Dependable Systems

3. Chapter

Fault Tolerance – Basics

Prof. Matthias Werner
Operating Systems Group

3.1 Motivation

Intolerance

How to deal with faults? ➡ avoid them

- Concept: **Fault intolerance**
- Eliminates causes of unreliability by
  - Fault avoidance
  - Fault removal (offline)
- No redundancy (online)
- Fault intolerance introduces reliability by:
  - Using very reliable components
  - Refined design techniques
  - Refined manufacturing techniques
  - Shielding
  - Comprehensive testing

Fault tolerance

How to deal with faults? ➡ tolerate them

- **Fault Tolerance**
  - Accepts that a system is not fault-free
  - Fault tolerance by redundancy in space and/or time
  - Automatic handling of faulty states
- **Advantages**:
  - Higher reliability
  - Sometimes lower total cost
  - Confidence of users (psychological assistance)
- **Disadvantages**:
  - Cost of redundancy
  - Sometimes higher complexity

Attention

Fault tolerant systems are **not automatically** more reliable than other systems.

Intolerance vs. Tolerance

- In "Dependable Systems" class, we consider both, fault intolerance and fault tolerance
- Some approaches are suitable for both areas
- **In general:** Never neglect fault intolerance because of use of fault tolerance

Please note

There is no need to tolerate a fault that not occurs.

- Now, we consider basic approaches of fault tolerance

3.2 Redundancy

Types of Redundancy

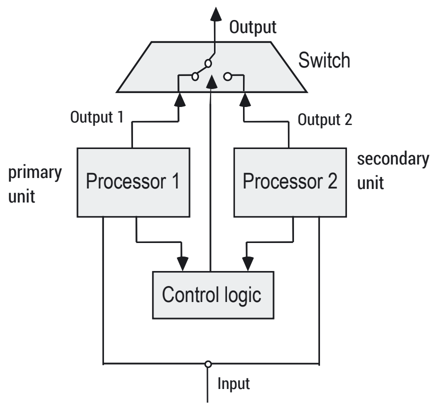
Fault tolerance needs redundancy.

- Redundancy in space
  ➡ Additional hardware, memory, ...
- Redundancy in time
  ➡ Additional time for computation, fault detection and recovery
- (Functional redundancy) ➡ More complex algorithms, monitoring functions, ...
  (can usually be mapped onto the other types)

Using Redundancy: Strategies

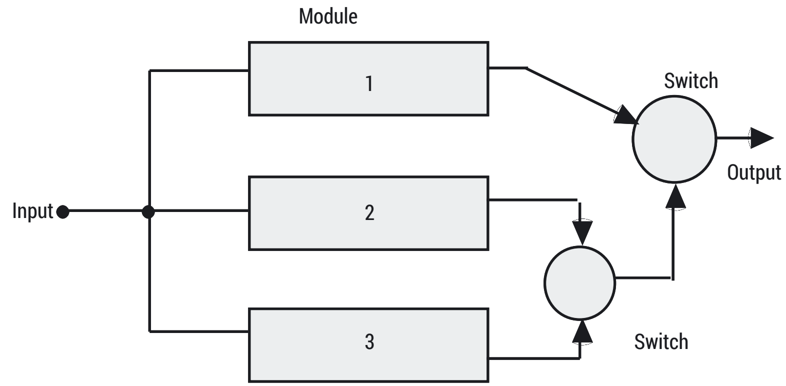
- **Containment.** Limiting impact of failures
- **Diagnosis.** Identification of faulty modules
- **Masking.** Dynamic correction of errors
- **Repair, Reconfiguration, Recovery.** Replacing, removing or avoiding faulty modules and bringing the system into an accectable state

Design Pattern: Duplex System

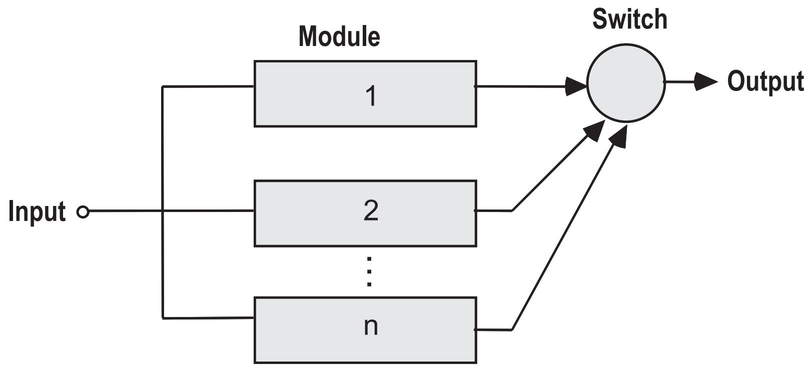


- In case of difference between modules faulty module can be detected with a diagnosis method, e.g.,
  - Using self diagnosis
  - Watchdog timer
  - Diagnosis by external arbiter

Design Pattern: Duplex System + Stand-by Module



Design Pattern: Multiple Stand-by Modules

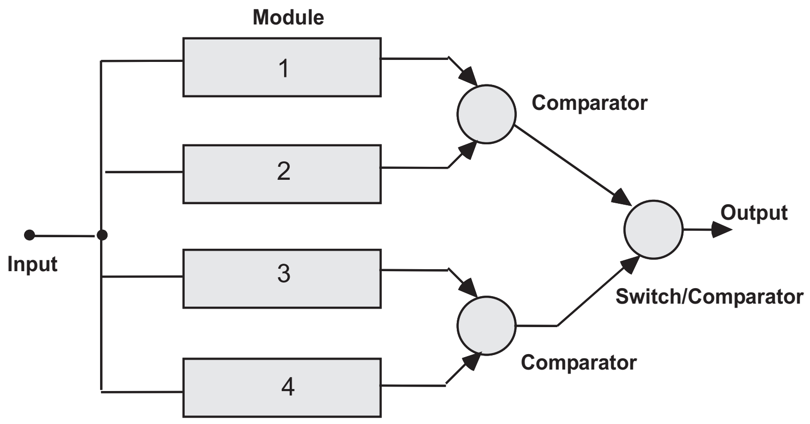


Stand-by Modules

Different kinds of redundancy can be distinguished:

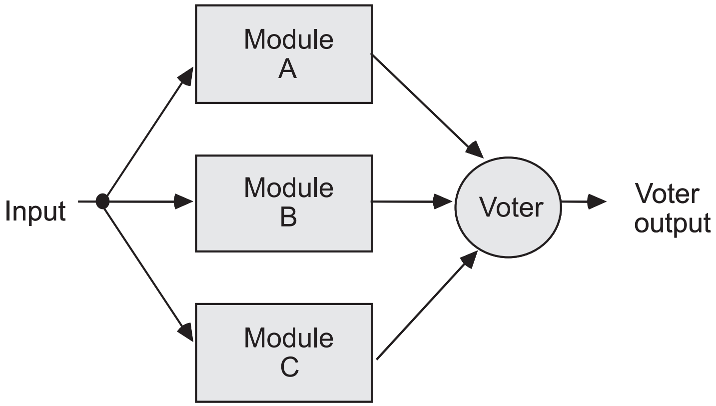
- **Hot Stand-by:** The redundant unit reads and computes all input at the same time as the primary unit.
- **Warm Stand-by:** Inputs are recorded (with delay). In case of take-over redundant unit has to go into appropriate state.
- **Cold Stand-by:** Redundant unit is disabled until take-over takes place.
- No clear distinction between “warm” and “cold”
  - In literature, “warm” is skipped sometimes

Design Pattern: Pair and Spare

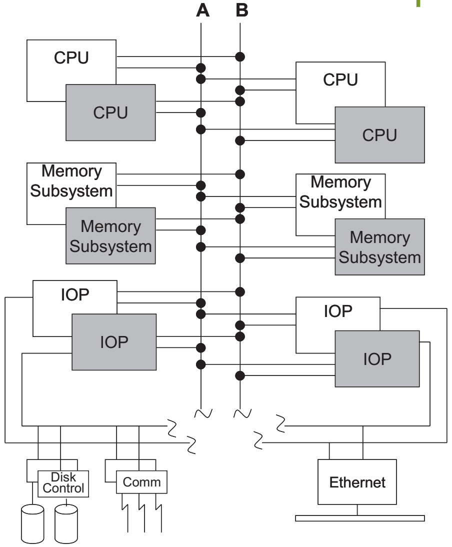


Design Pattern: TMR

- **Triple Modular Redundancy** (TMR)
- Majority (2 out of 3) wins

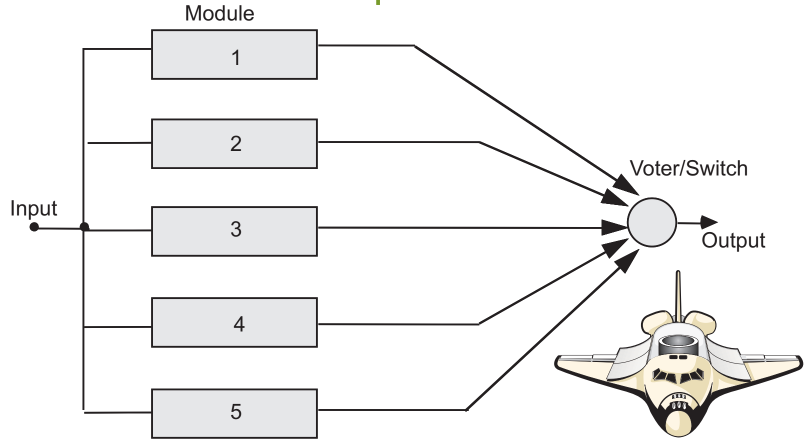


Redundant Hardware: Example I



- Stratus XA/R Series 300
- Prinziple: “Pair and Spare”
- Alternative modules operate parallel using two busses

Redundant Hardware: Example II



- Space Shuttle: Triple modular redundancy (TMR) with two stand-by modules
- Active modules: 1,2 und 3
- Module 4: warm standby, Module 5: cold standby

3.3 Evaluation

Measures

- In order to evaluate the impact of a failure or the success of a countermeasure measures and metrics are necessary
- Different application have different goals ➡ different measures are needed
- Examples:
  - **Phone switch:** Has to work always but single failures (e.g., wrong connection) are acceptable
  - **Space probe on mars:** Has to work for a defined interval of time
  - **Identification friend or foe (IFF):** Detailled consideration of trade-offs between possible “false friend” and “false enemy” failures

Reliability

Definition 3.1 (Reliability)

Reliability $R(t)$ of a system is the probability that the system will perform satisfactorily in a time interval $[0, t]$ given that the system performed successfully at $t_0 = 0$.

- With density of failure distribution $f(t)$:

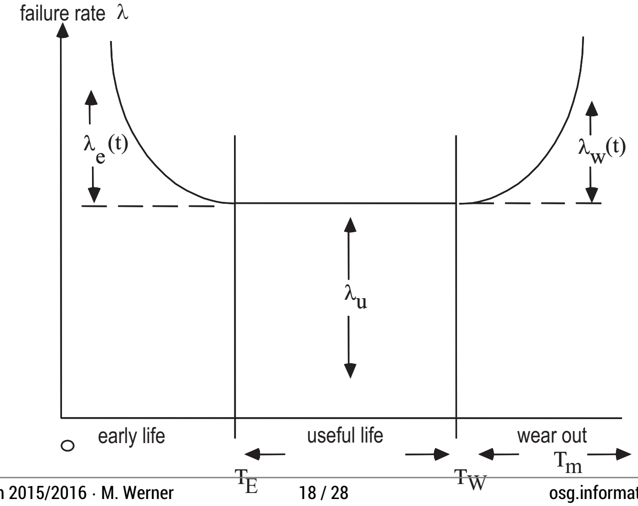
$$R(t) = \int_t^\infty f(\tau) d\tau$$

Reliability (cont.)

- Different distributions and therefore functions to describe reliability are possible
- **Hardware**
  - Exponential failure distribution: $R(t) = e^{-\lambda \cdot t}$,
    λ: failure rate
  - Weibull-distribution for failure: $R(t) = e^{-\lambda \cdot t^{\beta}}$,
    β: form parameter
- **Software**: No simple and sound models

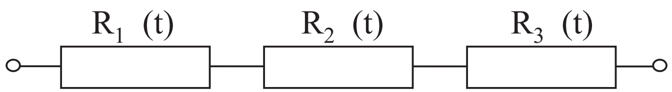
Bathtube Function

Failure rate is often assumed to be **constant** but may also change over time.
For hardware: Bathtube function



Series System

Failure of one module implies total failure



$$R_s(t) = R_1(t) \cdot R_2(t) \cdot R_3(t) = e^{-(\lambda_1+\lambda_2+\lambda_3)}$$

In general for n serial modules

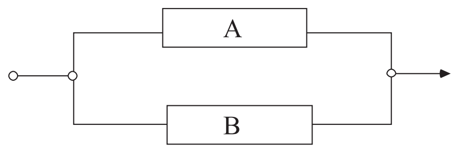
$$R_s(t) = \prod_{i=1}^{n} R_i(t) = e^{-\lambda_s t}, \lambda_s = \sum_{i=1}^{n} \lambda_i$$

λ_s: System failure rate
λ_i: Module failure rates
For n identical modules:

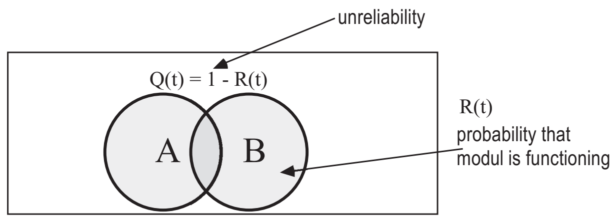
$$R_s(t) = (R(t))^n$$

Parallel System

Each module operates independently.
System operates if at least one module operates.



$$R_p(t) = R_A(t) + R_B(t) - R_{AB}(t) = R_A(t) + R_B(t) - R_A(t)R_B(t)$$



Venn Diagram

Parallel System (cont.)

In general for n modules

$$R(t) = 1 - (1 - R_1(t)) \cdot (1 - R_2(t)) \cdots (1 - R_n(t)))$$

- **For n identical modules**

$$R_p(t) = 1 - (1 - R_m(t))^n$$

- $n = 2 \Rightarrow R_A = R_B = 0.5, R_P = 1 - (0.5)^2 = 0.75$

Mean Times

Goal: Simpler Measure – mean times instead of rates

- **MTTF** (**Mean Time To Failure**)
  Expected value of failure distribution:
  - $MTTF = \int\limits_0^\infty t \cdot f(t)dt$
  - for exponential distribution: $MTTF = \frac{1}{\lambda}$
- **MTTR** (**Mean Time To Repair**)
  $MTTR = \frac{1}{\mu}$ for exponentiell distribution, μ: repair rate
- **MTBF** (**Mean Time Between Failure**)
  $$MTBF = MTTF + MTTR$$

Availability

Definition 3.2 (Availability)

Availability $A(t)$ of a system is the probability that the system is operational (delivers satisfactory service) at a given time t

Definition 3.3 (Steady-state Availability)

Steady-state Availability A_s of a system is the fraction of lifetime that the system is operational

- $A_s = \frac{\text{Uptime}}{\text{total Time}} = \frac{\text{MTTF}}{\text{MTTF + MTTR}} = \frac{\mu}{\mu+\lambda}$

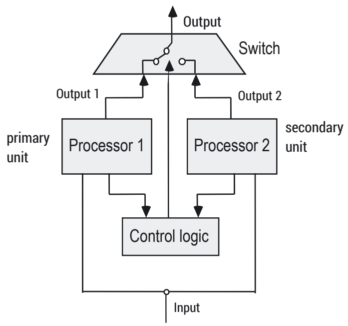
  λ: failure rate, μ: repair rate

Further Measures

Beside the measures given so far other measures exist that are mostly used for special cases.

- **Responsiveness** $\mathcal{R}(t, D)$ is the probability that a service that was started at time t correctly operates in the interval $[t, t + D]$
- **Mission time** $MT(r)$ gives the time at which system's reliability falls below a given level r
- **Maintainability** $M(t)$ is the probability that a system returns to function within t time units
- ...

Example: Reliability of a Duplex-System

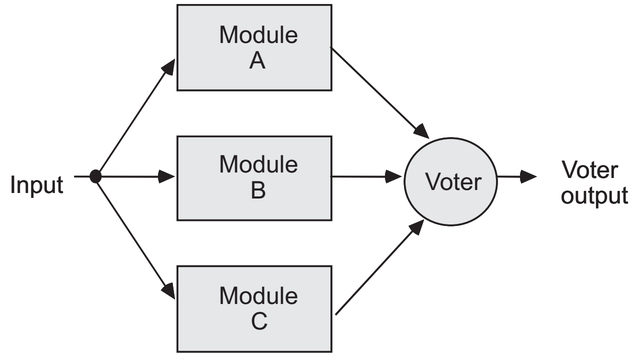


Reliability

$R = (R_m^2 + 2CR_m(1 - R_m))R_k$

- R_m : Reliability of a module
- R_k : Reliability of control, comparator, ...
- C: Coverage factor, represents combined probability of successful fault detection and reconfiguration

Example: Reliability of TMR



Voter gives a correct result if it is working correctly and at least two of three modules are working correctly.

Example: Reliability of TMR (cont.)

- **Reliability:**
  - $R_{TMR} = R_{Voter} \cdot R_{2-out-of-3}$

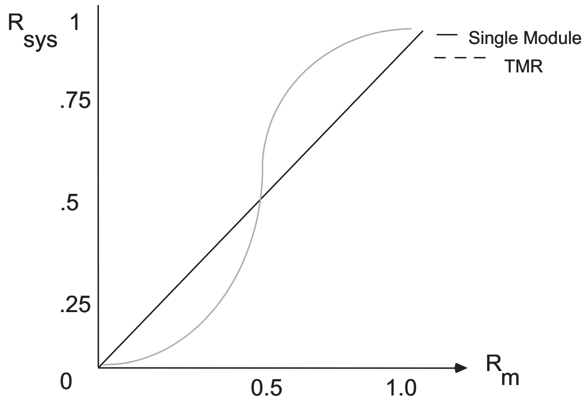
$$R_{TMR} = R_V \left(R_m^3 + 3R_m^2(1 - R_m)\right)$$

  - R_V: Reliability of voter
  - R_m: Reliability of each module
- **Under which conditions TMR gives a benefit?**
  - TMR is "better" than a single module if $R_{TMR} > R_m$

    that means: $R_V \left(R_m^3 + 3R_m^2(1 - R_m)\right) > R_m$

Example: Reliability of TMR (cont.)

- Assumption: Voter is perfect ($R_V = 1$)



- TMR is only better if $R_m > 0.5$
- Voter has to be very reliable ($R_m > 0.9$ or even $R_m > 0.99$)