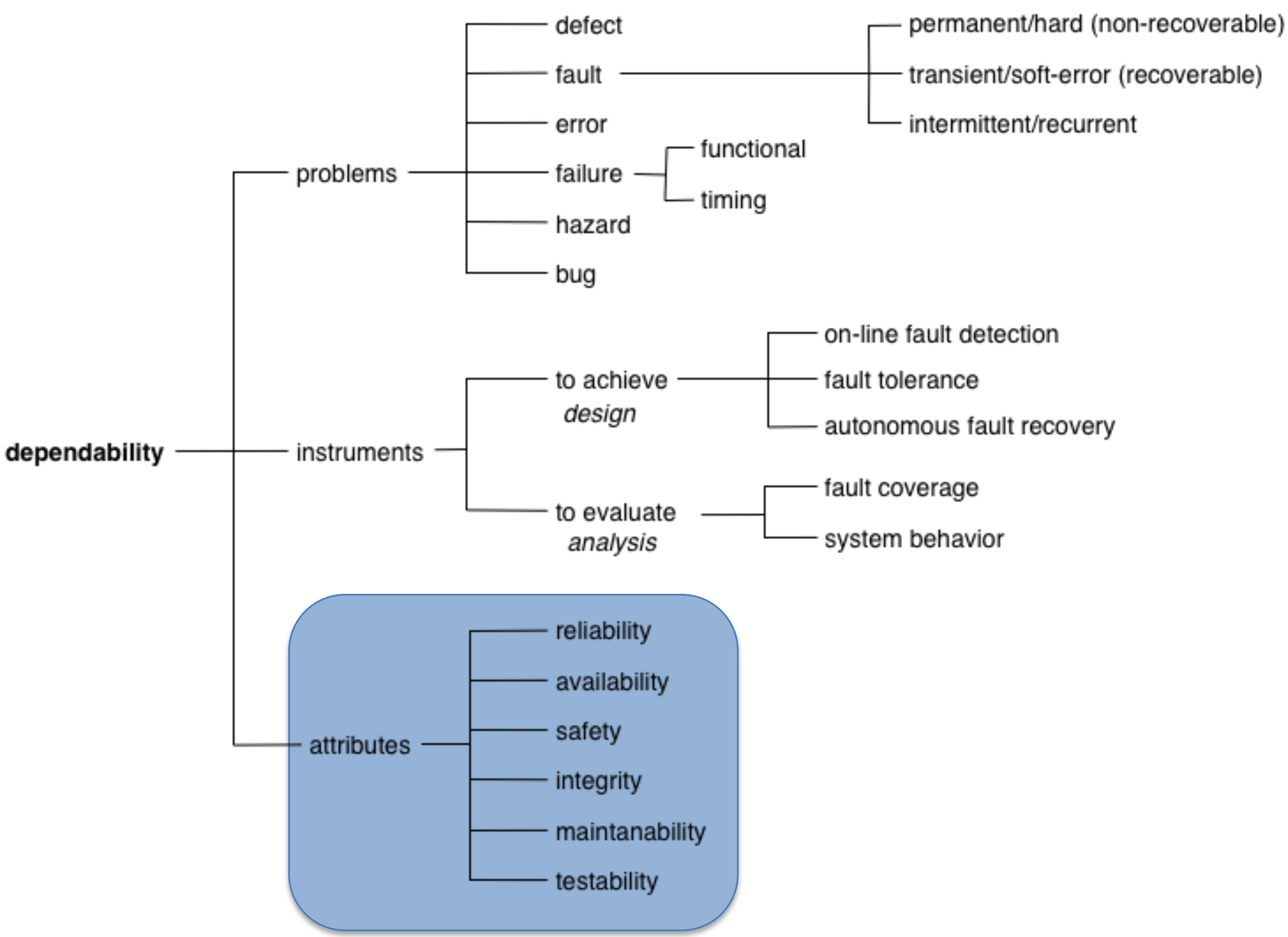
Computing Infrastructures
-
System Dependability

Roberto Sala
roberto.sala@polimi.it
Contribution: Luca Cassano

The scenario



Reliability

The ability of a system or component to perform its required functions under stated conditions for a specified period of time

[IEEE610]: IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990 (R2002)

definition

Reliability

R(t): probability that the system will operate correctly in a specified operating environment until time *t*

$$R(t) = P(\text{not failed during } [0, t])$$

assuming it was operating at time t = 0

t is important

If a system needs to work for slots of ten hours at a time, then ten hours is the reliability target

characteristics

1 – R(*t*): unreliability, also denoted Q(*t*)

R(*t*) is a non-increasing function varying from 1 to 0 over [0,+∞)

$$\lim_{x \to +\infty} R(t) = 0$$

adoption

Reliability

Often used to characterize systems in which even small periods of incorrect behavior are unacceptable

- Performance requirements
- Timing requirements
- Extreme safety requirements
- Impossibility or difficulty to repair

Availability

The degree to which a system or component is operational and accessible when required for use
[IEEE610]

Availability = Uptime / (Uptime + Downtime)

definition

A(*t*): probability that the system will be operational at time *t*

$$A(t) = P(\text{not failed at time } t)$$

Literally, readiness for service

Admits the possibility of brief outages

Fundamentally different from reliability

characteristics

$1 - A(t)$: unavailability

When the system is not repairable?

characteristics

Availability

$1 - A(t)$: unavailability

When the system is not repairable: $A(t) = R(t)$

In general (repairable systems): $A(t) \geq R(t)$

Some numbers

Availability

Availability as a function of the "number of 9's"

Number of 9's	Availability	Downtime (mins/year)	Practical meaning
1	90%	52596.00	~5 weeks per year
2	99%	5259.60	~4 days per year
3	99.9%	525.96	~9 hours per year
4	99.99%	52.60	~1 hour per year
5	99.999%	5.26	~5 minutes per year
6	99.9999%	0.53	~30 secs per year
7	99.99999%	0.05	~3 secs per year

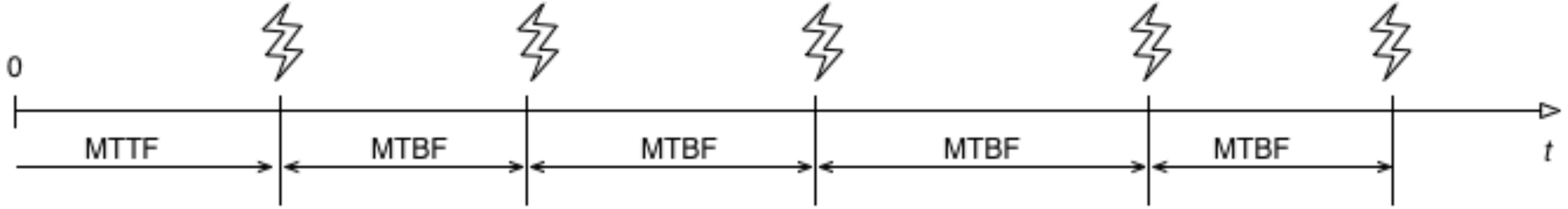
Some example

Number of 9's	Availability	Downtime/year	System
2	99%	~4 days	Generic web site
3	99.9%	~9 hours	Amazon.com
4	99.99%	~1 hour	Enterprise server
5	99.999%	~5 minutes	Telephone system
6	99.9999%	~30 seconds	Phone switches

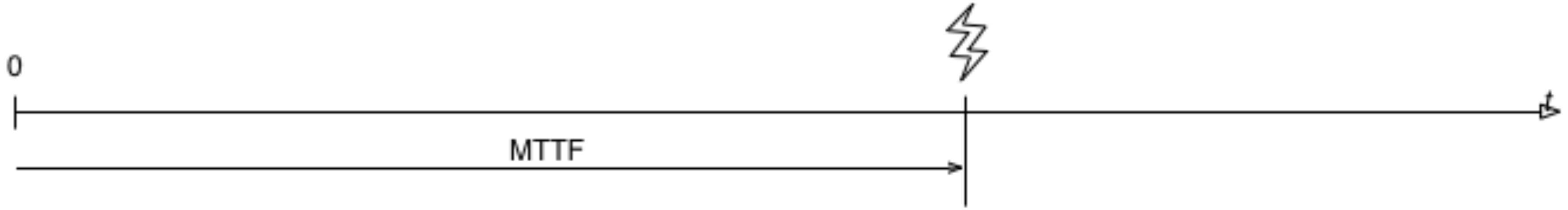
R(t) & A(t) related indices

MTTF (Mean Time To Failure): mean time before *any* failure will occur

MTBF (Mean Time Between Failures): mean time between two failures



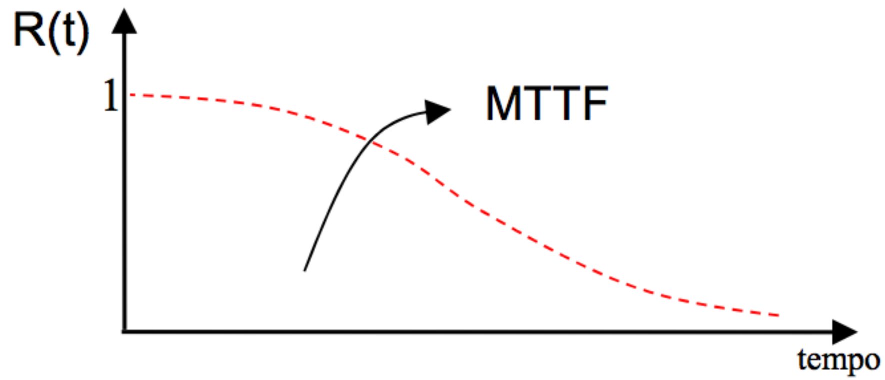
hypothesis: negligible repair time



R(t) & A(t) related indices

MTTF (Mean Time To Failure): mean time before *any* failure will occur

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



R(t) & A(t) related indices

MTTF: mean time to (first) failure, the up time before the first failure

MTBF: mean time between failures

$$\text{MTBF} = \frac{\text{total operating time}}{\text{number of failures}}$$

R(t) & A(t) related indices

Other indices

MTTF: mean time to (first) failure, the up time before the first failure

MTBF: mean time between failures

$$\text{MTBF} = \frac{\text{total operating time}}{\text{number of failures}}$$

FIT: failures in time

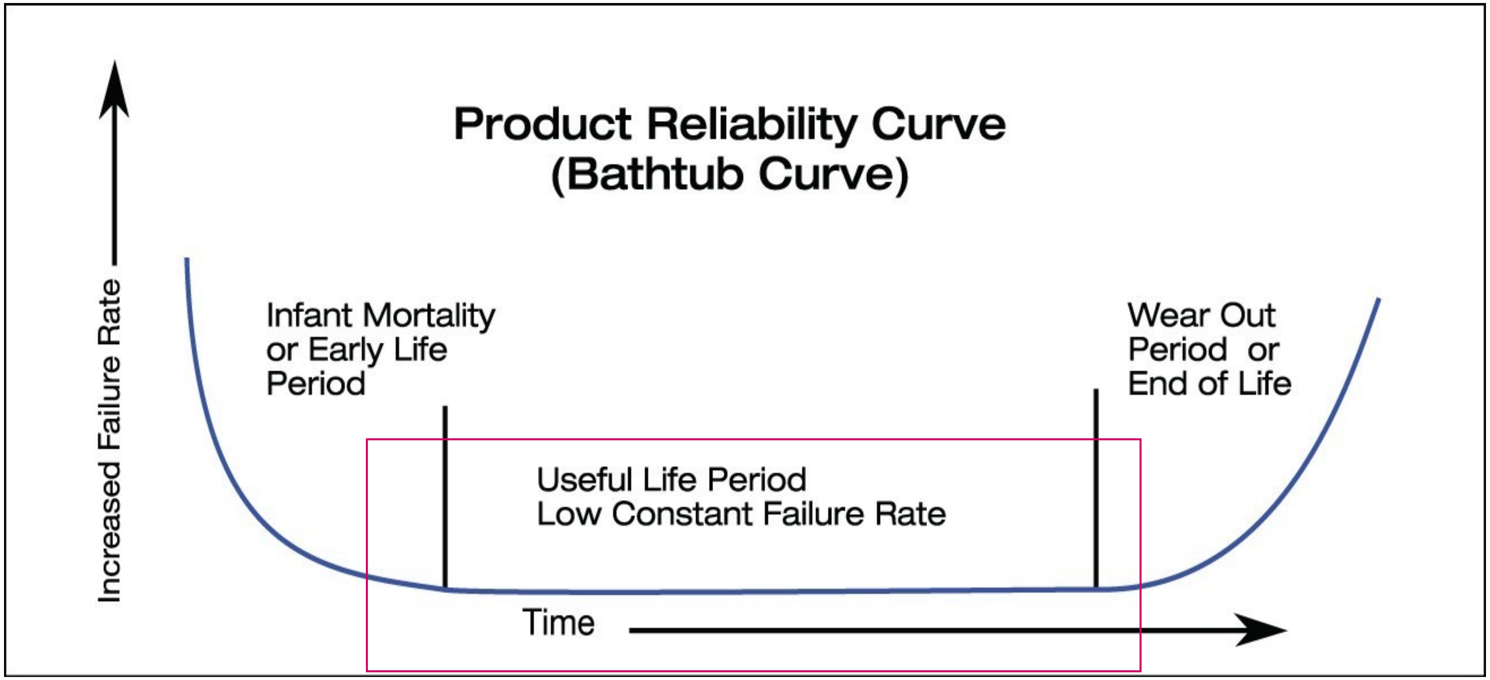
$$\text{Failure Rate } \lambda = \frac{\text{number of failures}}{\text{total operating time}}$$

- another way of reporting MTBF
- the number of expected failures per one billion hours (10^9) of operation for a device
- MTBF (in h) $=10^9/\text{FIT}$

$$\text{MTBF} = \frac{1}{\lambda}$$

R(t) & A(t) related indices

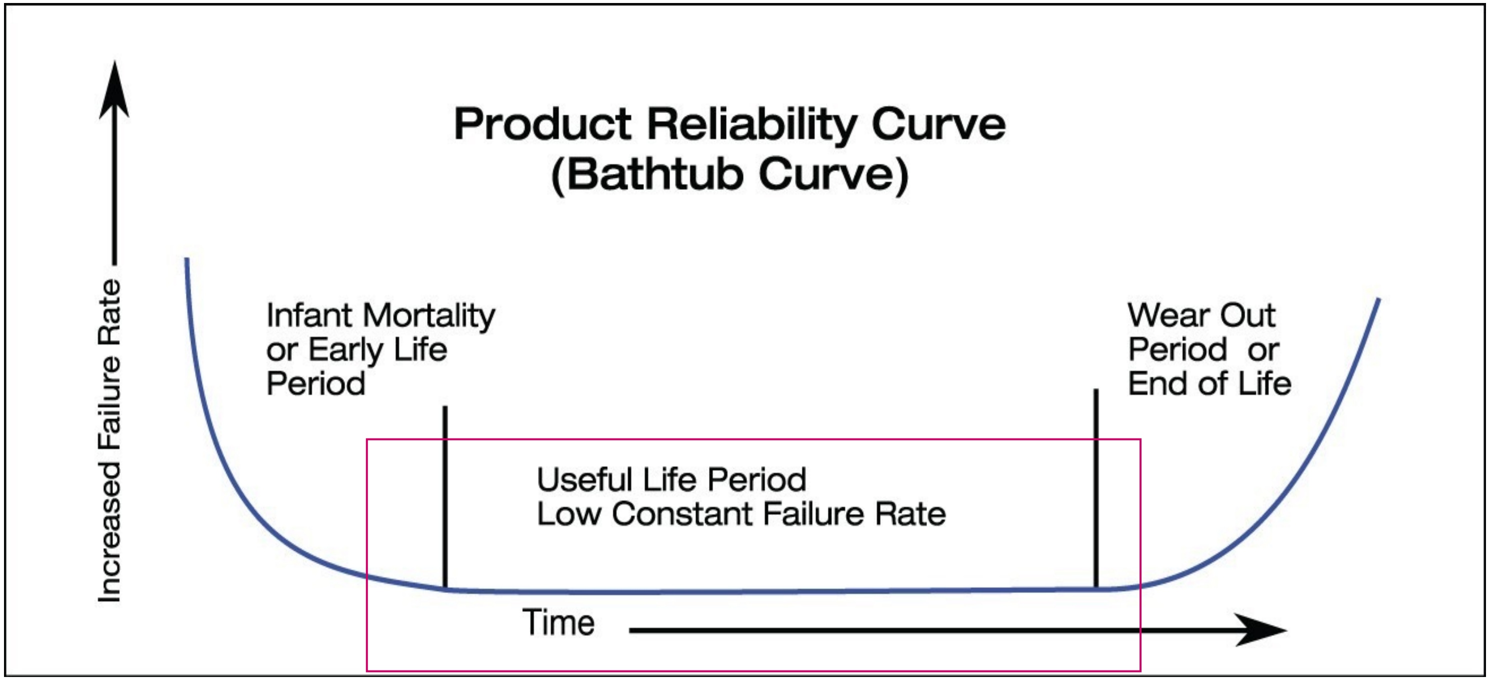
Other indices



- *Infant Mortality*: failures showing up in new systems.
  Usually this category is present during the testing phases, and not during production phases.
- *Random Failures*: showing up randomly during the entire life of a system.
  - Our main focus
- *Wear Out*: at the end of its life, some components can cause the failure of a system. Pre-emptive mainteinance can reduce the number of this type of failures.

R(t) & A(t) related indices

Other indices



How to identify defective products and calculate MTTF?

Burn-in test: *stress* the system with excessive temperature, voltage, current, humidity so to accelerate wear out.

Reliability & Availability

Two different points of view

"**reliability**: does not break down …"

"**availability**: even if it breaks down, it is working when needed …"

Could you provide an example of system with high availability but low reliability?

Reliability & Availability

Two different points of view

“**reliability**: does not break down …”

“**availability**: even if it breaks down, it is working when needed …”

Example:
a system that fails, on average, once per hour but which restarts automatically in ten milliseconds is not very reliable but is highly available

A(t)=0.9999972

Two points of view

Of course they are related:

- if a system is unavailable it is not delivering the specified system services

It is possible to have systems with low reliability that must be available

- system failures can be repaired quickly and do not damage data, low reliability may not be a problem (for example a database management system)

The opposite is generally more difficult…

R(t) … what to do?

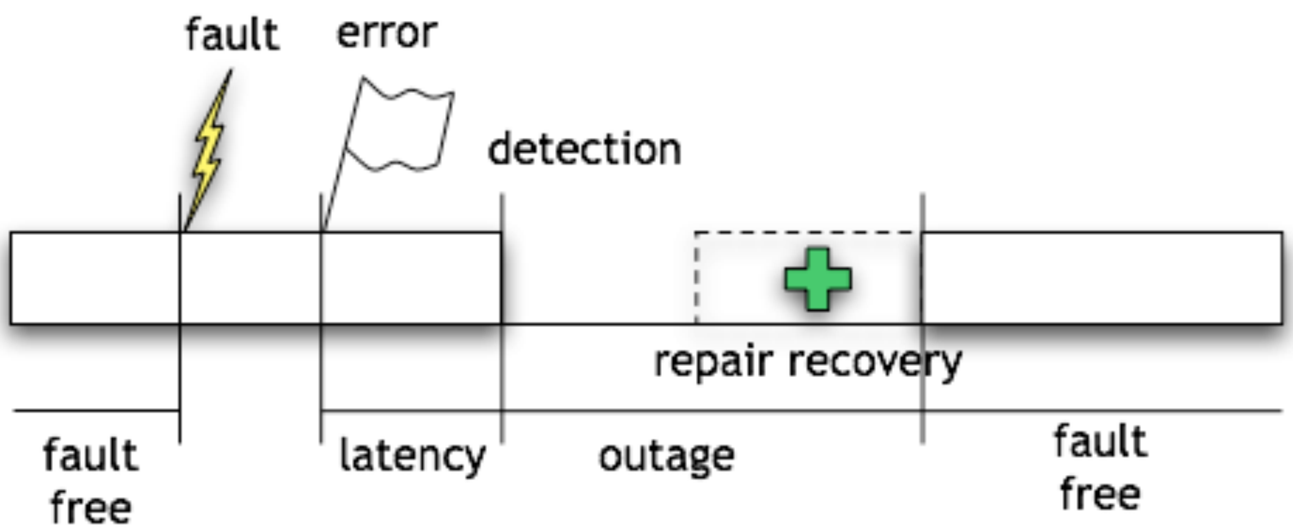
Exploitation of R(t) information is used to compute, for a complex system, its reliability in time, that is the <u>expected lifetime</u>

- computation of the MTTF

Computation of the overall reliability starting from the components' one

Reliability terminology

Term	Description
Fault	A defect within the system
Error	A deviation from the required operation of the system or subsystem
Failure	The system fails to perform its required function



Reliability terminology

An example: a flying drone with an automatic radar-guided landing system

Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory

Failure: the drone crashes to the ground

Reliability terminology

Another example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip)

Error: some frames of the video stream are corrupted

Failure: the surgeon kills the patient

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip) but the corrupted memory does not involve the video stream

Error: no frames are corrupted

Failure: the surgeon carries out the procedure

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a tele-surgery system

Fault: radioactive ions make some memory cells change value (bitflip) but the corrupted memory does not involve the video stream

Error: no frames are corrupted

Failure: the surgeon carries out the procedu

Non activated fault

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a flying drone with automatic radar-guided landing

Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory, but then, based on subsequent correct radar measurements it is able to recover the right trajectory

Failure: the drone safely lands

Reliability terminology

Not always the *fault – error – failure chain* closes

example: a flying drone with automatic radar-guided landing

Fault: electromagnetic disturbances interfere with a radar measurement

Error: the radar-guided landing system calculates a wrong trajectory, but then, based on subsequent correct rada[illegible]ments it is ab[illegible]

Non propagated
(or absorbed) error

Failure: the drone safely lands

Reliability Block Diagrams

Reliability Block Diagrams

An inductive model where a system is divided into blocks that represent distinct elements such as components or subsystems.

Reliability Block Diagrams

An inductive model where a system is divided into blocks that represent distinct elements such as components or subsystems.

Every element in the RBD has its own reliability (previously calculated or modelled)

Reliability Block Diagrams

An inductive model where a system is divided into blocks that represent distinct elements such as components or subsystems.

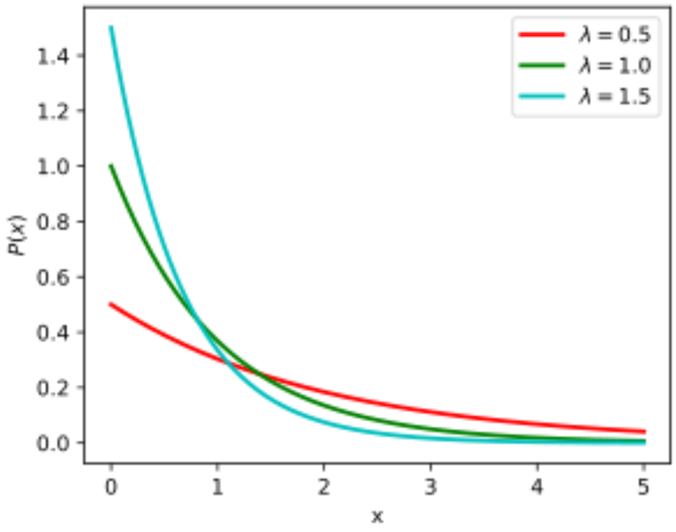
Every element in the RBD has its own reliability (previously calculated or modelled)

Blocks are then combined together to model all the possible *success paths*

Review: Exponential Distribution

Assuming that a failures occurs according to a Poisson model, it models the time between two successive failures:

- Probability density function: $f(t;\lambda) = \lambda e^{-\lambda t},\ t \geq 0,\ \lambda > 0$
- Cumulative density function: $P(T \leq t) = \int_0^t f(s;\lambda)\, ds = 1 - e^{-\lambda t}$
- Expected value: $E[T] = \frac{1}{\lambda}$
- Variance: $\sigma^2(T) = \frac{1}{\lambda^2}$

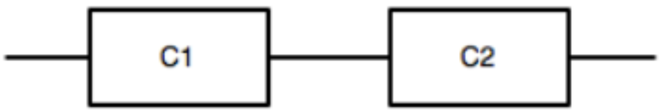


Reliability: $R(t) = P(T \geq t) = e^{-\lambda t}$

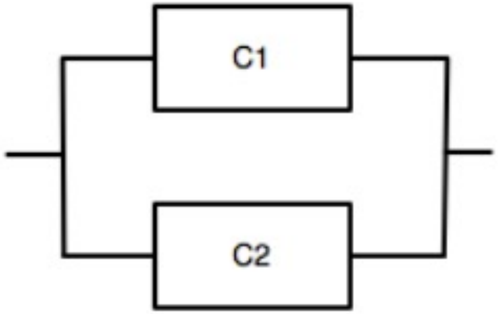
$\lambda(t)$: failure rate

Reliability Block Diagrams

RBDs are an approach to compute the reliability of a system starting from the reliability of its components



components in series



components in parallel

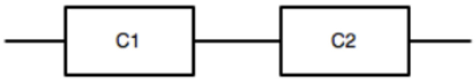
All components must be healthy for the system to work properly

If one component is healthy the system works properly

Reliability Block Diagrams

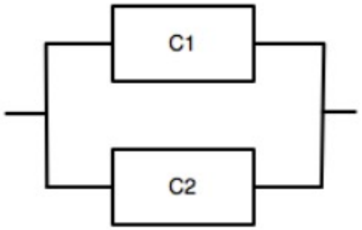
Series:

$$R_S(t) = R_{C1}(t) * R_{C2}(t)$$



Parallel:

$$R_S(t) = 1 - \left[(1 - R_{C1}(t)) * (1 - R_{C2}(t))\right]$$



$$R_S(t) = R_{C1}(t) + R_{C2}(t) - R_{C1}(t) * R_{C2}(t)$$

Reliability Block Diagrams

series

In general, if system S is composed by components with a reliability having an exponential distribution (very common case)

Failure in time

$$R_s(t) = e^{-\lambda_s t} \quad \text{where} \quad \lambda_s = \sum_{i=1}^{n} \lambda_i$$

Reliability Block Diagrams

series

In general, if system S is composed by components with a reliability having an exponential distribution (very common case)

Failure in time

$$R_s(t) = e^{-\lambda_s t} \quad \text{where} \quad \lambda_s = \sum_{i=1}^{n} \lambda_i$$

$$MTTF_S = \frac{1}{\lambda_S} = \frac{1}{\sum_{i=1}^{n} \lambda_i} = \frac{1}{\sum_{i=1}^{n} \frac{1}{MTTF_i}}$$

Reliability Block Diagrams

series

A special case: when all components are identical

$$R_s(t) = e^{-\lambda_s t}$$

$$R_S(t) = e^{-n\lambda t} = e^{-\frac{nt}{MTTF_1}}$$

$$MTTF_S = \frac{MTTF_1}{n}$$

Reliability Block Diagrams

series

Availability:

$$A_S = \prod_{i=1}^{n} \frac{MTTF_i}{MTTF_i + MTTR_i}$$

When all components are the same:

$$A_S(t) = A_1(t)^n \qquad A = \left(\frac{MTTF_1}{MTTF_1 + MTTR_1} \right)^n$$

Reliability Block Diagrams

parallel

System P composed by *n* components

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

Availability

$$A_P(t) = 1 - \prod_{i=1}^{n} \left(1 - A_i(t)\right)$$

$$A_P = 1 - \prod_{i=1}^{n} \left(1 - A_i\right) = 1 - \prod_{i=1}^{n} \frac{MTTR_i}{MTTF_i + MTTR_i}$$

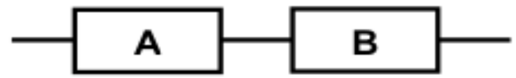
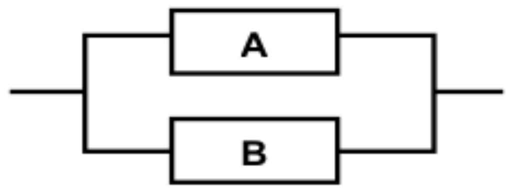
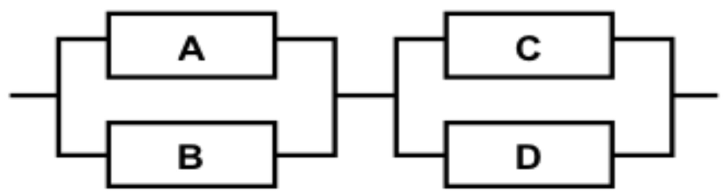
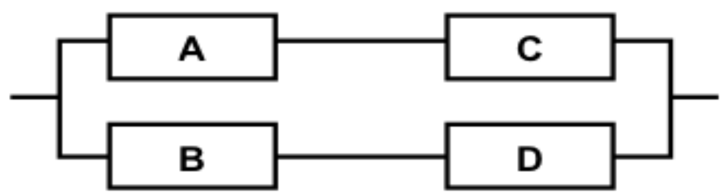
Reliability Block Diagrams (recap)

$$R_s = \prod_i^n R_i$$

$$R_s = 1 - \prod_i^n (1 - R_i)$$

Component redundancy

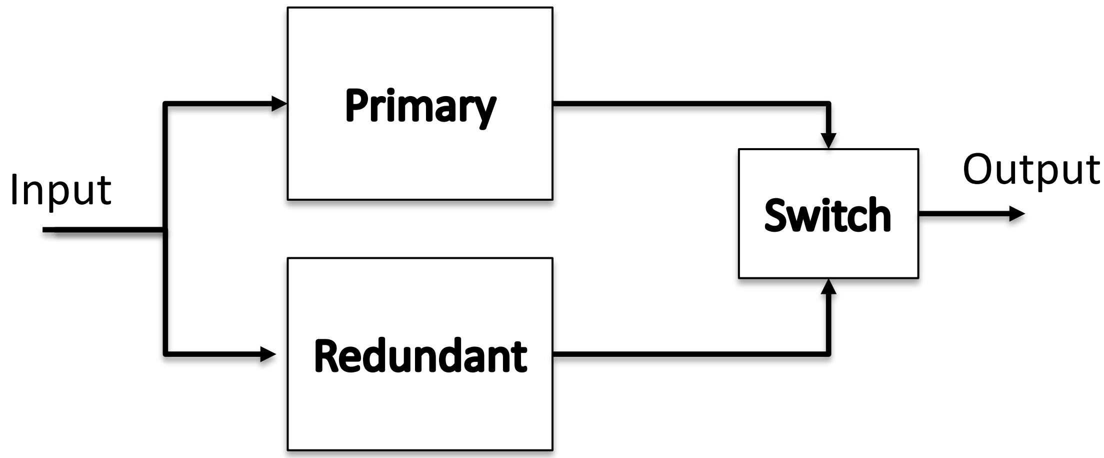
System redundancy

Type	Block Diagram Representation	System Reliability (R_S)
Series		$R_S = R_A R_B$ R_A = reliability, component A R_B = reliability, component B
Parallel		$R_S = 1\text{-}(1\text{-}R_A)(1\text{-}R_B)$
Series-Parallel		$R_S = [1\text{-}(1\text{-}R_A)(1\text{-}R_B)]*$ $[1\text{-}(1\text{-}R_C)(1\text{-}R_D)]$ R_C = reliability, component C R_D = reliability, component D
Parallel-Series		$R_S = 1\text{-}(1\text{-}R_A R_C)*$ $(1\text{-}R_B R_D)$

Standby redundancy

A system may be composed of two parallel replicas:

- The primary replica working all time, and
- The redundant replica (generally disable) that is activated when the primary replica fails



Standby redundancy

A system may be composed of two parallel replicas:

- The primary replica working all time, and
- The redundant replica (generally disable) that is activated when the primary replica fails

What do we need for such a redundancy to be operational?

Standby redundancy

A system may be composed of two parallel replicas:

- The primary replica working all time, and
- The redundant replica (generally disable) that is activated when the primary replica fails

Obviously we need:

- A mechanism to determine whether the primary replica is working properly or not (on-line self check)
- A dynamic switching mechanism to disable the primary replica and activate the redundant one

Standby redundancy

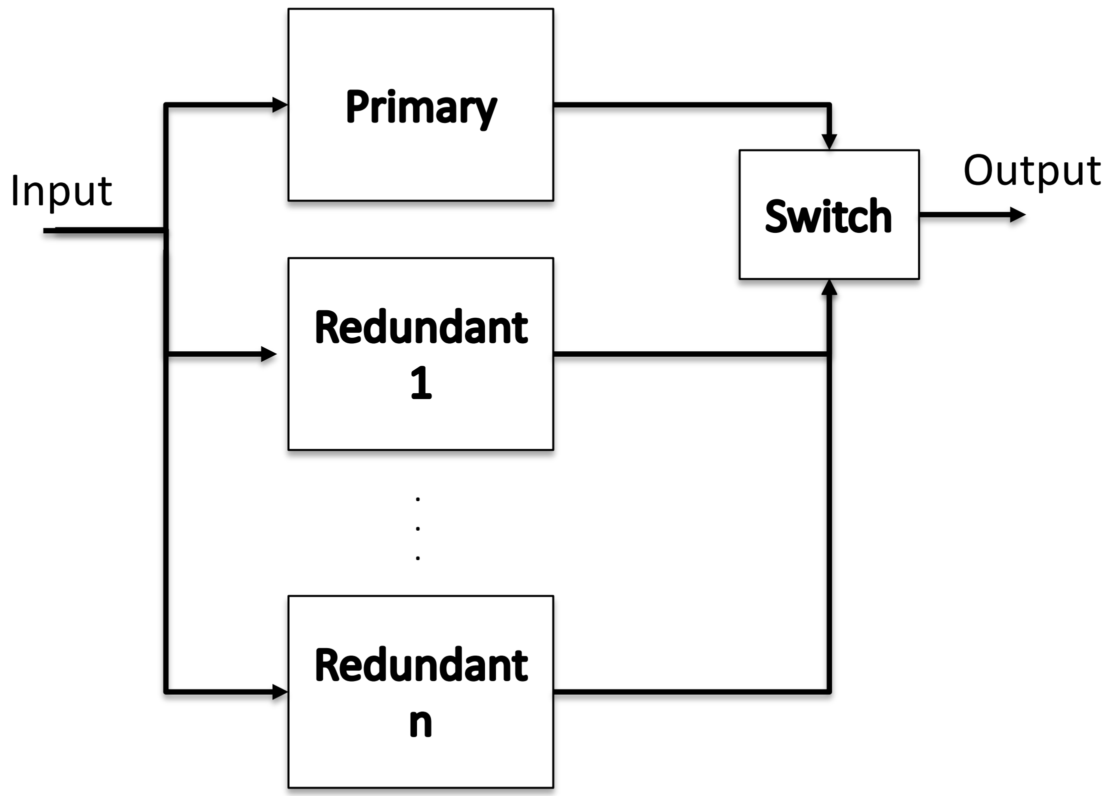
Standby Parallel Model	System Reliability
Equal failure rates, perfect switching	$R_s = e^{-\lambda t}(1+\lambda t)$
Unequal failure rates, perfect switching	$R_s = e^{-\lambda_1 t} + \lambda_1 (e^{-\lambda_1 t} - e^{-\lambda_2 t}) / (\lambda_2 - \lambda_1)$
Equal failure rates, imperfect switching	$R_s = e^{-\lambda t}(1+ R_{switch}\lambda t)$
Unequal failure rates, imperfect switching	$R_s = e^{-\lambda_1 t} + R_{switch}\lambda_1 (e^{-\lambda_1 t} - e^{-\lambda_2 t}) / (\lambda_2 - \lambda_1)$

where

R_s = System reliability
λ = Failure rate
t = Operating time
R_{switch} = Switching reliability

Standby redundancy

More in general, a system having one primary replica and *n* redundant replicas (with identical replicas and perfect switching)



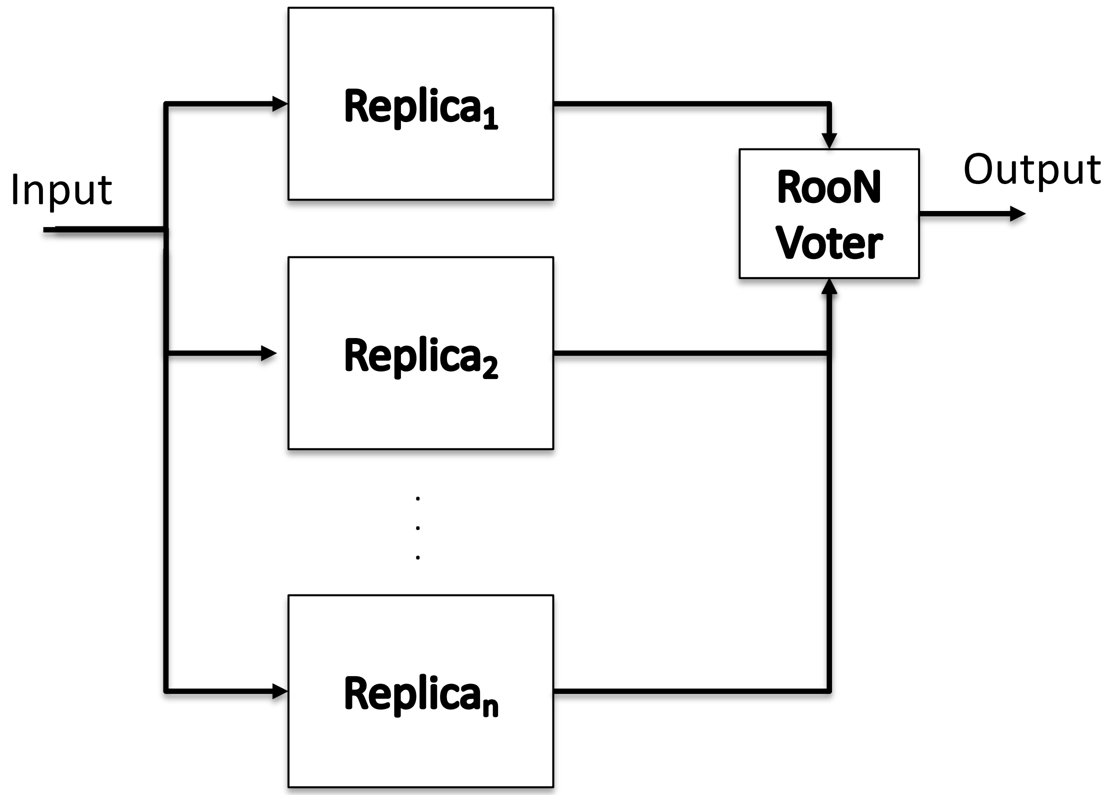
Standby redundancy

More in general, a system having one primary replica and *n* redundant replicas (with identical replicas and perfect switching)

$$R(t) = e^{-\lambda t} \sum_{i=0}^{n-1} \frac{(\lambda t)^i}{i!}$$

r out of *n* redundancy (RooN)

A system composed of *n* identical replicas where at least *r* replicas have to work fine for the entire system to work fine



r out of *n* redundancy (RooN)

R_s = System reliability
R_c = Component reliability
R_V= Voter Reliability
n = Number of components
r = Minimum number of components which must survive

$$R_S(t) = RV \sum_{i=r}^{n} R_c^i (1 - R_C)^{n-i} \frac{n!}{i!\,(n-i)!}$$

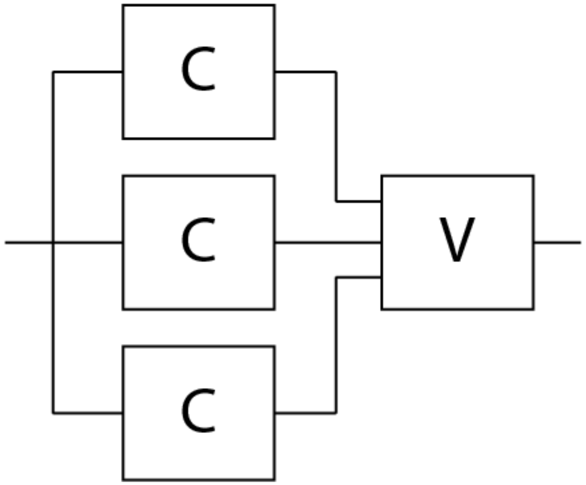
r out of *n* redundancy (RooN)

R_s = System reliability
R_c = Component reliability
R_V= Voter Reliability
n = Number of components
r = Minimum number of components which must survive

$$R_S(t) = RV \sum_{i=r}^{n} R_c^i (1 - R_C)^{n-i} \boxed{\frac{n!}{i!\,(n-i)!}}$$

Binomial coefficient
$\binom{n}{i}$

Triple Modular Redundancy – TMR



System works properly if

- 2 out of 3 components work properly AND the voter works properly

$$R_{TMR} = R_v[\sum_{i=2}^{3}\binom{3}{i}R_m^i(1-R_m)^{3-i}] = R_v[R_m^3 + 3R_m^2(1-R_m)] = R_v(3R_m^2 - 2R_m^3)$$

$$MTTF_{TMR} = \int_0^\infty R_{TMR}dt = \int_0^\infty R_v(3R_m^2 - 2R_m^3)dt = \int_0^\infty e^{-\lambda_v t}(3e^{-2\lambda_m t} - 2e^{-3\lambda_m t})dt$$

$$= \frac{3}{2\lambda_m + \lambda_v} - \frac{2}{3\lambda_m + \lambda_v} \cong \frac{3}{2\lambda_m} - \frac{2}{3\lambda_m} = \left(\frac{5}{6}\right)\left(\frac{1}{\lambda_m}\right) = \frac{5}{6}MTTF_{simplex}$$

TMR

- $MTTF_{TMR}$ is shorter than $MTTF_{symplex}$
- Can tolerate transient faults and permanent faults
- Higher reliability (for shorter missions)

When do we have the same reliability?

- $R_{TMR}(t) = R_C(t)$

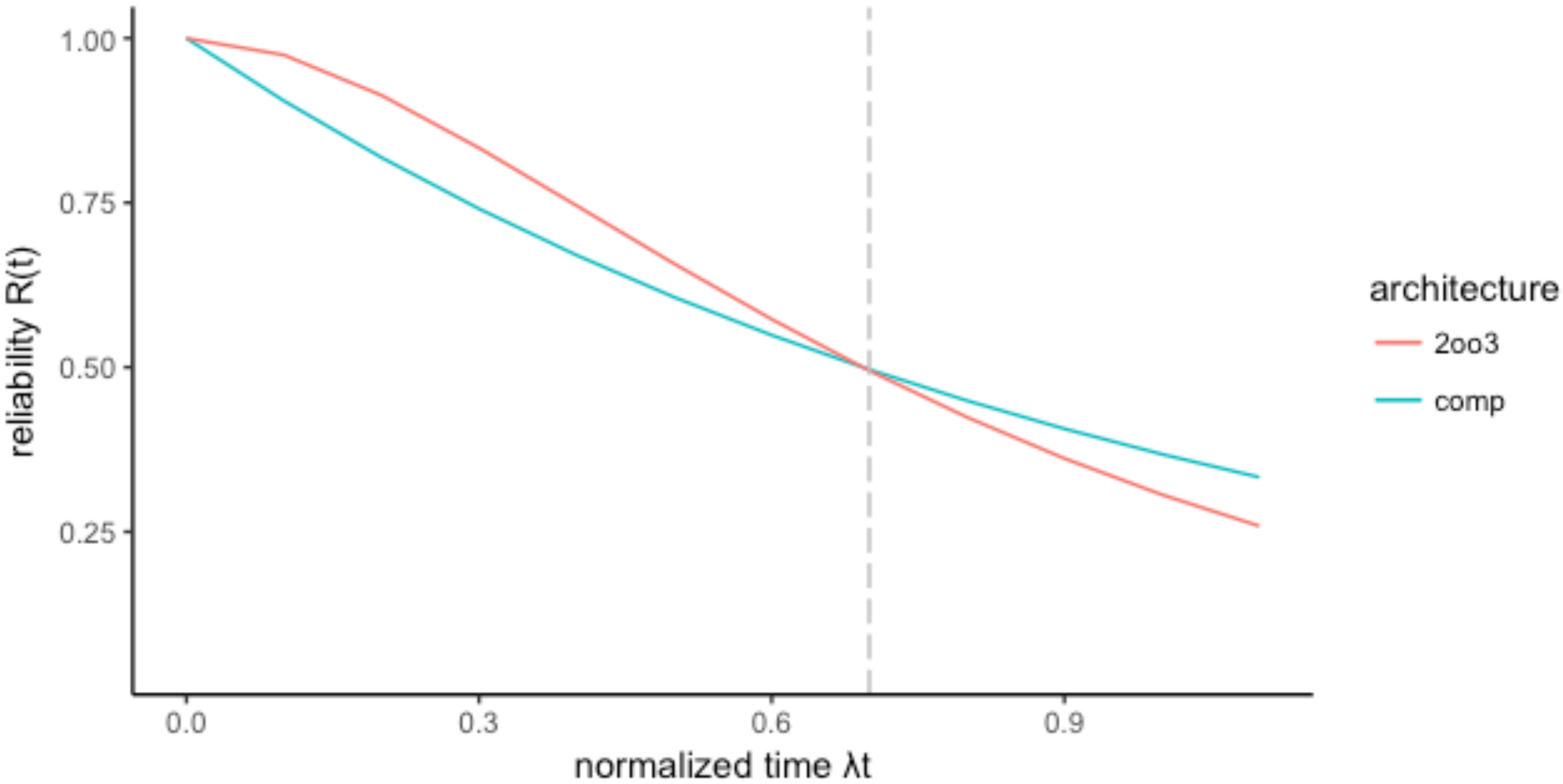
$$3e^{-2\lambda_m t} - 2e^{-3\lambda_m t} = e^{-\lambda_m t}$$

$$t = \frac{\ln 2}{\lambda_m} \cong 0.7\ MTTF_C$$

➡ $R_{TMR}(t) > R_C(t)$ when the mission time is shorter than 70% of $MTTF_C$

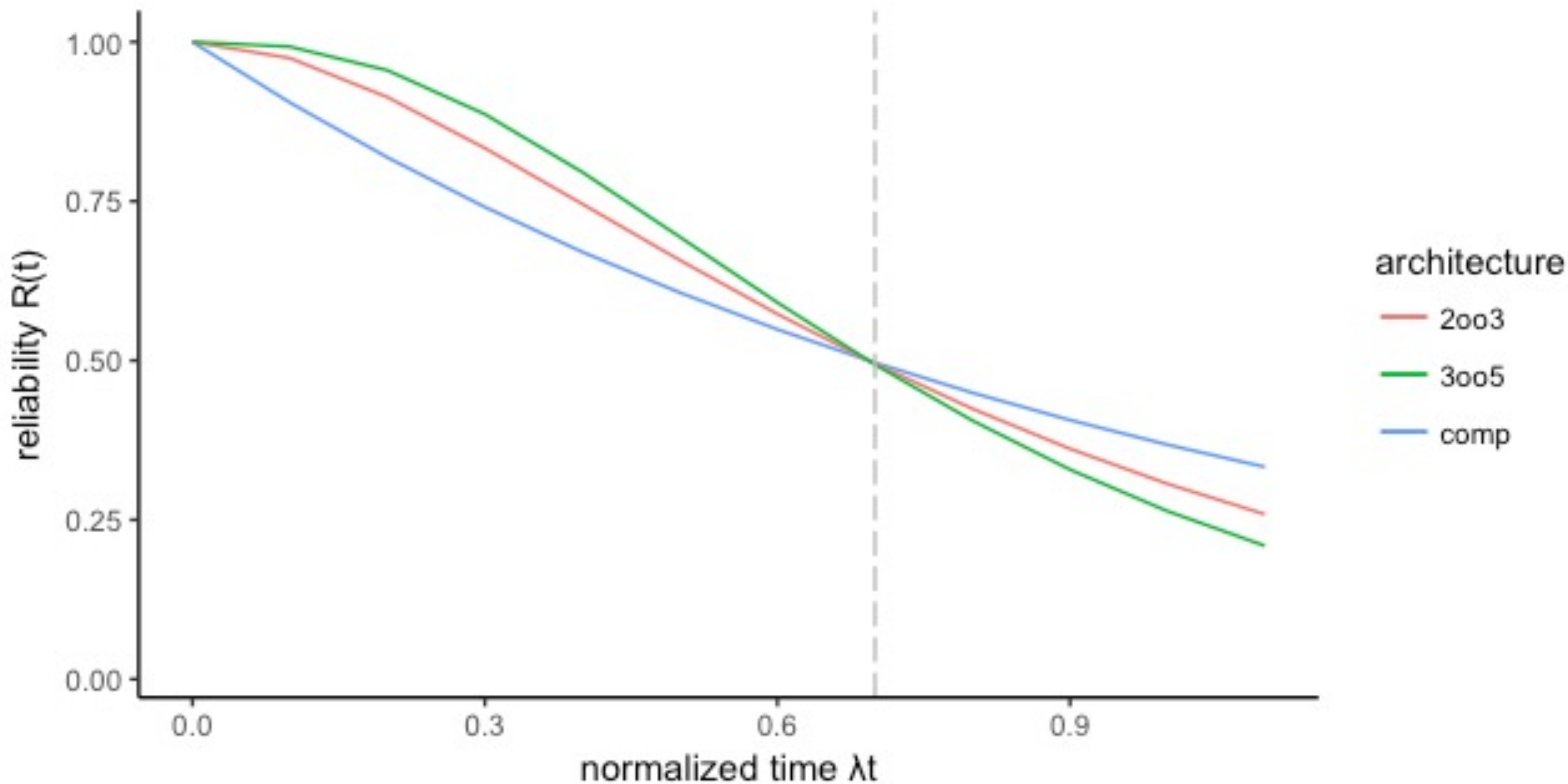
TMR

TMR: 2 out of 3 components (voter is a 'perfect' element)



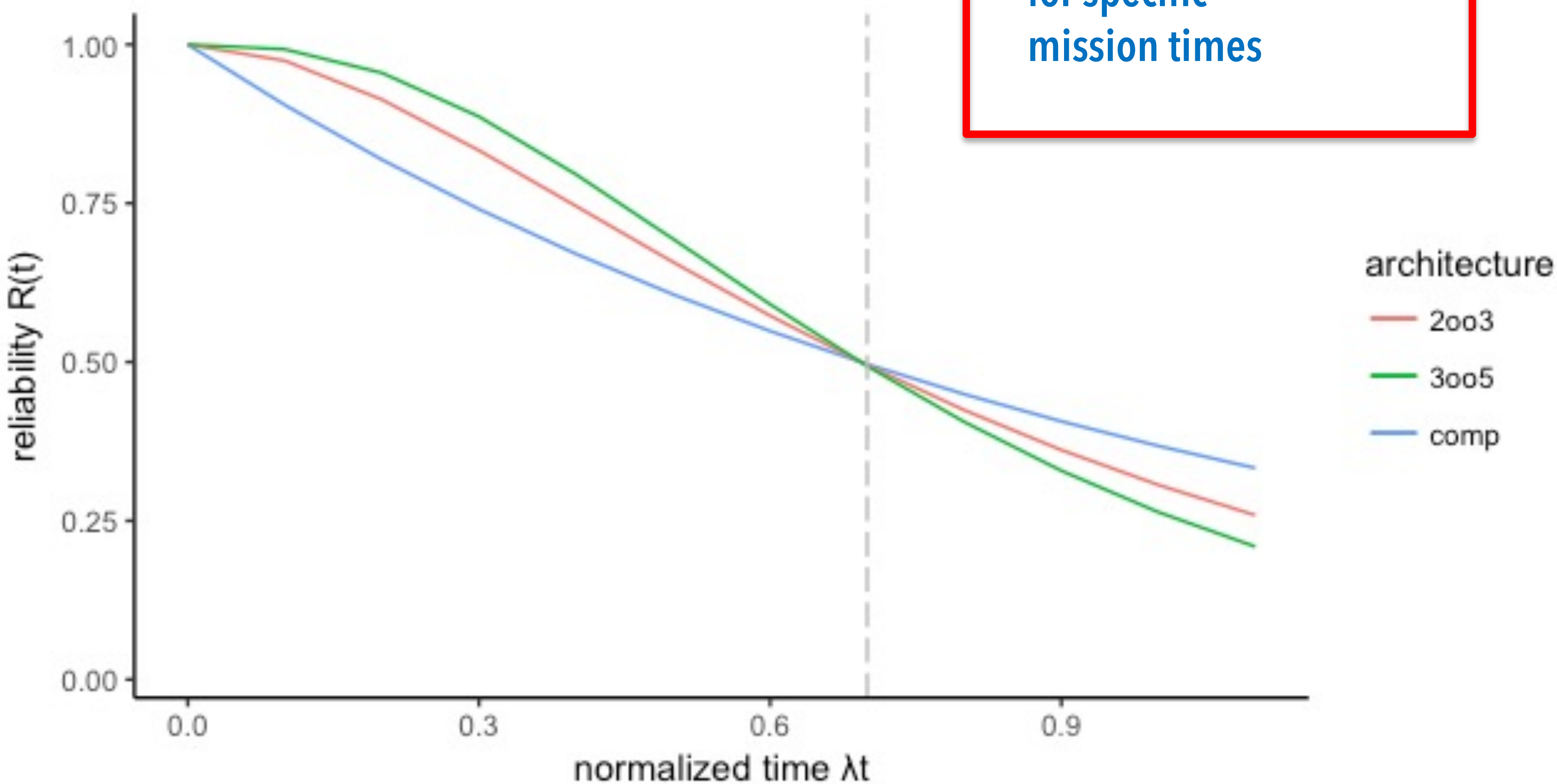
nMR

TMR: 2oo3 and nMR: 3oo5



nMR

TMR: 2oo3 and nMR: 3oo5



Redundancy is useful for specific mission times

Example 1

$R_A = 0.95$

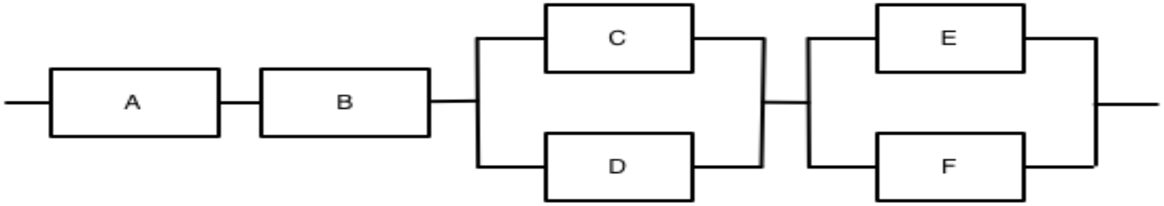
$R_B = 0.97$

$R_C = 0.99$

$R_D = 0.99$

$R_E = 0.92$

$R_F = 0.92$



Example 1

RBDs

$R_A = 0.95$

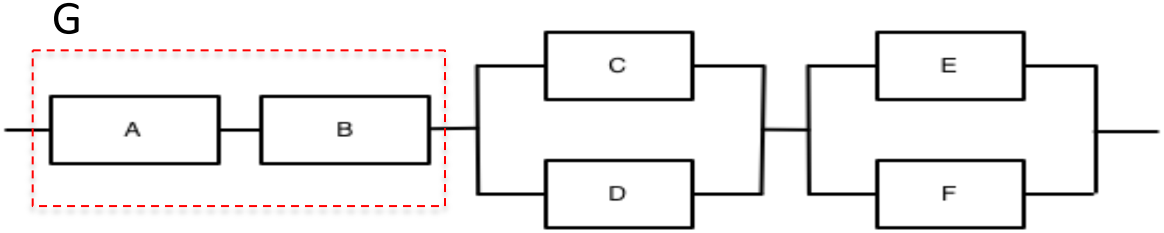
$R_B = 0.97$

$R_C = 0.99$

$R_D = 0.99$

$R_E = 0.92$

$R_F = 0.92$



$R_G = R_A * R_B$

$R_G = 0.9215$

Example 1

RBDs

$R_A = 0.95$

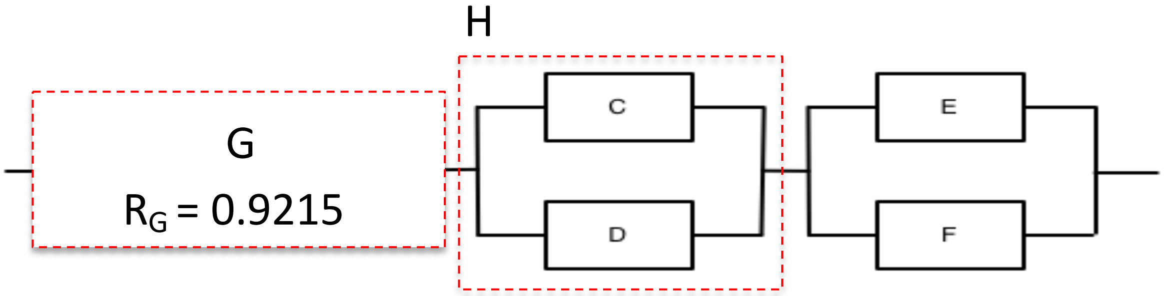
$R_B = 0.97$

$R_C = 0.99$

$R_D = 0.99$

$R_E = 0.92$

$R_F = 0.92$



$R_H = 1-[(1-R_c)*(1-R_D)]$

$R_H = 0.9999$

Example 1

$R_A = 0.95$

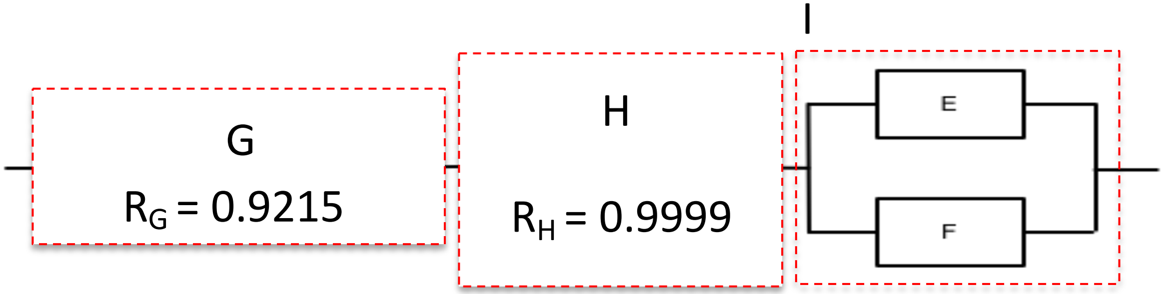
$R_B = 0.97$

$R_C = 0.99$

$R_D = 0.99$

$R_E = 0.92$

$R_F = 0.92$



$R_I = 1-[(1-R_E)*(1-R_F)]$

$R_I = 0.9936$

Example 1

RBDs

$R_A = 0.95$

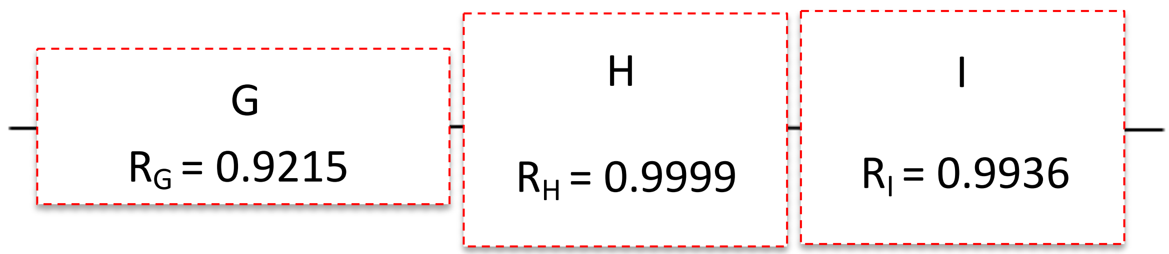
$R_B = 0.97$

$R_C = 0.99$

$R_D = 0.99$

$R_E = 0.92$

$R_F = 0.92$

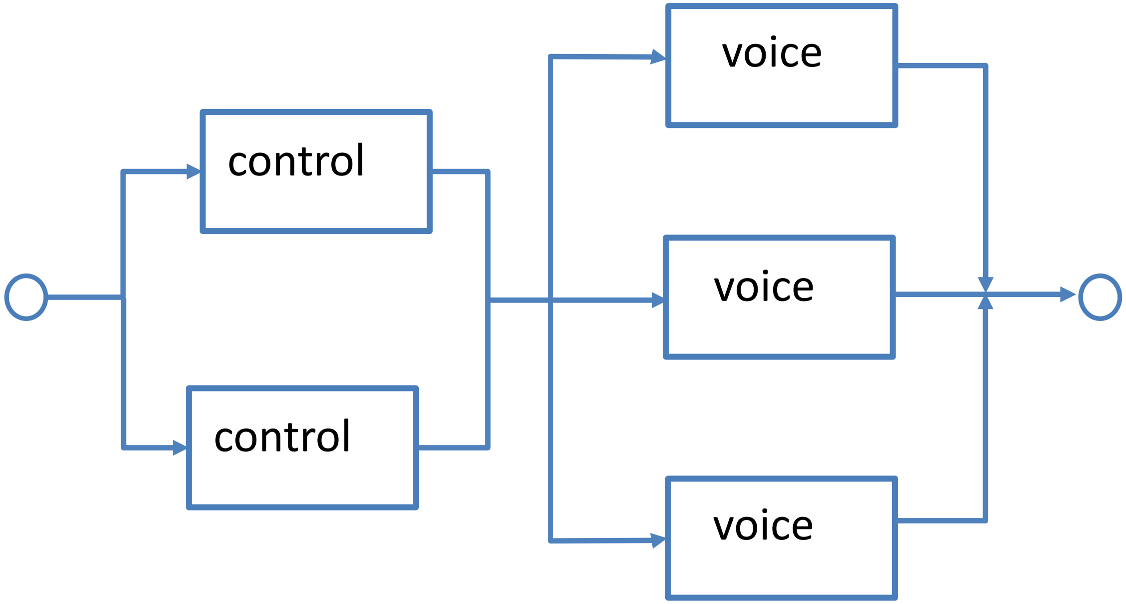


$R_S = R_G * R_H * R_I = 0.9155$

Example 2

2 control blocks and 3 voice channels:

- system is up if at least 1 control channel and at least 1 voice channel are up



Example 2 – cont'd

- Each control channel has reliability R_c
- Each voice channel has reliability R_v

- Reliability:

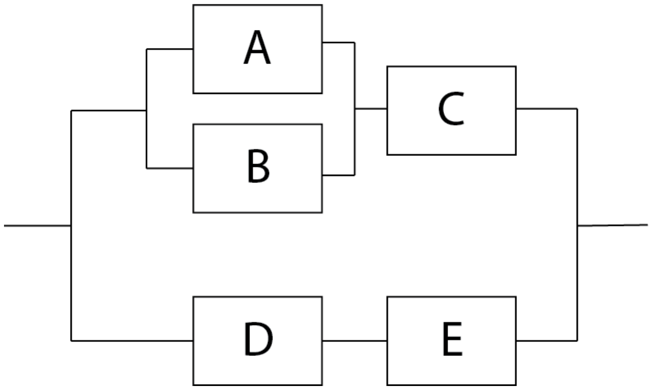
Example 2 – cont'd

- Each control channel has reliability R_c
- Each voice channel has reliability R_v

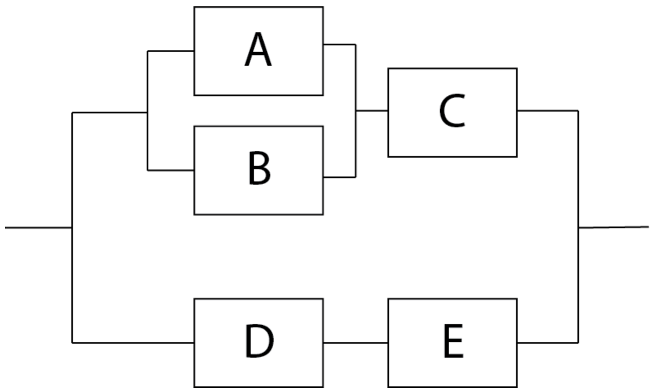
- Reliability:

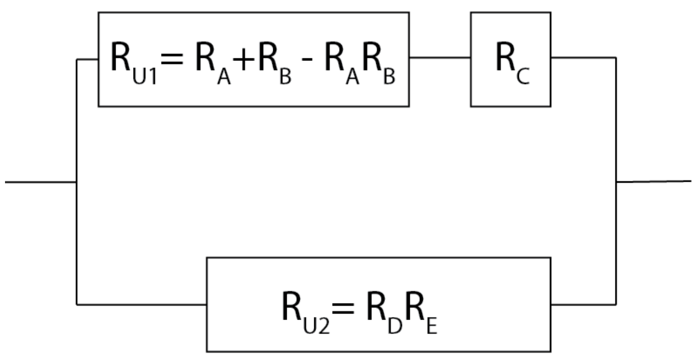
$$R = [1-(1-R_c)^2][1-(1-R_v)^3]$$

Example 3

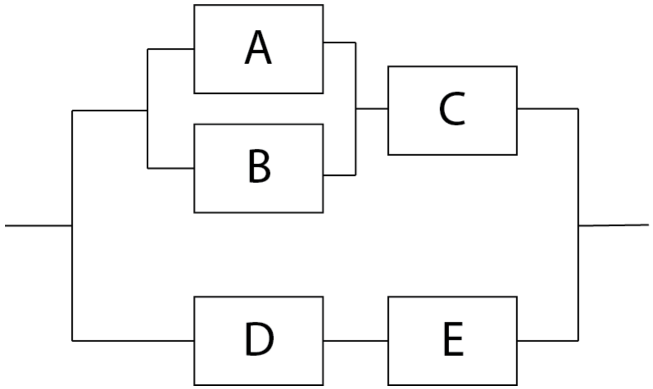


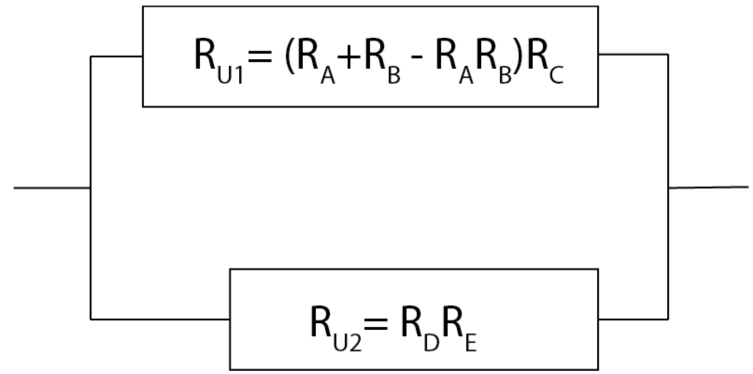
Example 3

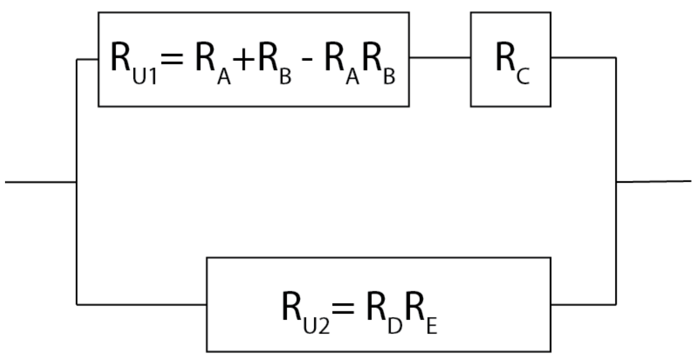




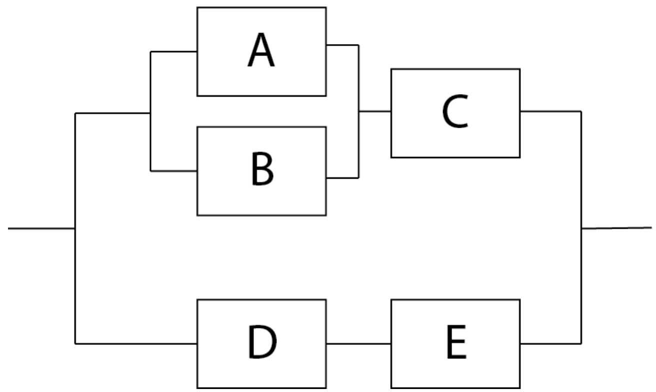
Example 3

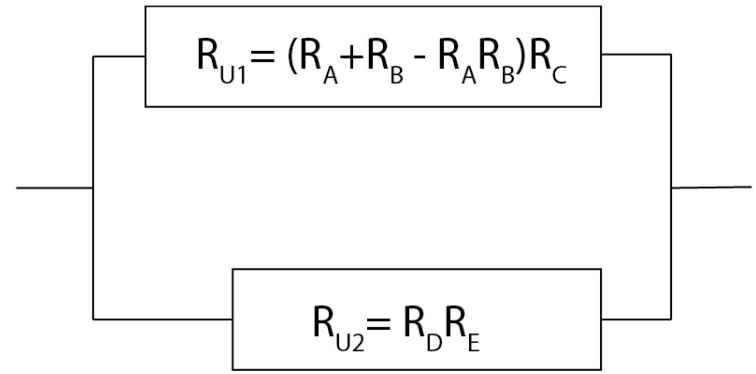


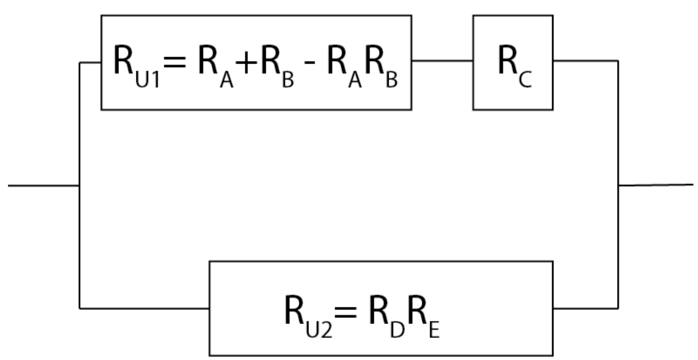


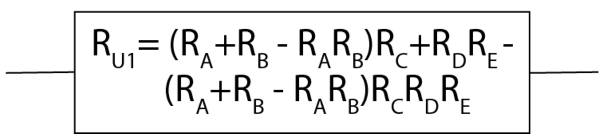


Example 3


A
B
C
D
E


$R_{U1} = (R_A + R_B - R_A R_B) R_C$
$R_{U2} = R_D R_E$


$R_{U1} = R_A + R_B - R_A R_B$
R_C
$R_{U2} = R_D R_E$


$R_{U1} = (R_A + R_B - R_A R_B) R_C + R_D R_E -$
$(R_A + R_B - R_A R_B) R_C R_D R_E$

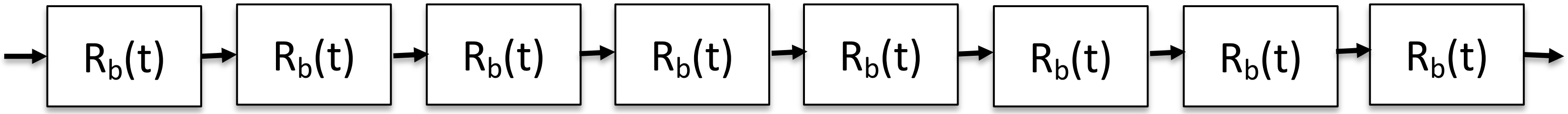
RBD: used to model a system and calculate its reliability

We have an 8-bit parallel bus within a System-on-Chip; each line of the bus may fail independently of the others; the reliability of each line of the bus is $R_b(t)$.

How would you model the entire bus using a RBD?

RBD: used to model a system and calculate its reliability

We have an 8-bit parallel bus within a System-on-Chip; each line of the bus may fail independently of the others; the reliability of each line of the bus is $R_b(t)$.

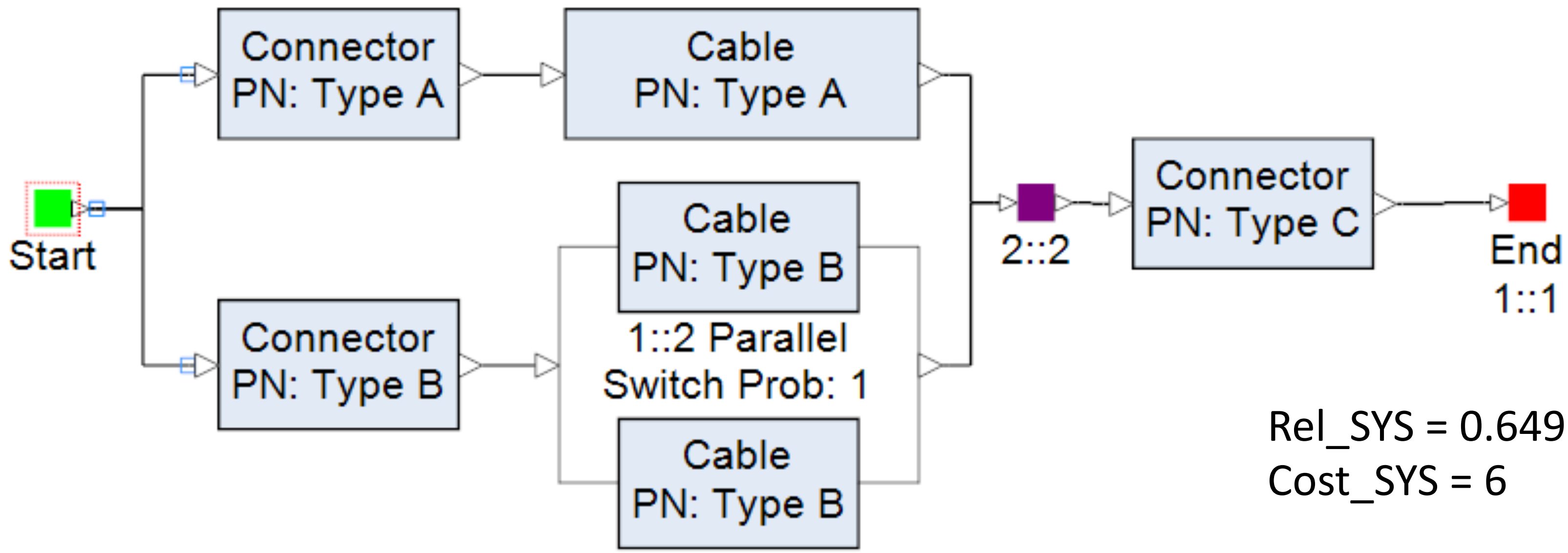


RBD: used to compare different alternatives

Cable Bundle

Each block has R = 0.9

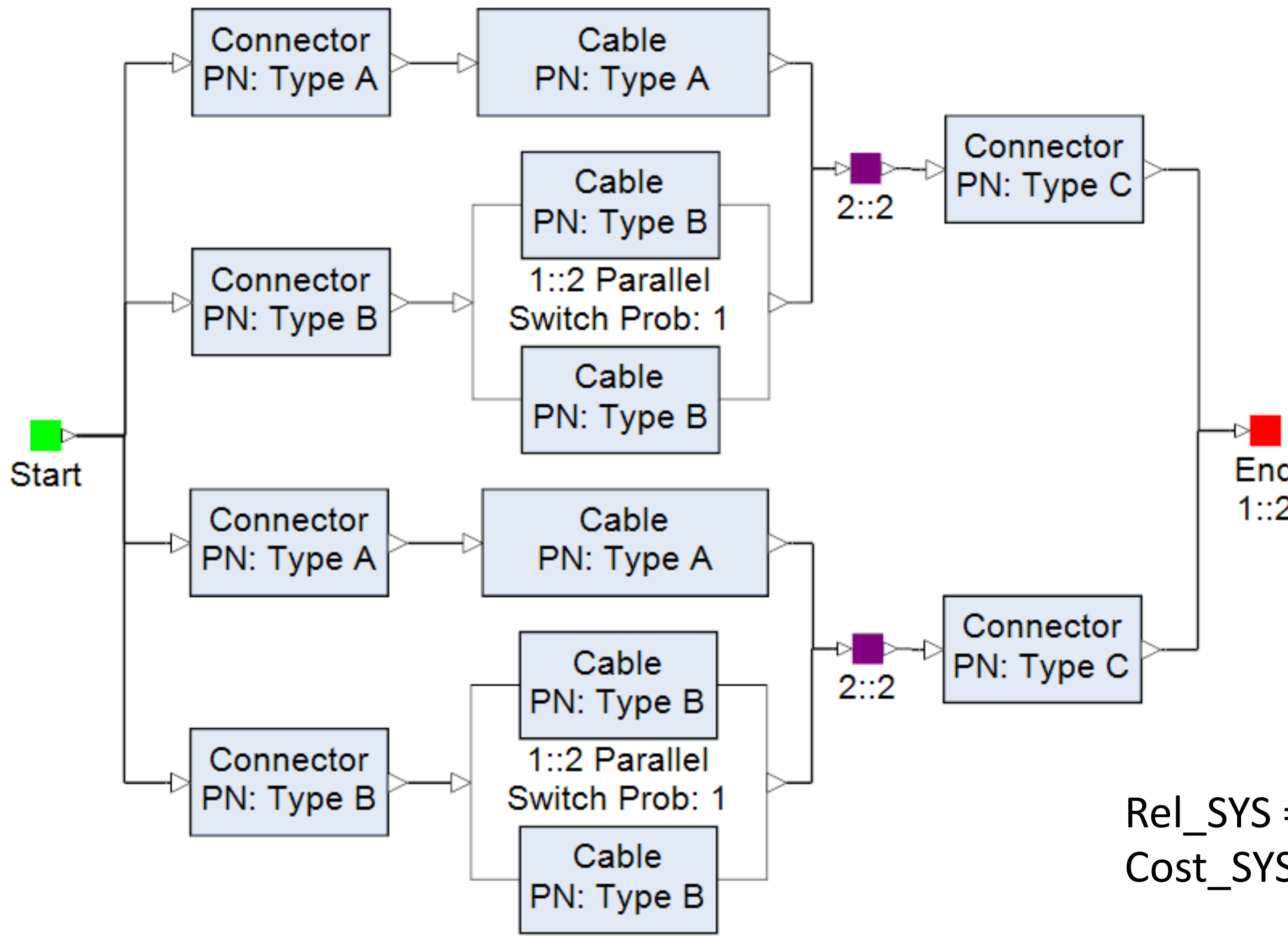
Each block costs 1



Rel_SYS = 0.649539

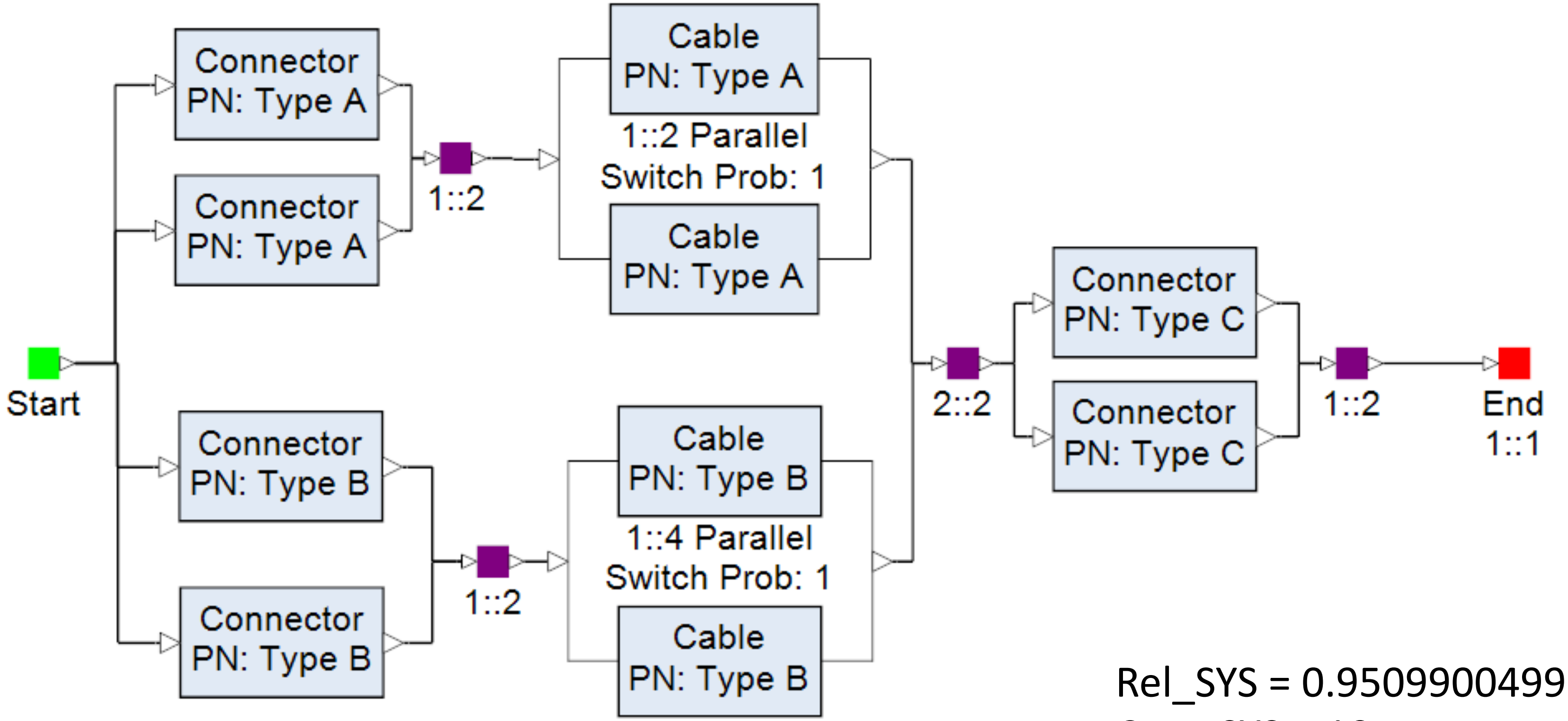
Cost_SYS = 6

Alternative 1



Rel_SYS = 0.877177
Cost_SYS = 12

Alternative 2



Rel_SYS = 0.9509900499
Cost_SYS = 10