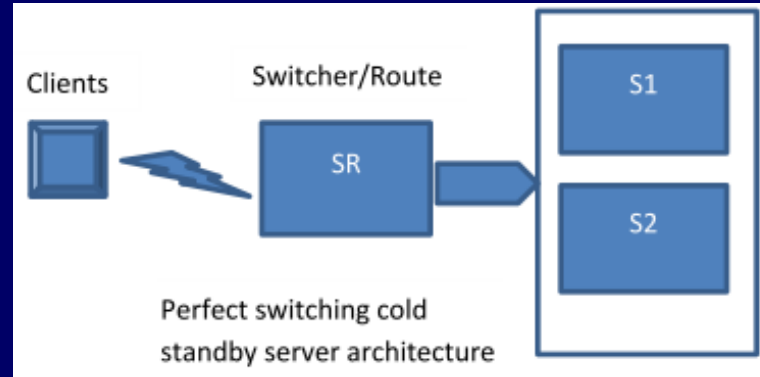
MTTF\_SR := 20000  
 MTTF\_S1 := 15000  
 MTTF\_S2 := 15000  
 MTTR\_S1 := 24

SystemUnreliability =  $P(\#SR\_F=1 \text{ OR } (\#S1\_F=1 \text{ AND } \#S2\_F=1))$   
 SystemReliability =  $1 - P(\#SR\_F=1 \text{ OR } (\#S1\_F=1 \text{ AND } \#S2\_F=1))$



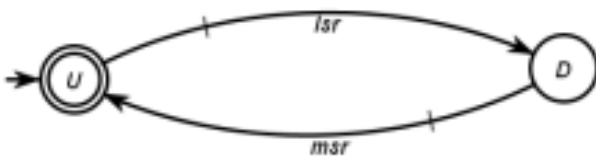
# SPN

## Example CTMC availability model

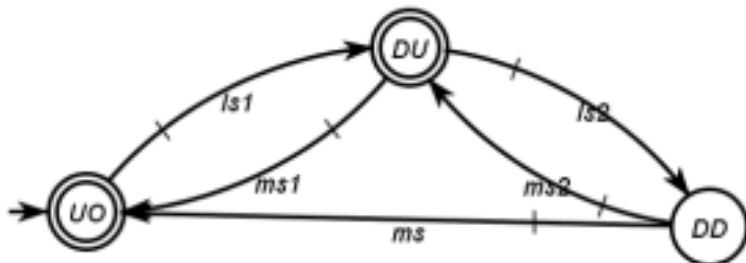


The component's state machines are:

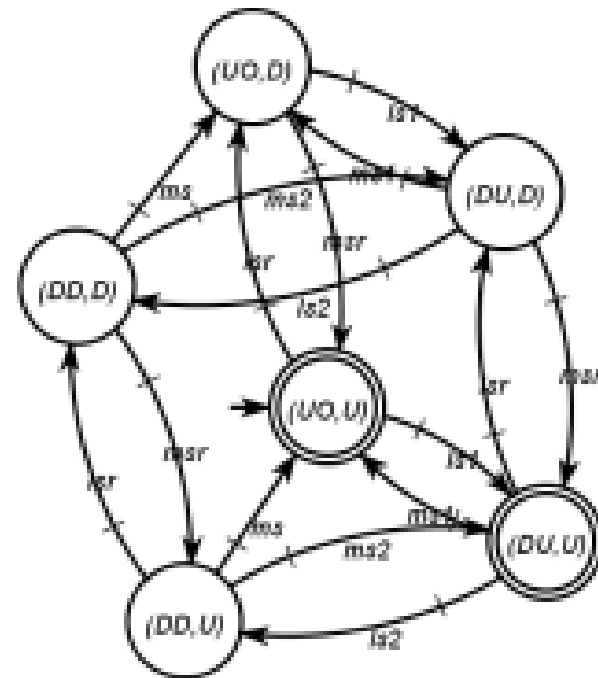
1) SR state machine (SR)



2) Server's state machine (CS)



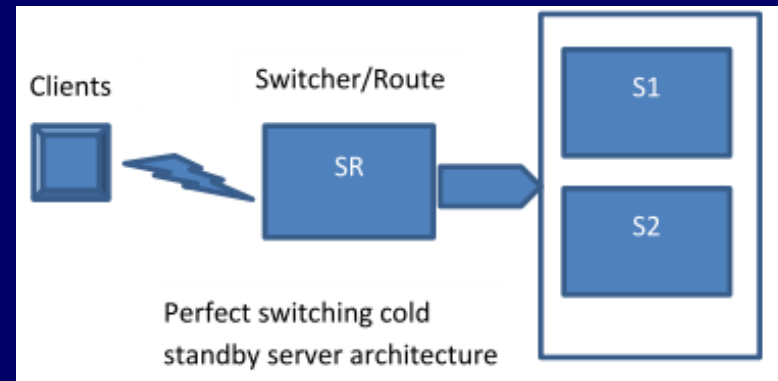
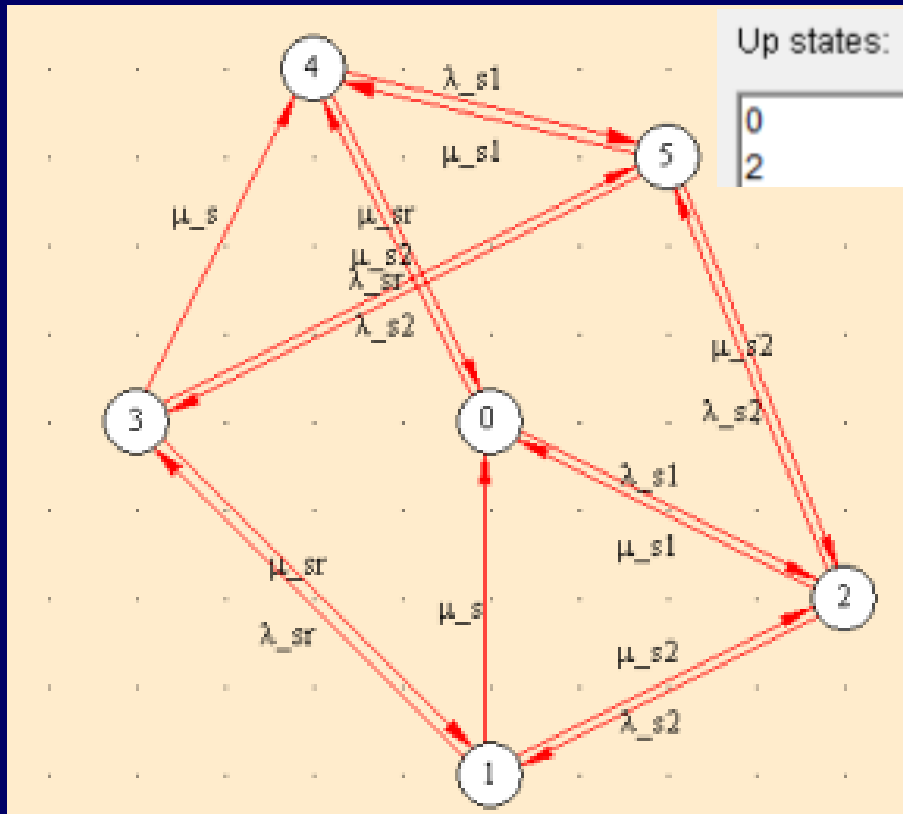
The respective CSM = Sync(SR,CS) is



# SPN

## Example CTMC availability model

The respective CTMC availability model is



System availability:

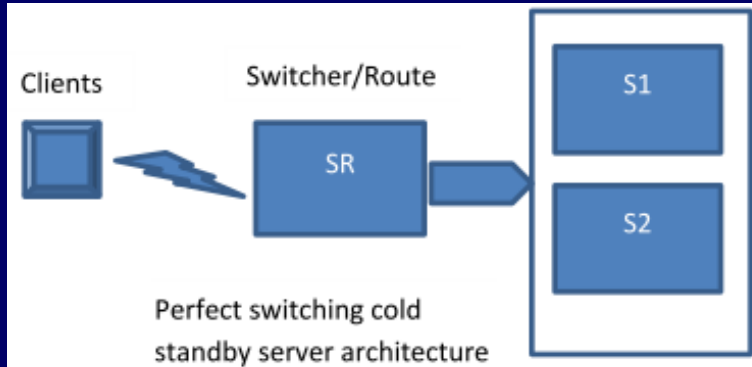
$$A=0.998799526$$

System unavailability:

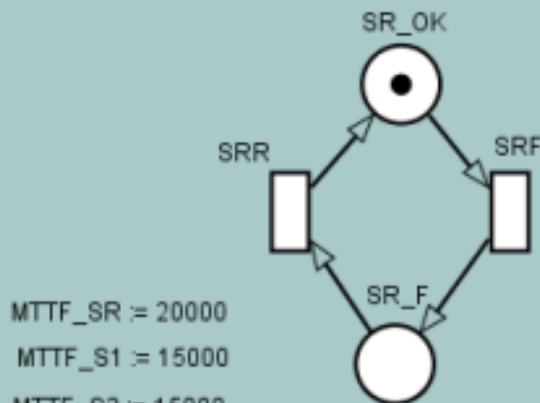
$$UA=0.00120047377$$

# SPN

## Example



### SPN availability model



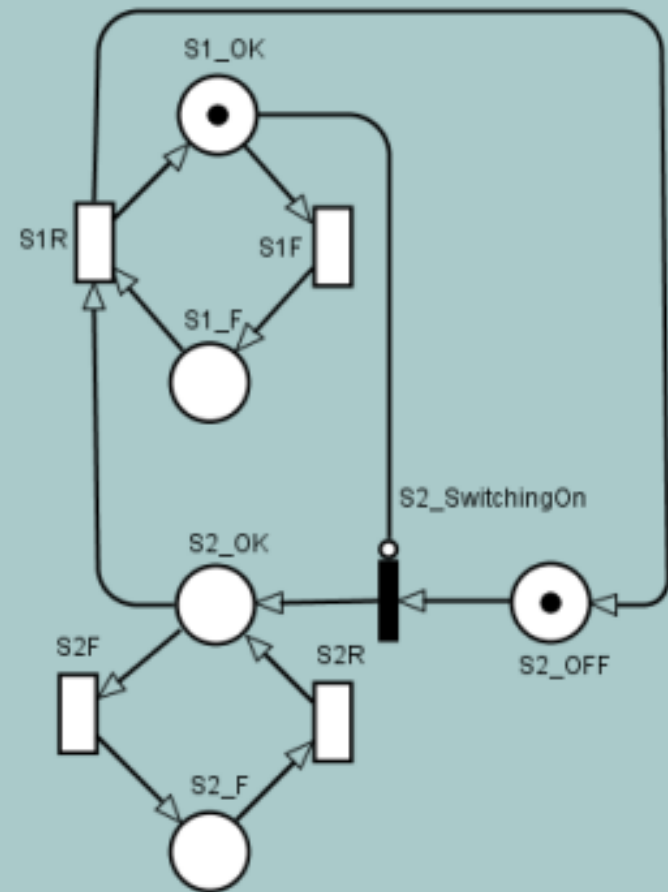
MTTF\_SR := 20000  
MTTF\_S1 := 15000  
MTTF\_S2 := 15000  
MTTR\_S1 := 24  
MTTR\_SR := 24  
MTTR\_S2 := 24

SystemUnavailability =  $P\{\#SR\_F=1 \text{ OR } (\#S1\_F=1 \text{ AND } \#S2\_F=1)\}$   
SystemAvailability =  $1 - P\{\#SR\_F=1 \text{ OR } (\#S1\_F=1 \text{ AND } \#S2\_F=1)\}$

#### Availability Results:

Steady State Unavailability = 0.0012011

Steady State Availability = 0.9987989

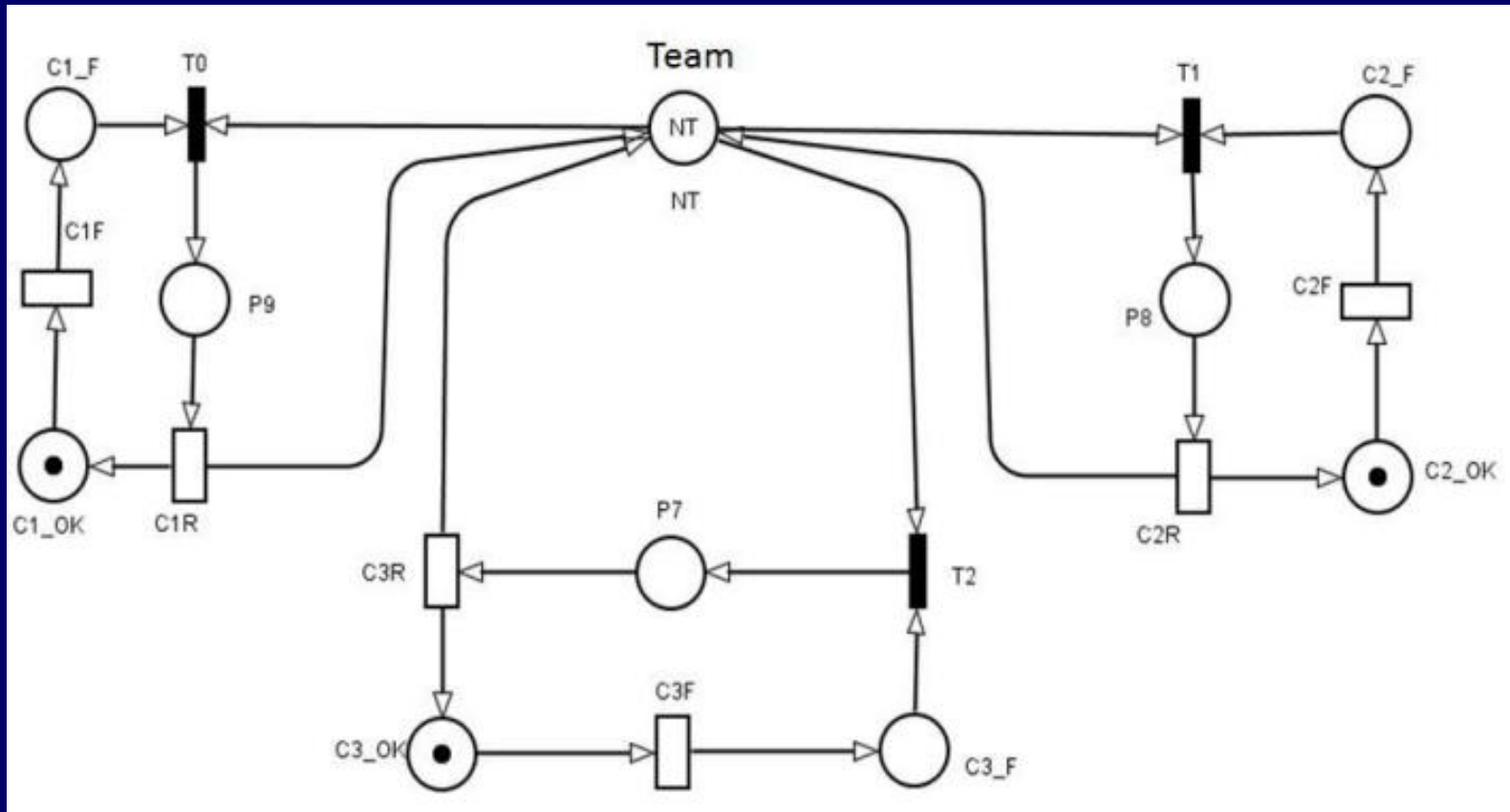




# SPN

Shared repair

**Corrective Maintenance**

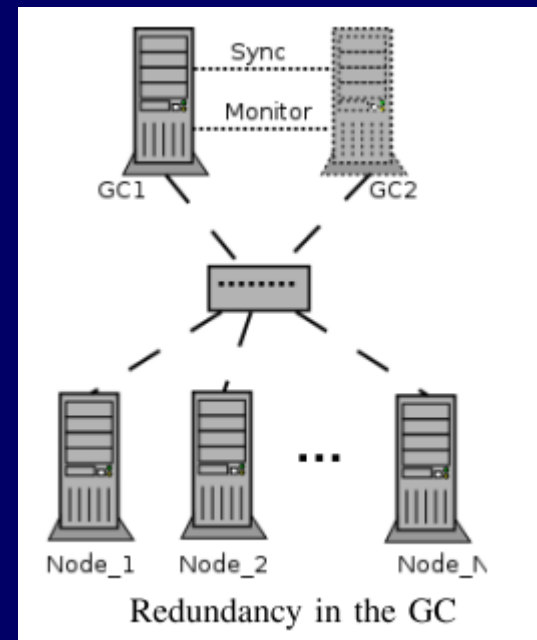
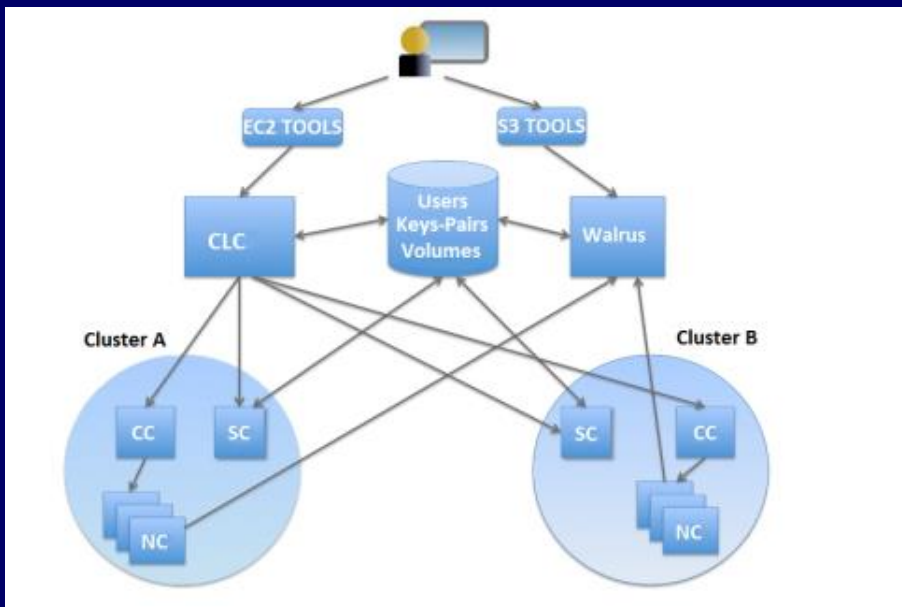




# **HIERARCHICAL MODELING**

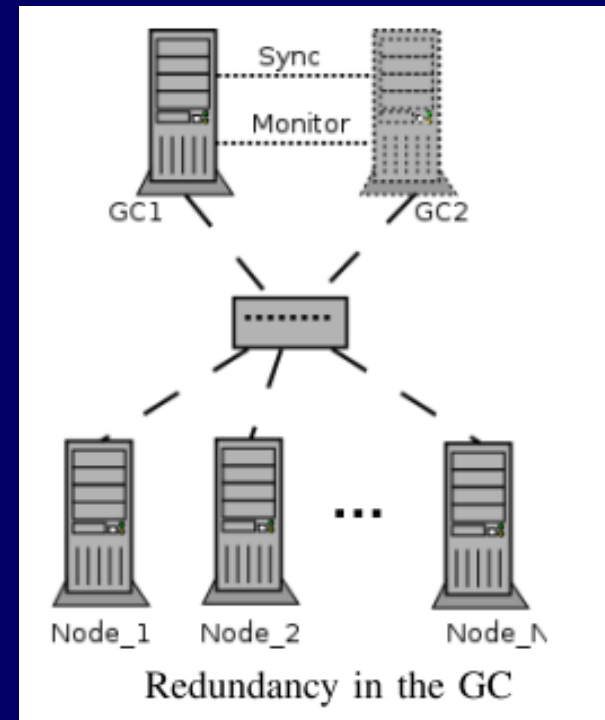
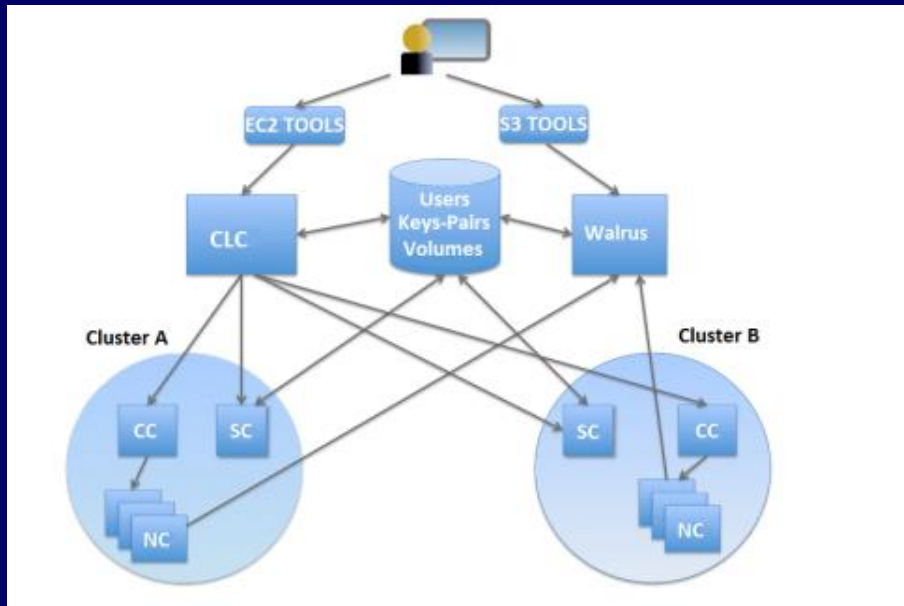
# Hierarchical Modeling

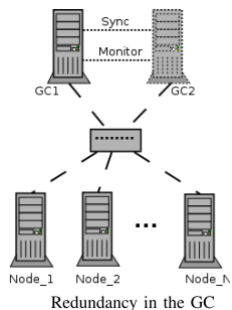
EUCALYPTUS is composed by five high-level components: Cloud Controller, Cluster Controller, Node Controller, Storage Controller, and Walrus. The Cloud Controller (CLC) is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage).



# Hierarchical Modeling

EUCALYPTUS is composed by five high-level components: Cloud Controller, Cluster Controller, Node Controller, Storage Controller, and Walrus. The Cloud Controller (CLC) is responsible for exposing and managing the underlying virtualized resources (servers, network, and storage).

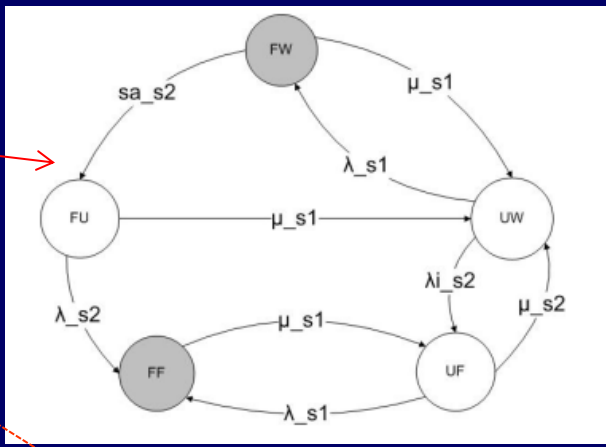
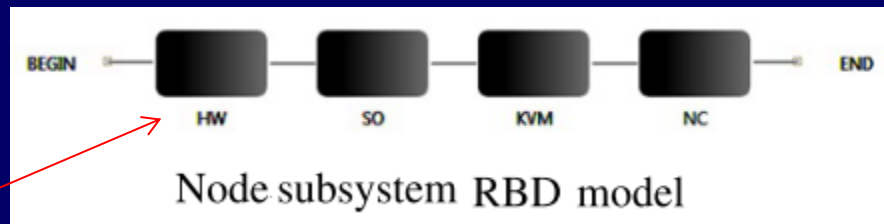
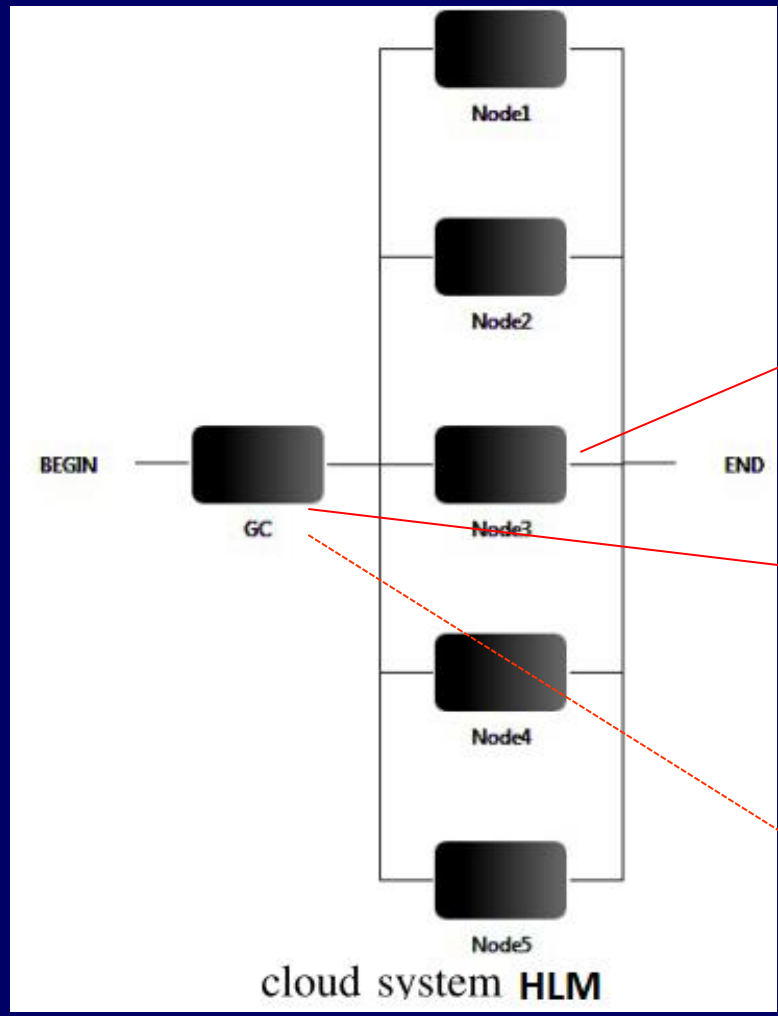




# Hierarchical Modeling

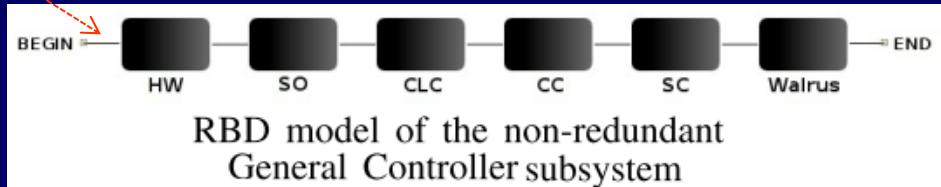
Component	MTTF	MTTR
KVM	2990 h	1 h
NC	788.4 h	1 h

Parameter	Description	Value
$\lambda_{s1} = \lambda_{s2} = 1/\lambda$	Mean time for host failure	1/180.721
$\lambda_{i_s2} = 1/\lambda_i$	Mean time for inactive host failure	1/216.865
$\mu_{s1} = \mu_{s2} = 1/\mu$	Mean time for host repair	1/0.9667
$sa_{s2} = 1/sa$	Mean time to system activate	1/0.005

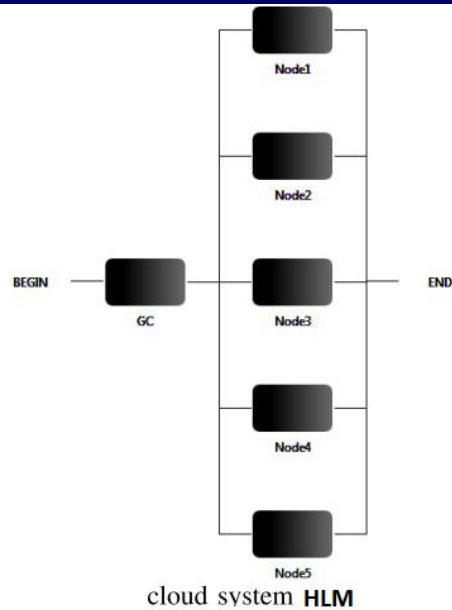


Component	MTTF	MTTR
HW	8760 h	100 min
SO	2893 h	15 min
CLC	788.4 h	1 h
CC	788.4 h	1 h
SC	788.4 h	1 h
Walrus	788.4 h	1 h

Redundant general controller subsystem



# Hierarchical Modeling



$$A_{GC} = \frac{\mu(\lambda_i(\mu + sa) + \mu^2 + sa(\lambda + \mu))}{\lambda_i(\lambda + \mu)(\mu + sa) + \mu^2(\lambda + \mu) + sa(\lambda^2 + \lambda\mu + \mu^2)}$$

$$A_{cloud} = A_{GC} * \left(1 - \prod_{i=1}^n (1 - A_{Node\_i})\right)$$

Measure	GC without redundancy	GC with redundancy
Steady-state availability	0.99467823178	0.99991793
Number of 9's	2.273944	4.08581
Annual downtime	46.66 h	0.72 h

