Resilient and Survivable Networking
The University of Kansas EECS 983
Fault Tolerance

James P.G. Sterbenz

Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

jpgs@eecs.ku.edu

http://www.ittc.ku.edu/~jpgs/courses/rsnets
Resilient and Survivable Networking
Fault Tolerance

FT.1  Overview and definitions
FT.2  Techniques and mechanisms
Fault Tolerance

FT.1 Overview and Definitions

FT.1 Overview and definitions
   FT1.1 Fault $\rightarrow$ error $\rightarrow$ failure chain
   FT1.2 Types of faults
   FT1.3 Fault tolerance and relationship to other disciplines

FT.2 Techniques and mechanisms
Fault Tolerance

FT.1.1 Fault → Error → Failure chain

FT.1 Overview and definitions
  FT1.1 Fault → error → failure chain
  FT1.2 Types of faults
  FT1.3 Fault tolerance and relationship to other disciplines

FT.2 Techniques and mechanisms
Fault $\rightarrow$ Error $\rightarrow$ Failure Chain

Challenges and Threats

- **Challenge**
  - adverse event or condition that impacts normal operation
    - unintentional mis-configuration or operational mistakes
    - disasters: large-scale natural disasters and man-made
    - malicious attacks from intelligent adversaries
    - environmental challenges
    - unusual but legitimate traffic load such as a flash crowd
    - a service failure at a lower level

- **Threat**
  - potential *challenge* that might exploit a *vulnerability*

[L1994], [ALRL2004TR]
Fault → Error → Failure Chain

Faults and Vulnerabilities

• **Fault**
  - property of a system based on its design
  - cause of an *error*
    - *dormant* (or latent) when does not yet cause an error
    - *active* when it causes an error
  - may be internal or external to a given system
  - cannot be directly observed
    - no such thing as “fault detection”

• **Vulnerability**
  - internal fault that allows an external fault to cause an error

[L1994], [ALRL2004TR]
Fault $\rightarrow$ Error $\rightarrow$ Failure Chain

Challenges and Faults

- **Challenges**
  - Environmental: mobile, wireless, delay
  - Natural Disasters
  - Non-malicious: ops., traffic, accidents
  - Malicious attacks
  - Lower-level failure

- **Faults**
  - Dormant Faults
  - Active Faults:
    - External Fault
    - Internal Fault

- **System Operation**

- **Errors**
  - Detect
  - Refine

- **Defend**
  - Errors passed on to operational state
Fault $\rightarrow$ Error $\rightarrow$ Failure Chain

Errors

- **Fault** $\rightarrow$
- **Error**
  - stochastic event in either space (system) or time
  - manifestation of a *fault*
  - system state that may lead to a subsequent failure
  - errors can be detected
    - and used for *fault diagnosis*

Fault → Error → Failure Chain

Challenges and Faults

Environmental: mobile, wireless, delay
Natural Disasters
Non-malicious: ops., traffic, accidents
Malicious attacks
Lower-level failure

Challenges

Defend

Defend

Defend

Errors passed on to operational state

Detect

Detect

Diagnose

Refine

Errors

System Operation

Dormant Faults

Active

External Fault

Internal Fault
Fault $\rightarrow$ Error $\rightarrow$ Failure Chain

Failures

- **Fault $\rightarrow$ Error $\rightarrow$**
- **Service failure**
  - deviation of delivered service from service specification
  - may result from an error (but may not)
  - transition from correct to incorrect service state
    - *service outage*: incorrect service state
    - *timing failure*: performance degradation
    - *content failure*: incorrect information
- **Service restoration**
  - transition from incorrect to correct service state

[L1994], [ALRL2004TR]
Fault → Error → Failure Chain

Challenges and Faults

Environmental: mobile, wireless, delay
Natural Disasters
Non-malicious: ops., traffic, accidents
Malicious attacks
Lower-level failure

Challenges

Detect

Defend

Errors passed on to operational state

Detect

System Operation

Defend

Errors

Diagnose Refine

Dormant Faults

Active

External Fault

Internal Fault

Defend
Fault Tolerance

FT.1.2 Types of Faults

FT.1 Overview and definitions
  FT1.1 Fault, error, failure chain
  FT1.2 Types of faults
  FT1.3 Fault tolerance and relationship to other disciplines

FT.2 Techniques and mechanisms
Types of Faults
Classification and Taxonomy

• Classification and taxonomy of faults
  - based on [ALBL 2004] and IFIP 10.4 related publications

• Fault groups: major overlapping types of faults
  - development faults occur during system development
  - physical faults include all fault classes that affect hardware
  - interaction faults include all external faults

• Elementary fault classes
  - elementary orthogonal classification within fault groups
Types of Faults

Elementary Fault Classes

- Elementary fault classes
  - phase of creation or occurrence
  - system boundaries
  - phenomenological cause
  - dimension
  - objective
  - intent
  - capability
  - persistence

- Not all $2^8 = 256$ combinations possible
  - e.g. natural faults (phenom.) can’t be malicious (objective)
Types of Faults: Elementary Classes
Phase of Creation of Occurrence

Phase of creation or occurrence

• Developmental faults
  - during system development
  - maintenance during use phase
  - generation of procedures to operate or maintain system

• Operational faults
  - during service delivery of use phase
    - configuration faults
      • human-made faults result from incorrect system parameters
    - reconfiguration faults
      • human-made faults from incorrect upgrade or change
Types of Faults: Elementary Classes

System Boundaries

System boundaries

- **Internal faults**
  - originate within system boundaries
  - e.g. execution of code

- **External faults**
  - originate outside system boundary from a challenge
  - interact or interfere with system operation
Types of Faults: Elementary Classes

Phenomenological Cause

Phenomenological cause

- **Natural faults**
  - *physical faults* caused by natural phenomena
  - without (direct) human participation
  - include *production defects* during development
  - include physical deterioration *internal hardware faults*

- **Human-made faults**
  - result from human actions
  - *omission faults* result from absence of required actions
  - *commission faults* result from wrong actions
Types of Faults: Elementary Classes

Dimension

- **Hardware faults**
  - originate in hardware
  - affect hardware
- **Software faults**
  - affect software
  - programs, data, protocols
Objective

- **Malicious faults**
  - introduced by human with intent to harm system
  - may be *developmental faults* or *operational faults*
  - may be automated (e.g. Botnet)

- **Non-malicious faults**
  - introduced without malicious objective
  - include all *natural faults*
  - include *human-made deliberate faults* due to bad decisions
  - include *human-made non-deliberate faults* due to mistakes
Types of Faults: Elementary Classes

Intent

• *Deliberate faults*
  - result of a harmful decision

• *Non-deliberate faults*
  - introduced without awareness
  - include mistakes
Types of Faults: Elementary Classes

Capability

- **Accidental faults**
  - introduced inadvertently

- **Incompetence faults**
  - from lack of professional competence by authorised humans
  - inadequacy of development or deployment organisation
Types of Faults: Elementary Classes

Persistence

- **Permanent faults**
  - presence assumed to be continuous in time
  - include physical deterioration *internal hardware faults*

- **Transient or temporary faults**
  - presence bounded in time
Fault Tolerance

FT.1.3 Relationship to Other Disciplines

FT.1 Overview and definitions

FT1.1 Fault, error, failure chain
FT1.2 Types of faults
FT1.3 Fault tolerance and relationship to other disciplines

FT.2 Techniques and mechanisms
Fault Tolerance
Type of Discipline

- Fault-tolerance is related to challenges
  - subset of survivability
  - peer to disruption-tolerance and traffic-tolerance
- FT measured by trustworthiness disciplines
  - dependability: availability, reliability, etc.
  - performability
  - security
Fault Tolerance

Relationship to Other Disciplines

- Trustworthiness
  - Dependability
    - reliability
    - maintainability
    - safety
  - availability
  - integrity
  - confidentiality
  - Security
    - nonrepudiability
    - AAA
      - auditability
      - authorisability
      - authenticity
  - Performability
    - QoS measures

- Robustness Complexity
  - Challenge Tolerance
    - Survivability
      - many ∨ targeted failures
    - Disruption Tolerance
      - environmental
      - delay
      - mobility
      - connectivity
      - energy
    - Traffic Tolerance
      - legitimate flash crowd
      - attack DDoS
Fault Tolerance

Definition

• *Fault tolerance*
  - avoid service *failures* in the presence of *faults*

• Mature discipline
  - generally assumes *independent random* faults
  - traditional fault models do *not* hold under
    - malicious attack and large-scale natural disaster
    - generally the domain of *survivability*
Byzantine Fault Tolerance

Definition

• Conflicting information to different parts of system
  – may be malicious

• **Byzantine fault tolerance**
  – avoid *failures* in the presence of *Byzantine faults*

• Preventing Byzantine failures
  – less than 1/3 Byzantine components if not authentication

[LSP1982]
Fault Tolerance
Analytical Relation to Reliability

• Robustness:
  challenge tolerance improves trustworthiness
    – fault tolerance improves dependability

• \[
  R = \Pr[\text{no fault}] + \Pr[\text{correct operation} \mid \text{fault}] \times \Pr[\text{fault}]
\]
  – \(\Pr[\text{no fault}]\) is correct (fault intolerant) design
    • if system is good enough, fault tolerance isn’t needed
  – \(\Pr[\text{correct operation} \mid \text{fault}]\) is coverage of fault tolerance
    • conditional probability of correction operation given faults
  – \(\Pr[\text{fault}]\) is probability faults will occur (dormant→active)

[N1990]
Fault Tolerance
Series and Parallel Availability Composition

- Series availabilities multiply: \( A(t) = \prod R_i(t) \)
  - conservative approximation as sum of unavailabilities
    \( U(t) = \sum U_i(t) \)

- Parallel availabilities: \( A(t) = 1 - \prod [1-R_i(t)] \)
  - exact product of unavailabilities
    \( U(t) = \prod U_i(t) \)

- Series-parallel iterative reductions
  - not possible for many network topologies (diagonals)
Fault Tolerance
Confusion with Survivability

• Some communities use survivability to mean FT
  - optical networking community: “survivable optical rings”
Fault Tolerance

FT.2 Techniques and Mechanisms

FT.1 Overview and definitions

FT.2 Techniques and mechanisms

FT2.1 Fault masking

FT2.2 Redundancy
Fault Tolerance Techniques

FT.2.1 Redundancy

FT.1 Overview and definitions
FT.2 Techniques and mechanisms
  FT2.1 Fault masking
  FT2.2 Redundancy
Fault Tolerance Techniques
Redundancy and Diversity

- Redundancy is the fundamental FT technique
  - redundancy: multiple components or mechanisms
  - diversity: alternatives in components or mechanisms
    - primarily a survivability technique *Lecture SV*
Redundancy Techniques

• Redundancy: *replication of parts or modules of system*
  • components, e.g. electronic circuits, switches, links
  • information, e.g. packets, communication circuits
  • algorithms, e.g. *N-version programming*
    - permit operation even when some parts have failed
Redundancy

RB vs. NVP

• RB: recovery blocks (software)
  standby sparing (hardware)
  - sequential processing
  - recovery invoked if acceptance test fails

• NVP: n-verison programming (software)
  n-modular redundancy (hardware)
  - parallel processing
  - results compared or voted
Redundancy

\( M\) -of- \( N \) Redundancy

- \( M\)-of-\( N \) redundancy
  - \( N \) modules in system
  - \( M \) modules needed to maintain operational state
  - \((N - M) + 1\) module failures
    - cause error
    - that may result in system failure (at the next higher level)
Redundancy

$M$-of-$N$ Redundancy Example

- 3-of-5 redundancy
Redundancy

$M$-of-$N$ Redundancy Example

- 3-of-5 redundancy
  - up to $(3 - 5) = 2$ module failures are tolerated
Redundancy

$M$-of-$N$ Redundancy Example

- 3-of-5 redundancy
  - up to $(3 - 5) = 2$ module failures are tolerated

![Diagram of 3-of-5 redundancy example]
Redundancy

3-of-5 Redundancy Example

- 3-of-5 redundancy
  - $(3 - 5) + 1 = 3$ module failures cause error
  - that may result in system failure (at the next higher level)
Redundancy

$M$-of-$N$ Important Cases

- **1+1 redundancy**
  - **1:1 redundancy**
    - every component has a backup
    - duplex or dual modular redundancy (DMR)
    - 1-of-2 in $M$-of-$N$ terminology

- **N+1 redundancy**
  - **1:N redundancy**
    - one redundant component for a group of $N$
    - $N$-of-$(N+1)$ in $M$-of-$N$ terminology
Redundancy

$M$-of-$N$ Alternative Use of Redundancy

- **Hot standby**
  - unused; ready for substitution

- **Dynamic redundancy**
  - used but ready to load balance

*problem?
Redundancy

$M$-of-$N$ Alternative Use of Redundancy

- **Hot standby**
  - unused; ready for substitution

- **Dynamic redundancy**
  - used but ready to load balance
    - useful for processors and networking
    - may result in service degradation

- **Choice based on**
  - probability of error
  - degradation-tolerance of service
Redundancy Techniques

1+1 Redundancy Checking

- 1+1 redundancy checking
  - comparator raises alarm if outputs don’t match

_is this fault tolerant?_
Redundancy Techniques

1+1 Redundancy Checking

- 1+1 redundancy checking
  - comparator raises alarm if outputs don’t match
  - doesn’t indicate *which* module has failed
• 1+1 self-checking
  - comparator built into modules
  - module raises alarm if outputs don’t match
  - doesn’t indicate *which* module has failed

*Modification for fault tolerance?*
Redundancy Techniques

2+2 Duplexed Self-Checking

- Duplexed self-checking
  - error signals control switchpoints
  - error alarm indicates *which* self-check module had failed

*is this fault tolerant?*
Redundancy Techniques
2+2 Duplexed Self-Checking

- Duplexed self-checking
  - error signals control switchpoints
  - error alarm indicates *which* self-check module had failed
  - fault tolerant to a single self-check module failure

*More efficient solution possible?*
Redundancy Techniques

Triple Modular Redundancy

- Triple modular redundancy (TMR)
  - TMR is special case 2-of-3 triplex redundancy
  - three modules perform same task
    - typically identical modules in lock step or loose synchronisation
    - modules may not be identical (form of diversity)

*what is missing?*
Redundancy Techniques

Triple Modular Redundancy

- Triple modular redundancy (TMR)
  - TMR is special case 2-of-3 triplex redundancy
  - three modules perform same task
  - *voter* compares result
Redundancy Techniques
Triple Modular Redundancy

- Triple modular redundancy (TMR)
  - TMR is special case 2-of-3 triplex redundancy
  - three modules perform same task
  - voter compares result
    - identical result: system operational
    - 2 to 1 vote: fault masked; alarm raised
    - 3 different votes: unrecoverable fault; system fails

issues and assumptions?
Redundancy Techniques
TMR Issues and Assumptions

- Triple modular redundancy (TMR)
  - failures must be independent and uncorrelated

  *why?*
Redundancy Techniques
TMR Issues and Assumptions

- Triple modular redundancy (TMR)
  - failures must be independent and uncorrelated
    - otherwise fault tolerance will be *reduced*
  - voter must be significantly reliable than modules
    - typically very simple circuit or program fragment
  - 3 times resource needed
Redundancy Techniques

Multistage TMR

- Triple modular redundancy (TMR)
  - can be extended by triplexing voters in-between stages
Redundancy Techniques
TMR Use in Networks

• Possible application of TMR to network design
  - TMR component design in switches and routers
  - erasure coding: multiple copies of information
    • in time: multiple copies on a path
    • in space: multiple copies spread across paths
Redundancy Techniques
Network Multilevel

- Physical layer
- HBH Link layer
- Topology layer (sublayer of network layer)
- Path layer (sublayer of network layer)
- E2E Transport layer
- Application layer
Redundancy Techniques

Network Multilevel: Link Layer

• Link layer: redundant links
  - redundant links in case one fails

*Problem?*
Redundancy Techniques
Network Multilevel: Link Layer

• Link layer: redundant links
  – redundant links in case one fails

• Problem:
  – geographically entwined links generally fail together
    • back hoe fade
Redundancy Techniques

Dual Rings

- Dual rings for fault tolerance
  - e.g. SONET APS (automatic protection switching)
Diversity
Definition and Measure

- **Diversity** consists of providing different alternatives
  - when challenges impact particular alternatives
    other alternatives prevent degradation
  - generally a survivability technique _Lecture SV_
Fault Tolerance

References and Further Reading

References on [wiki.ittc.ku.edu/resilinets/Fault_Tolerance]

[N1990] Victor P. Nelson,
   *IEEE Computer*, vol.23, #7, July 1990, pp. 19–25

   “Definition and Analysis of Hardware- and Software-Fault-
   Tolerant Architectures”,
   *IEEE Computer*, vol.23, #7, July 1990, pp. 39–51

[MS1956] E. F. Moore and Claude E. Shannon,
   “Reliable Circuits using Less Reliable Relays”,
   *Journal of the Franklin Institute*, Sep. 1956, pp. 191–208
Fault Tolerance

References and Further Reading

[L1994] Jean-Claude Laprie (ed.),
Dependability: Basic Concepts and Terminology,
IFIP WG 10.4 – Dependable Computing and Fault Tolerance,
(draft), Aug. 1994

Basic Concepts and Taxonomy of Dependable and Secure
Computing, LAAS TR 2004-47, 2004

[T-P2000] Wilfredo Torres-Pomales,
Software Fault Tolerance: A Tutorial,

[LSP1982] Leslie Lamport, Robert Shostak, and Marshall Pease,
“The Byzantine Generals Problem”,
ACM TOPLAS, vol.4, #3, July 1982, pp. 382–401
End of Foils