Sistemas Críticos

Paulo Maciel

Centro de Informática - UFPE

Programa

- Sistemas de Tempo Real (até o 08/10)
- Dependabilidade (de 15/10 até 26/11)

Objetivo

- É o estudo, fixação e aplicação de métodos e modelos para avaliação de sistemas críticos.

Programa

- Sistemas de Tempo Real (até o 08/10)
  - Características e requisitos
  - Categorias
  - Alocação de tarefas e escalonamento
  - Métricas de desempenho para sistemas de tempo real
  - Modelos
    - Álgebras de Procesos Temporizada
    - Redes de Petri Temporizadas
  - Análise e verificação e estimativa

Pré-requisitos

- Avaliação de Desempenho de Sistemas
- Modelos para Sistemas Comunicantes

Programa

- Dependabilidade (de 15/10 até 26/11)
  - História
  - Conceitos básicos e terminologia
  - Fundamentos

  Análise de Dados
  - Análise de tempo de vida
  - Modelos de aceleração de tempo de vida

Programa

- Dependabilidade (de 15/10 até 26/11)
  Modelagem
  - Mecanismos de detecção, recuperação e tolerância à falhas
  - Mantenabilidade
  - Sistemas coerentes
  - Modo de falha e operacional
  - Modelos combinacionais: RBD, FT, RG
    - Função estrutural e lógica
    - Métodos de análise
    - Modelagem
  - Cadeias de Markov e Redes de Petri Estocásticas
    - Modelagem
  - Modelagem hierárquica e heterogênea

Bibliografia Básica

- **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, 2nd 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 Computacão, 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. Avižienis, 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

Metodologia

- Aulas expositivas
- Aulas práticas.

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)].

Avaliação

- Resolução de listas.

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

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.]

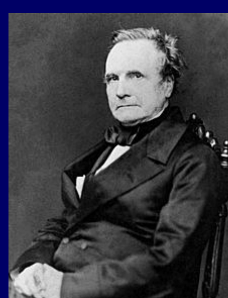
In 1905, H. G. Stott and H. R. Stuart: discuss "Time-Limit Relays and Duplication of Electrical Apparatus to Secure Reliability of Services at New York and at Pittsburg.

In these works the concept of reliability was primarily qualitative.

A Brief History

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 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 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

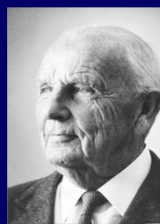
In 1910s, A. K. Erlang studied telephone traffic planning problems for reliable service provisioning [Erlang, A. K. (1909)].

[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.

Agner Karup Erlang

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)].



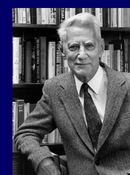
Waloddi Weibull 1887-1979

Gumbel, Emil Julius (18.7.1891 - 10.9.1966)

A Brief History

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. Shanon

John von Neumann

Edward Forrest Moore

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 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

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

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 Reliiability Theory Still Alive?. e-journal "Reliability: Theory& Applications", 1(2).

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.

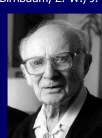
A. Avizienis

A Brief History

In 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)].



Zygmunt William Birnbaum

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.

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.

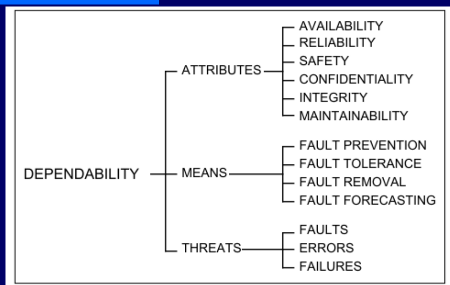
Afterwards, in 1962, Boeing and AVCO expanded use of FTA to the entire Minuteman II.

Minuteman I

Minuteman II

BASIC CONCEPTS

Basic Concepts



The dependability tree

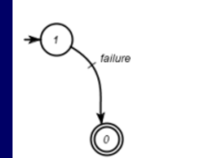
Avizienis, A., Laprie, J.-C., Randell, B. (2001). Fundamental Concepts of Dependability. LAAS-CNRS, Technical Report N01145.

Basic Concepts

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

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

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



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

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 it is specified.

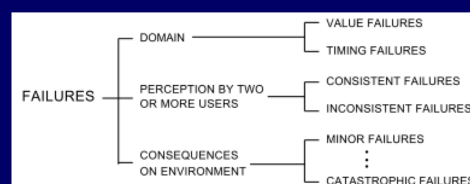
- A system failure is an event that occurs when the delivered service deviates from correct service.

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

- Failure Modes



Basic Concepts

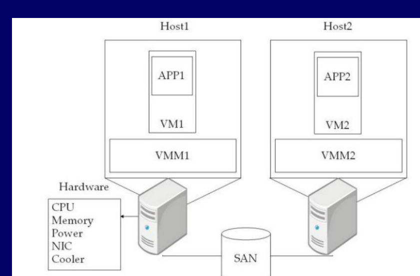
- An error is that part of the system state that may cause a subsequent failure.

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

$$\text{fault} \xrightarrow{\text{activation}} \text{error} \xrightarrow{\text{propagation}} \text{failure}$$

$$\cdots \rightarrow \text{fault} \xrightarrow{\text{activation}} \text{error} \xrightarrow{\text{propagation}} \text{failure} \xrightarrow{\text{causation}} \text{fault} \rightarrow \cdots$$

A motivational example

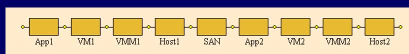


36

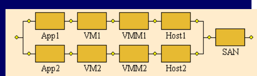
A motivational example

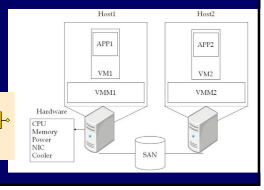
- What is the respective RBD?

This?



Or this?

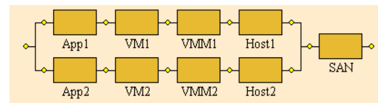




Operational Mode

- Operational Mode 2

$$OM_2 = ((App_1 \land VMM_1 \land VM_1 \land H_1) \lor (App_2 \land VMM_2 \land VM_2 \land H_2)) \land SAN$$



$R(t) = 0.975215145, \qquad t = 0.002 \ tu$

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)

38

Basic Concepts

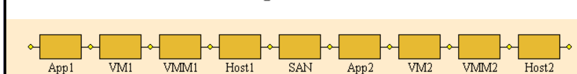
- Fault prevention: how to prevent the occurrence or introduction of faults;
- Fault tolerance: how to deliver correct service in the presence of faults;
- Fault removal: how to reduce the number or severity of faults;
- Fault forecasting: how to estimate the present number, the future incidence, and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

Operational Mode

is a condition that defines the system as operational.

- Operational Mode 1

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



$R(t) = 0.805735302, \qquad t = 0.002 \ tu$

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

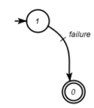
Fault Tolerance is intended to preserve the delivery of correct service in the presence of active faults.

- Active strategies
  Phase:
  1) Error detection
  2) Recovery
- Passive strategies
  Fault masking

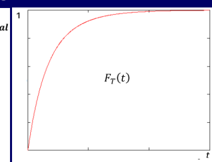
Basic Concepts

- Time to Failure

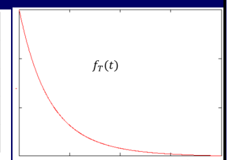
$X_S(t) = \begin{cases} 0, & \text{if } S \text{ has failed} \\ 1, & \text{if } S \text{ is operational} \end{cases}$



States of $X_S(t)$



$F_T(t)$ – Cumulative Distribution Function



$f_{T)}(t)$ - Density Function 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, $\boldsymbol{F_T(t)}$ its **cumulative distribution function**, and $\boldsymbol{f_T(t)}$ the respective **density function**, where:

$$F_T(0) = 0 \;\; and \;\; \lim_{t \to \infty} F_T(t) = 1,$$

$$\boldsymbol{f_T(t)} = \frac{dF_T}{dt}, \qquad \int_0^{\infty} f_T(t) \times dt = 1$$

Basic Concepts

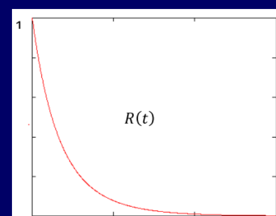
Fault Removal is performed both during the development phase, and during the operational life of a system.

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

- Reliability



$R(t)$ Reliability Function t

The **probability that the system S does not fail up to time** $\boldsymbol{t}$ (**reliability**) is

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

$$R(0) = 1 \;\; and \;\; \lim_{t \to \infty} R(t) = 0.$$

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).

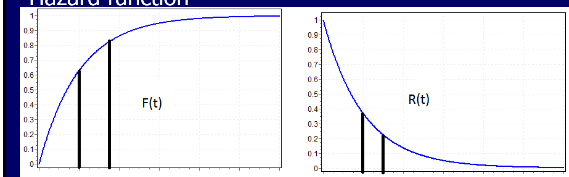
Basic Concepts

- Reliability

Reliability (Survivor function) - Complementary of the distribution function: $R(t) = 1 - F(t)$

Basic Concepts

- Hazard function

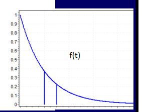


The probability of the system S fail within the interval $[t, t + \Delta t]$ may be calculated by:

$$P\{t \le T \le t + \Delta t\} = F_T(t + \Delta t) - F_T(t) =$$

$$R(t) - R(t + \Delta t) =$$

$$\int_t^{t+\Delta t} f_T(t)\, dt.$$



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

- 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

$$P\{t \le T(0) \le t + \Delta t | T > t\} =$$

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

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

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

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

Basic Concepts

- 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,

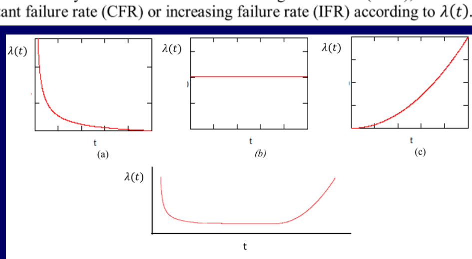
$$MTTF = E[T] = -\int_0^\infty \frac{dR(t)}{dt} \times t\, dt.$$

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

- 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

- Mean Time To Failure

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

$$-\left[0 - \int_0^\infty R(t) \times dt\right] = \int_0^\infty R(t) \times dt,$$

hence

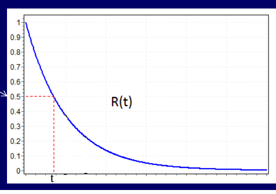
$$MTTF = \int_0^\infty R(t) \times dt$$

Basic Concepts

- Median Time To Failure

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

R(t)



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

- Mean Time To Repair

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

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

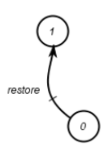
An alternative often easier to compute $MTTR$ is

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

Basic Concepts

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

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



States of $X_S(t)$

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

Therefore, the random variable **D** represents the system **time to repair**, $\boldsymbol{F_D(t)}$ its **cumulative distribution function**, and $\boldsymbol{f_D(t)}$ the respective **density function**

$F_D(0) = 0\ \ and\ \ \lim_{t \to \infty} F_D(t) = 1,$

$\boldsymbol{f_D(t)} = \frac{\boldsymbol{dF_{D(t)}}}{\boldsymbol{dt}},$

$f_D(t) \geq 0$, and

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

Basic Concepts

- Repairable Systems

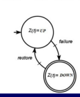
Consider a repairable system S that is either operational (Up) or faulty (Down). Whenever the system fails, a set of activities are conducted in order to allow the restoring process.

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.

$Downtime = TR = NRT + TTR.$

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



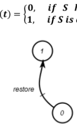
Basic Concepts

- Maintainability

The **probability that the system S will be repaired by** $\boldsymbol{t}$ is defined as **maintainability**.

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

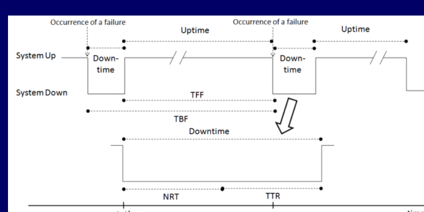
$$X_S(t) = \begin{cases} 0, & if\ \ S\ \ has\ failed \\ 1, & if\ S\ is\ operational \end{cases}$$



States of $X_S(t)$

Basic Concepts

- Downtime and Uptime



Downtime and Uptime

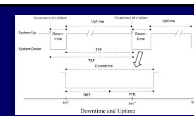
Basic Concepts

- 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

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:

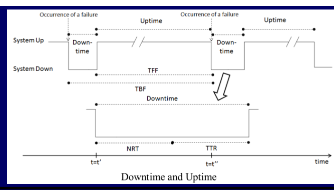
$$A = \frac{MTBF}{MTBF+MTTR}$$

Basic Concepts

- Availability

Consider that the system started operating at time $t = t'$ and fails at $t = t''$, thus $\Delta t = t'' - t' = Uptime$.
Therefore, the system availability may also be expressed by:

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



Basic Concepts

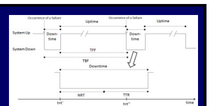
- Instantaneous Availability

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

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

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

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

- Steady State Availability

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

$$A = \lim_{t\to\infty} A(t), \; t \geq 0$$

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

68

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

69

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)

70

Remember these formulae

Exponential Distribution: EXP(λ)

- 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 λ is a paramter 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

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

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

71

Exponential Distribution: EXP(λ)

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

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

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

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

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

72

Exponential Distribution: EXP(λ)

The memoryless property can be demonstrated with conditional reliability:

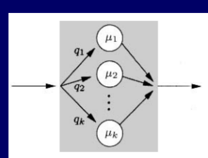
$$R(x \mid t) = \Pr(T > x + t \mid T > t) = \frac{\Pr(T > x + t)}{\Pr(T > t)}$$

$$= \frac{e^{-\lambda(t+x)}}{e^{-\lambda t}} = e^{-\lambda x} = R(x), \qquad x \geq 0.$$

73

Hyperexponential Distribution

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



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

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

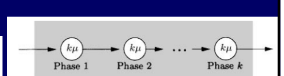
variance: $\mathrm{var}(X) = 2 \sum_{j=1}^{k} \frac{q_j}{\mu_j^2} - \frac{1}{\mu^2},$

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

74

Erlang Distribution

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



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

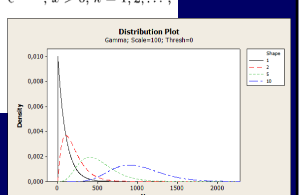
mean: $\overline{X} = \frac{1}{\mu},$

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

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

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

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



75

Hypoexponential Distribution

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

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

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

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

76

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 α

scale parameter $\lambda > 0$

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

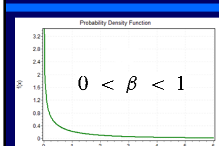
$$\overline{X} = \frac{1}{\lambda} \Gamma\left(1 + \frac{1}{\alpha}\right),$$

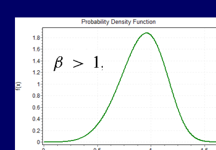
$$c_X^2 = \frac{\Gamma(1 + 2/\alpha)}{\{\Gamma(1 + 1/\alpha)\}^2} - 1$$

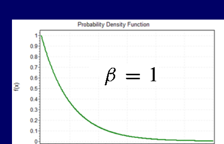
Weibull distribution is often used to model times to failure

77

Weibull Distribution







Lognormal Distribution

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

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

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

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

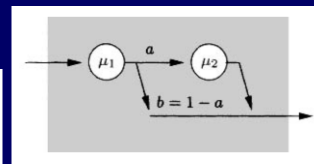
$$\alpha = \sqrt{\ln(c_X^2 + 1)}, \qquad \lambda = \ln \overline{X} - \frac{\alpha^2}{2}$$

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 $n \to \infty$.

Cox Distribution

Case 2: $c_X > 1$



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

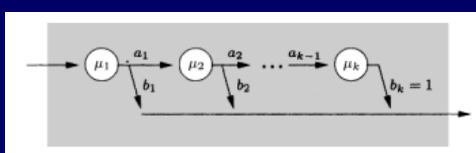
$$\mathrm{var}(X) = \frac{\mu_2^2 + a\mu_1^2(2-a)}{\mu_1^2 \cdot \mu_2^2},$$

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

$$\mu_1 = \frac{2}{\overline{X}} \qquad a = \frac{1}{2c_X^2}$$

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

Cox Distribution



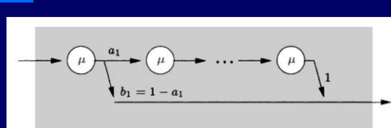
The model consists of k phases in series with exponentially distributed times and rates $\mu_1, \mu_2, \ldots, \mu_k$. After phase j, another phase $j+1$ follows with probability a_j and with probability $b_j = 1 - a_j$ the total time span is completed.

REDUNDANCY MECHANISMS

Cox Distribution

Case 1: $c_X \le 1$

$$\mu_j = \mu \quad j = 1, \ldots, k,$$
$$a_j = 1 \quad j = 2, \ldots, k-1$$



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

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

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

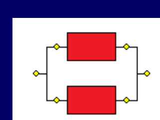
$$k = \left\lceil \frac{1}{c_X^2} \right\rceil$$

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

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

Redundancy Mechanisms

- Parallel Redundancy

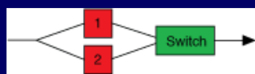


Parallel Redundancy refers to the approach of having multiply units running in parallel. All units are highly synchronized and receive the same input information at the same time.

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

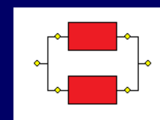


Deciding which unit is correct can be challenging if you only have two units. Sometimes you just have to choose which one you are going to trust the most and it can get complicated.

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

Redundancy Mechanisms

- Hot Standby

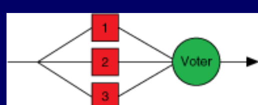


Some flavors of *Hot Standby* are similar to *Parallel Redundancy.*
These naming conventions are commonly interchanged.

For us, Hot Standby and Parallel Redundancy are the same mechanism!
But, attention!

Redundancy Mechanisms

- Triple Modular Redundancy (TMR)



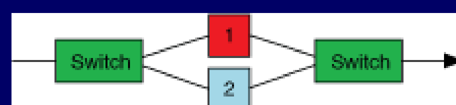
Deciding which unit is correct can be challenging if you only have two units. Sometimes you just have to choose which one you are going to trust the most and it can get complicated.

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

- Cold Standby



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

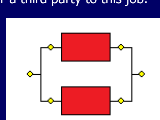
The drawback of this design is that standby unit have to power up, since it is initially powered off.

Perfect switching AND non-prefect switching

Redundancy Mechanisms

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

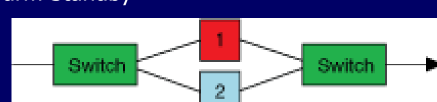
If you use the secondary unit as the watchdog and/or voter to decide when to switch over, you can eliminate the need for a third party to this job.



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

- Warm Standby



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

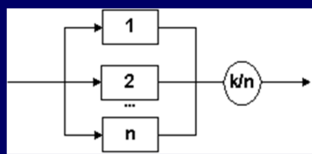
It is common to assume that in such a state the standby component has higher reliability than when receiving the workload (properly working).

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

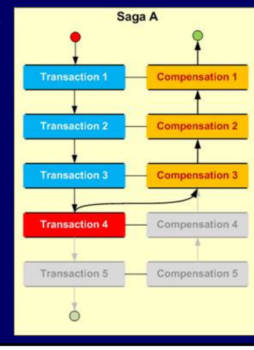


Consider a system composed of n identical and independent components that is operational if at least k out of its n components are working properly.

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

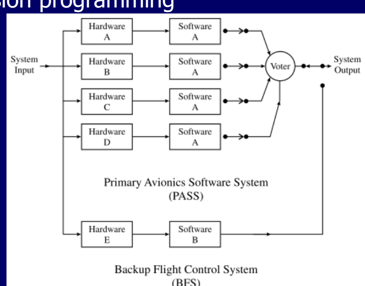
Redundancy Mechanisms

- Backward Recovery



Redundancy Mechanisms

- N-version programming



Hardware and software redundancy in the Space Shuttle's avionics control system.

Redundancy Mechanisms

- Reboot

The simplest - but weakest - recovery technique.
From the implementation standpoint is to reboot or restart the system.

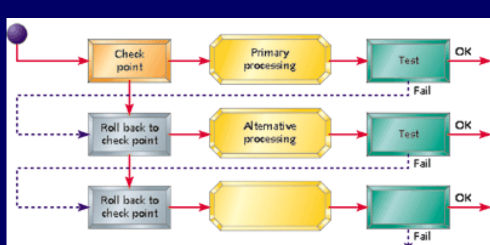
- Journaling - To employ these techniques requires that:

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.

Redundancy Mechanisms

- Checkpoints and recovering



COHERENT SYSTEM

Coherent System

Structure Function

Operations
- {+,-,×,÷} – 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:

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

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

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

$\phi(\mathbf{x})$ 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, $\phi(\mathbf{x}(\mathbf{t}))$, where $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_i(t), \dots, x_n(t))$.

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

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

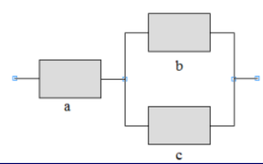
The first term $(x_i \, \phi(1_i, \mathbf{x}))$ represents a state where the component c_i is operational and the state of the other components are random variables $(\phi(x_1, x_2, \dots, 1_i, \dots, x_n))$. The second term $((1 - x_i) \, \phi(0_i, \mathbf{x}))$, on the other hand, states the condition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

Example - Structure Function

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



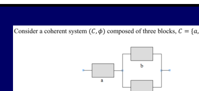
Coherent System

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



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,

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

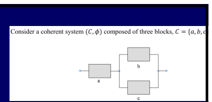
As $\phi(1_a, 1_b, x_c) = 1$, thus:

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

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

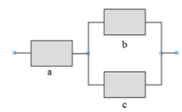
Since $\phi(1_a, 0_b, 1_c) = 1$ and $\phi(1_a, 0_b, 0_c) = 0$, thus:
$\phi(1_a, 0_b, x_c) = x_c$.
So
$\phi(x_a, x_b, x_c) = x_a \times [x_b + (1 - x_b)\, \phi(1_a, 0_b, x_c)] =$
$x_a \times [x_b + (1 - x_b)\, x_c] =$
$\phi(x_a, x_b, x_c) = x_a x_b + x_a x_c (1 - x_b) =$
$\phi(x_a, x_b, x_c) = x_a [1 - (1 - x_b)(1 - x_c)]$.

Coherent System

- Example – Converting a Logical Function into a Structure Function

Using the notation described, s_i is equivalent to x_i, $\overline{s_i}$ 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, ϕ) composed of three blocks, $C = \{a, b, c\}$



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

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

Coherent System

- Logical Function

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

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

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

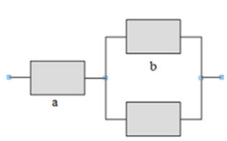
$\mathbf{bs} = (s_1, s_2, \ldots, s_i, \ldots, s_n)$ represents the Boolean state of each component of the system.
The system state could be either operational or failed.
The operational system state is represented by $\varphi(\mathbf{bs})$, whereas $\overline{\varphi(\mathbf{bs})}$ denotes a faulty system.

MODELING

Coherent System

- Example – Logical Function

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



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

Modeling Techniques

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

Combinatorial models

Reliability Block Diagram

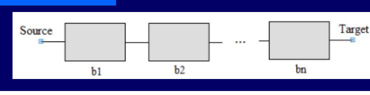
- Series

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

- 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

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

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

Likewise, the system instantaneous availability is

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

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

The steady state availability is

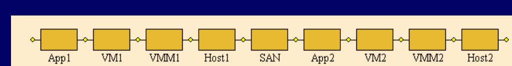
$A_S = P\{\phi(\mathbf{x}) = 1\} = \prod_{i=1}^{n} P\{x_i = 1\} = \prod_{i=1}^{n} A_i,$

where A_i is steady state availability of block b_i.

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.

Computing the Reliability



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

$$R(t) = 0.805735302, \qquad t = 0.002 \ tu$$

4

Reliability Block Diagram

- Series

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

- Example:

Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are λ1 = 0.00001 failures per hour, λ2 = 0.00002 failures per hour, λ3 = 0.00003 failures per hour, and λ4 = 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

- Series

R.v. X: series system life time

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

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

Case of *weakest link*

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

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

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

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.

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

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

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

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

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

Reliability Block Diagram

- Example:

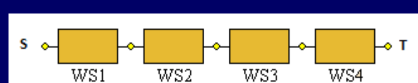
Assume that the constant failure rates of web services 1, 2, 3, and 4 of sw system are λ1 = 0.00001 failures per hour, λ2 = 0.00002 failures per hour, λ3 = 0.00003 failures per hour, and λ4 = 0.00004 failures per hour, respectively. The sw system cannot work when any one of the web services is down.

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:

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

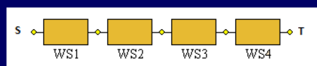


– a) $\sum_{i=1}^{n} \lambda_i$ $\quad \lambda_s = 0.00001 + 0.00002 + 0.00003 + 0.00004 = 0.0001$ failures per hour

– b) MTTF=$1/\sum_{i=1}^{n} \lambda_i$ $\quad \text{MTTF}_s = \frac{1}{0.0001} = 10{,}000$ h

Reliability Block Diagram

- Example:
  - c)



$$\phi(x_1,x_2,x_3) = x_1x_2x_3x_4$$

$$P\{\ \phi(x_1,x_2,x_3) = 1\} = E\{\phi(x_1,x_2,x_3)\} = E\{x_1x_2x_3x_4\}$$

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

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

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

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

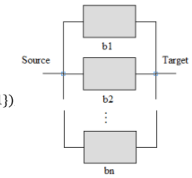
Reliability Block Diagram

- 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 svstem reliabilitv is then:

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

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

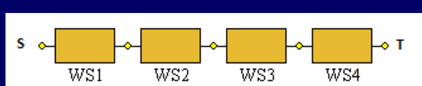
such that,

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

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

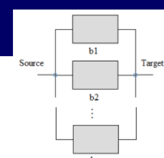
Reliability Block Diagram

- Problem:



Reliability Block Diagram

- Parallel



Similarly, the system instantaneous availability is

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

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

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

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

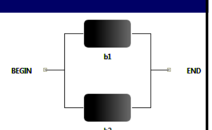
where A_i and UA_i are the steady availability and unavailability of block b_i, respectively.

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

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

Reliability Block Diagram

- Parallel



$$P\{\phi(\mathbf{x}) = 1\} + P\{\phi(\mathbf{x}) = 0\} = 1$$

$$P\{\phi(\mathbf{x}) = 1\} = 1 - P\{\phi(\mathbf{x}) = 0\}$$

$$\phi(\mathbf{x}) = 0$$

Hence, $\phi(\mathbf{x}) = 0 \Leftrightarrow \overline{\psi(\mathbf{s})}$

So, $\overline{\psi(\mathbf{s})} = \bar{s_1} \wedge \bar{s_2}$

Therefore: $\phi(\mathbf{x}) = 0 = (1 - x_1)(1 - x_2)$

So,

$$P\{\phi(\mathbf{x}) = 0\} = P\{(1 - x_1)(1 - x_2)\} =$$

$$P\{\phi(\mathbf{x}) = 0\} = E\{(1 - x_1)(1 - x_2)\} =$$

$$E\{(1 - x_1)\}E\{(1 - x_2)\} = (1 - p_1)(1 - p_2) = q_1q_2$$

$$P\{\phi(\mathbf{x}) = 1\} = 1 - P\{\phi(\mathbf{x}) = 0\}$$

$$P\{\phi(\mathbf{x}) = 1\} = 1 - q_1q_2$$

Or

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

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

Reliability Block Diagram

- Parallel

For a parallel system with n independent and identical components with rate λ

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

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

- Example

Reliability Block Diagram

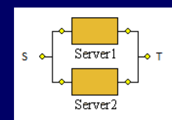
- Example

We know that

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

As

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



– b)

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

– c)

$$R(730h) = 0.9997906870$$
$$MTTF = 105\ 000h$$

Reliability Block Diagram

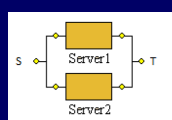
- Example

The system works when at least one server works.

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

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

$\varphi(s_1, s_2) = s_1 \vee s_2 \Leftrightarrow \overline{\varphi(s_1, s_2)} = \overline{s_1} \wedge \overline{s_2}$



We know that

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

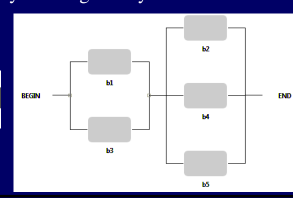
As

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

Reliability Block Diagram

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

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



Reliability Block Diagram

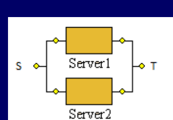
- Example

We know that

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

As

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



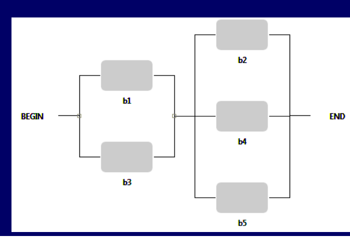
– a)

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

Reliability Block Diagram

- Series-Parallel System

Example:

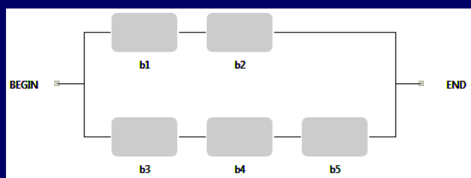


$$P = (1-(1-p_1)(1-p_3)) \times (1-(1-p_2)(1-p_4)(1-p_5))$$

Reliability Block Diagram

- Series-Parallel System

Example:



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

Reliability Block Diagram

- Problem

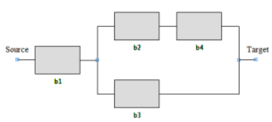
Now, considering the previous example, suppose that the repairing time of each web service is exponentially distributed with average 2h.

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

Reliability Block Diagram

- 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

- 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

$$S_1 = \{0, 1\}$$
$$S_2 = \{(0,0), (0,1), (1,0), (1,1)\}$$
$$S_n = \{2^n \ n\text{-tuples of 0s and 1s}\}$$

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

- K out of N

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

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

$$P(s) = P(A_1)P(A_2)\ldots P(A_k)P(\overline{A}_{k+1})\ldots P(\overline{A}_n)$$
$$= p^k q^{n-k}$$

$P(s)$: Prob. of sequence of k successes followed by $(n\text{-}k)$ 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,

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

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

- 2 out of 3

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

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

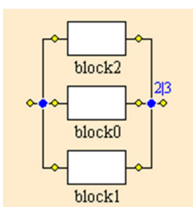
$$E[X] = \int_0^\infty R_{2oo3}(t)dt = \int_0^\infty 3e^{-2\lambda t}dt - \int_0^\infty 2e^{-3\lambda t}dt = \frac{5}{6\lambda} = MTTF_{2oo3}$$

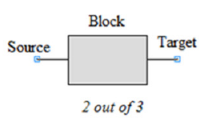
Comparing with expected life of a single component: $MTTF_{2oo3} = \frac{5}{6\lambda} < \frac{1}{\lambda} = MTTF_{Simplex}$

Reliability Block Diagram

Example: 2 out of 3 system

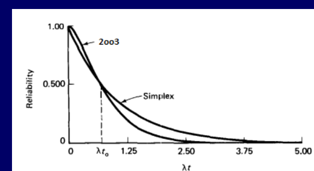
n statistically identical components; also statistically independent





Reliability Block Diagram

- 2 out of 3



Comparison of 2oo3 and simplex reliabilities

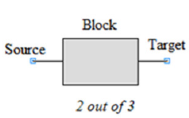
Thus 2oo3 actually reduces (by 16%) the MTTF over the simplex system.

Although 2oo3 has lower MTTF than does Simplex, it has higher reliability than Simplex for "short" missions, defined by mission time *t<(ln2)/λ*.

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

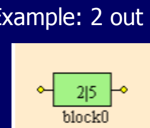
$$\sum_{i=2}^{3} \binom{3}{i} p^i (1-p)^{n-i} =$$

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

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

Reliability Block Diagram

- Example: 2 out of 5



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

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

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

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

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

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

Block Availability =

********* Outputs asked for the model: 2oo5 **************

Steady-State Availability

SS_Avail: 9.99540000e-001

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

- Structure Function

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

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

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

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

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

Fault Tree

- 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

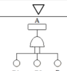
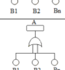
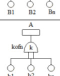
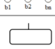
- Logical Function

FT Logic Function Ψ denotes the counterpart that represents the FT structure function (ψ) According to the notation previously introduced, s_i (a Boolean variable) is equivalent to x_i and $\overline{s_i}$ 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

- Basic Symbols

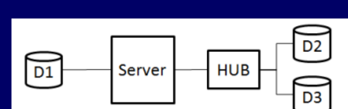
Basic Symbols and their description

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

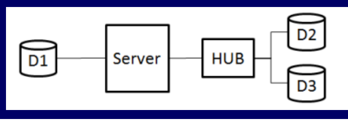
- Example

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.

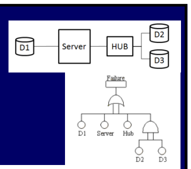
Fault Tree

- 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

Fault Tree



- Example

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

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

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

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.

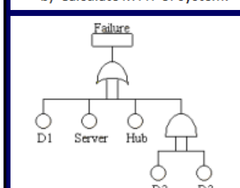
ANALYSIS METHODS

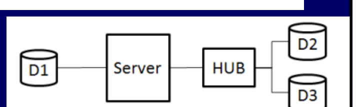
Fault Tree

- 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.

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

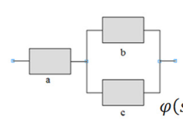




Analysis by Expected Value of the Structure Function

- The method by an example

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



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

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

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

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

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

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

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

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.

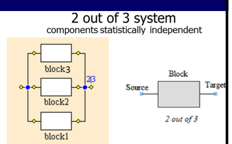
Summarizing, the main steps for computing the system failure probability, by adopting this method are:

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



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

Therefore

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

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

$$\Leftrightarrow$$

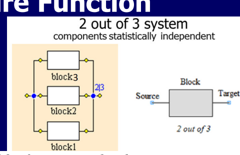
$$\phi(\mathbf{x}) = 1 - (1 - x_1x_2)(1 - x_1x_3)(1 - x_2x_3).$$

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

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

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

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

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 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:

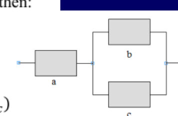
$$P\{\phi(\mathbf{x}) = 1\} = p_i \times P\{\phi(1_i, \mathbf{x}) = 1\} + (1 - p_i) \times P\{\phi(0_i, \mathbf{x}) = 1\}.$$

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:



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

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

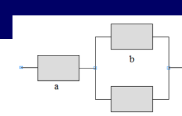
$$P\{\phi(x_a, x_b, x_c) = 1\} = p_a \times E[\phi(1_a, x_b, x_c)].$$

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

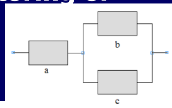
$$E[\,\phi(1_a, 0_b, x_c)] = E[x_c]\,E[\phi(1_a, 0_b, 1_c)] + E[(1 - x_c)]E[\phi(1_a, 0_b, 0_c)],$$

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:

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

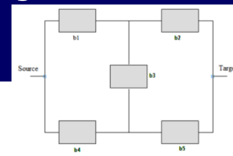
$$P\{\phi(x_a,x_b,x_c) = 1\} = p_a p_b + p_a p_c (1-p_b),$$

which is

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

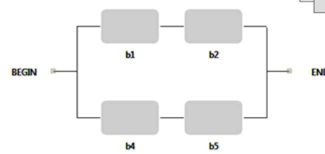
Pivotal Decomposition, Factoring or Conditioning

- Example – Bridge Structure



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

Configuration 2:

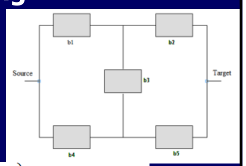


$$P\{\phi(\mathbf{x}) = 1\} = E[\phi(0_3,\mathbf{x})] = P\{\phi(0_3,\mathbf{x}) = 1\}$$

$$P\{\phi(0_3,\mathbf{x}) = 1\} = \big(1-(1-p_1p_2)(1-p_4p_5)\big)$$

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

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

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

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

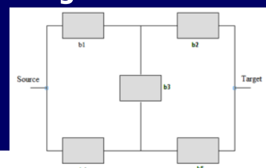
By independency

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

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

Pivotal Decomposition, Factoring or Conditioning

- Example – Bridge Structure



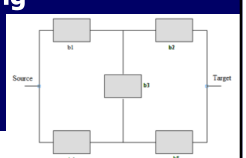
Therefore:

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

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

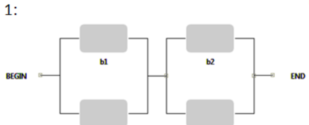
Pivotal Decomposition, Factoring or Conditioning

- Example – Bridge Structure



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

Configuration 1:

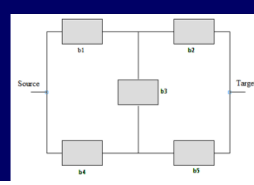


$$P\{\phi(\mathbf{x}) = 1\} = E[\phi(1_3,\mathbf{x})] = P\{\phi(1_3,\mathbf{x}) = 1\}$$

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

Pivotal Decomposition, Factoring or Conditioning

- Example – Bridge Structure



$$R_{bridge}(t) = e^{-(\lambda_2+\lambda_3+\lambda_4)t} - e^{-(\lambda_2+\lambda_3+\lambda_4+\lambda_5)t} + e^{-(\lambda_1+\lambda_3+\lambda_5)t}$$

$$+\,2e^{-\sum_i \lambda_i t} - e^{-(\lambda_1+\lambda_2+\lambda_3+\lambda_5)t} - e^{-(\lambda_1+\lambda_2+\lambda_3+\lambda_4)t} - e^{-(\lambda_1+\lambda_3+\lambda_4+\lambda_5)t}$$

$$+\,e^{-(\lambda_1+\lambda_2)t} + e^{-(\lambda_4+\lambda_5)t} - e^{-(\lambda_1+\lambda_2+\lambda_4+\lambda_5)t}$$

Pivotal Decomposition, Factoring or Conditioning

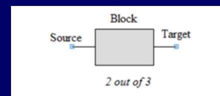
- Example – Bridge Structure

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

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

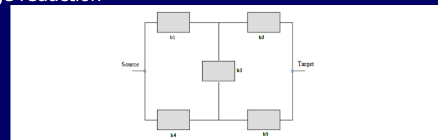
Reductions

- 2 out of 3 reduction



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

- 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

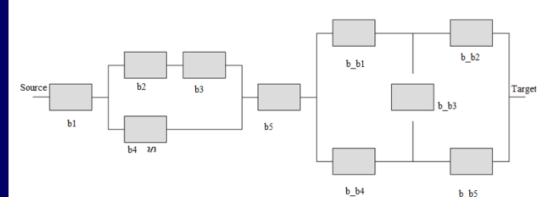
The dependability evaluation of complex system structures might be conducted iteratively by indentifying series, parallel, *k out of n* and *bridge* subsystems, evaluating each of those subsystems, and then reducing each subsystem to one respective equivalent block.

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

Reductions

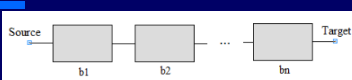
- Example

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.



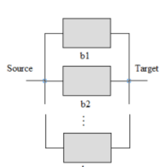
Reductions

- Series reduction



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

- Parallel reduction

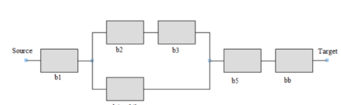


$$P\{\phi(\mathbf{x}) = 1\} = 1 - \prod_{i=1}^{n}(1 - p_i).$$

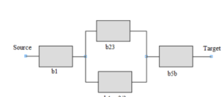
Reductions

- 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 $p_{bb} = (1 - (1 - p_{b1}p_{b2})(1 - p_{b4}p_{b5})(1 - p_{b1}p_{b3}p_{b5})(1 - p_{b2}p_{b3}p_{b4}))$.



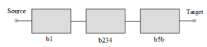
After that, two series reductions may be applied, one reducing blocks b_2 and b_3 into block b_{23}; and a second that combines blocks b_5 and b_b and reduces it to the block b_{5b}. The reliability of block b_{23} is $p_{23} = p_2 \times p_3$, and the block reliability of block b_{5b} is $p_{5b} = p_5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



Reductions

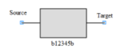
- Example

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 $p_{234} = 1 - (1 - p_2 \times p_3) \times (1 - 3p^2 - 2p^3)$.

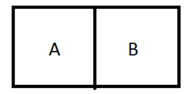


Finally, a final series reduction may be applied to RBD and one block RBD is generated, whose reliability is

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



Sum-of-Disjoint-Products (SDP) method



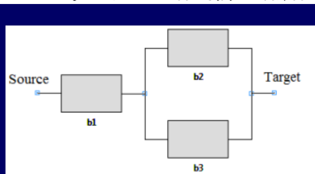
Disjoint Terms: Addition Law The addition law of probabilities is the underlying justification for the SDP method. If two or more events have no elements in common, the probability that at least one of the events will occur is the sum of the probabilities of the individual events. If two events A and 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 $\overline{A}B$, where $\overline{A}$ denotes the complement of A. Then we have the following equation for evaluation of the probability of $A \cup B$:

$$\Pr(A \cup B) = \Pr(A) + \Pr(\overline{A}B).$$

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 $SCS = \{c_1, c_2, \ldots, c_n\}$ is the set of components. A state vector $\mathbf{x}$ is named a **path vector** is $\phi(\mathbf{x}) = 1$, 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 $\phi(\mathbf{x}) = 0$, for any $\mathbf{y} < x$, and the respective path set is named **minimal path set**, that is $MPS(\mathbf{x}) = \{c_i | c_i \in PS(\mathbf{x}), \phi(\mathbf{x}) = 0 \ \forall \ \mathbf{y} < x\}$.



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

Sum-of-Disjoint-Products (SDP) method

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

$$\Pr(A \cup B \cup C) = \Pr(A) + \Pr(\overline{A}B) + \Pr(\overline{A}\,\overline{B}\,C).$$

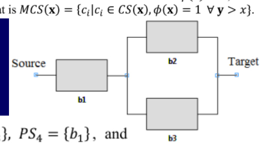
With n events $A_1, A_2, \ldots, A_n$, we have

$$\Pr(A_1 \cup \cdots \cup A_n) = \Pr(A_1) + \Pr(\overline{A}_1 A_2) + \Pr(\overline{A}_1 \overline{A}_2 A_3) + \cdots + \Pr(\overline{A}_1 \cdots \overline{A}_{n-1} A_n).$$

Computation Based on Minimal Paths and Minimal Cuts

- Cut and Minimal Cut

A state vector $\mathbf{x}$ is named a **cut vector** is $\phi(\mathbf{x}) = 0$, 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 $\mathbf{y} > x$, and the respective path set is named **minimal cut set**, that is $MCS(\mathbf{x}) = \{c_i | c_i \in CS(\mathbf{x}), \phi(\mathbf{x}) = 1 \ \forall \ \mathbf{y} > x\}$.



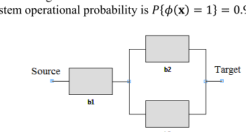
$PS_1 = \{b_1, b_2\}$, $PS_2 = \{b_1, b_3\}$, $PS_3 = \{b_1, b_2, b_3\}$, $PS_4 = \{b_1\}$, and $PS_5 = \{b_2, b_3\}$

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

- Example

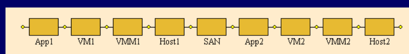
Consider the RBD presented, where the operational probabilities are $p_1 = p_2 = p_3 = 0.9$.
The minimal path sets and cuts are $S_1 = \{b_1, b_2\}$ and $S_2 = \{b_1, b_3\}$, and $S_4 = \{b_1\}$ and $S_5 = \{b_2, b_3\}$, respectively.
The operational probability computed in the first interaction of 48 when considering the minimal path S_1 is 0.980296.
If the minimal cuts are adopted instead of paths, and if S_4 is the first, operational probability is 0.990099. So, $P\{\phi(\mathbf{x}) = 1\} \in [0.980296, 0.990099]$. In the second interaction, operational probability calculated considering the ${S_1}^c \cap S_2$ is 0.9900019. When adopting the cuts, the next (and sole) disjoint product is ${S_4}^c \cap S_5$. The operational probability computed considering the additional term is 0.9900019. The reader may observe that the two bounds converged. Thus, the system operational probability is $P\{\phi(\mathbf{x}) = 1\} = 0.9900019$.



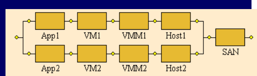
Computing the Reliability

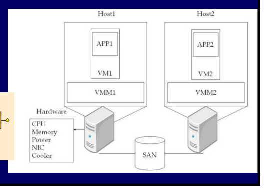
- What is the respective RBD?

This?

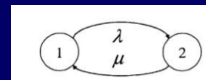


Or this?





CTMC



A simple 2-state CTMC

Single Component System Availability Model

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

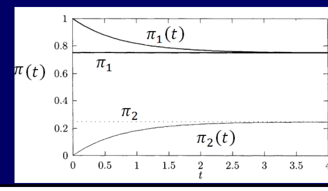
$$\pi_2(t) = \pi_2 = \frac{\lambda}{\lambda + \mu}, t \to \infty$$

$A = \pi_1$

Steady state availability

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

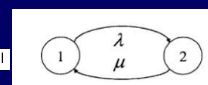
Figure shows the transient and steady-state behavior of the 2-state CTMC for $3\lambda = \mu = 1$.



State-space based models

CTMC

Single Component System Availability Model



A simple 2-state CTMC

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

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

$A = \pi_1$

Steady state availability

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

T – time period

Downtime

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

hours in a year

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

minutes in a year

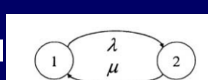
CTMC

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 λ and μ, respectively, the CTMC that represents the system availability model is



A simple 2-state CTMC

$$\pi_1(0) = 1$$

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

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

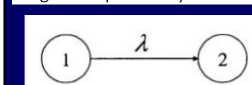
$$\pi_1(t) + \pi_2(t) = 1$$

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

Instantaneous availability

CTMC

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

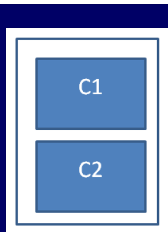
$$R(t) = \pi_1(t) = 0, t \to \infty$$

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

CTMC

Two Component System

Availability Model



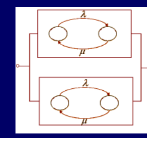
Two-component parallel redundant system with the same repair rate μ and the same failure rate for both components is (λ).

When both the components fail, the system fails.

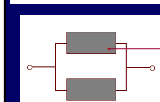
CTMC

Two Component System

Availability Model



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



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

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

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

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

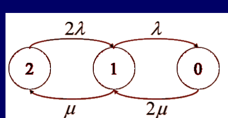
$$A_{sys} = 1 - \left(1 - \frac{\mu}{\lambda + \mu}\right)^2 = 1 - \left(\frac{\lambda}{\lambda + \mu}\right)^2$$

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

CTMC

Two Component System

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

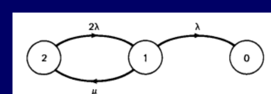
Downtime

CTMC

Two Component System

Reliability Model

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



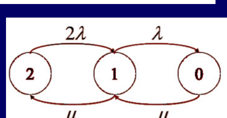
$$R(t) = 1 - \pi_0(t)$$

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

CTMC

Two Component System

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

- Example – Availability model

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

Failure rate of each machine is λ

Repair rate is μ

CTMC

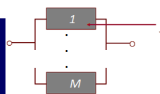
- Example – Availability model

M similar machines independent repair facility

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

Failure rate of each machine is λ

Repair rate is μ



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

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

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

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

System availability is then computed using a combinatorial approach

CTMC

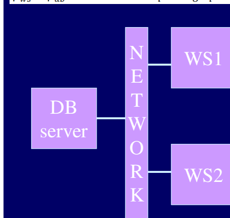
- Example

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 (λ_{ws}, λ_{db} respectively), and the respective time to repair are exponentially distributed with rate μ_{ws} and μ_{db}.

$\lambda_{ws} = 1.14 \times 10^{-4}$ *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

- Example – Availability model

An equivalent 2-state availability model

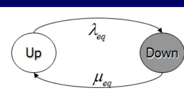
It is interesting to consider an equivalent 2-state availability model that has the same steady state availability as the given multi-state availability model.

To represent system availability in the simple form of equivalent 2-state system, we need to properly define equivalent failure rate λ_{eq} and equivalent repair rate μ_{eq}, such that

$$A = \frac{MTTF_{eq}}{MTTF_{eq} + MTTR_{eq}} = \frac{\mu_{eq}}{\lambda_{eq} + \mu_{eq}}$$

Failure rate of each machine is λ

Repair rate is μ

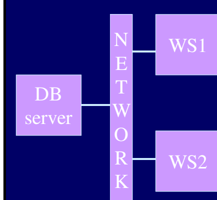


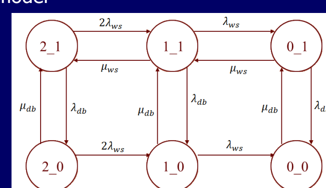
$$\lambda_{eq} = \frac{\lambda \pi_{M-1}}{\pi_0 + \pi_1 + \pi_2 + ... + \pi_{M-1}}$$

$$\mu_{eq} = \mu$$

CTMC

- Example – Availability model





$A = \pi_{2\_1} + \pi_{1\_1} = 0.994547080$

$Downtime = (1 - A) \times T = 2866.05467$ *minutes*

$T = 8760h \times 60min = 525{,}600$ *minutes* in one year.

$\lambda_{ws} = 1.14 \times 10^{-4}$ *failures per hour*

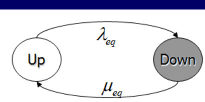
$\lambda_{db} = 2.28 \times 10^{-4}$ *failures per hour*

$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2}$ *repairings per hour*

CTMC

- Example – Availability model

An equivalent 2-state availability model



$$\lambda_{eq} = \frac{\lambda \pi_{M-1}}{\pi_0 + \pi_1 + \pi_2 + ... + \pi_{M-1}}$$

$$\mu_{eq} = \mu$$

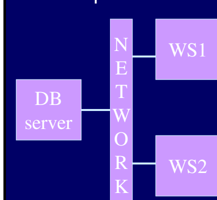
Let U be the set of up states, D the set of down states, R the set of all transitions from U to D, G the set of all transition from D to U, t_{ij} the transition from state i to j

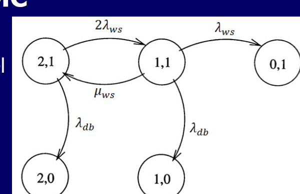
$$\lambda_{eq} = \sum_{t_{ij} \in R} P(\text{system in state } i \mid \text{system is up}) \times q_{ij} = \frac{\sum_{t_{ij} \in R} \pi_i \times q_{ij}}{\sum_{k \in U} \pi_k}$$

$$\mu_{eq} = \sum_{t_{ij} \in G} P(\text{system in state } i \mid \text{system is down}) \times q_{ij} = \frac{\sum_{t_{ij} \in G} \pi_i \times q_{ij}}{1 - \sum_{k \in U} \pi_k}$$

CTMC

- Example – 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.

$$R(t) = \pi_{2,1}(t) + \pi_{1,1}(t)$$

$\lambda_{ws} = 1.14 \times 10^{-4}$ *failures per hour*

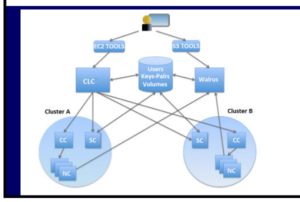
$\lambda_{db} = 2.28 \times 10^{-4}$ *failures per hour*

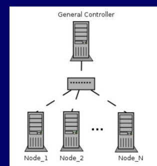
$\mu_{ws} = \mu_{db} = 4.17 \times 10^{-2}$ *repairings per hour*

CTMC

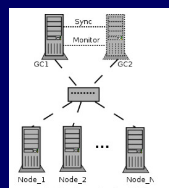
■ Example - Availability model

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 th underlying virtualized resources (servers, network, and storage).





No redundancy



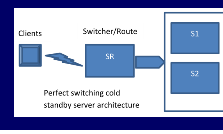
Redundancy in the GC

CTMC

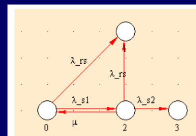
■ Example – Reliability model

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}.

λ_{rs} is failure rate of the Switcher/Router.

λ_{s1} is failure rate of the Server 1.

λ_{s2} is failure rate of the Server 2.

μ is the repair rate assigned to Server 1 repair activity.

System Unreliability:

UR(4000h)= 0.181615244

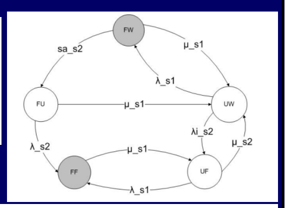
System Reliability:

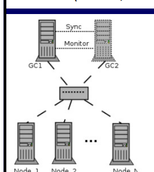
R(4000h) = 0.818384756

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 th underlying virtualized resources (servers, network, and storage).





Redundancy in the GC

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

■ 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 HYPO(λ_1, λ_2) distribution.

Time to trigger inspection is assumed to be EXP(λ_{in}),

Time to carry out inspection is EXP(μ_{in}),

Time to repair is EXP(μ),

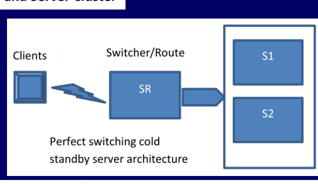
Time to carry out PM is EXP($\gamma\mu$).

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

■ Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

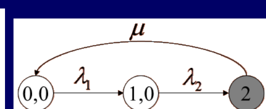
CTMC with corrective maintenance only

Time to failure is HYPO(λ_1, λ_2);

(0,0) & (1,0) are up states;

2 is a down state

Time to corrective maintenance is EXP(μ)



CTMC

- Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.

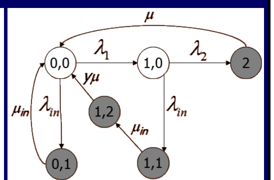
CTMC with preventive maintenance

Inspection triggered after EXP(λ_{in}) intervals

Time to carry out inspection is EXP(μ_{in})

Time to carry out PM is EXP($\gamma\mu$)

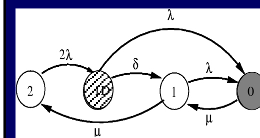
PM carried out if inspection finds the system to be in degraded state (1,0)



$A = \pi_{0,0} + \pi_{1,0}$

CTMC

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



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

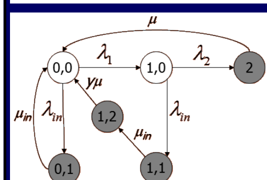
$$U(\delta, t_{th}) = \pi_0 + e^{-\delta t_{th}} \pi_{1D}$$

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

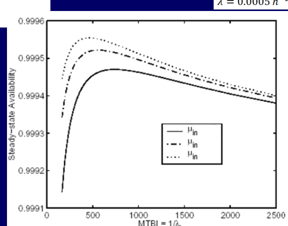
CTMC

- Preventive Maintenance

Preventive maintenance is useful when the time to failure distribution has an increasing failure rate.



$\lambda_1 = 0.001\ h^{-1}$
$\lambda_2 = 0.001\ h^{-1}$
$\mu_{in} = 10\ h^{-1}$
$\mu = 0.1\ h^{-1}$
$y = 5$
$\lambda = 0.0005\ h^{-1}$



$A = \pi_{0,0} + \pi_{1,0}$

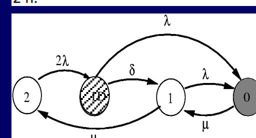
MTBI – mean time between inspections

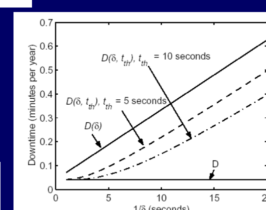
Availability as function of MTBI=$1/\lambda_{in}$

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.





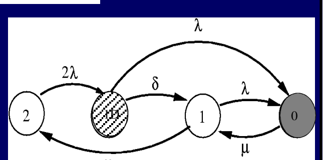
$$U(\delta, t_{th}) = \pi_0 + e^{-\delta t_{th}} \pi_{1D}$$

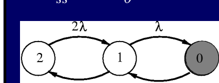
CTMC

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

- 2-equal component cold standby availability model with perfect switching

  $A_{ss} = 1 - \pi_0$



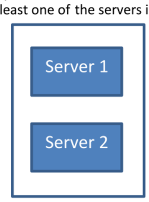


$$A = \pi_2 + \pi_1 + r\pi_{1D}$$

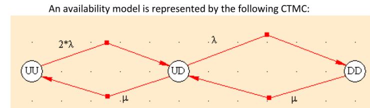
CTMC

Capacity oriented availability

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

Capacity oriented availability

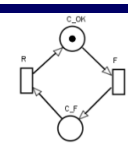
Now, if the users are interested not as much whether the system is operational or not, but rather in the service capacity the system may deliver. Considering the depicted architecture, it is assumed that if the two servers are operational, the system may deliver its full service capacity. If only one server is operational, the system may deliver only half of it service capacity. And when none of the servers is operational, the system may not deliver the service. Therefore Capacity Oriented Availability (COA) is:

$$COA = 2 \times \pi(UU) + \pi(DU).$$

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

SPN

Single Component System Availability Model



Transition	Time	Time	Type of Service
F	*MTTF*	λ	*single Server*
R	*MTTR*	μ	*single Server*

The instantaneous availiability :

$$A(t) = P\{(m(C\_OK) = 1)(t)\}$$

$$= \sum_{\forall M_i \in RS} r_i \times \pi_i(t) = \frac{\lambda e^{-t(\lambda+\mu)} + \mu}{\lambda + \mu}$$

Downtime in period T :

$$DT = T \times P\{(m(C\_F) = 1)\} = T \times \left(1 - \frac{\lambda}{\lambda+\mu}\right)$$

The stationary availability :

$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i = \frac{\lambda}{\lambda + \mu}$$

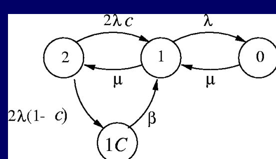
$$r_i = \begin{cases} 1 & se \; m_i(C\_OK) = 1 \\ 0 & se \; m_i(C\_OK) = 0 \end{cases}$$

CTMC

2-equal component cold standby 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.

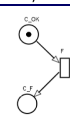


$$U(\beta, c) = \pi_0 + \pi_{1C} = \frac{\lambda\beta + \mu^2(1-c)}{\mu\beta E} \qquad (E = \rho\pi_0^{-1})$$

$$D(\beta, c) = U(\beta, c) \times 8760 \times 60 \text{ (down time in min/year)}$$

SPN

Single Component System Reliability Model



Transition	Time	Time	Type of Service
F	*MTTF*	λ	*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 transiton 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

$$r_i = \begin{cases} 1 & se \; m_i(C\_OK) = 1 \\ 0 & se \; m_i(C\_OK) = 0 \end{cases}$$

CTMC

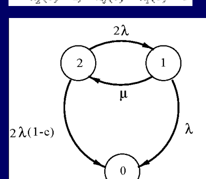
2-equal component cold standby without perfect switching with imperfect coverage reliability model

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

1C state is a reboot (down) state.

Assume that the initial state is 2 so that:

$$\pi_2(0) = 1, \quad \pi_0(0) = \pi_1(0) = 0$$



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

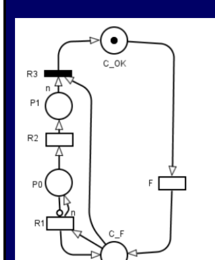
$$MTTF = \frac{\lambda(1+2c) + \mu}{2\lambda[\lambda + \mu(1-c)]}$$

It should be clear that the system *MTTF* and system reliability are critically dependent on the coverage factor.

SPN

Basic Model with Erlang Distributed Repair Time

Availability Model



$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i,$$

$$r_i = \begin{cases} 1 & se \; m_i(C\_OK) = 1 \\ 0 & se \; m_i(C\_OK) = 0 \end{cases}$$

$E[T_E] = \overline{X}$ e $DP[T_E] = SD$

$$n = \left(\frac{\overline{X}}{DP}\right)^2$$

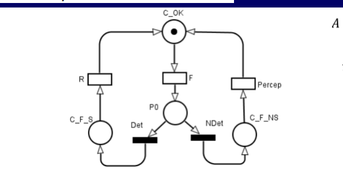
$$\lambda = \frac{n}{\overline{X}}$$

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



$$A = P\{(m(C\_OK) = 1)\} = \sum_{\forall M_i \in RS} r_i \times \pi_i$$

$$r_i = \begin{cases} 1 & se \ m_i(C\_OK) = 1 \\ 0 & se \ m_i(C\_OK) = 0 \end{cases}$$

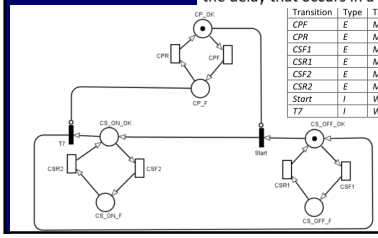
Transition	Type	Time or Wieght	Rate	Type of Service
F	*E*	*MTTF*	λ	*single server*
Det	*I*	W_{Det}		
Ndet	*I*	W_{NDet}		
Percep	*E*	*MTTP*	β	*single server*
R	*E*	*MTTR*	μ	*single server*

Failure Coverage Basic Model

SPN

Warm Standby
Availability Model

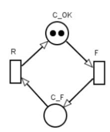
The *Warm Standby* model is similar to the *Cold Standby* model. However, in a system with *Warm Standby* redundancy, the reserve component remains energized (but inoperative), so that, when the main component fails, the reserve component takes over operations without the delay that occurs in a *Cold Standby* system.



Transition	Type	Time or Weight	Rate	Type of Service	Priority
CPF	*E*	*MTTF_CP*	λ	*single server*	
CPR	*E*	*MTTR_ CP*	μ	*single server*	
CSF1	*E*	*MTTF1_CS*	α	*single server*	
CSR1	*E*	*MTTR1_ CS*	β	*single server*	
CSF2	*E*	*MTTF2_CS*	α	*single server*	
CSR2	*E*	*MTTR2_ CS*	β	*single server*	
Start	*I*	*W=1*			*1*
T7	*I*	*W=1*			*1*

SPN

Hot Standby Model
Availability Model



$$A = P\{(m(C\_OK) = 2) \vee (m(C\_OK) = 1)\}$$

$$= \sum_{\forall M_i \in RS} r_i \times \pi_i = 1 - \frac{2\lambda^2}{2\lambda^2 + 2\lambda\mu + \mu^2},$$

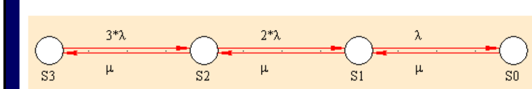
$$r_i = \begin{cases} 1 & se \ (m(C_{OK}) = 2) \vee (m(C_{OK}) = 1) \\ 0 & se \ m_i(C\_OK) = 0. \end{cases}$$

Transition	Type	Rate	Type of Service
F	*MTTF*	λ	*infinity Server*
R	*MTTR*	μ	*single Server*

SPN

2 out of 3 with shared repair
Availability Model

The CTMC model:



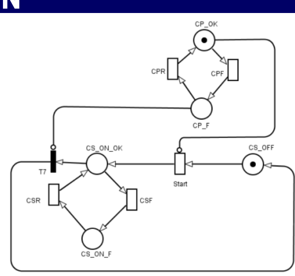
$$\lambda = \frac{1}{8760} h^{-1}$$

$$\mu = \frac{1}{24} h^{-1}$$

Availability=$\pi(S_3) + \pi(S_2) =$ 9.99955210e-001

SPN

Cold Standby
Availability Model



Transition	Type	Time or Weight	Rate	Type of Service
CPF	*E*	*MTTF_CP*	λ	*single server*
CPR	*E*	*MTTR_ CP*	μ	*single server*
CSF	*E*	*MTTF_CS*	α	*single server*
CSR	*E*	*MTTR_ CS*	β	*single server*
Start	*E*	*TTS*	μ	*single server*
T7	*I*	*W=1*		

The stationary availability of the component is calculated by the expression:

$$A = P\{((m(CP\_OK) = 1) \vee (m(CS\_ON\_OK) = 1))\} = \sum_{\forall M_i \in RS} r_i \times \pi_i$$

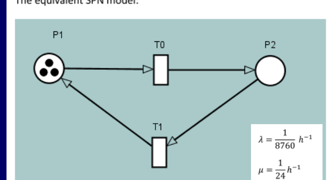
where r_i is a function that

$$r_i = \begin{cases} 1 & se \ (m(CP\_OK) = 1) \vee (m(CS\_ON\_OK) = 1) \\ 0 & se \ (m(CP\_OK) = 0) \wedge (m(CS\_ON\_OK) = 0) \end{cases}$$

SPN

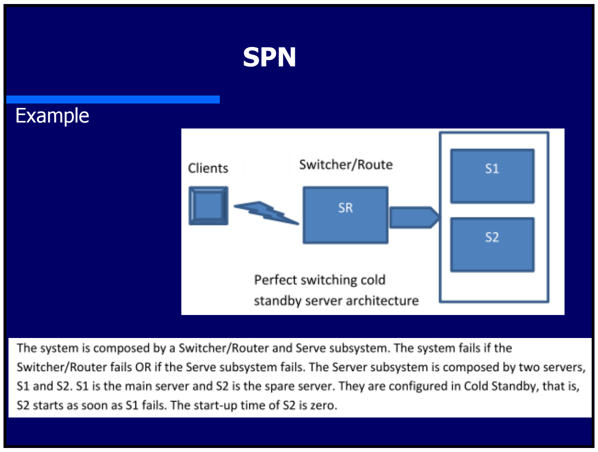
2 out of 3 with shared repair
Availability Model

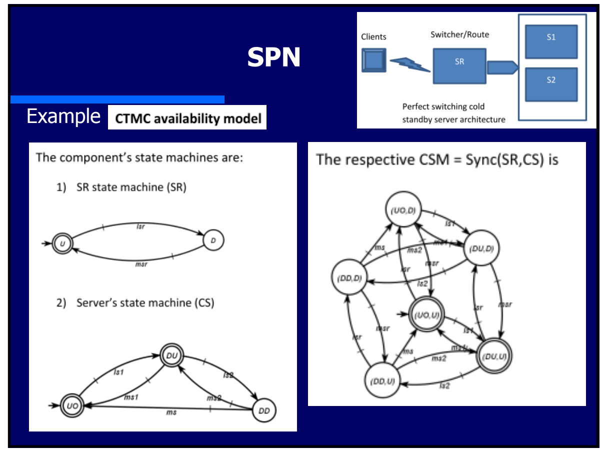
The equivalent SPN model:

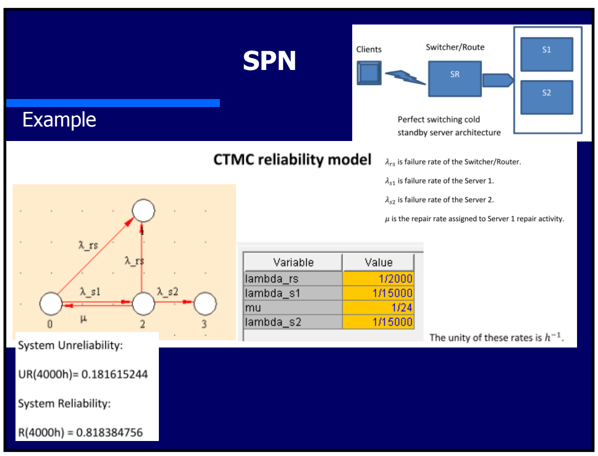


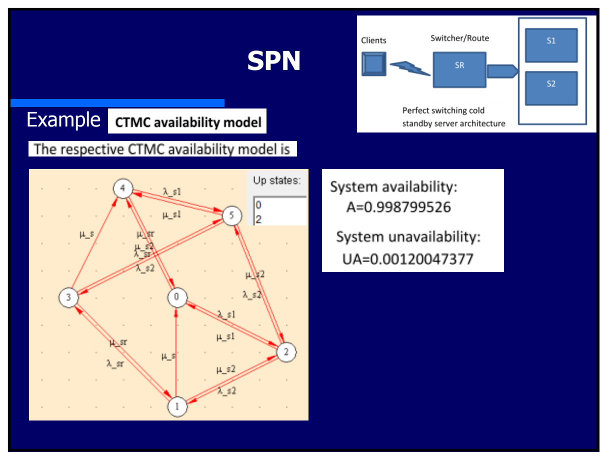
The result obtained through TimeNET:

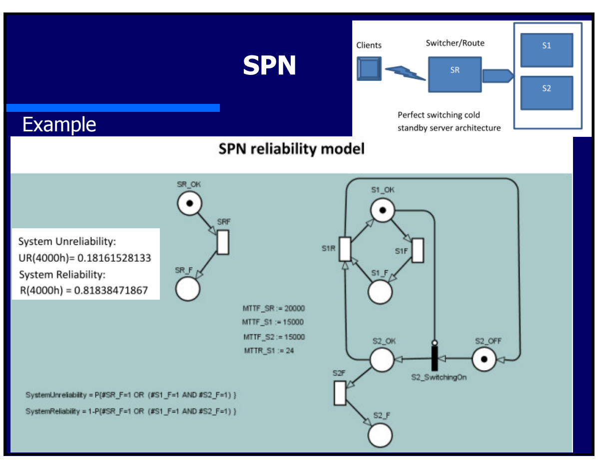
The Availability = P{#P1≥ 2} = 0.9999552.

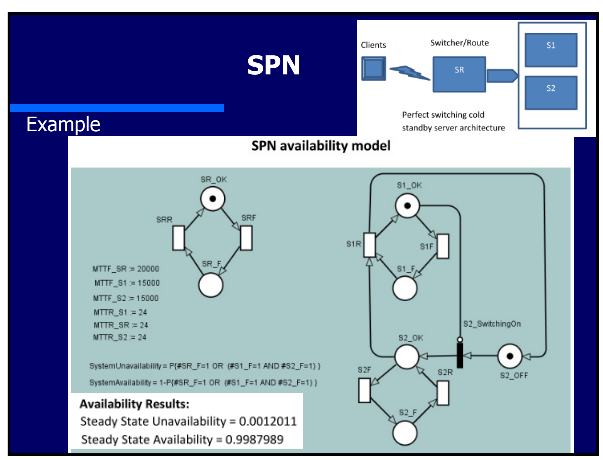

SPN
Example
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
The system is composed by a Switcher/Router and Serve subsystem. The system fails if the Switcher/Router fails OR if the Serve subsystem fails. The Server subsystem is composed by two servers, S1 and S2. S1 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 availability model
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
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
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
CTMC reliability model
λ_{rs} is failure rate of the Switcher/Router.
λ_{s1} is failure rate of the Server 1.
λ_{s2} is failure rate of the Server 2.
μ 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
CTMC availability model
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
The respective CTMC availability model is
Up states:
0
2
System availability:
A=0.998799526
System unavailability:
UA=0.00120047377

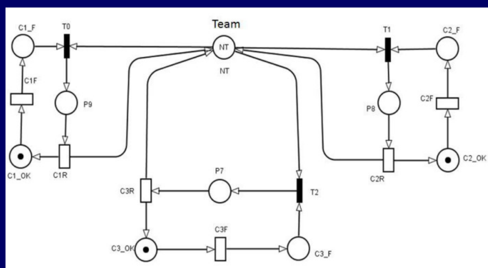

SPN
Example
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
SPN reliability model
System Unreliability:
UR(4000h)= 0.18161528133
System Reliability:
R(4000h) = 0.81838471867


SPN
Example
Clients
Switcher/Route
SR
S1
S2
Perfect switching cold standby server architecture
SPN availability model
Availability Results:
Steady State Unavailability = 0.0012011
Steady State Availability = 0.9987989

SPN

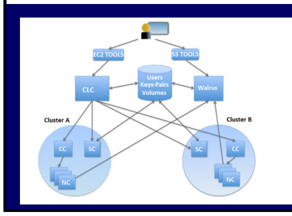
Shared repair

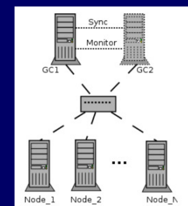
Corrective Maintenance



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 th underlying virtualized resources (servers, network, and storage).

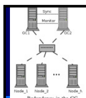




Redundancy in the GC

HIERARCHICAL MODELING

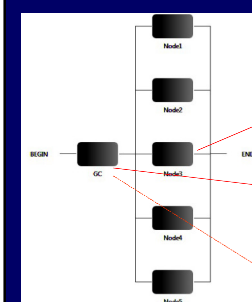
Hierarchical Modeling



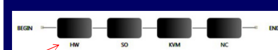
Input Parameters for the nodes

Component	MTTF	MTTR
KVM	2990 h	1 h
NC	788.4 h	1 h

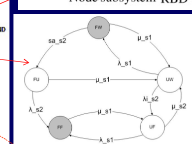
Parameter	Description	Value
$\lambda\_s1 = \lambda\_s2 = 1/\lambda$	Mean time for host failure	$1/180.721$
$\lambda_i\_s2 = 1/\lambda_i$	Mean time for inactive host failure	$1/216.865$
$\mu\_s1 = \mu\_s2 = 1/\mu$	Mean time for host repair	$1/0.9667$
$sa\_s2 = 1/sa$	Mean time to system activate	$1/0.005$



cloud system **HLM**

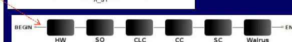


Node subsystem RBD model



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

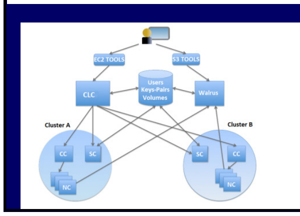
Redundant general controller subsystem

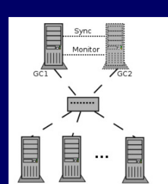


RBD model of the non-redundant General Controller subsystem

Hierarchical Modeling

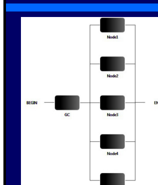
EUCALYPTUS is composed by five high-level components: Cloud Controller, Cluster Controller, Node Controller, Storage Controller, and Walrus. The Cloud Controller (CLC) is responsible for exposing and managing th underlying virtualized resources (servers, network, and storage).



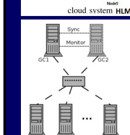


Redundancy in the GC

Hierarchical Modeling



cloud system **HLM**



Redundancy in the GC

$$A_{GC} = \frac{\mu(\lambda_i(\mu + sa) + \mu^2 + sa(\lambda + \mu))}{\lambda_i(\lambda + \mu)(\mu + sa) + \mu^2(\lambda + \mu) + sa(\lambda^2 + \lambda\mu + \mu^2)}$$

$$A_{cloud} = A_{GC} * (1 - \prod_{i=1}^{n}(1 - A_{Node\_i}))$$

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