# Dependable and Distributed Systems

Professor Matthew Leeke
School of Computer Science
University of Birmingham

Topic 4 - Dependability Analysis and Evaluation

# The Role of Dependability Analysis

Tradeoffs need to be identified and resolved

Wrong to leave analysis until the end of the development process

- Complete redesign is costly
- The earlier potential problems are identified, the better it is for all concerned

**Dependability analysis tries to identify acceptable levels of safety and reliability**

# Hazards, Risk and Safety Cases

# Safety Requirements

Safety-critical systems will have to satisfy some safety requirements

Safety requirements deal with elimination / reduction of accidents (bad consequences)

Loss of life, property, etc.

Safety requirements are met by identifying and addressing **hazards**

# Hazards

A hazard is a state or set of conditions of a system that, together with environmental conditions, will lead to an accident

Usually defined in terms of states to be avoided

No two aircraft should share the same airspace

Care should not be within 2 seconds of each other when travelling at high speed

# Dealing With Hazards

Need to be able to rank hazards to address the most severe

Hence, we need to know:

The **severity** of the hazard

The potential **frequency** of occurrence

# Hazard Severity - An Example

Civil aviation standards in Europe and the USA define five hazard severity levels

**Catastrophic** - Failure conditions preventing safe flight

**Hazardous** - Failure conditions significantly reducing the ability of the aircraft / crew / passengers to ensure safety

**Major** - Failure conditions reducing the ability of the aircraft / crew / passengers to ensure safety

**Minor** - Failure conditions resulting in minor loss of safety or little increase in workload

**No effect** - Failure conditions with no impact

# Hazard Frequency - An Example

Three classes for aircraft, similar in Europe and USA

**Probable** - $10^{-4}$ to $10^{0}$ per operating hour

**Improbable** - $10^{-8}$ to $10^{-5}$ per operating hour

**Extremely improbable** - $10^{-9}$ to $10^{-8}$ per operating hour

Different domains will have different interpretations of severity and frequency

# Risks

**Hazard level** is the combination of hazard severity and likelihood of occurrence

**Risk** is hazard level combined with:

Likelihood of hazard leading to accidence (higher likelihood, higher risk)

Hazard exposure or duration / latency (more exposes, the more likely the accident will happen)

# Safety Cases

Highly critical systems need certification from authority before the start of operation

Safety case needed as part of certification

> It should communicate a **clear**, comprehensive **argument** that a **system** is **acceptably** safe to operate in a particular **context**

Not meant to be formal (as in formal methods, logic, etc.)

> Typically a document that explains how risks have been mitigated

# Safety Cases

**Clear** - Be convincing to an informed reader

**Argument** - Demonstrate how an informed reader could reasonably conclude that a system is safe, given the available evidence

**System** - The case must refer to the specific system, as opposed to its model or class of system

**Acceptably** - Absolute safety is impossible, hence acceptable safety is taken with respect to risk levels

**Context** - Context-free safety is impossible and acceptable safety is situational

# Safety Cases

UK Defence Standard 00-56 requires that safety case development to be treated as an evolutionary activity that is integrated with design and safety analysis

Specifies that at least three versions of a safety case should be developed:

**Preliminary safety case** - After definition and review of system requirements specification

**Interim safety case** - After initial system design and preliminary validation activities

**Operational safety case** - Prior to in-service use, including complete evidence of requirements satisfaction

# Dependability Cases

Different from safety cases

Since dependability is a multidimensional attribute, tradeoffs must be resolved across dimensions

| | Objective | Typical Argument | Typical Evidence |
|---|---|---|---|
| **Safety** | System is adequately safe | Hazard mitigation argument | Hazard analysis and causal analysis |
| **Reliability** | System meets reliability requirements | Sufficient redundancy among corporate components | Testing, simulation and Markov analysis |
| **Maintainability** | System meets maintainability requirements | Modular design, plug-and-play, etc. | Simulation, expert opinion and software metrics |
| **Security** | Mission critical information protected | Asset identification and protection arguments | Access control policies |

# Hazard Analysis

Identification of way in which a system can cause harm

Consists of a set of techniques with distinctive attributes, hence they are often combined

- FMEA
- FMECA
- HAZOP
- Fault trees
- Event trees

# Failure Modes and Effects Analysis (FMEA)

Considers the failure of a component (including hardware) with the effect of this failure being assessed at system-level to detect hazardous situations

Advantages:

- Can detect hazard situations arising as a result of single failures

Disadvantages:

- Does not consider multiple failures (not all single failures results in hazards)
- Expensive to performs exhaustively (usually only done for critical systems)

# Failure Modes, Effects and Criticality Analysis (FMECA)

Similar to FMEA, except the important of each failure is ranked by accounting for the consequences and likely frequency of occurrence

Advantages:

Similar to FMEA, though FMECA does permit meaningful cost-benefit analysis

Disadvantages:

Similar to FMEA

# Hazard and Operability Studies (HAZOP)

Originally developed for the chemical industry

Focuses on answers to “What if…?”  Questions

Can effectively demonstrate the effects of parametric changes and out-of-range values

Advantages:

- Effective for situations where “What if…?” questions are constrained

Disadvantages:

- Time consuming, especially for the experts used for question generation

# Event Trees

Starting from events that can affect system, tracking **forward** to determine their effects

Events can be normal (within specification) or bad (faulty)

Root node - Associated with event under scrutiny
Other nodes - Associated with other system events
Edges - Link two directly related system events

Advantages:

Allows outcomes of events to be determined

Disadvantages:

Exponential complexity

# Fault Trees

Commonly used for safety-critical systems

Differs from event trees, in that fault trees track **backwards**

Starting from hazardous situations, propagate backwards to determine possible causes, using boolean connectives to group effects of events

Advantages:

- Resultant trees are simplified, compared to event trees

Disadvantages:

- Identification of high-level hazards is challenging

# The Exponential Failure Law

# Reliability

Reliability of a system is the probability of the system to deliver correct services over a specified time period under given conditions

This leaves a few aspects for us to clarify

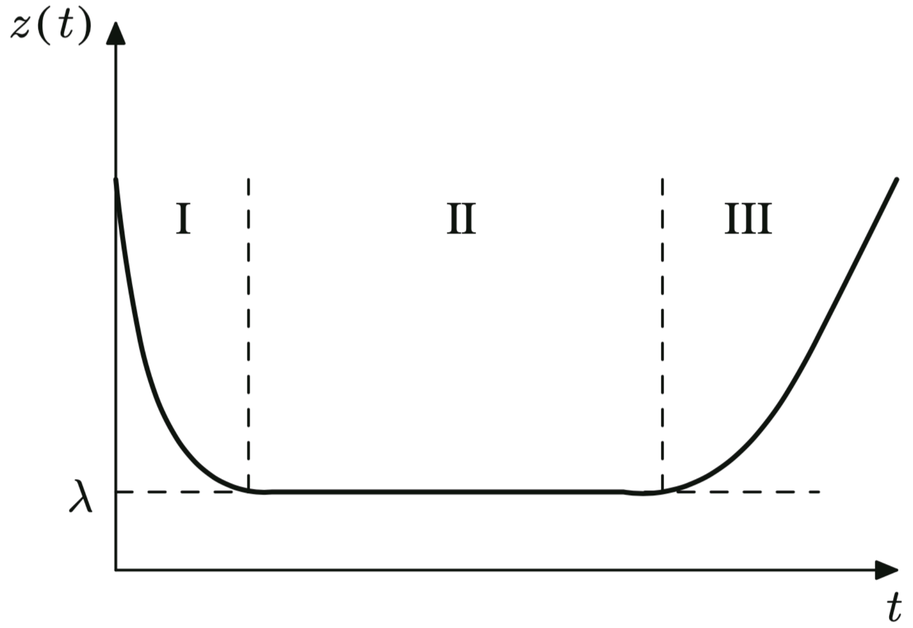
- How can system reliability be computed?
- How is a system modelled?
- What are the parameters?

# Failure Rate

Failure rate is the expected number of failures of a certain device over a given time frame

Typical evolution of failure rate over the lifetime of a system is illustrated by the bathtub curve

This curve has three phases; (i) infant mortality, (ii) useful life and (iii) wear out




During the useful life phase of the system, the failure rate function $z(t)$ is assumed to be constant $\lambda$

Them, the reliability of the system decreases exponentially with time

# Failure Rate

To see where we get this from, we can calculate the failure rate directly:

Consider a sample of $N$ devices and run them at time $t_0$

At time $t$, compute the number of devices still running

Denote by $U(t)$ the number of up devices at time $t$

Denote by $F(t)$ the number of failed devices at time $t$

# Failure Rate

Reliability of component over time period $(t_1, t)$

Reliability $= R(t) = \dfrac{U(t)}{U(t) + F(t)} = \dfrac{U(t)}{N}$

Unreliability = $U(t) = \dfrac{F(t)}{U(t) + F(t)}$

However, time is a continuous variable rather than a discrete one

$$R(t) = 1 - U(t) \Rightarrow \frac{dR(t)}{dt} = -\frac{1}{N}\frac{dF(t)}{dt} \Rightarrow \frac{dF(t)}{dt} = -(N)\frac{dR(t)}{dt}$$

# Failure Rate

Derivative of $F(t)$ is the instantaneous fail rate of the device

$$\Rightarrow \frac{dF(t)}{dt} = -N\frac{dR(t)}{dt}$$

$$z(t) = \frac{1}{U(t)}\frac{dF(t)}{dt}$$

This is called **hazard function**, **hazard rate** or **failure rate function**

The derivative of **unreliability** is called the **failure density function**

# Where Do Failure Rates Appear?

Military standardisation handbook

US Department of Defence DoD MIL - HDBK - 217F

Reliability prediction of electronic equipments

Explain how this is done, what temperature etc

Every electronic component including basic ones such as gates

# Failure Rates

We are only concerned about the failure rate during the useful phase of the device

Given it is constant, it will be assumed to be equal to $\lambda$

$$z(t) = -\frac{\frac{dR(t)}{dt}}{R(t)} \Rightarrow -z(t)R(t) = \frac{dR(t)}{dt} \Rightarrow R(t) = e^{-\int z(t)dt} \Rightarrow R(t) = e^{-\lambda t}$$

**The result is the exponential failure law:** $R(t) = e^{-\lambda t}$

# The Exponential Failure Law

Widely used when failure rate is constant

If failure rate is changing, then it cannot be used

Used in analysis of electronic components

Examples where it cannot be used

Software where bugs are removed when discovered, failure rate decreased with time and exponential failure law

# The Exponential Failure Law

It is important to analyse its purpose and usefulness

Allows to plot a graph of reliability vs time as well as analysing the reliability of the device

As reliability is defined over a specific period of time, it could be detected if reliability decreases below a threshold within that time frame.

# Reliability Modelling - MTTF, MTTR and MTBF

# Mean Time To Failure (MTTF)

Another reliability measure

Expected time a system will operate before first failure

If there are $N$ identical systems which run correctly at $t = 0$ but they all fail at time $t_i$, then the MTTF formula has the form:

$$MTTF = \sum_{i=1}^{N} \frac{t_i}{N}$$

# Mean Time To Failure (MTTF)

In the continuous case, the expected value is

$E[X] = \int_{-\infty}^{\infty} xf(x)dx$ , where $X$ is a random variable and $f(x)$ is the probability distribution function

Thus, MTTF can be calculated using:

$MTTF = \int_{-\infty}^{\infty} tf(t)dt$ , where $f(t)$ is failure density function, i.e., the derivative of unreliability

# Deriving Mean Time To Failure (MTTF)

$$MTTF = \int_0^\infty t \frac{dU(t)}{dt} dt$$

$$MTTF = -\int_0^\infty t \frac{dR(t)}{dt} dt$$

Integrating by parts and assuming $R(t) = 0$ at $t = \infty$

$$MTTF = -\left[tR(t) - \int R(t)dt\right]_0^\infty$$

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

# Mean Time To Repair (MTTR)

Repair is important, otherwise the system will die at $t = \infty$

MTTR is the average time to repair a failed computer system

Includes time for detecting and locating the fault, repairing the fault and reconfiguring the system

It might be difficult to predict analytically

Thus, it might required experimental measurement for computation

# Mean Time To Repair (MTTR)

Repair is important, otherwise the system will die at $t = \infty$

MTTR is the average time to repair a failed computer system

Includes time for detecting and locating the fault, repairing the fault and reconfiguring the system

It might be difficult to predict analytically. Therefore, it might required experimental measurement for computation

Just as in failure, there is a repair rate which is denoted by $\mu$

As in the failure, the MTTR is equal to: $\frac{1}{\mu}$

# Mean Time Between Failures (MTBF)

This is the average time elapsed between failures

An assumption is that, after repair, the system works as it was originally doing ( without substitution)

This leads to a definition of MTBF based on MTTR and MTTF

$$MTBF = MTTR + MTTF$$

# Reliability Modelling - Combinatorial Modelling

# Reliability Modelling

In the development of safety-critical systems, system requirements mandate a certain level of reliability to be reached

For certification / safety, it needs to be demonstrated

To this end, two of the most popular analytical techniques used are

Combinatorial models

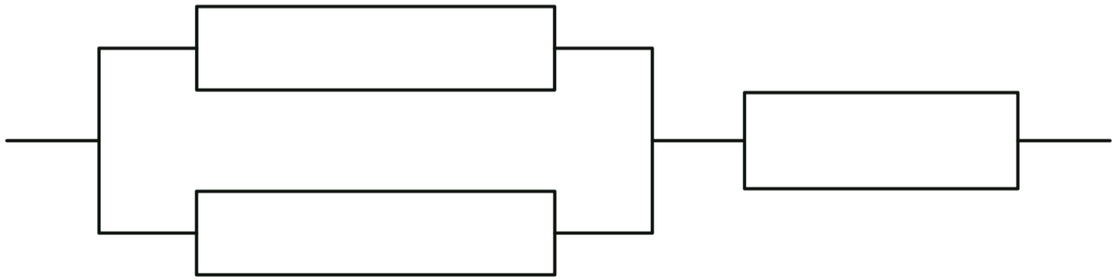
Markov models

# Combinatorial Models

Reliability of the system is calculated from the reliability of its components

Construction of a reliability block diagram (RBD) to provide an abstract view of the system

Composition of RBD is necessarily related to
the physical connections between components



Can be modelled in two ways

**Series systems** - failure of any part will cause an outage

**Parallel systems** - failure of a combination of parts will cause an outage

# Series Systems

All components are required to work for the overall system to operate correctly

Probabilistic techniques are used to enumerate combinations that will ensure a working system

Assume a system S-of-N components

$$P(S_t) = Pr\{C_1(t) \cap C_2(t) \cap \ldots \cap C_N(t)\}$$

$$\Rightarrow P(S_t) = P(C_1(t)) \times P(C_2(t)) \times \ldots \times P(C_N(t))$$

$$\Rightarrow R_1(t) \times R_2(t) \times \ldots \times R_N(t) = \prod_{i=1}^{N} R_i(t)$$

# Parallel Systems

Failure of one component does not result in a system failure

Assume a system S-of-N components

$$P(S_t) = 1 - P(all\ components\ failing)$$

$$= 1 - P\{f_1(t) \cap f_2(t) \cap \ldots \cap f_N(t)\}$$

$$= 1 - P(f_1(t)) \times Pr(f_2(t)) \times \ldots \times Pr(f_N(t))$$

$$= 1 - \prod_{i=1}^{N} (1 - R_i(t))$$

# M-of-N systems

Generalisation of parallel systems

Instead of at least one component working (parallel) at M-out-of-N components should work

Reliability of such a system is expressed as:

$$P(S_t) = \sum_{i=0}^{N-M} \frac{N!}{(N-i)!i!}(R(t))^{N-i}(1-R(t))^i$$

# The Reality of Combinatorial Modelling

Real-world systems are usually a combination of:

Series

Parallel

M-of-N systems

In practice, any system can ultimately be reduced to a combination of series / parallel systems and then be further reduced to a single equivalent element

# Reliability Modelling - Cut Set and Tie Sets

# Cut Sets

Given a reliability block diagram

Draw lines through the diagram to represent combinations of elements in which simultaneous failures would lead to a system failure

Interested in the **minimal cut sets** - subsets of minimal cut set will not lead to system failure

**Provide lower bound on reliability**

In a series systems, each cut set is of size 1

In a parallel system, each cut set is of size system

# Tie Sets

Given a reliability block diagram

Draw lines through the diagram to represent combinations of elements in which simultaneous operations would lead to a correctly working system

Interested in the **minimal tie sets** - subsets of the minimal tie set will not lead to a correctly working system

**Provide upper bounds on reliability**

In a series systems, tie set is the size of the system

In a parallel system, each tie set is of size 1

# An Example - Triple Mode Redundancy (TMR)

Three processors run in parallel - outputs are compared and any discrepancy indicates a fault

Voting is done according to the majority, thus at least two out of three must work correctly

# An Example - Triple Mode Redundancy (TMR)

Assume each processor has a failure rate $\lambda$

Reliability of each processor is $e^{-\lambda t}$

The expression for M-of-N systems is being used

$$P(S_t) = \sum_{i=0}^{N-M} \frac{N!}{(N-i)!i!}(R(t))^{N-i}(1-R(t))^i$$

Reliability of TMR is therefore:

$$\frac{3!}{3!0!}((e^{-\lambda t})^3)(1-e^{-\lambda t})^0 + \frac{3!}{2!1!}((e^{-\lambda t})^2)(1-e^{-\lambda t})^1$$

# MTTF of TMR

MTTF of TMR:

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

$$\Rightarrow \int_0^\infty 3e^{-2\lambda t} - 2e^{-3\lambda t}dt$$

$$\Rightarrow [\frac{-3}{2\lambda}e^{-2\lambda t} + \frac{2}{3\lambda}e^{-3\lambda t}]_0^\infty$$

$$\Rightarrow (\frac{3}{2\lambda} - \frac{2}{3\lambda}) \Rightarrow \frac{5}{6\lambda}$$

# Reliability Modelling - Markov Models

# Markov Models

Derive their name from the Russian mathematician Andrey A. Markov (1856–1922), who was the first to describe stochastic processes

Markov processes are a special class of stochastic processes, where it is assumed that:

The behaviour of the system in each state is memoryless

The time spent in each state (before a transition occurs) follows an exponential distribution

Assumptions satisfied for dependability analysis if all events (e.g., failures, repairs, etc.) have a constant rate of occurrence

# Markov Models

Markov processes are classified by their to state space and time space properties

In most forms of dependability analysis, it is usual for the state space to be discrete and the time space to be continuous

| State space | Time space | Common model name |
|---|---|---|
| Discrete | Discrete | Discrete time Markov chains |
| Discrete | Continuous | Continuous time Markov chains |
| Continuous | Discrete | Continuous state, discrete time Markov processes |
| Continuous | Continuous | Continuous state, continuous time Markov processes |

This focuses our analyst on **Continuous Time Markov Chains**

# Continuous Time Markov Chains

Two components:

**State** - Comprise all that is needed to characterise it

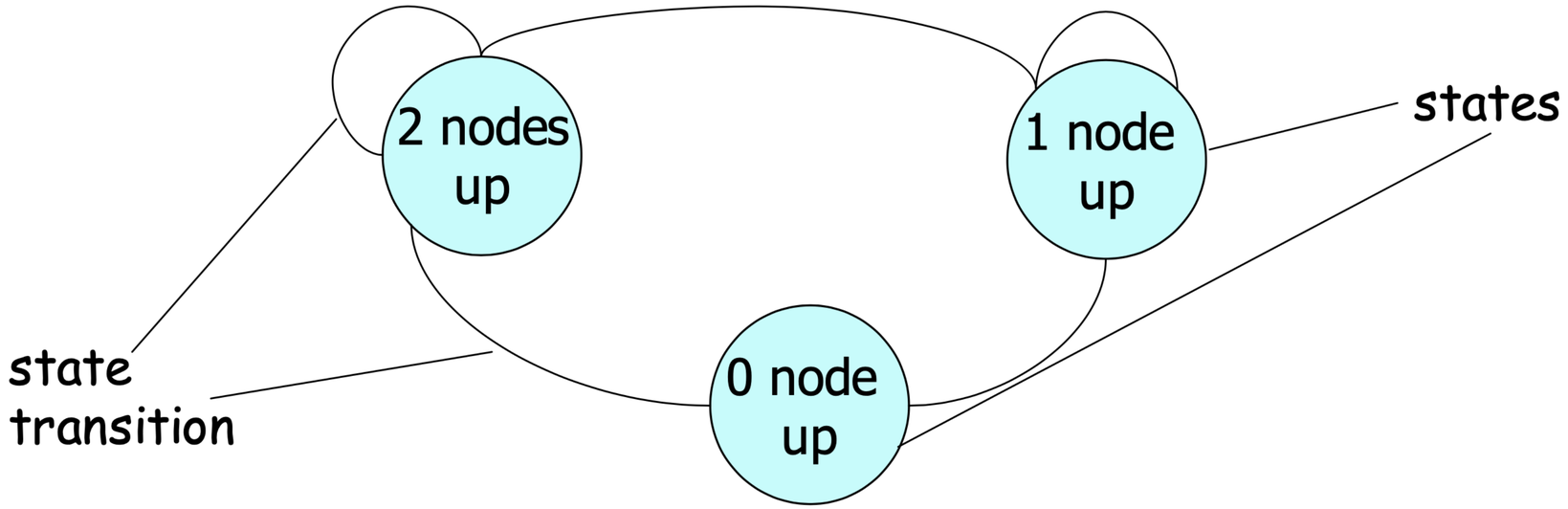
**State transition** - Govern how changes in state will occur

In reliability modelling we ofter consider state to be the number of failed nodes / processors and state transition to be dictated by the failure or repair of nodes / processors

# Continuous Time Markov Chains - An Example

A system is composed of two components running in parallel

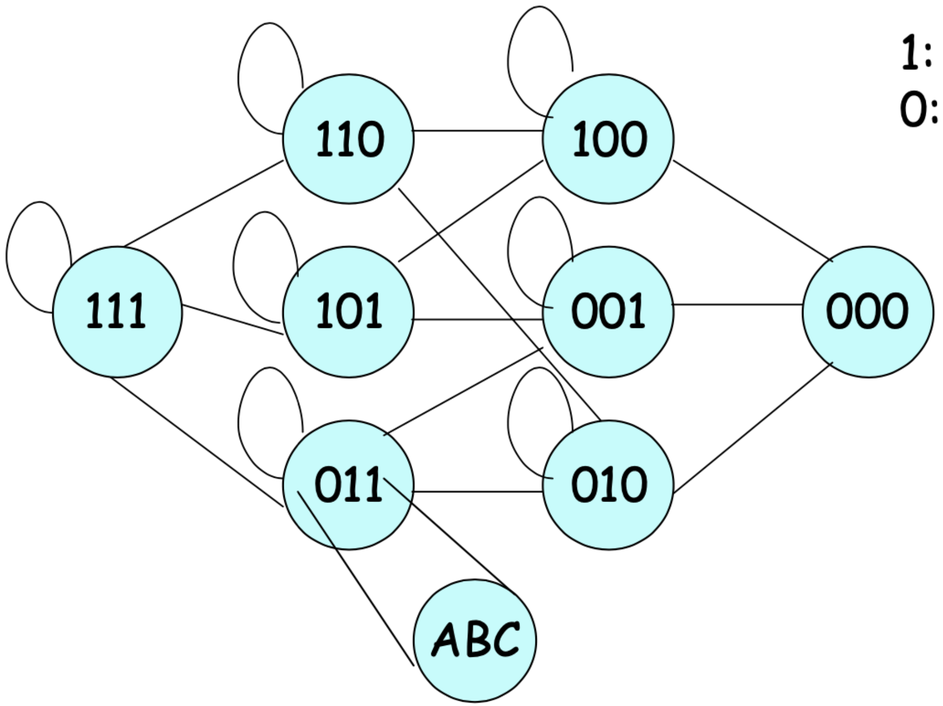
At any instant of time, any number of components may fail by crashing or be repaired



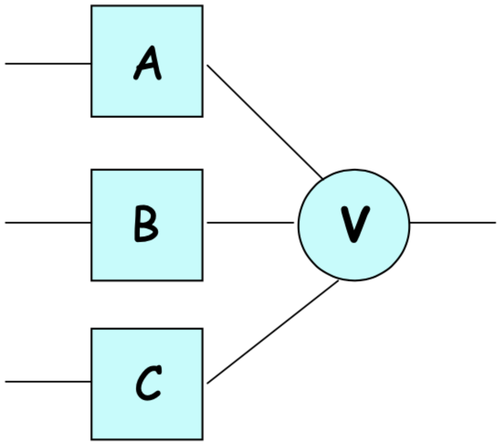
# Continuous Time Markov Chains - A More

A triple mode redundancy (TMR) system

Three replicas are run in parallel on the same input, with the outputs being compared




1: up
0: crash




At least two nodes must be up for successful operation

# Continuous Time Markov Chains - Probabilities

State transitions are labeled with probabilities (accounting for factors influencing transition)

Probability of a crash

Coverage

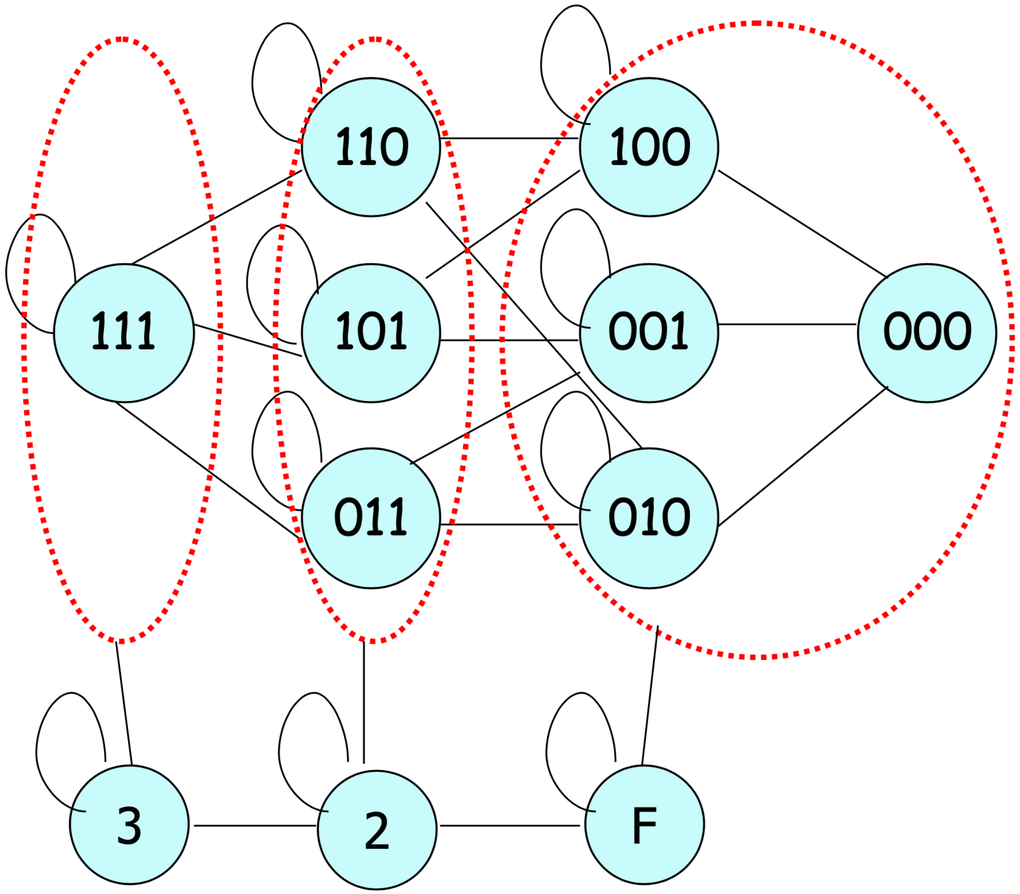
Repair probability

# Condensing Continuous Time Markov Chains

You may be wondering why we have not grouped 110, 101 and 011

The previous example was the complete Markov model

Condensed Markov models are less computationally expensive

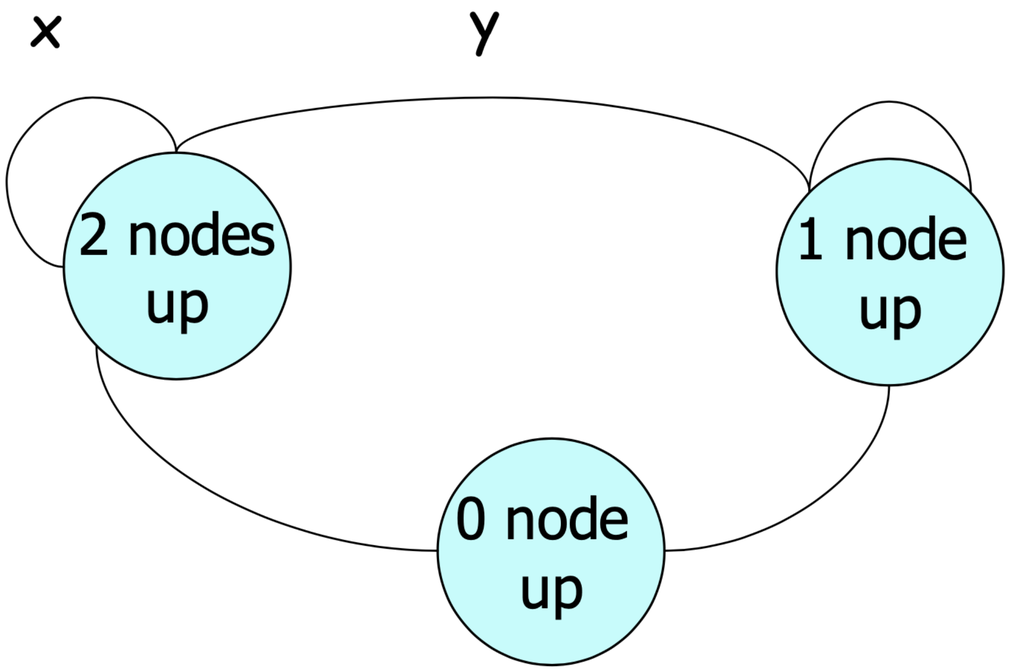


# Labelling The Model

Label state transitions with probabilities

Assume systems obey exponential law with failure ra

System must in one state or another but how do we l



# Calculating Model Probabilities

Assuming a failure rate of $\lambda$ and the exponential failure law, the probability of a module failure at time $t + \Delta t$ given that it was working at time $t$ is:

$$1 - \frac{R(t + \Delta t)}{R(t)}$$

$$= 1 - \frac{e^{-\lambda(t+\Delta t)}}{e^{-\lambda t}}$$

$$= 1 - e^{-\lambda \Delta t}$$

# Calculating Model Probabilities

$$1 - e^{-\lambda \Delta t}$$

We can expand this expression to obtain:

$$1 - [1 + (-\lambda t) + \ldots]$$

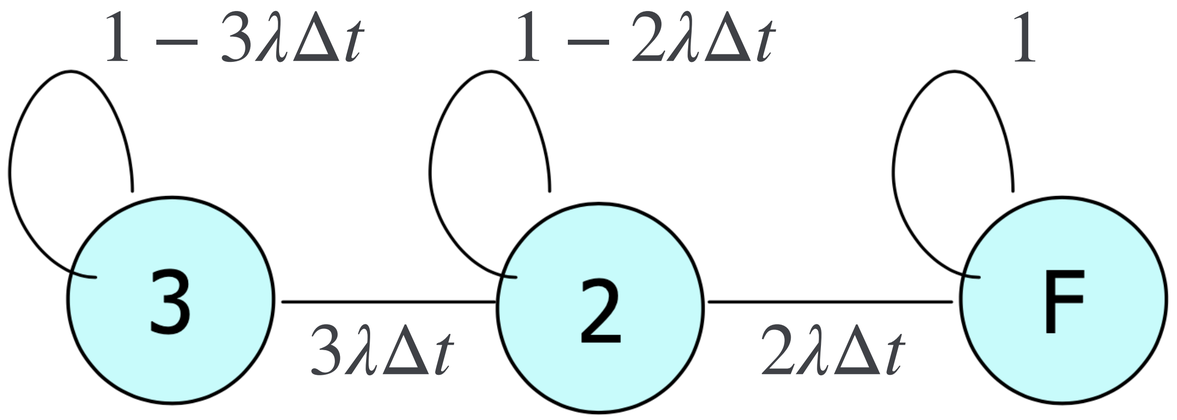
$$= \lambda \Delta t$$

Hence, a component with a failure rate of $\lambda$ that obeys exponential failure law will fail with probability $\lambda \Delta t$, given that it was working at time $t$

# Calculating Model Probabilities

When states are condensed, transition probability in the condensed model is the sum of the individual probabilities in the complete model

In State 3, three nodes can crash, each with probability $\lambda\Delta t$, giving a probability of $3\lambda\Delta t$



# Solving The Model

We can use simple mathematics to solve the Markov model once we have it

For example, if at time $t + \Delta t$ we want to know the probability of the system being in state $s$ then we need to know the probability of the system being in state $s'$ at time $t$, where $s$ can be reached by a state transition from $s'$, and the probability of the transition happening

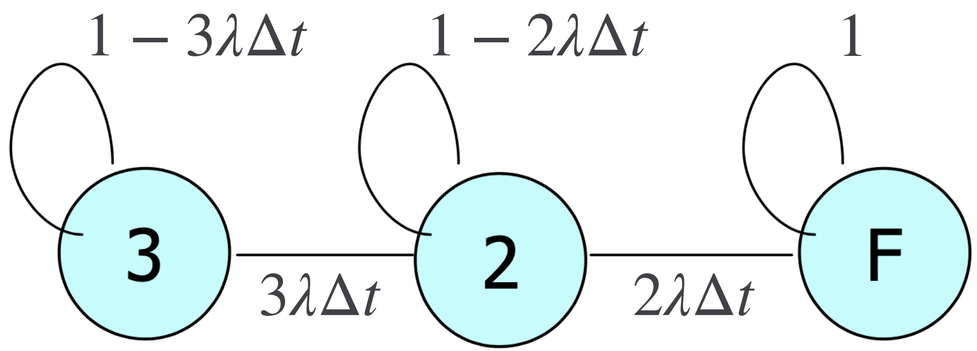
$$P_s(t + t\Delta) = \Sigma_{\forall x,(x,s)} P_x(t) \,.\, P_{(x,s)}$$

# Solving The Model

For such a TMR system, that is:

$$P_2(t + t\Delta) = 3\lambda\Delta t P_3(t) + (1 - 2\lambda\Delta t)P_2(t)$$

We do that for every state to solve



# Solving The Model

Once we have all the questions for the system, we can build a matrix to represent the information

For our TMR example, this is:

$$\begin{pmatrix} p_3(t+\Delta t) \\ p_2(t+\Delta t) \\ p_F(t+\Delta t) \end{pmatrix} = \begin{pmatrix} 1-3\lambda\Delta t & 0 & 0 \\ 3\lambda\Delta t & 1-2\lambda\Delta t & 0 \\ 0 & 2\lambda\Delta t & 1 \end{pmatrix} \begin{pmatrix} p_3(t) \\ p_2(t) \\ p_F(t) \end{pmatrix}$$

In compact form that would be: $P(t + t\Delta) = AP(t)$

Each column sums to 1

# Solving The Model

We can do some algebraic manipulation on our current equations, for example:

$$P_2(t + t\Delta) = 3\lambda\Delta t P_3(t) + (1 - 2\lambda\Delta t)P_2(t)$$

Simplifies to give:

$$\frac{P_2(t + t\Delta) - P_2(t)}{\Delta t} = 3\lambda P_3(t) - 2\lambda P_2(t)$$

In the limit, this simplifies to:

$$\frac{dP_2(t)}{dt} = 3\lambda P_3(t) - 2\lambda P_2(t)$$

Same process is performed for all other equations to give a set of differential equations to be solved in the limit

# Solving The Model

Thus we have a set of differential equations for our TMR system:

$$\frac{dP_3(t)}{dt} = -3\lambda P_3(t)$$

$$\frac{dP_2(t)}{dt} = 3\lambda P_3(t) - 2\lambda P_2(t)$$

$$\frac{dP_F(t)}{dt} = 2\lambda P_2(t)$$

# Solving The Model

Solving the first equation is trivial:

$$P_3(t) = -e^{-3\lambda t}$$

Solving the second is a little trickier:

$$P_2(t) = 3e^{-2\lambda t} - 3e^{-3\lambda t}$$

Solving the absorbing state is trivial:

$$P_F(t) = 1 - (P_1(t) + P_2(t)) = 1 - (3e^{-2\lambda t} - 2e^{-3\lambda t})$$

# An Alternative Way To Obtain The Differential

Rather than having self-looping state transitions, we consider only the transitions between distinct states

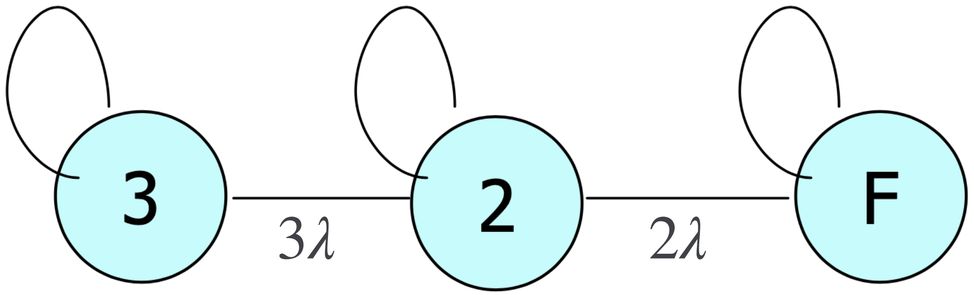
We can assume that: **Rate Of Buildup = Rate Of Flow In - Rate Of Flow Out**

For TMR that would give us:

$$\frac{d\pi_3(t)}{dt} = -3\lambda\pi_3(t)$$

$$\frac{d\pi_2(t)}{dt} = 3\lambda\pi_3(t) - 2\lambda\pi_2(t)$$

$$\frac{d\pi_F(t)}{dt} = 2\lambda\pi_2(t)$$



# So We Solved The TMR Markov Model?

We have looked at how to solve the Markov model for TMR

We are not really interested in stead–state, since the system be the absorbing state with probability 0

Rather, we are interested in performing a transient analysis, where the system is in states other than the absorbing state

# Modelling More Sophisticated Systems

We have considered the modelling of **passive redundancy**

**Dynamic redundancy** presents challenges in the development of a Markov model

Rather than having a set of replica processors (TMR = 3), we have some processors running and a set of shadow processors that are switched on when a primary processor fails

# Modelling More Sophisticated Systems

We will look at the principles for constructing Markov models and how to develop the associated system of equations

These are the challenging parts when it comes to the analysis of more complex systems

Solving the systems of equations is identical to what we've seen

# A More Sophisticated Example

Consider a system with three active components:

Active configuration is run as a TMR

Whenever a node in TMR fails, a spare components is swapped in

An active component has a failure rate of $\lambda$, whilst a spare component has a failure rate of $\mu$

Two possible types of spare

**Cold spare** - Different failure rate to active node

**Hot spare** - Same failure rate as active node

# A More Sophisticated Example

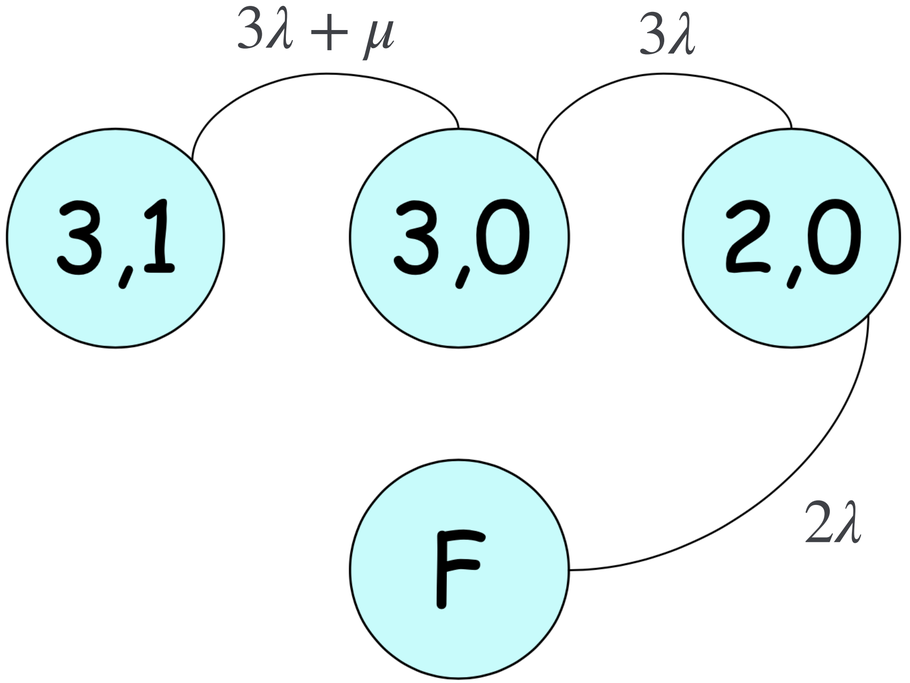
$$\frac{d\pi_{3,1}(t)}{dt} = -(3\lambda + \mu)\pi_{3,1}(t)$$

$$\frac{d\pi_{3,0}(t)}{dt} = (3\lambda + \mu)\pi_{3,1}(t) - 3\lambda\pi_{3,0}(t)$$

$$\frac{d\pi_{2,0}(t)}{dt} = 3\lambda\pi_{3,0}(t) - 2\lambda\pi_{2,0}(t)$$

$$\frac{d\pi_{F}(t)}{dt} = 2\lambda\pi_{2,0}(t)$$

Note: Still sums to zero



# New Problems With Dynamic Redundancy

Dynamic redundancy involved the reconfiguration of a live system

System needs to correctly handle the faulty component so that the shadow takes over

**Coverage** - Probability the fault will be correctly handled

What happens if enabling of the spare does not occur?

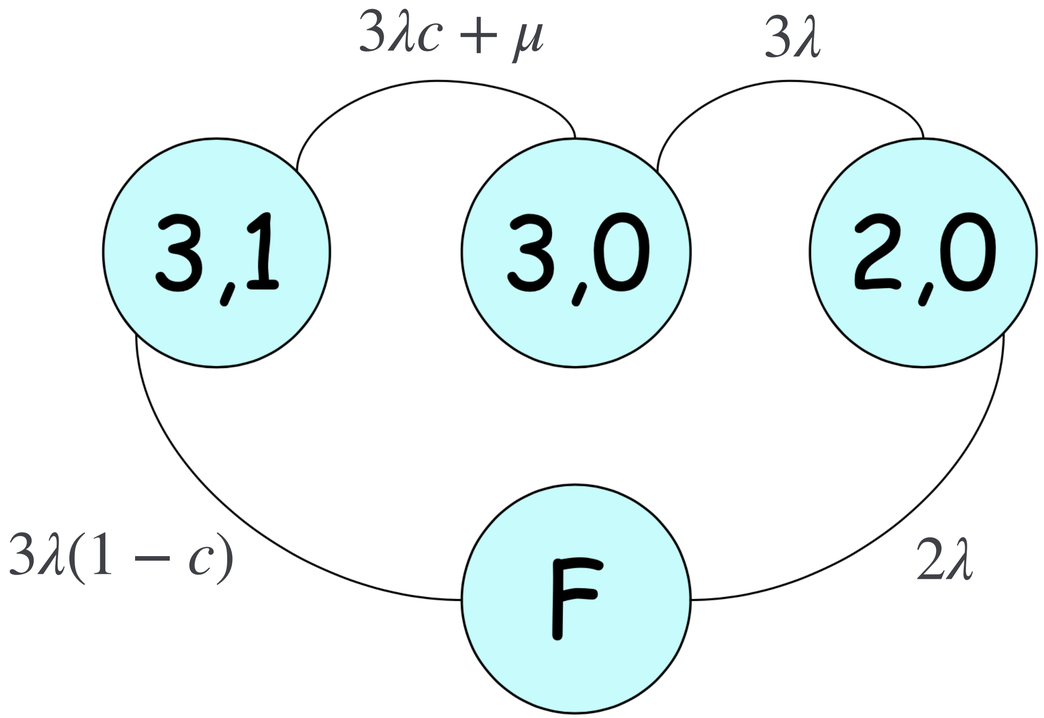
# TMR Model With Spare Coverage

$$\frac{d\pi_{3,1}(t)}{dt} = -(3\lambda + \mu)\pi_{3,1}(t)$$

$$\frac{d\pi_{3,0}(t)}{dt} = -3\lambda\pi_{3,0}(t) + (3\lambda c + \mu)\pi_{3,1}(t)$$

$$\frac{d\pi_{2,0}(t)}{dt} = -2\lambda\pi_{2,0}(t) + 3\lambda\pi_{3,0}(t)$$

$$\frac{d\pi_{F}(t)}{dt} = 3\lambda(1-c)\pi_{3,1}(t) + 2\lambda\pi_{2,0}(t)$$



# Examples For You To Try

Obtain a reliability expression for each of the following systems:

Single processor node with a failure rate of $\lambda$ (we have already seen this one)

A single processor system with a cold spare, where the failure rate of the main processor is $\lambda$ and the failure rate of the spare is $\mu$

A single processor with a home spare with coverage $c$

# Example Exam Question

# A Hard Question - Dependability Modelling and Analysis

A Triple Modular Redundancy (TMR) system is developed with three processors $P_1$, $P_2$ and $P_3$. The output of each processor is fed into a voter.

(a) Explain the concept of a minimal cut set.

(b) Enumerate the minimal cut sets of the TMR system.

(c) It is stated that the union of two minimal cut sets is a minimal cut set. Explain whether you agree with this statement.

(d) Processors $P_1$, $P_2$ and $P_3$ are known to have failure rates of $\lambda_1$, $\lambda_2$ and $\lambda_3$ respectively. Assuming the exponential failure law, derive an expression for the reliability of the TMR system. You should explain each step in your derivation.

(e) Assuming the exponential failure law, derive the continuous time Markov model for the reliability of the TMR system. State any assumptions. You should explain each step in your derivation.