Fault-Tolerant Design of Digital Systems

EGE 534

Introduction:

What is fault and fault-tolerant computing?

Dr. Baback Izadi

Department of Electrical and Computer Engineering and

State University of New York – New Paltz

bai@engr.newpaltz.edu

Class Information

- **Class Times:**
  - TF 9:25 AM – 10:40 AM REH 111
- **Instructor**
  - Dr. Baback Izadi
- **Office Hours**
  - Tuesday 11:00 AM- 12:00 PM; 1:30 PM – 2:30 PM
  - Wednesday 10:00 AM - 12:00 PM
  - Friday 11:00 AM- 12:00 PM; 1:30 PM – 2:30 PM
  - And, by appointment
- **Web site:**
  - http//www.engr.newpaltz.edu/~bai/

Outline

- Grading policy
- Overview and course objectives
- Recommended reading
- Why reliable computing?
- Fault – Tolerant computing
- Faults and its manifestation
- Hardware and software fault model
- Sources of failure

Grading Policy

Homework	20 %	
Research Presentation	10 %	
Midterm Exam	35 %	March 7
Final	35 %	May 16

- Attendance
  - Attendance may be taken during the first 10 minutes
  - Three missing classes is allowed.
  - 4th absence -2%
  - 5th absence -5%

Recommended Reading

- *Design and Analysis of Fault-Tolerant Digital Systems*, B. W. Johnson: Addison-Wesley, 1989.
- *Fault-Tolerant Computer System Design*, D. Pradhan, Prentice-Hall, 1996.
- *Reliable Computer Systems-Design and Evaluation, 2nd edition*, D. Siewiorek and R. Swarz: Digital Press - Butterworth, 1992.
- *Fault Tolerance in Distributed Systems*, P. Jalote: Prentice Hall, 1994
- *Performance and Reliability Analysis of Computer Systems*, R. Sahner, K. Trivedi: Kluwer Academic, 1996
- *Fault Tolerance through reconfiguration of VLSI and WSI arrays*, R. Negrini: MIT Press, 1989.

Course Overview

- Introduction: What is fault and fault-tolerant computing?
- Hardware Redundancy – Basic Approaches & Models
- Information Redundancy
- Evaluation Techniques
- MIDTERM EXAM   March 7, 2014 (Tentative)
- Testing
- Check Pointing & Recovery
- Software Fault Tolerance
- Fault Tolerant Architecture
- Trends in Fault Tolerant Architecture
- Student Presentations
- FINAL EXAM,   May 16, 2014   10:15 AM - 12:15 PM

Why Study Reliable Computing!!!

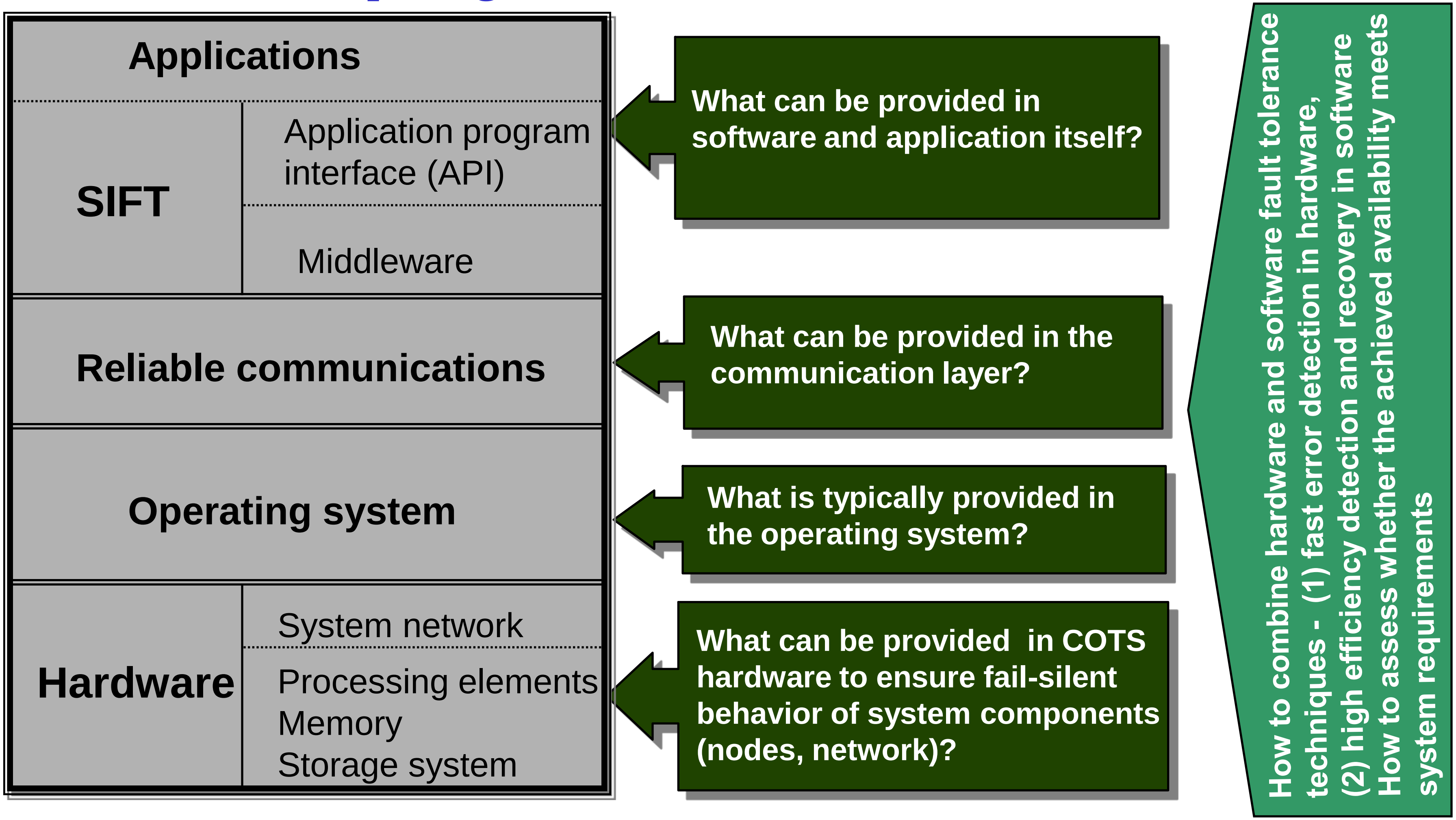
- **Traditional needs**
  - Long-life applications (e.g., unmanned and manned space missions)
  - Life-critical, short-term applications (e.g., aircraft engine control, fly-by-wire)
  - Defense applications (e.g., aircraft, guidance & control)
  - Nuclear industry
  - Telecommunications Switching systems (1 ESS – 5 ESS)
- **Mission-critical applications**
  - Health Care industry
  - Automotive industry
  - Industrial control systems, production lines
  - Banking, reservations, commerce

Why Study Reliable Computing!!! (cont.)

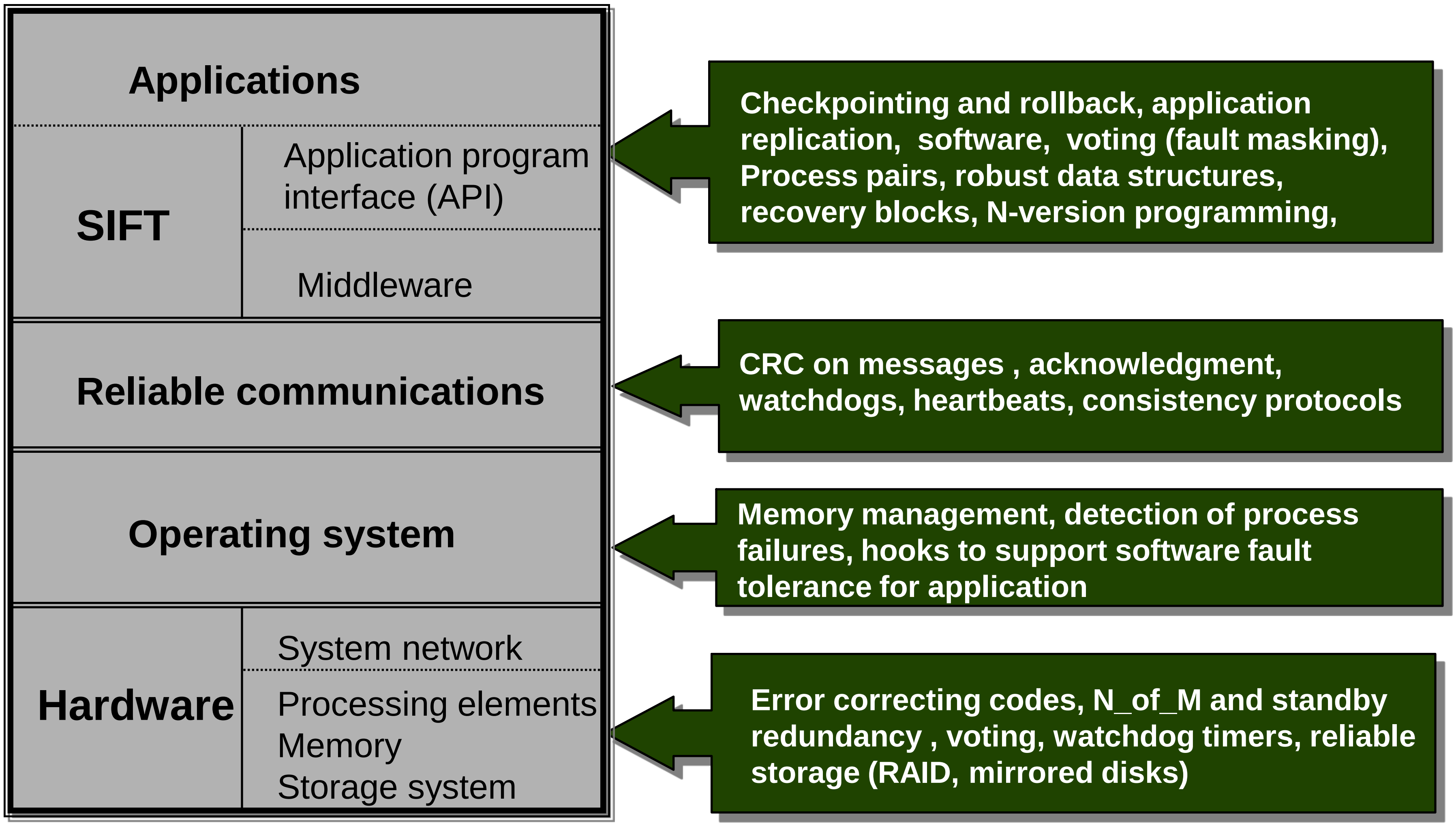
- **Networks**
  - Wired and wireless networked applications
  - Data mining
  - Distributed, networked systems (reliability and security are the major concerns)
  - Commerce: stores, catalog industry
- **Scientific computing, education**
  - Typically reliability has not an issue till recently.
  - IBM Petaflop Blue gene computers reliability a major concern.

Objectives

- **System** (hardware, software) **perspective/view** on design issues in reliable computing



How do We Achieve the Objectives?

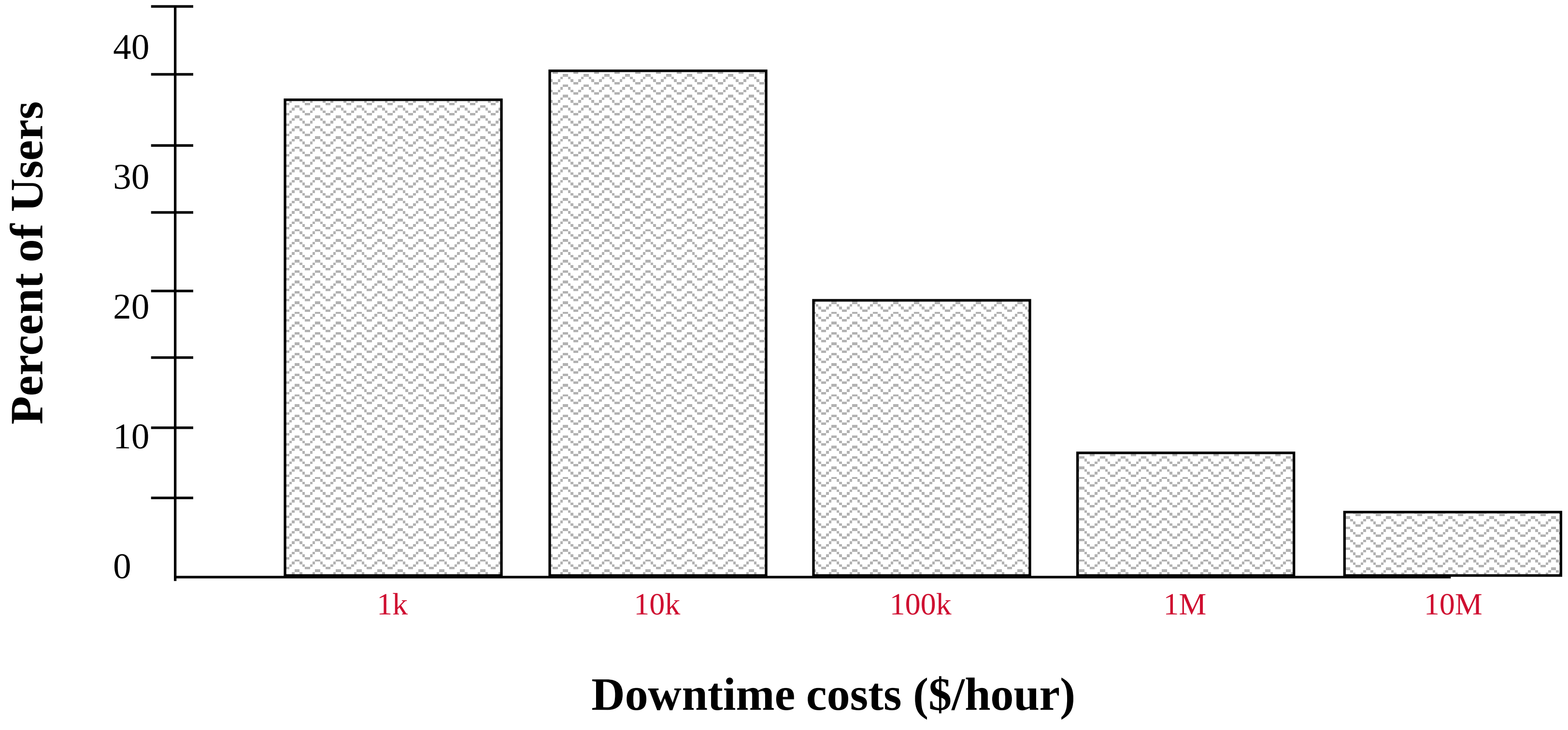


Examples of Computer-related Failures

Date	Event	FAULTS: Physical	FAULTS: Design	FAULTS: Interaction	FAILURES: Localized	FAILURES: Distributed	Availability / Reliability	Safety	Confidentiality
June 1980	False alerts at the North American Air Defense (NORAD) [Ford 85]	√			√		√		
April 1981	First launch of the Space Shuttle postponed [Gaman 81]		√		√		√		
June 1985 - January 1987	Excessive radiotherapy doses (Therac-25) [Leveson & Turner 93]		√		√			√	
August 1986 - 1987	The “wily hacker” penetrates several tens of sensitive computing facilities [Stoll 88]		√	√	√				√
November 1988	Internet worm [Spatford 89]		√	√		√	√		
15 January 1990	9 hours outage of the long-distance phone in the USA [Neumann 95]		√			√	√		
February 1991	Scud missed by a Patriot (Dhahran, Gulf War) [Neumann 95]		√	√	√		√	√	
November 1992	Crash of the communication system of the London ambulance service [HA 93]		√	√		√	√	√	
26 and 27 June 1993	Authorization denial of credit card operations in France	√	√			√	√		
4 June 1996	The maiden flight of the Arine 5 launcher ended in a failure (France)		√		√		√	√	

Effect of major network outages on business

Large Insurance Carriers	**$20k/hour**
Major Airlines	**$2.5M/hour**
Trading / Investment Banking	**$6M/hour**



Fault-tolerant Computing

- Fault-tolerant system
  - One that can continue to correctly perform it specific tasks in the presence of hardware failure or software errors
- Fault-tolerance: the attribute that enables a system to achieve fault-tolerant operation
- Fault-tolerant computing: the process of performing calculation in a fault-tolerant manner

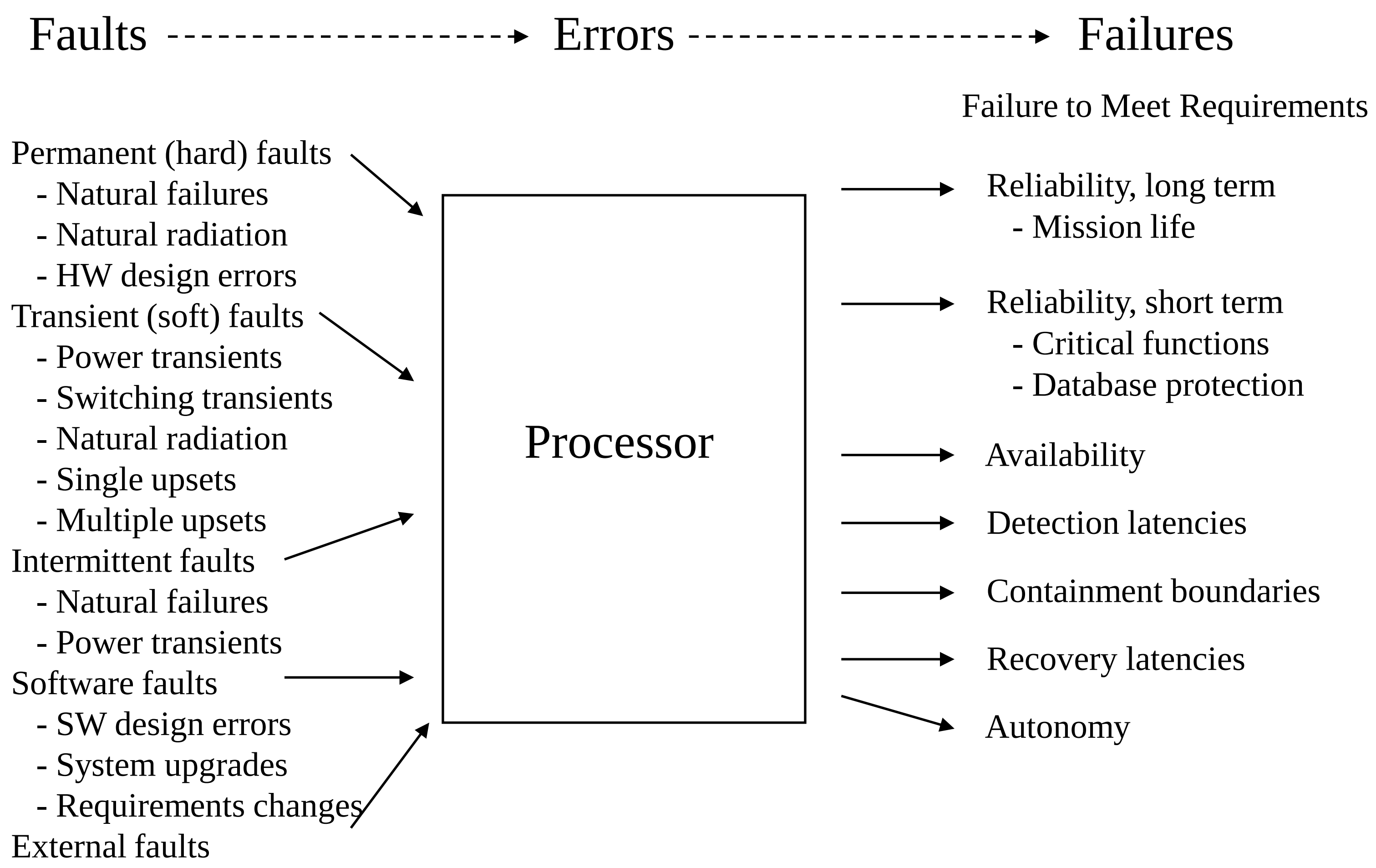
Faults, Errors, and Failures

- **Fault** is the physical defect, imperfection, or flaw that occurs within some hardware or software component
- **Error** is the manifestation of a fault. It is a deviation from accuracy or correctness
- **Failure** is an incorrect performance of one of the functions of the system.

Faults ----------> Errors ----------> Failures

Physical Universe | Information Universe | User Universe

Faults, Errors, and Failures in Computing Systems



Fault Classes

Based on the temporal persistence

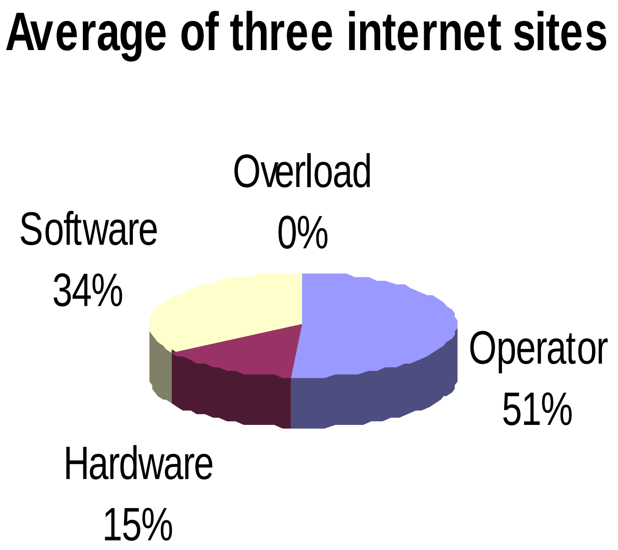
- **Permanent** faults, whose presence is continuous and stable.
- **Intermittent** faults, whose presence is only occasional due to unstable hardware or varying hardware and software states (e.g., as a function of load or activity).
- **Transient** faults, resulting from temporary environmental conditions.

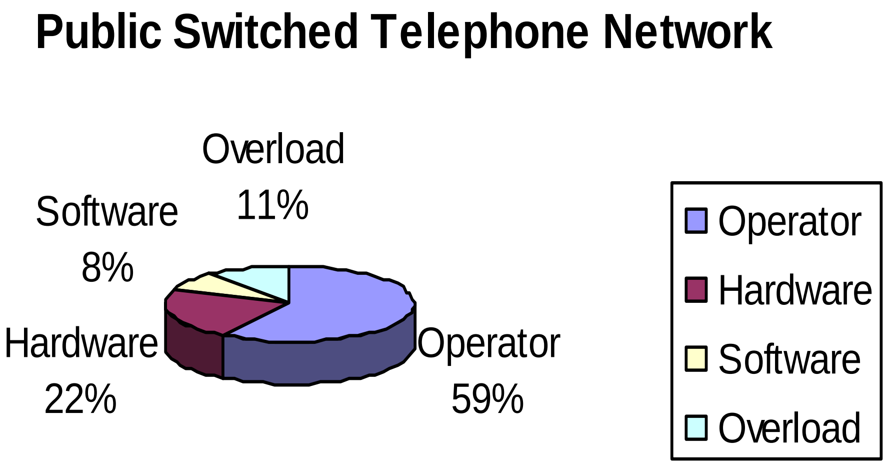
Based on the origin

- **Physical** faults, stemming from physical phenomena internal to the system, such as threshold change, shorts, opens, etc., or from external changes, such as environmental, electromagnetic, vibration, etc.
- **Human-made** faults, which may be either design faults, introduced during system design, modification, or establishment of operating procedures, or interaction faults, which are violation of operating or maintenance procedures.

Faults Due to Human

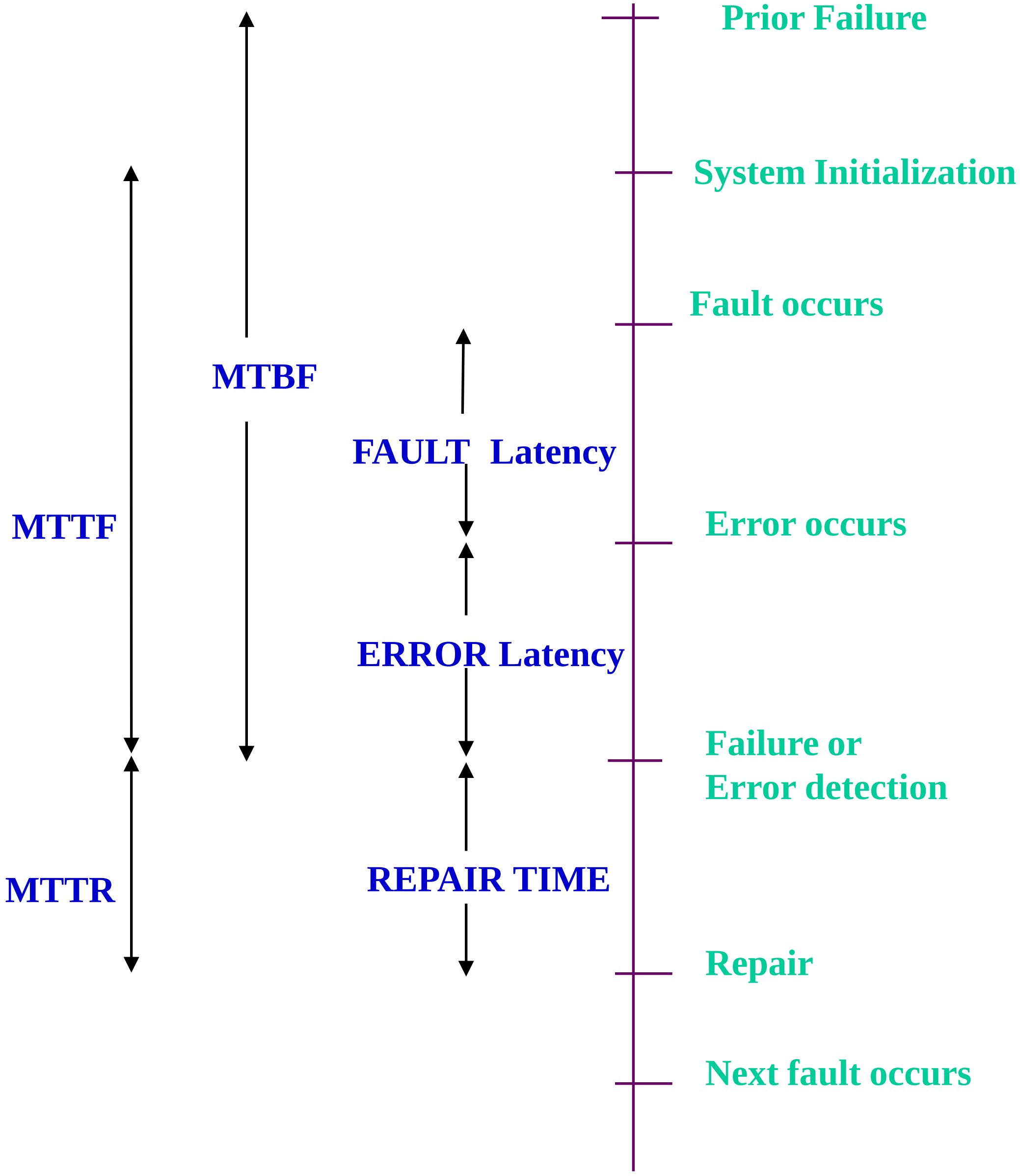
Human operators are both a major cause of failures and a major agent of recovery for *non-transient* failures





* Oppenheimer, Ganapathy et al, 'Why internet services fail and what can be done about it

Fault Cycle & Dependability Measures



Reliability:
a measure of the continuous delivery of service;
R(t) is the probability that the system survives (does not fail) throughout [0, t];
expected value: ***Q(t) = 1 – R(t)***
MTTF(Mean Time To Failure)

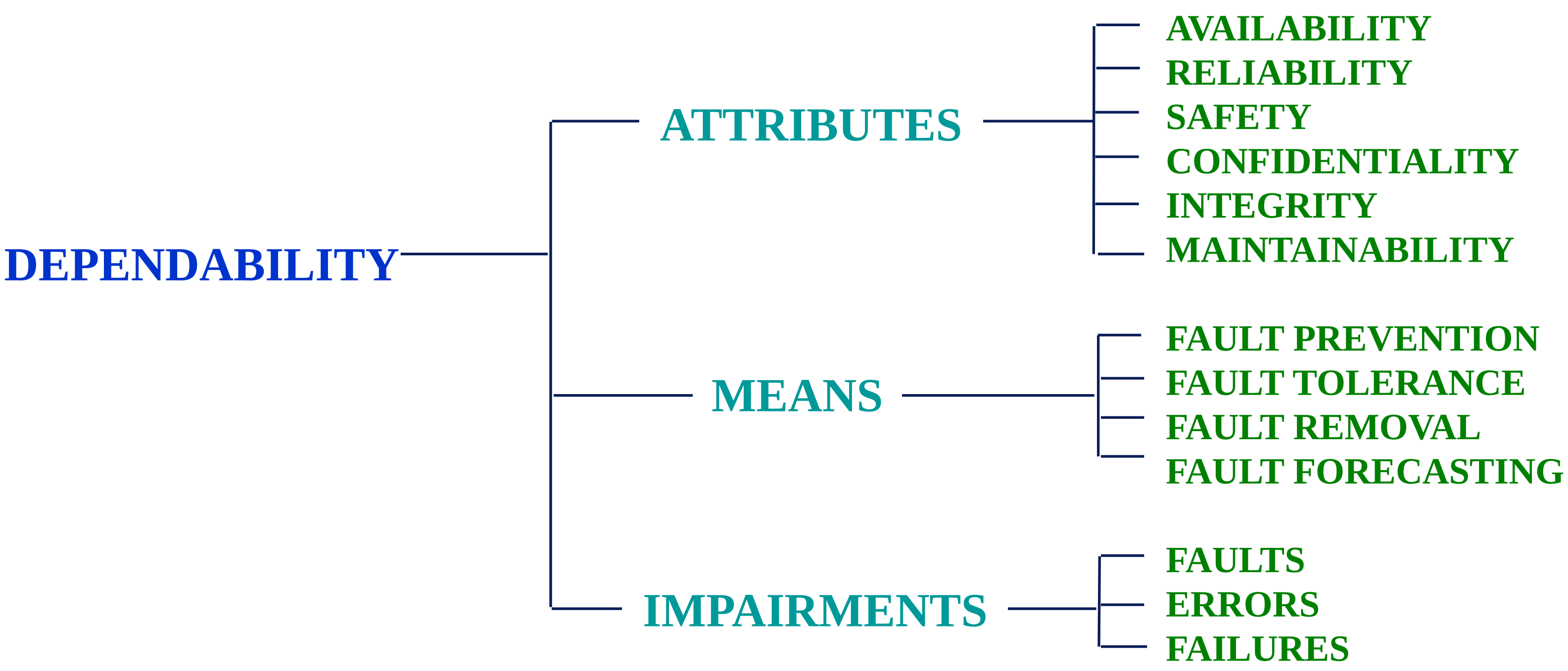
Availability:
a measure of the service delivery with respect to the alternation of the delivery and interruptions
A(t) is the probability that the system delivers a proper (conforming to specification)service at a given time t.
expected value: ***EA = MTTF / (MTTF + MTTR)***

Maintainability:
a measure of the service interruption
M(t) is the probability that the system will be repaired within a time less than t;
expected value: ***MTTR (Mean Time To Repair)***

Safety:
a measure of the time to catastrophic failure
S(t) is the probability that no catastrophic failures occur during [0, t];
expected value:
MTTCF(Mean Time To Catastrophic Failure)

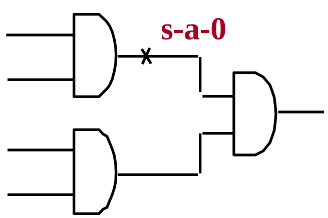
Dependable Computing

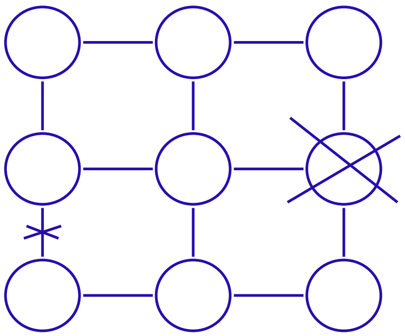
- Dependability is property of computer system that allows reliance to be placed justifiably on service it delivers. The service delivered by a system is its behavior as it is perceptible by its user



Hardware Fault Models

Stack-at	Module level	Functional level	System level
Example: physical failures in circuits Lines in a gate level stuck at 0 or 1 Faulty contact Transistor stuck open or closed Metal lines open Shorts between adjacent metal lines	**Example:** decoder No output lines activated An incorrect line activated instead of desired line An incorrect line activated in addition to desired line	**Example:** Memories One or more cells are stuck at 0 or 1 One or more cells fail to undergo 0-1 or 1-0 transition Two or more cells are coupled A 1-0 transition in one cell changes contents in another cell More than one cell is accessed during READ or WRITE A wrong cell is accessed during READ or WRITE	**Example:** a parallel processor topology View machine as a graph - nodes correspond to processors - edges correspond to links Fault Model: A processor (node) or link (edge) faulty





Software Fault Models

IBM OS

Allocation management : Memory region used after deallocation

Copying overrun: Program copies data past end of a buffer

Pointer management: Variable containing data address corrupted

Wrong algorithm: Program executes but uses wrong algorithm

Uninitialized variable: Variable used before initialization

Undefined state: System goes into unanticipated state

Data error: Program produces or reads wrong data

Statement logic: Statements executed in wrong order or omitted

Interface error: A module's interface incorrectly defined or incorrectly used

Memory leak: Program does not deallocate memory it has allocated

Synchronization: Error in locking or synchronization code

GUARDIAN 90

Incorrect computation: Arithmetic overflow or an incorrect arithmetic function

Data fault: Incorrect constant or variable

Data definition fault: Fault in declaring data or data structure

Missing operation: Omission of a few lines of source code

Side effect of code update: Not all dependencies between software modules considered when updating software

Unexpected situation: Not providing routines to handle rare but legitimate operational scenarios

Software Fault Models (Myrinet Network Switch)

Message dropped	A message was dropped.
Data corrupted	A message with incorrect data was sent.
Restart	The Myrinet Control Program restarted itself.
Interface hung	The interface (on local or remote node) was not able to operate properly.
Computer crash	The system (local or remote node) crashed.

Failure Sources and Frequencies

Non-Fault-Tolerant Systems

- Japan, 1383 organizations (Watanabe 1986, Siewiorek & Swarz 1992)
- USA, 450 companies (FIND/SVP 1993)

Mean time to failure: 6 to 12 weeks

Average outage duration after failure:

1 to 4 hours

Fault-Tolerant Systems

–Tandem Non-Stop Computers (Gray 1990, HP now)

– Mean time to failure: 21 years (Tandem)

Failure Sources: