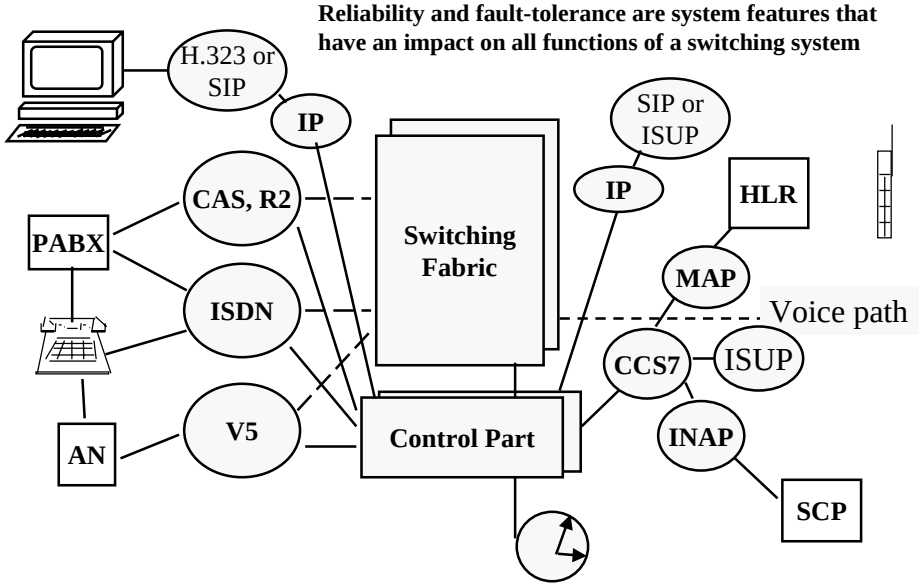
Fault tolerance and Reliability

Reliability measures
Fault tolerance in a switching system
Modeling of fault tolerance and reliability

Summary of course scope

Reliability and fault-tolerance are system features that have an impact on all functions of a switching system



Definition of basic terms

- ✔ ***Failure, malfunction*** **- is a deviation from the intended/specified behavior of a system**
- ✔ ***Fault*** **- such a state of a program or a device that can lead to a failure**
- ✔ ***Error*** **- a wrong response produced by a program or module. An error is an indication (evidence) that the module may be faulty, that the module has received wrong input or that it has been misused. Can lead to failure if the system is not tolerant to this type of an error. A Fault can exist without any errors occuring.**

What is fault tolerance?

- ✔ ***FT is the capability of a system to continue its intended function in spite of having a fault or faults.***
- ✔ ***A switching system is an example of a fault tolerant system.***
- ✔ ***Fault tolerance always requires redundancy of some sort.***

Faults can be categorized

- ✔ **Based on their duration:**
  - ***permanent or stuck-at*** **(stuck at zero, stuck at one)**
  - ***intermittent*** **- the fault requires repair action but its impact is not observable always**
  - ***transient,*** **can be observed for a short period of time but then disappears without repair.**
- ✔ **Observable and latent.**
- ✔ **Based on the scope of the impact (how serious the fault is).**
- ✔ **In general we talk about fault models.**
  - **Many types of models are used depending on the purpose.**

Graceful degradation is

- ✔ ***The capability of the system to continue to function under one or more faults but on a reduced level of performance.***
- ✔ **For example in some RAID (redundant arrays of inexpensive disks) configurations write speed drobs in case of a disk fault but can still continue on a lower level of performance even while the fault has not yet been repaired.**

Reliability and availability performance

- ✔ ***Reliability, R(t)*** **= Probability that the system does not fail within time *t* under the condition that it was functioning correctly at *t* = 0.**
  - **For all known man-made systems always approaches zero when time approaches infinity**
- ✔ ***Availability, A(t)*** **= Probability that the system will function correctly at time *t*.**
  - **For a system that can be repaired will approach some stable value asymptotically during the useful lifetime of the system.**

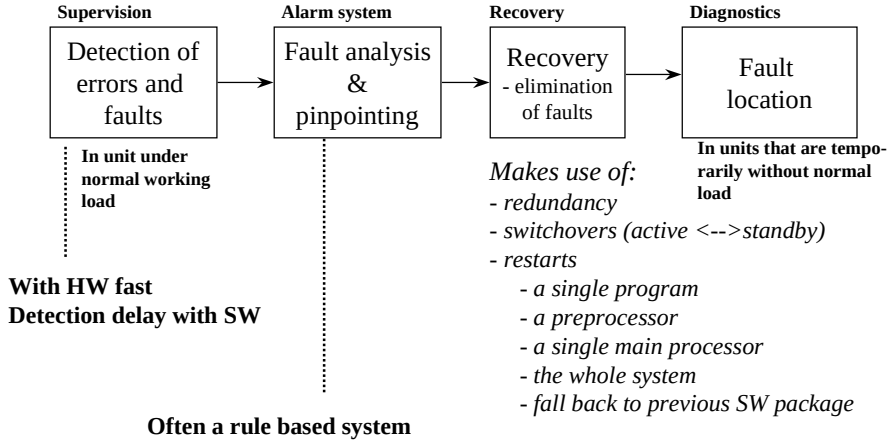
For a repairable system we define

- ✔ ***Maintainability, M(t)*** **= probability that the system is returned to the correct functioning state during time *t* under the condition that it was faulty at time *t* = 0.**

MTTF, MTTR, MTBF

- ✔ **MTTF = Mean-Time-To-Failure = expected value of the time duration from "now" to the next failure**
- ✔ **MTTR = Mean-Time-To-Repair = expected value of the time duration from a fault until the system has been restored into a correct functioning state**
- ✔ **MTBF = Mean-Time-Between-Failure = expected value of the time duration from an occurrence of a fault until the next occurrence of a fault**
  - **MTBF = MTTF + MTTR.**

A high availability performance is targeted by maintenance software in the exchange

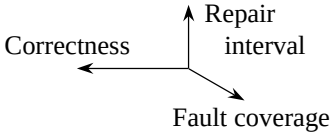


This is one of the most important software subsystems in an exchange in parallel with call control and charging

Main types of Redundancy are

- *Hardware redundancy*, e.g. duplication (1+1), r/n -principle duplication requires in principle *self-checking* - recovery blocks blocks that are able to detect their own faults.
- *Software redundancy* (always required).
- *Information redundancy* (e.g. parity bit).
- *Time redundancy* (e.g. delayed re-execution of transactions).

Fault tolerance goals and priorities vary in different application areas



- Common goal to all areas is increasing availability.
- The importance of other goals depend on application:
  - *Correctness* (computational integrity) - important e.g. for charging.
  - *Repair interval* (long life systems) - e.g. space based systems
  - *Fault coverage* or efficiency of recovery - how well the fault tolerance mechanisms perform.
  - *In addition, performance and price are important limiting factors.*

- Life-critical, mission critical, long life systems. Commercial systems.
- It is important to understand the goals and set the right priorities! Implementations vary significantly depending on the priorities set on the goals.

Most important reliability requirements for an exchange are

Q.543: Premature release of a call:
Probability of premature release of a call during any *1 min* interval due to failure of an exchange

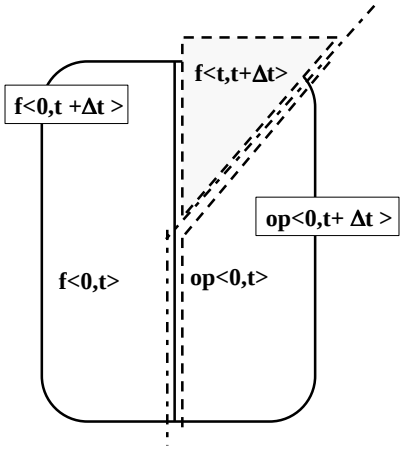
$$P \leq 2 \times 10^{-5}$$

Average intrinsic Down-time: 2 min/a.

If a system restart takes e.g. 20 min

$\Rightarrow$ *MTTF (time between restarts) > 10 years.*

R, F, Λ, MTTF

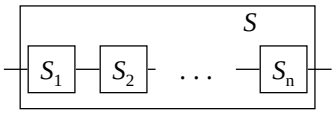


Event space:

f<0,t> - the system fail in the interval [0,t)
op<0,t> - the system functions in the interval [0,t)

You should learn and be able to show the relationships between the variables
R - reliability,
F - probability of failure,
Λ – failure intensity,
MTTF - mean time to failure

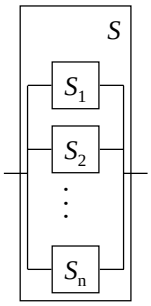
Combinatorial reliability



A serial system *S* functions, if and only if all its parts S_i will function

$$R_s = \prod_{i=1}^{n} R_i$$

Failures in subsystems are independent



A parallel (a replicated) system fails, if all its subsystems fail:

$$F_s = \prod_{i=1}^{n} (1 - R_i)$$

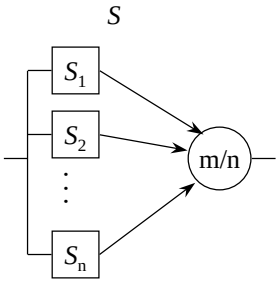
$$R_s = 1 - F_s = 1 - \prod_{i=1}^{n} (1 - R_i)$$

E.g. a duplicated system:

$$R_s = 1 - (1 - R)^2$$

More reliability combinatorics

A load sharing system will function, if *m* of the total of *n* subsystems will function



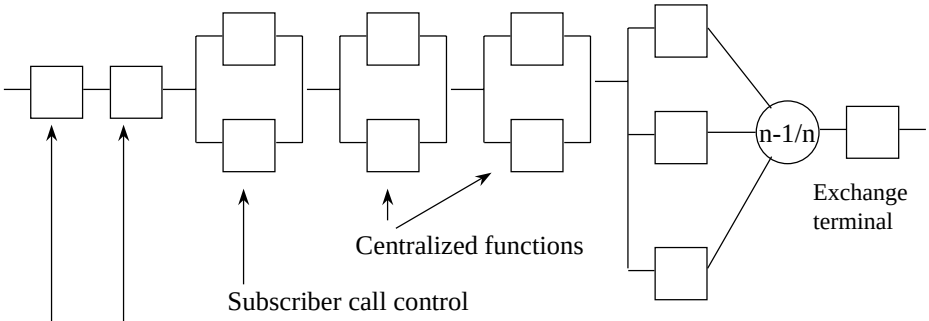
For Bernoull trials binomial distribution applies

$$R_{m/n} = \sum_{k=m}^{n} C_n^k \, R^k (1 - R)^{n-k}$$

Esim. $R_{2/3} = \sum_{k=2}^{3} \frac{3!}{k!(3-k)!} R^k (1 - R)^{3-k} = -2R^3 + 3R^2$

For example: $R = 0.9 \Rightarrow R_{2/3} = 0.972$

Exchange reliability from the subscriber point of view



CCS7 signaling processor - to be more exact: under condition that n-1/n operational processors at call setup, the chosen processor will function during the call

$R > 1 - 2 \times 10^{-5}$ *Premature release requirement applies*

Unit of failure intensity λ is

$[\lambda]$ = fit = number of faults/10^9h

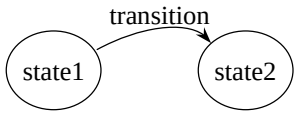
Failure intensities of replaceable plug-in units are 0.1 - 10 kfit.

Example: The failure intensity of the line card in an exchange is 2 kfit. What is MTTF?

$$MTTF = 1/\lambda = \frac{10^9 h}{2000} = \frac{1000\,000}{2 \times 24 \times 360} = 580 \text{ years.}$$

Reliability modeling based on Markov chains

The system is modeled as a set of states and a set of transitions. Each state corresponds to the fulfillment of a set of conditions and each transition corresponds to an event of the systems changing from one state to another.

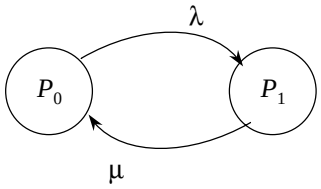


Using this method it is possible to find the reliability behavior of a complex system with many states and non-independent failure modes.

Goal of this subject: learn to formulate Markov reliability models and solve them.

Markov chain modeling

(discrete event, continuous time)



λ = failure intensity
μ = repair intensity
(repair time is exponentially distributed)

P_i = probability of state *i*

e.g. $P_0 = R(t)$, $P_1 = F(t)$

- Modeling leads to groups of linear differential equations.
- For a given modeling goal it is essential to choose a minimal set of states for the equations to be easily solvable.
- By setting the derivatives of the probabilities to zero we get an asymptotic, state if one exists.