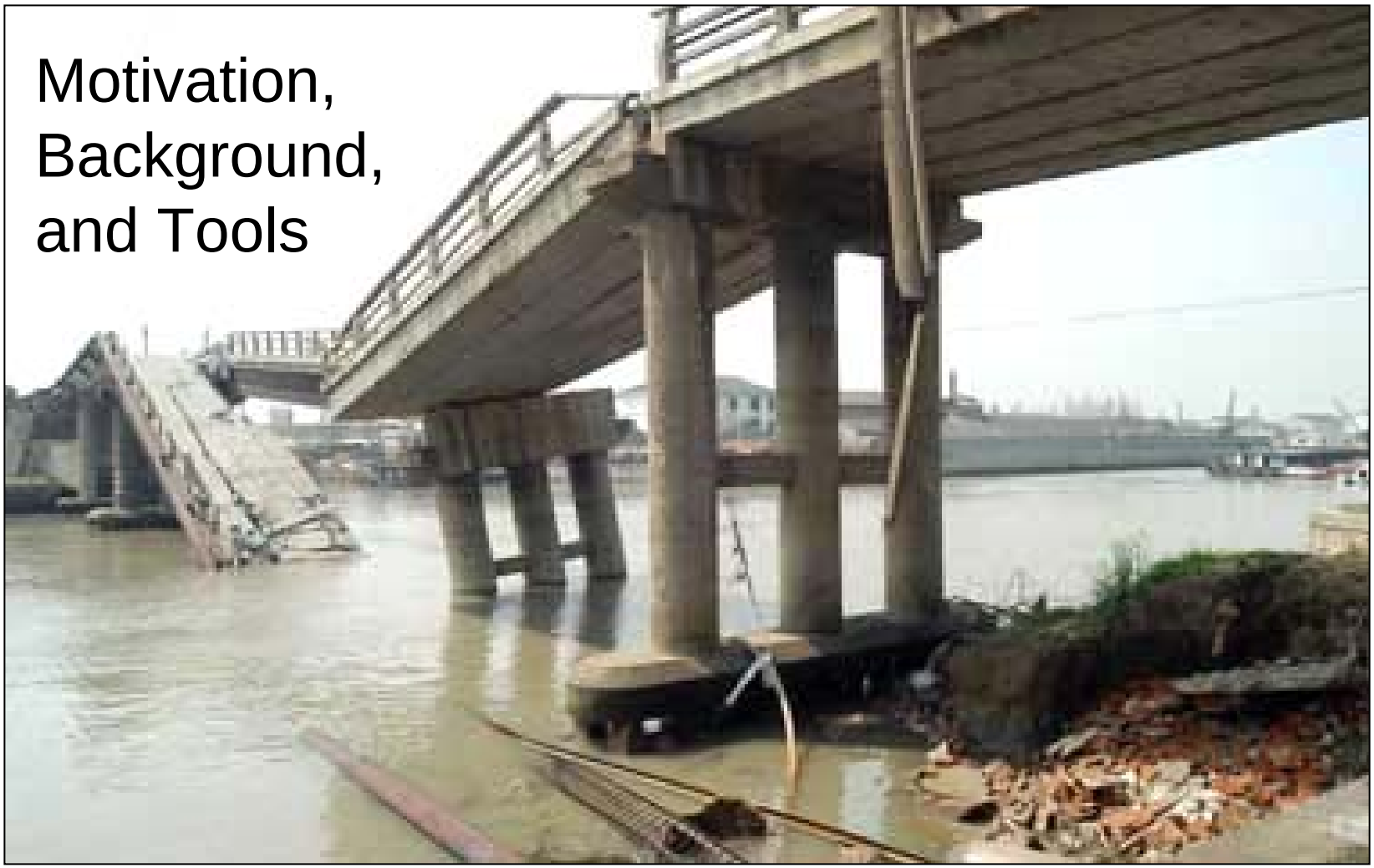
Fault-Tolerant Computing

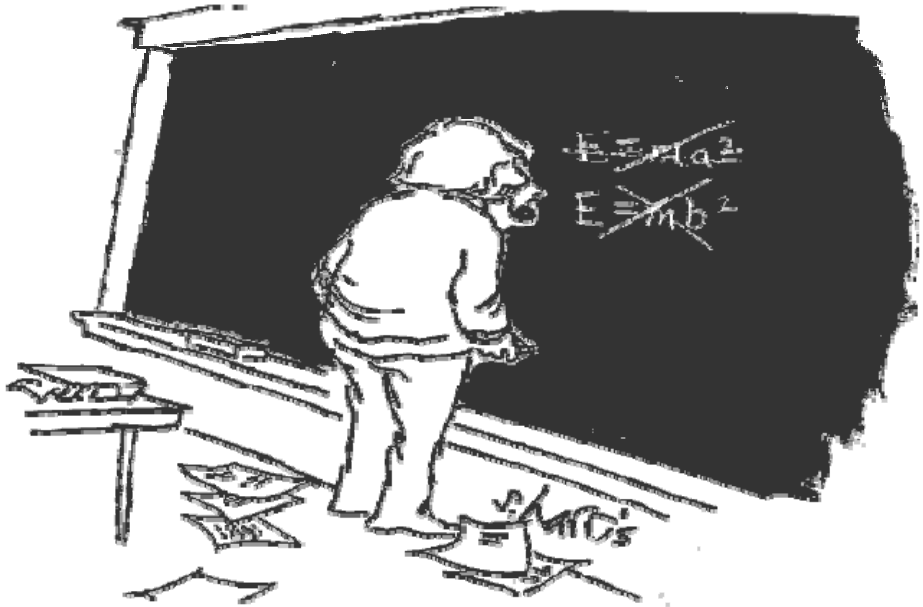


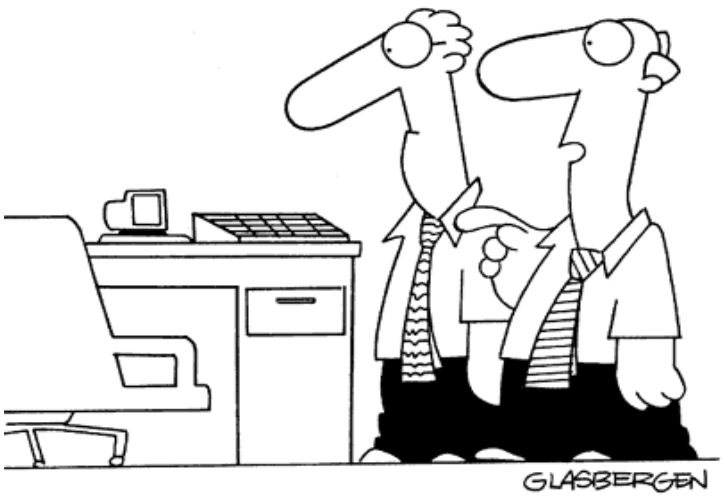
About This Presentation

This presentation has been prepared for the graduate course ECE 257A (Fault-Tolerant Computing) by Behrooz Parhami, Professor of Electrical and Computer Engineering at University of California, Santa Barbara. The material contained herein can be used freely in classroom teaching or any other educational setting. Unauthorized uses are prohibited.

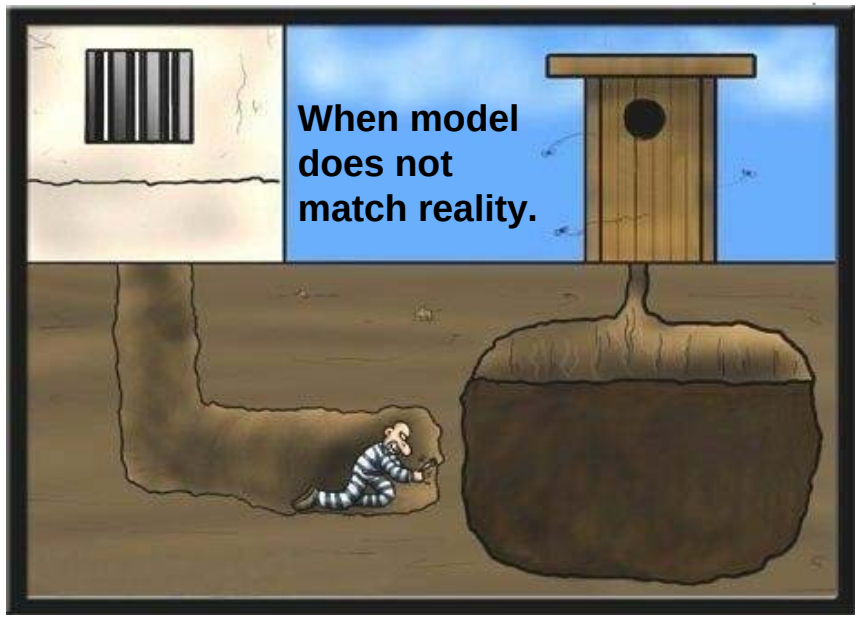
Edition	Released	Revised	Revised
First	Oct. 2006		

Combinational Modeling





"We installed little monitors because they make all of our problems look smaller."



A Motivating Example

Data files to be stored on the sites so that they remain available despite site and link malfunctions

S = Site availability (a_S in handout)

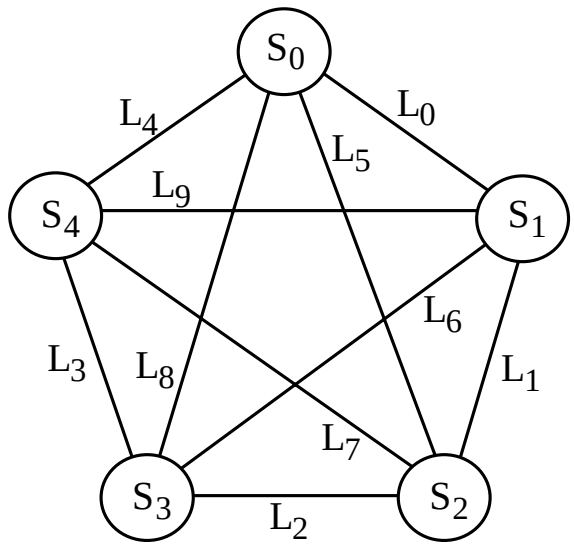
L = Link availability (a_L in handout)

Some possible strategies:

- Duplication on home site and mirror site
- Triplication on home site and 2 backups
- Data dispersion through coding

Here, we ignore the important problem of keeping the replicas consistent and do not worry about malfunction detection and attendant recovery actions

Five-site distributed computer system

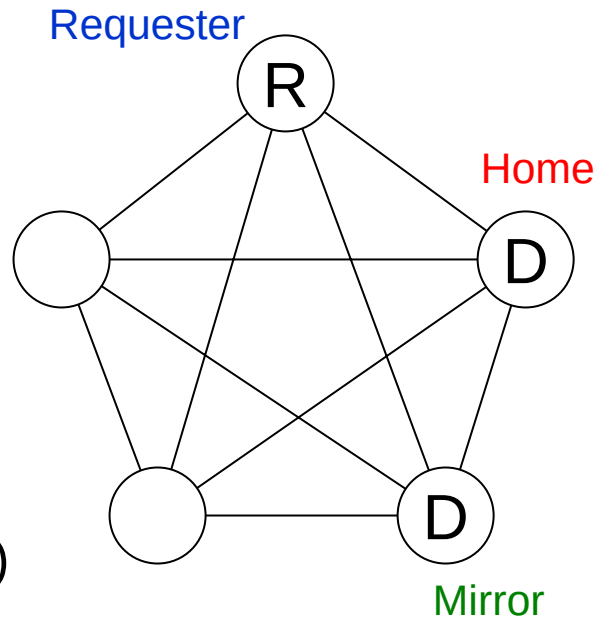


Data Availability with Home and Mirror Sites

Assume data file must be obtained directly from a site that holds it

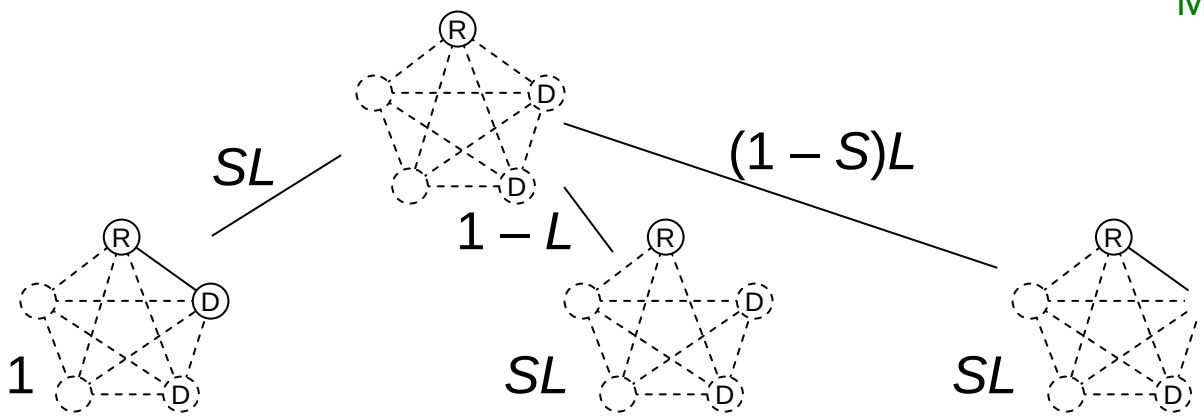
$A = SL + (1 - SL)SL = 2SL - (SL)^2$

For example, $S = 0.99$, $L = 0.95$, $A = 0.9965$
With no redundancy, $A = 0.99 \times 0.95 = 0.9405$



Combinational modeling:
Consider all combinations of circumstances that lead to availability/success (unavailability/failure)

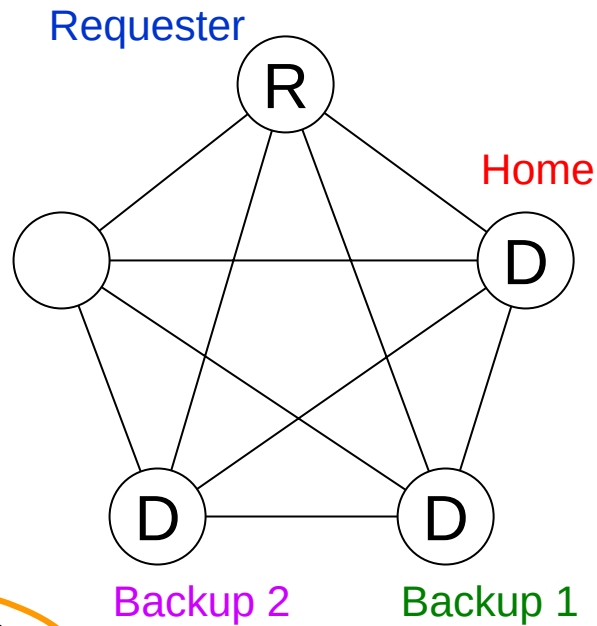
Analysis by considering mutually exclusive subcases

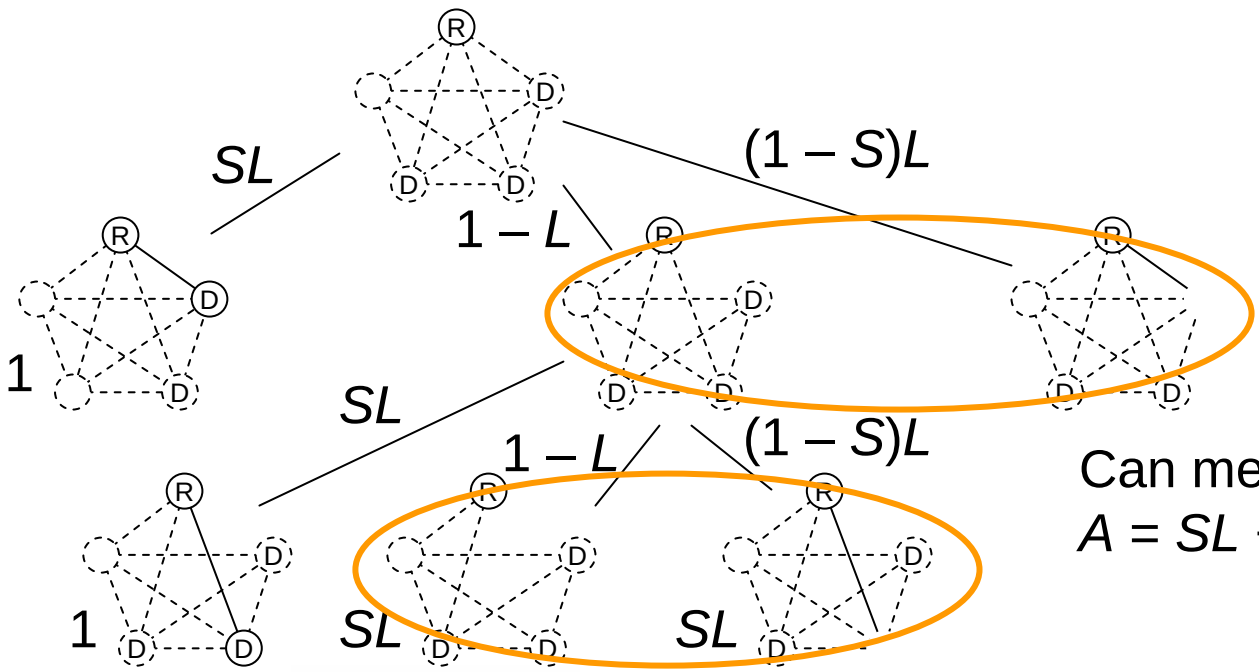


Data Availability with Triplication

$A = SL + (1 - SL)SL + (1 - SL)^2SL$
$= 3SL - 3(SL)^2 + (SL)^3$

For example, $S = 0.99$, $L = 0.95$, $A = 0.9998$
With duplication, $A = 0.9965$
With no redundancy, $A = 0.9405$



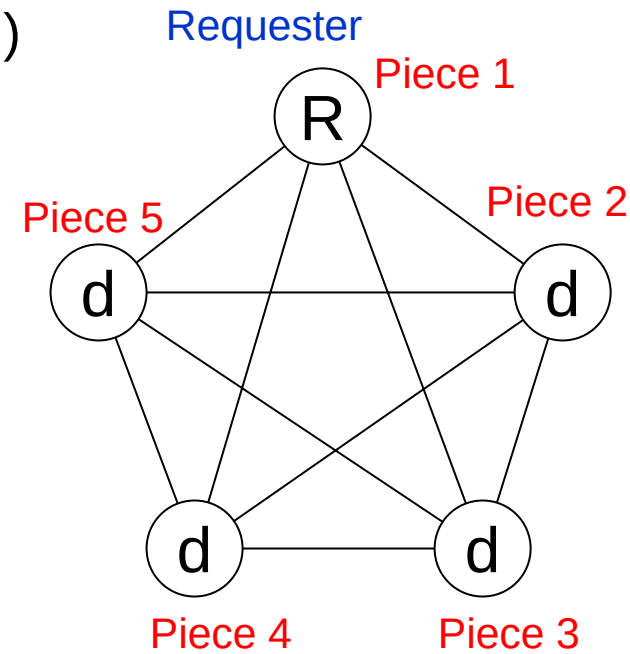


Can merge these two cases
$A = SL + (1 - SL)$
$[SL + (1 - SL)SL]$

Data Availability with File Dispersion

Encode an l-bit file into $\approx 5l/3$ bits (67% redund.)
Break encoded file into 5 pieces of length $l/3$
Store each piece on one of the 5 sites

Any 3 of the 5 pieces can be used to reconstruct the original file



File accessible if 2 out of 4 sites accessible

$$A = (SL)^4 + 4(1 - SL)(SL)^3 + 6(1 - SL)^2(SL)^2$$

$$= 6(SL)^2 - 8(SL)^3 + 3(SL)^4$$

For example, $S = 0.99$, $L = 0.95$, $A = 0.9992$, Redundancy = 67%
With duplication, $A = 0.9965$, Redundancy = 100%
With triplication, $A = 0.9998$, Redundancy = 200%
With no redundancy, $A = 0.9405$

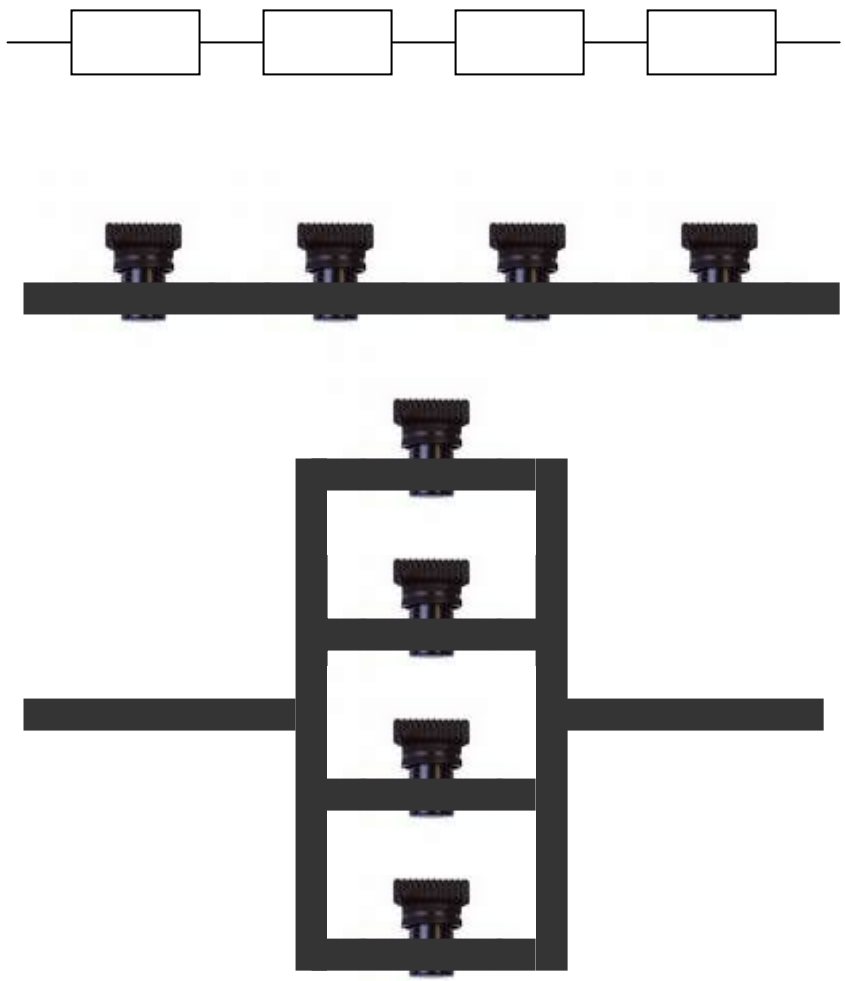
Series System

A system composed of *n* units all of which must be healthy for the system to function properly

$R = \Pi R_i$

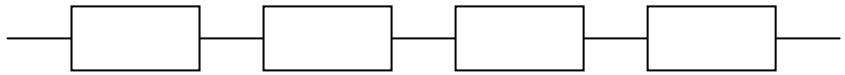
Example: Redundant system of valves in series with regard to stuck-on-shut malfunctions (tolerates stuck-on-open valves)

Example: Redundant system of valves in parallel with regard to to stuck-on-open malfunctions (tolerates stuck-on-shut valves)



Series System: Implications to Design

Assume exponential reliability law



$R_i = \exp[-\lambda_i t]$

$R = \prod R_i = \exp[-(\Sigma\lambda_i) t]$

Given the reliability goal r, find the required value of $\Sigma\lambda_i$

Assign a failure rate “budget” to each unit and proceed with its design

May have to reallocate budgets if design proves impossible or costly

Parallel System

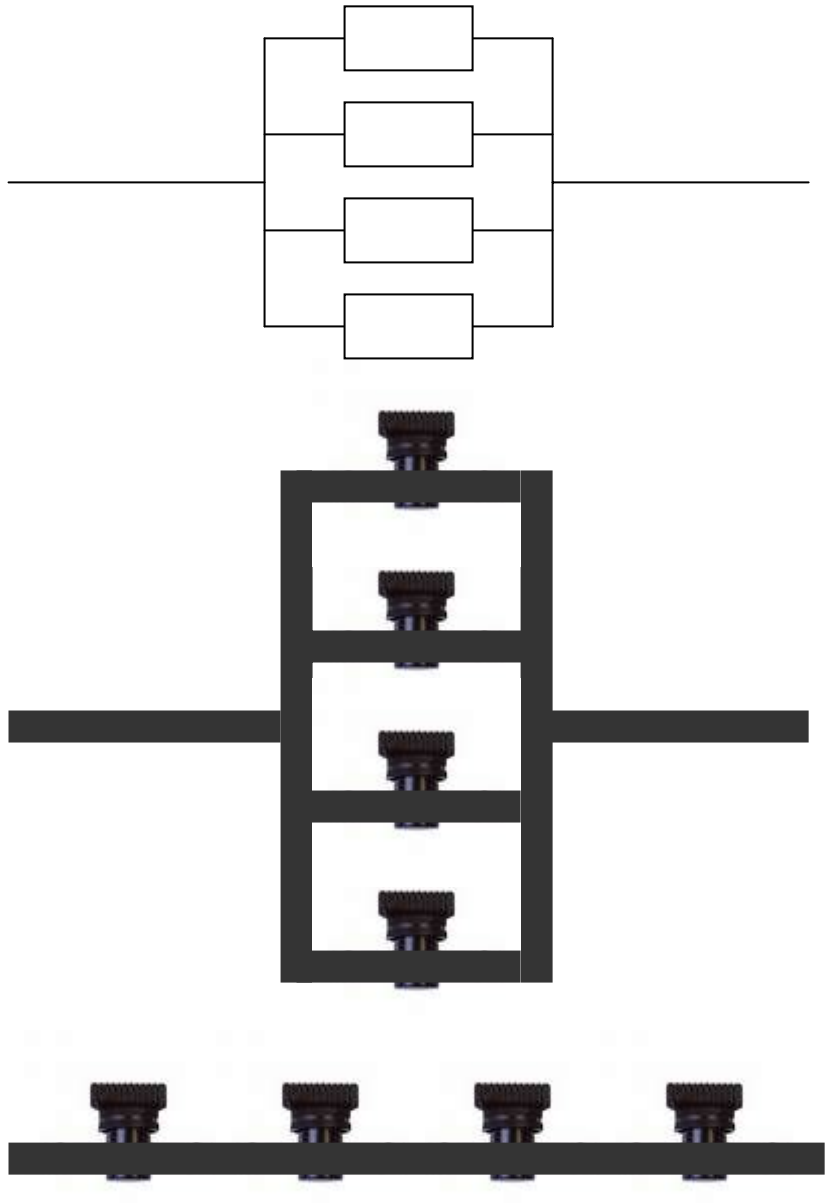
A system composed of *n* units, the health of one of which is enough for the system to function properly

$$1 - R = \prod (1 - R_i)$$

That is, the system fails only if all units malfunction

Example: Redundant system of valves in parallel with regard to stuck-on-shut malfunctions (tolerates stuck-on-shut valves)

Example: Redundant system of valves in series with regard to stuck-on-open malfunctions (tolerates stuck-on-open valves)

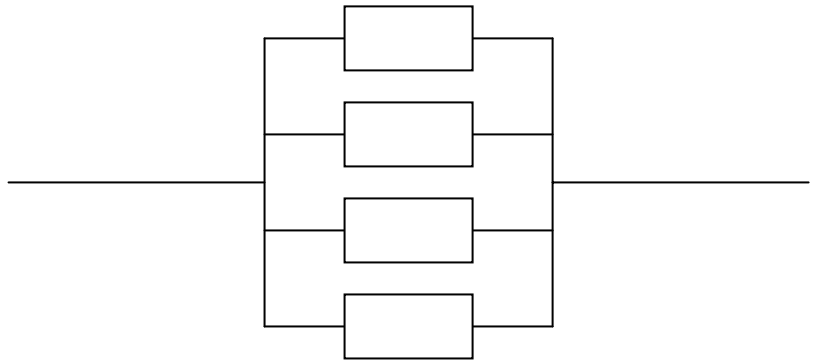


Parallel System: Implications to Design

Assume exponential reliability law

$R_i = \exp[-\lambda_i t]$

$1 - R = \prod (1 - R_i)$



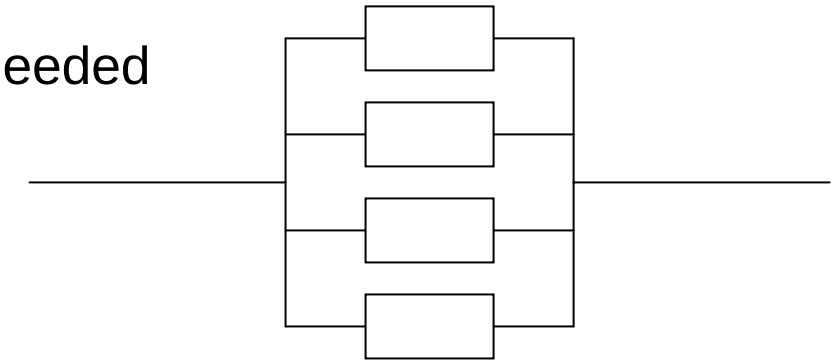
Given the reliability goal r, find the required value of $\prod (1 - R_i)$

Assign a failure probability "budget" to each unit

For example, with identical units, $1 - R_m = \sqrt[n]{1 - r}$
Assume $r = 0.9999$, $n = 4$ → $1 - R_m = 0.1$ (module reliability must be 0.9)
Conversely, for $r = 0.9999$ and $R_m = 0.9$, $n = 4$ is needed

The Concept of Coverage

For $r = 0.9999$ and $R_i = 0.9$, $n = 4$ is needed



Standby sparing: One unit works; others are also active concurrently or they may be inactive (spares)

When a malfunction of the main unit is detected, it is removed from service and an alternate unit is brought on-line; our analysis thus far assumes perfect malfunction detection and reconfiguration

$$R = 1 - (1 - R_m)^n = R_m \frac{1 - (1 - R_m)^n}{1 - (1 - R_m)}$$

Let the probability of correct malfunction detection and successful reconfiguration be c (coverage factor)

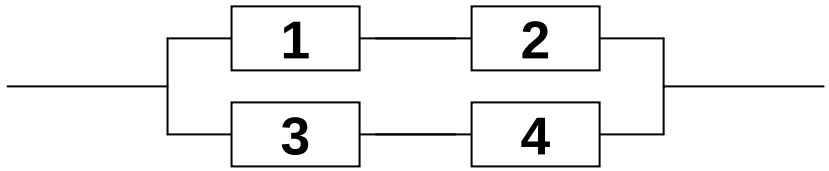
$$R = R_m \frac{1 - c^n(1 - R_m)^n}{1 - c(1 - R_m)}$$

See [Siew92], p. 288

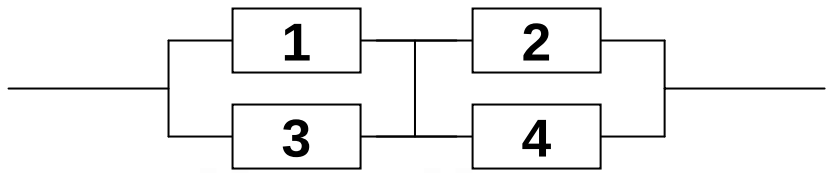
Series-Parallel System

The system functions properly if a string of healthy units connect one side of the diagram to the other

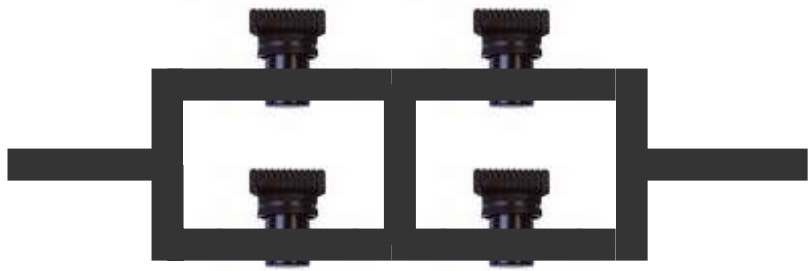
$1 - R = (1 - R_1 R_2)(1 - R_3 R_4)$



Example: Parallel connection of series pairs of valves (tolerates one stuck-on-shut and one stuck-on-open valve)



Example: Series connection of parallel pairs of valves (tolerates one stuck-on-shut and one stuck-on-open valve)

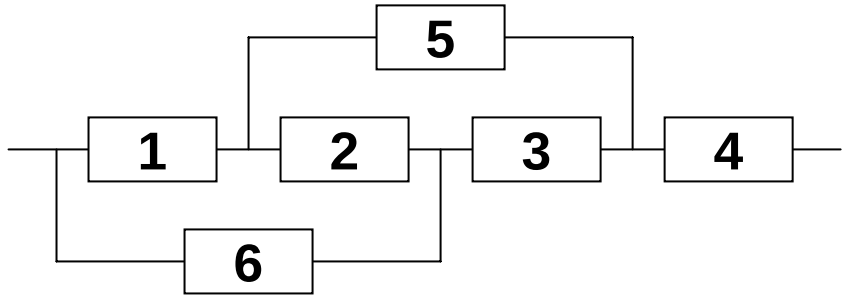


$$R = [1 - (1 - R_1)(1 - R_3)] \times [1 - (1 - R_2)(1 - R_4)]$$

More Complex Series-Parallel Systems

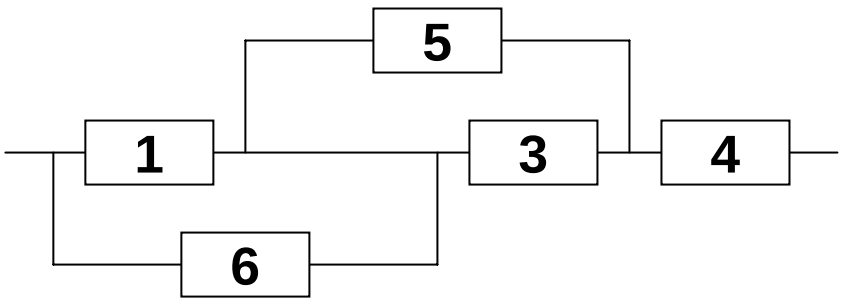
The system functions properly if a string of healthy units connect one side of the diagram to the other

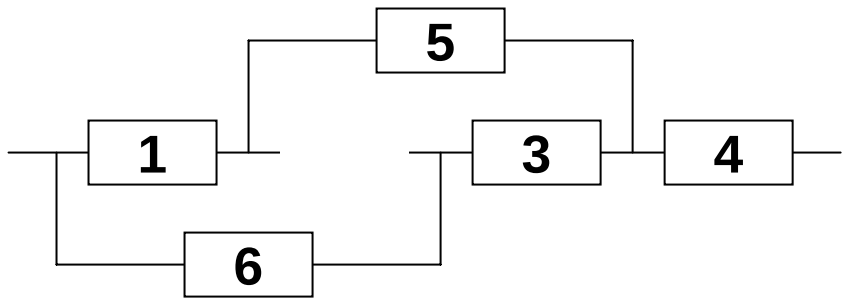
We can think of Unit 5 as being able to replace Units 2 and 3



$R = R_2 \times \text{prob(system OK | Unit 2 OK)}$ ..... R_{2OK}

$\quad + (1 - R_2) \times \text{prob(system OK | Unit 2 not OK)}$ ..... $R_{2\neg OK}$





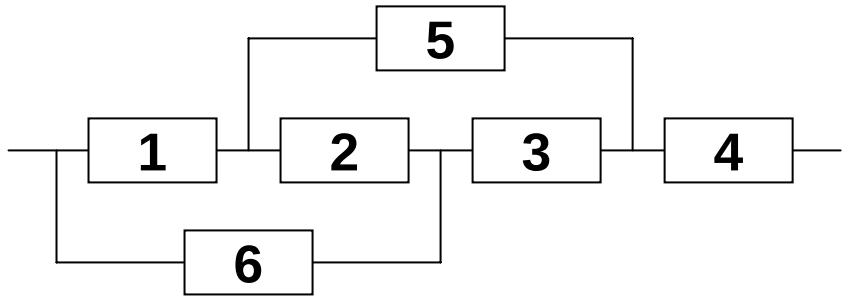
$$R_{2OK} = [1 - (1 - R_1)(1 - R_6)] \times [1 - (1 - R_3)(1 - R_5)] \times R_4$$

$$R_{2\neg OK} = [1 - (1 - R_1 R_5)(1 - R_3 R_6)] \times R_4$$

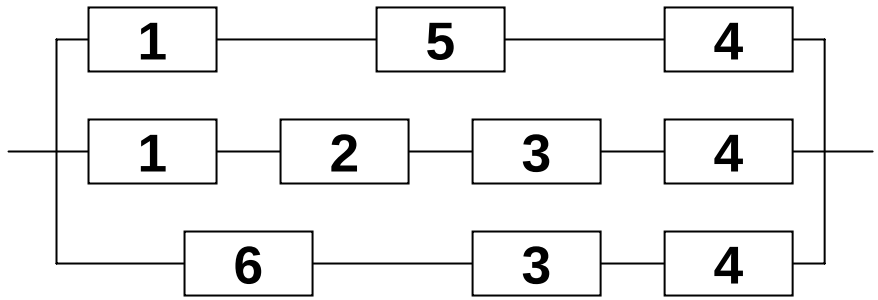
Analysis Using Success Paths

$R \leq 1 - \Pi_i (1 - R_{i\text{th success path}})$

This yields an upper bound on reliability because it assumes the paths to be independent



$R \leq 1 - (1 - R_1R_5R_4)$ [*]
$(1 - R_1R_2R_3R_4)(1 - R_6R_3R_4)$



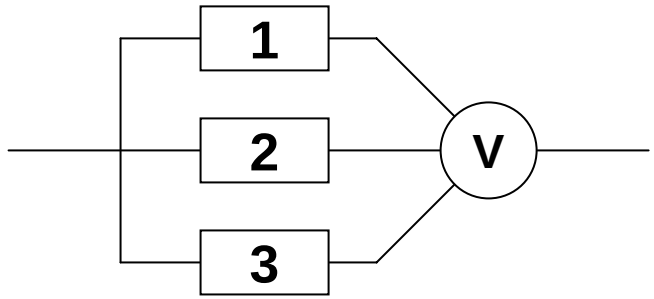
With equal module reliabilities:
$R \leq 1 - (1 - R_m^3)^2 (1 - R_m^4)$

If we expand [*] by multiplying out, removing any power for the various reliabilities, we get an exact reliability expression

$R = 1 - (1 - R_1R_4R_5)(1 - R_3R_4R_6 - R_1R_2R_3R_4 - R_1R_2R_3R_4R_6)$
$= R_3R_4R_6 + R_1R_2R_3R_4 + R_1R_2R_3R_4R_6 + R_1R_4R_5 - R_1R_3R_4R_5R_6$
$-R_1R_2R_3R_4R_5 - R_1R_2R_3R_4R_5R_6$ (Verify for the case of equal R_j)

k-out-of-*n* System

There are *n* modules, any *k* of which are adequate for proper system functioning



Example: System with 2-out-of-3 voting
Assume perfect voter

$$R = R_1 R_2 R_3 + R_1 R_2 (1 - R_3) + R_2 R_3 (1 - R_1) + R_3 R_1 (1 - R_2)$$

With all units having the same reliability R_m and imperfect voter:

$R = (3R_m^2 - 2R_m^3) R_v$ Triple-modular redundancy (TMR)

$R = \sum_{j = k \text{ to } n} \binom{n}{j} R_m^j (1 - R_m)^{n-j}$ *k*-out-of-*n* system in general

Assuming that any 2 malfunctions in TMR lead to failure is pessimistic
With binary outputs, we can model compensating errors
(when two malfunctioning modules produce 0 and 1 outputs)

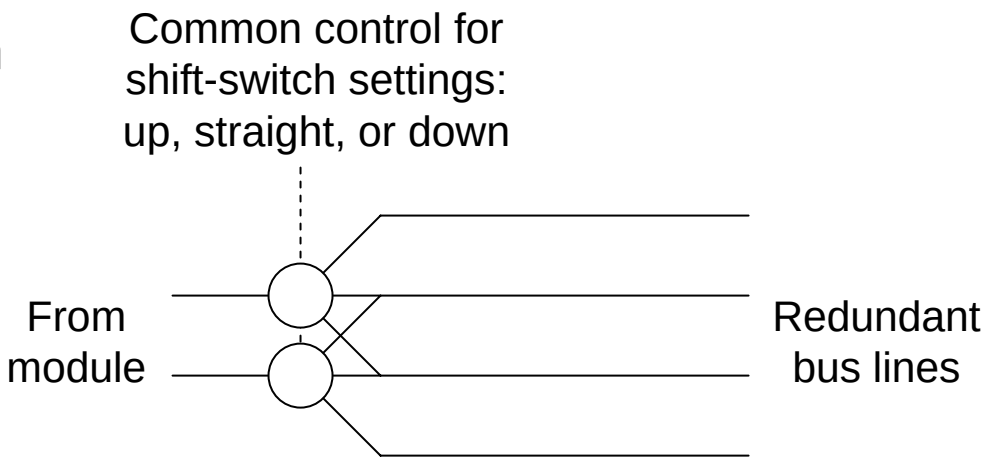
Consecutive *k*-out-of-*n*:G (*k*-out-of-*n*:F) System

Units are ordered and the functioning (failure) of *k* consecutive units leads to proper system function (system failure)

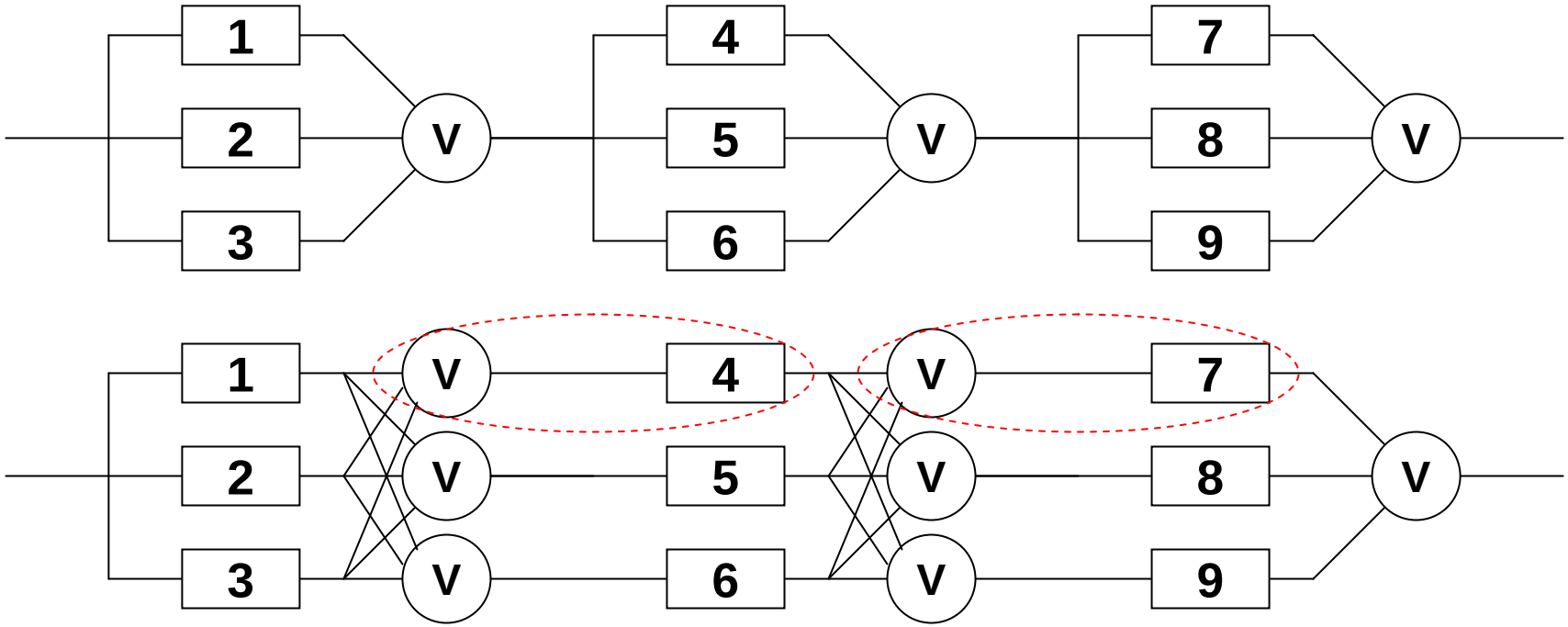
Ordering may be linear (usual case) or circular

Example: System of street lights may be considered a consecutive 2-out-of-*n*:F system

Example: The following redundant bus reconfiguration scheme is a consecutive 2-out-of-4:G system



n-Modular Redundancy with Replicated Voters



Voters (all but the final one in a chain) no longer critical components

Can model as a series system of 2-out-of-3 subsystems