Introduction to Dependability

slides made with the collaboration of: Laprie, Kanoon, Romano

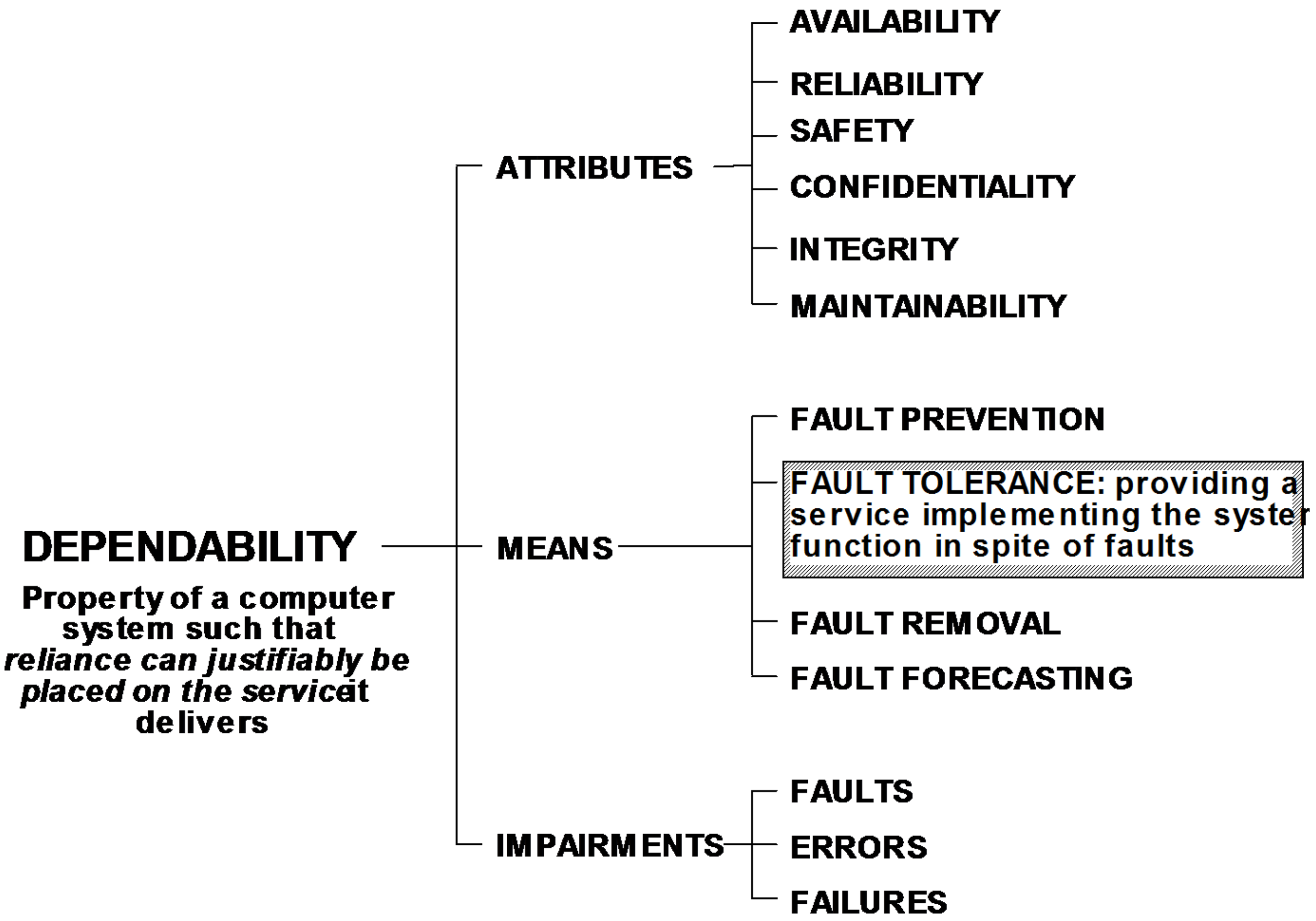
Overview

Dependability: "*[..] the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers [..]*"

IFIP 10.4 Working Group on Dependable Computing and Fault Tolerance

- Introduction
- Dependability attributes
- Applications with dependability requirements
- Impairments
- Techniques to improve dependability
  - **Fault tolerant techniques**

Introduction



Dependability attributes

- Reliability R(t): continuity of correct service
- Availability: readiness for correct service
  - A(t) (transient value),
  - A (steady state value)
- Safety S(t): absence of catastrophic consequences on the user(s) and the environment
- Performability P(L,t): ability to perform a given performance level
- Maintainability: ability for a system to undergo modifications and repairs
- Testability: attitude of a given system to be tested
- Security: degree of protection against danger, damage, loss, and criminal activity.

Reliability R(t), Availability A(t) & A

- **Reliability**, R(t): the conditional probability that a system performs correctly throughout the interval (t_0,t), given that the system was performing correctly at time t_0.
- **Istantaneous Availability**, A(t): the probability that a system is operating corretly and is available to perform its functions at the instant of time t
- **Limiting or steady state Availability**, A: the probability that a system is operating correctly and is available to perform its functions.

Reliability versus Availability

- Availability differs from reliability in that reliability involves an interval of time, while availability at an istant of time.

- A system can be highly available yet experience frequent periods of inoperability.

- The availability of a system depends not only on how frequently it becomes inoperable but also how quickly it can be repaired.

Safety S(t)

- **Safety**, S(t): the probability that a system will either perform its functions correctly or will discontinue its functions in a manner that does not disrupt the operation of other systems or compromise the safety of any people associated directly or inderectly with the system.
- The Safety is a measure of the fail-safe capability of a system, i.e, if the system does not operate correctly, it fails in a safe manner.
- Safety and availability differ because availability is the probability that a system will perform its function corretly, while Safety is the probability that a system will either perform its functions correctly or will discontinue the functions in a manner that causes no harm.

Performability P(L,t)

- **Performability**, P(L,t): the probability that a system performance will be at, or above, some level L, at instant t (Fortes 1984).
- It is a measure of the system ability to achieve a given performance level, despite the occurrence of failures.
- Performability differs from reliability in that reliability is a measure of the likehooh that all of the functions are performed correctly, while performability is a measure of likehood that some subset of the functions is performed correctly.

Security

- **Security** is the degree of protection against danger, damage, loss, and criminal activity.

- Security as a form of protections are *structures and processes that provide or improve security as a condition.*

- The key difference between security and reliability-availability-safety is that security must take into account the actions of people attempting to cause destruction.

Maintainability

- **Maintainability** is the probability *M(t)* that a malfunctioning system can be restored to a correct state within time *t*.
- It is a measure of the speed of repairing a system after the occurrence of a failure.
- It is closely correlated with availability:
  - The shortest the interval to restore a correct behavior, the highest the likelihood that the system is correct at any time *t*.
  - As an extreme, if *M(0)* = 1.0, the system will always be available.

Testability

- **Testability** is simply a measure of how easy it is for an operator to verify the attributes of a system.

- It is clearly related to maintainability: the easiest it is to test a malfunctioning system, the fastest will be to identify a faulty component, the shortest will be the time to repair the system.

Applications with dependability requirements (from Pradhan's book)

- Long life applications
- Critical-computation applications
- Hardly maintainable applications (Maintenance postponement applications)
- High availability applications

- ***Long life applications***: applications whose operational life is of the order of some year. The most common examples are the unmanned space flights and satellites. Typical requirements are to have a 0.95 or greater probability of being operational at the end of ten year period. This kind of system should or not have maintenance capability

Applications with dependability requirements (2/3)

- ***Critical-computation applications:*** applications that should cause safety problem to the people and to the business. Examples: aircraft, air-traffic flight control system, military systems, infrastructures for the control of industrial plants like nuclear or chemical plants. Typical requirements are to have a 0.999999 or greater probability of being operational at the end of three hour period. In this period normally it is not possible a human maintenance.
- ***Hardly Maintainable Applications*** *:* applications in which the maintenance is costly or difficult to perform. Examples: remote processing systems in not human region (like Antarctic continent). The maintenance can be scheduled independently by the presence of failure

Applications with dependability requirements (3/3)

- ***High availability applications:*** applications in which the availability is the key parameter.

  Users expect that the service is operational with high probability whenever it is requested.

  Examples: banking computing infrastructures. The maintenance can be done immediately and "easily".

Number of Nines as an Availability Metric

Availability %	Downtime per year	Downtime per month*	Downtime per week
90%	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
98%	7.30 days	14.4 hours	3.36 hours
99%	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 min
99.8%	17.52 hours	86.23 min	20.16 min
99.9% ("three nines")	8.76 hours	43.2 min	10.1 min
99.95%	4.38 hours	21.56 min	5.04 min
99.99% ("four nines")	52.6 min	4.32 min	1.01 min
99.999% ("five nines")	5.26 min	25.9 s	6.05 s
99.9999% ("six nines")	31.5 s	2.59 s	0.605 s

Impairments to dependability

IMPAIRMENTS TO DEPENDABILITY

- **delivered service deviates from fulfilling the system function** → **FAILURE**
- **part of system state liable to lead to failure** → **ERROR**
- **adjudged or hypothesized cause of error(s)** → **FAULT**

Causes and effects

FAULT

↓

ACTIVATION (INTERNAL) OR OCCURRENCE (EXTERNAL)

↓

ERROR

↓

PROPAGATION

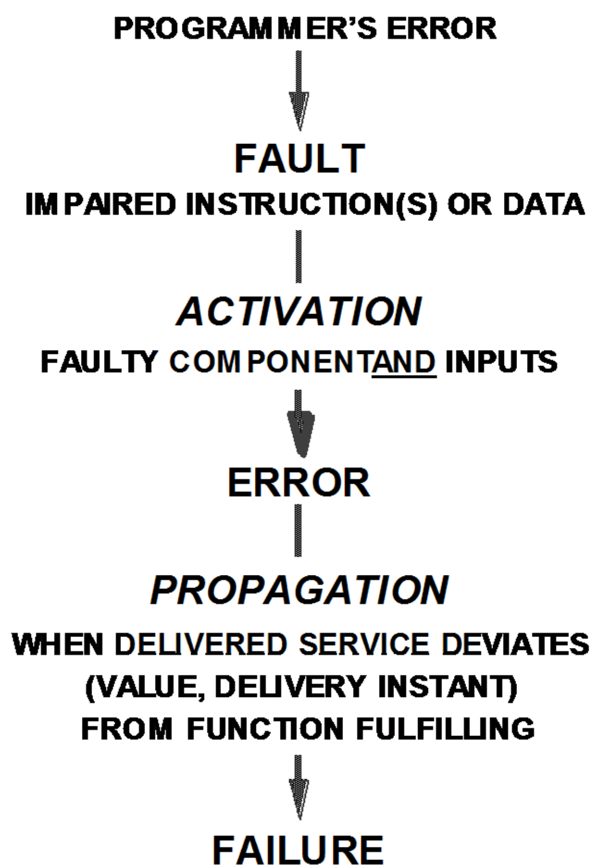
↓

FAILURE

ERROR : FAULT MANIFESTATION *IN SYSTEM*

FAILURE : ERROR MANIFESTATION *UPON SERVICE*

Example of human causes at design phase



Example of physical cause (permanent)

SHORT-CIRCUIT IN INTEGRATED CIRCUIT

FAILURE

↓

FAULT

STUCK-AT CONNECTION, FUNCTION MODIFICATIOI

ACTIVATION

FAULTY COMPONENT AND INPUTS

↓

ERROR

PROPAGATION

WHEN DELIVERED SERVICE DEVIATES
(VALUE, DELIVERY INSTANT)
FROM FUNCTION FULFILLING

↓

FAILURE

Example of human cause at operational phase

OPERATOR ERROR
IMPROPER HUMAN-MACHINE INTERACTION

FAULT

↓

ERROR

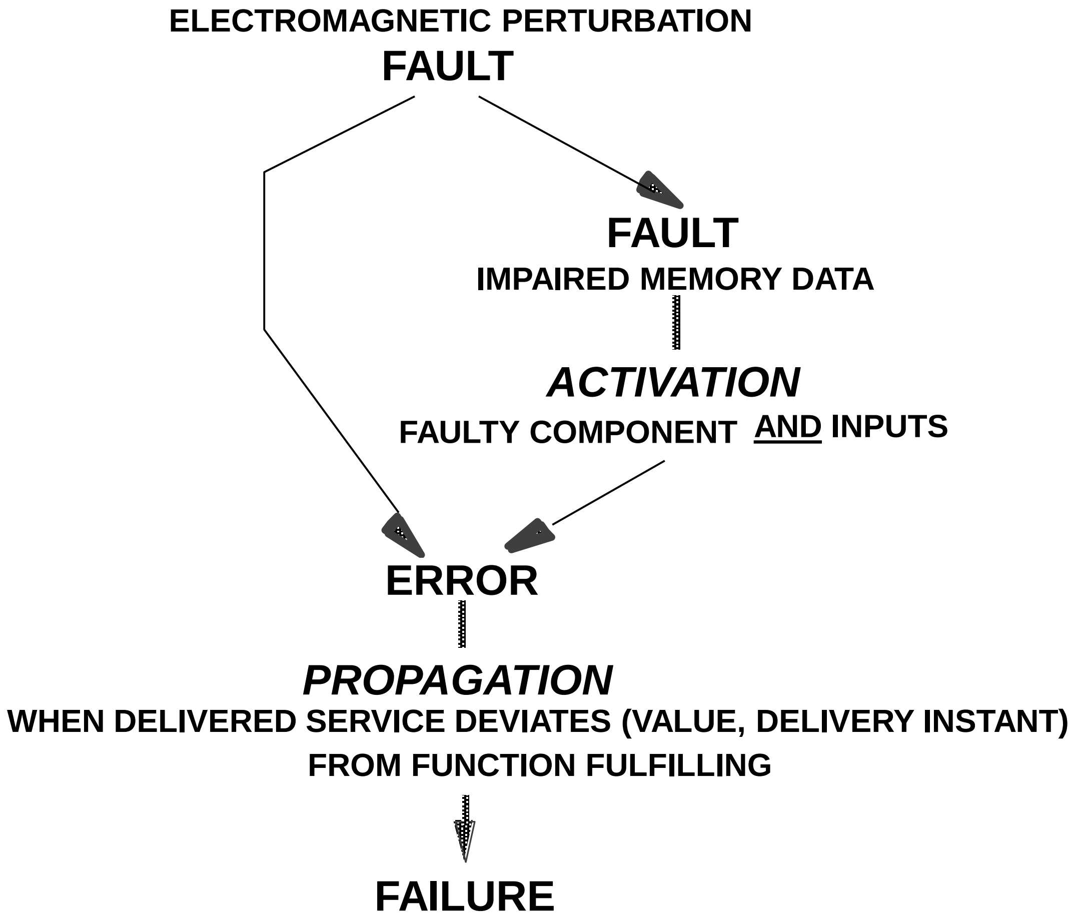
PROPAGATION

WHEN DELIVERED SERVICE DEVIATES
(VALUE, DELIVERY INSTANT)
FROM FUNCTION FULFILLING

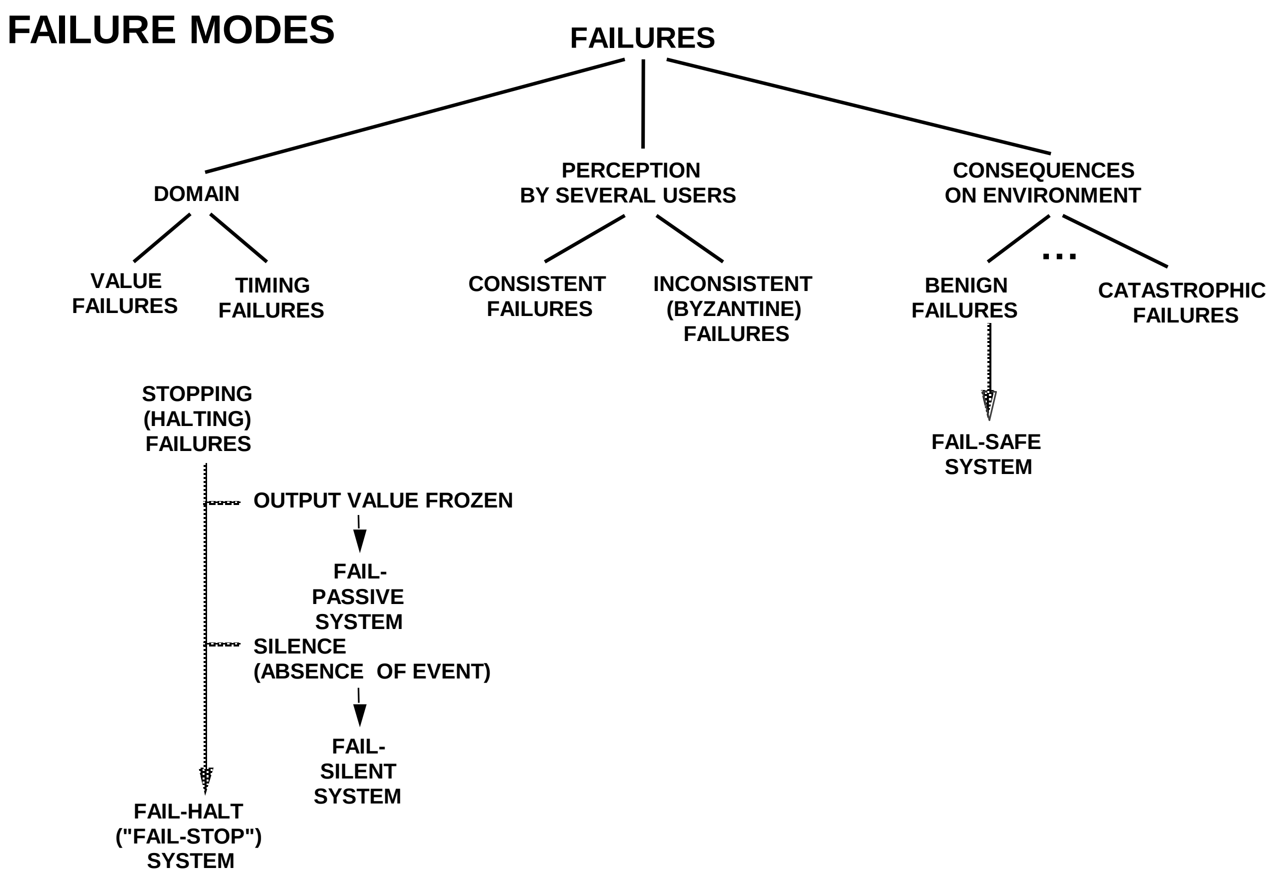
↓

FAILURE

Example of physical cause (transient)



Failure modes: taxonomy



Fault classification

- FAULTS
  - Phenomenological cause
    - Physical faults
    - Human-made faults
  - Nature
    - Accidental faults
    - Non-malicious intentional faults
    - Malicious faults
  - Phase of creation or occurrence
    - Development faults
    - Operational faults
  - System boundaries
    - Internal faults
    - External faults
  - Persistence
    - Permanent faults
    - Temporary faults

Fault classification (1/2)

PHENOMENOLOGICAL CAUSES

- physical faults: due to adverse physical phenomena
- human-made faults: result from human imperfections

NATURE

- accidental faults: appear or are created fortuitously
- intentional faults: created deliberately, with or without a malicious intention

PHASE OF CREATION WITH RESPECT TO THE SYSTEM'S LIFE

- development faults: result from imperfections
  - during the development of the system (from requirement specification to implementation) or during subsequent modificati
  - the establishment of the procedures for operating or main tainin system
- operational faults: appear during the system's exploitation

Fault classification (2/2)

SYSTEM BOUNDARIES

- **internal faults: those parts of the system state which, when invoke the computation activity, will produce an error**
- **external faults: result from system interference or interaction with physical (electromagnetic perturbations, radiation, temperature, vibration, etc.) or human environment**

TEMPORAL PERSISTENCE

- **permanent faults: presence is not related to pointwise conditions**
  - **internal (computation activity)**
  - **external (environment)**
- **temporary faults: present for a limited amount of time**

Human-made faults

- human-made fault classes

Intentional, non-malicious, faults

- design faults: result generally from tradeoffs
  - aimed at preserving acceptable performances, at facilitating the system utilization
  - induced by financial considerations
- interaction faults: may result from the action of an operator
  - aimed at overcoming an unforeseen situation
  - deliberately violating an operating procedure without having deve the consciousness of the possibly damaging consequences

realized often they were faults only after an unacceptable system behavior, thus a failure, has occurred

Malicious faults: specific labels

- design faults: malicious logics
  - development faults: Trojan horses, logic or timing bombs, trapdoor
  - operational faults: viruses, worms
- interaction faults: intrusions

Human-made faults: statistics

- **Human-made interaction faults**
  - **result from operators errors**
    - **errors: negative side of human activities**
    - **positive side: adaptability → aptitude to address unforecasted situations**
  - **Growing relative importance**

Causes of accidents in commercial flights in the USA

	Accidents per million take-offs			
	1970-78		1979-86	
Technical defects	1,49	(45%)	0,43	(33%)
Weather conditions	0,82	(25%)	0,33	(26%)
Human errors	1,03	(30%)	0,53	(41%)
Total	3,34		1,29	

- **Consciousness that most interaction faults have their source in the system *design***

Fault natures: some statistics (1/3)

Traditional systems, non fault-tolerant

- **USA, 450 companies, 1993 (FIND/SVP)**

  MTBF : 6 weeks

  Average downtime after failure: 3.5 h

Hardware		51%
Processors	24%	
Disks	27%	
Software		22%
Communication processors		11%
Communication network		10%
Procedures		6%

- **Japan, 1383 organizations, 1986**

  MTBF : 10 weeks

  Average downtime after failure: 1.5 h

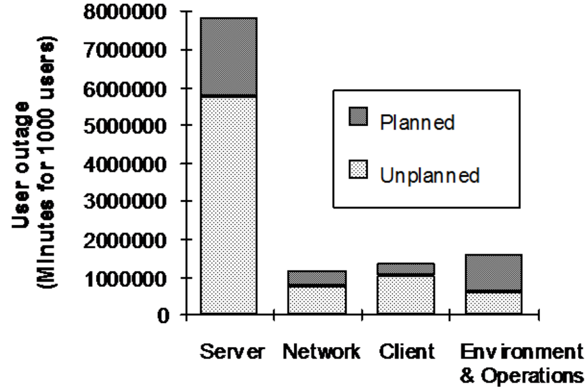
Vendor hardware and software, maintenance	42%	5 months
Application software	25%	9 months
Communication network	12%	18 months
Environment	11%	24 months
Operations	10%	24 months

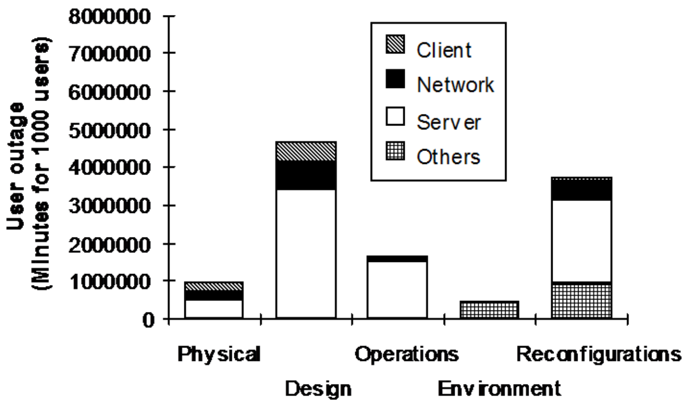
Fault natures: some statistics (2/3)

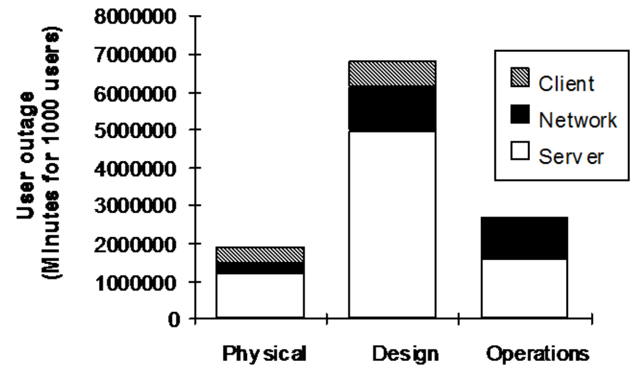
Tandem survey on Client-Server Networks

- **non fault-tolerant networks**
- **large networks: thousands workstations**

Yearly outage per user: 200 hours
Availability per user: 97,7 %
Planned outages (reconfigurations) : 32 %
Unplanned outages (failures) : 68 %







Redistribution {environment, reconfigurations}
on {physical, design, operations}

Fault natures: some statistics (3/3)

MTBF: **M**ean **T**ime **B**etween **F**ault
In the table **MTBE** and **MTFF** denotes **MTBF** for all kind of faults and for permanent ones, respectively

System, Technology	MTBE for all fault classes (h)	MTFF for permanent faults (h)	MTBE/MTFF
PDP-10, ECL	44	800-1600	0,03 - 0,06
CM* LSI-11, NMOS	128	4200	0,03
C.vmp TMR LSI-11, NMOS	97 - 328	4900	0,02 - 0,07
Telettra, TTL	80 - 170	1300	0,06 - 0,13
SUN-2, TTL-MOS	689	6552	0,11
1 Mx37 RAM, MOS	106	1450	0,07

13 stations of CMU Andrew network, 21 stations.years

	Number manifestations	Mean time to manifestation (h)
Permanent faults	29	6552
Intermitent faults	610	58
Transient faults	446	354
System crashes	298	689