

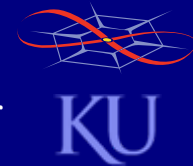
Resilient and Survivable Networking

The University of Kansas EECS 983

Fault Tolerance – Spring 2008

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 → error → 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 → Error → Failure Chain

Challenges and Threats

- *Challenge*
 - adverse event or condition that impacts normal operation
 - unintentional mis-configuration or operational mistakes
 - large-scale natural disasters
 - 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*

Fault → Error → Failure Chain

Faults and Vulnerabilities

- *Fault*

- property of a system based on its design
- cause of an *error*
 - *dormant* (or latent) when it 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”

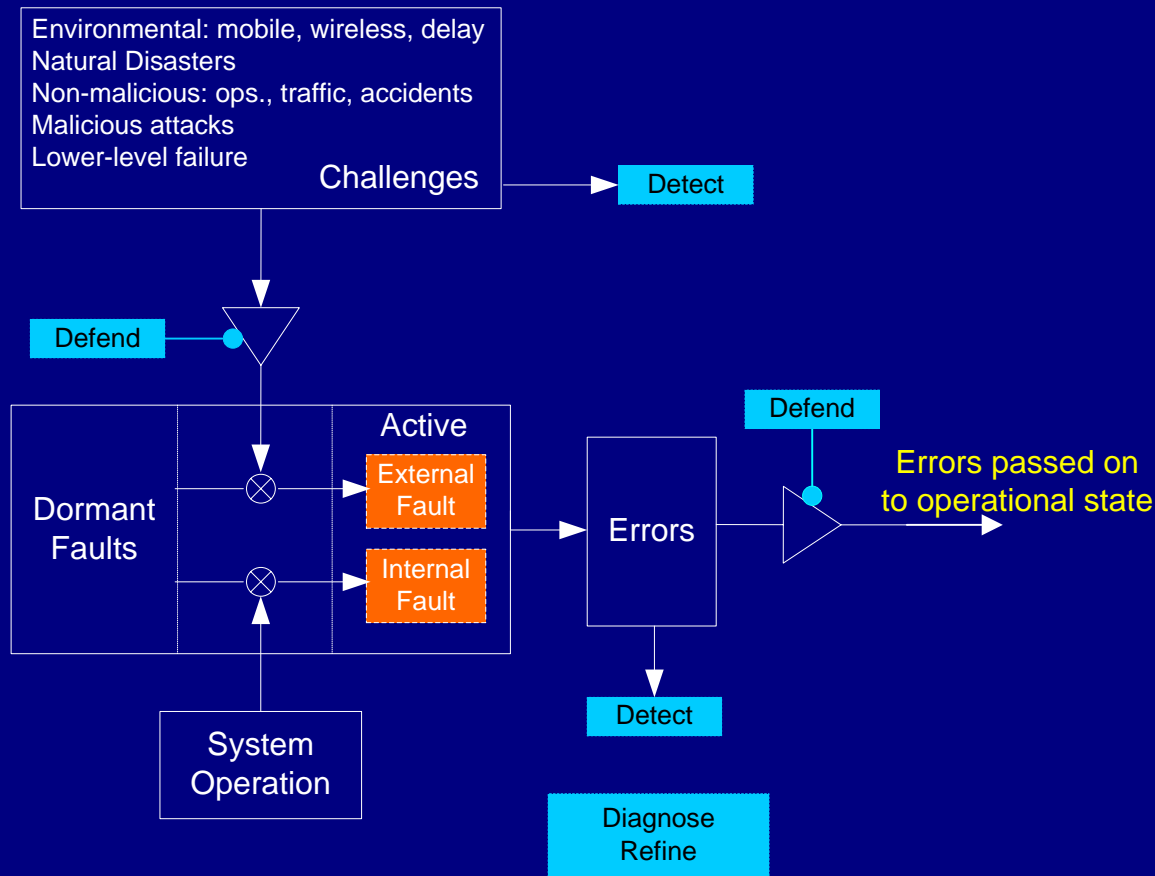
- *Vulnerability*

- internal fault that allows an external fault to cause an error

[Laprie-1994], [Avizienis-Laprie-Randell-Landwehr-2004TR]

Fault → Error → Failure Chain

Challenges and Faults



Fault → Error → Failure Chain

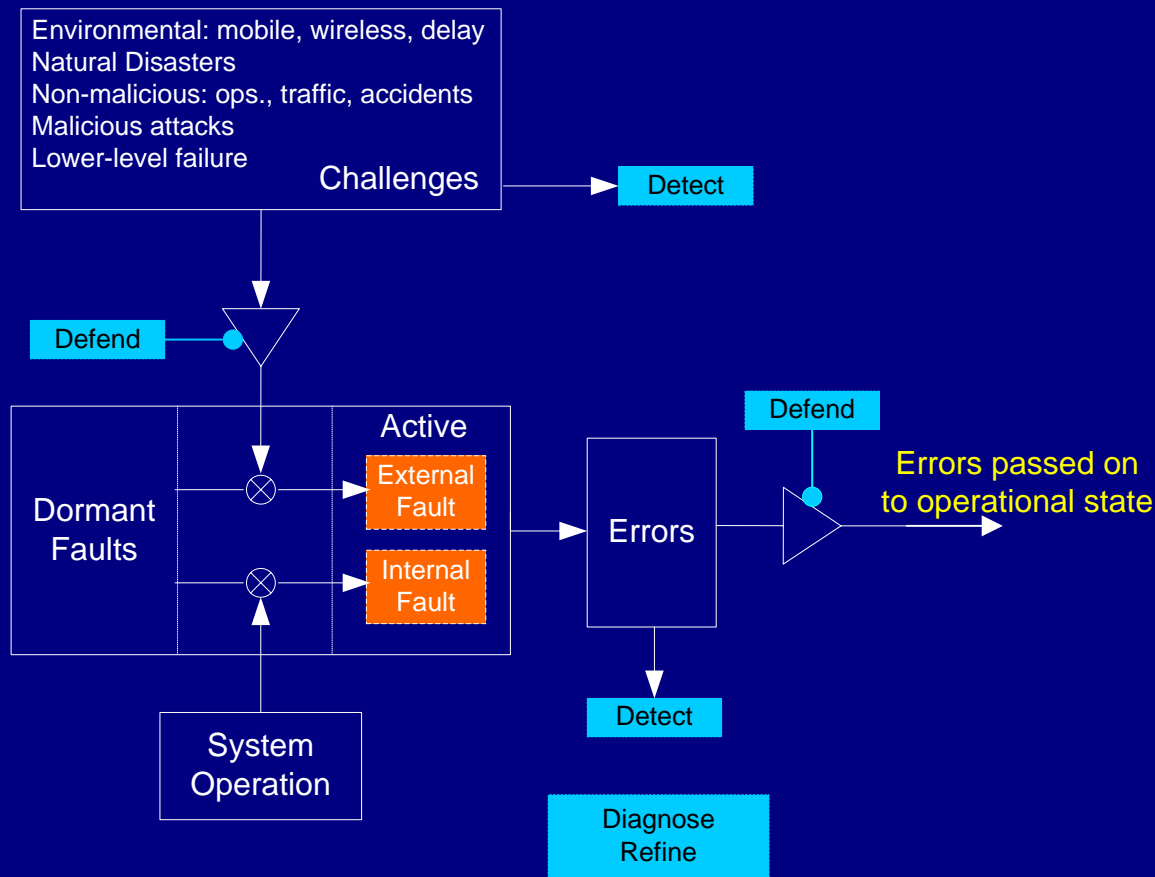
Errors

- *Fault* →
- *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*

[Laprie-1994], [Avizienis-Laprie-Randell-Landwehr-2004TR]

Fault → Error → Failure Chain

Challenges and Faults



Fault → Error → Failure Chain

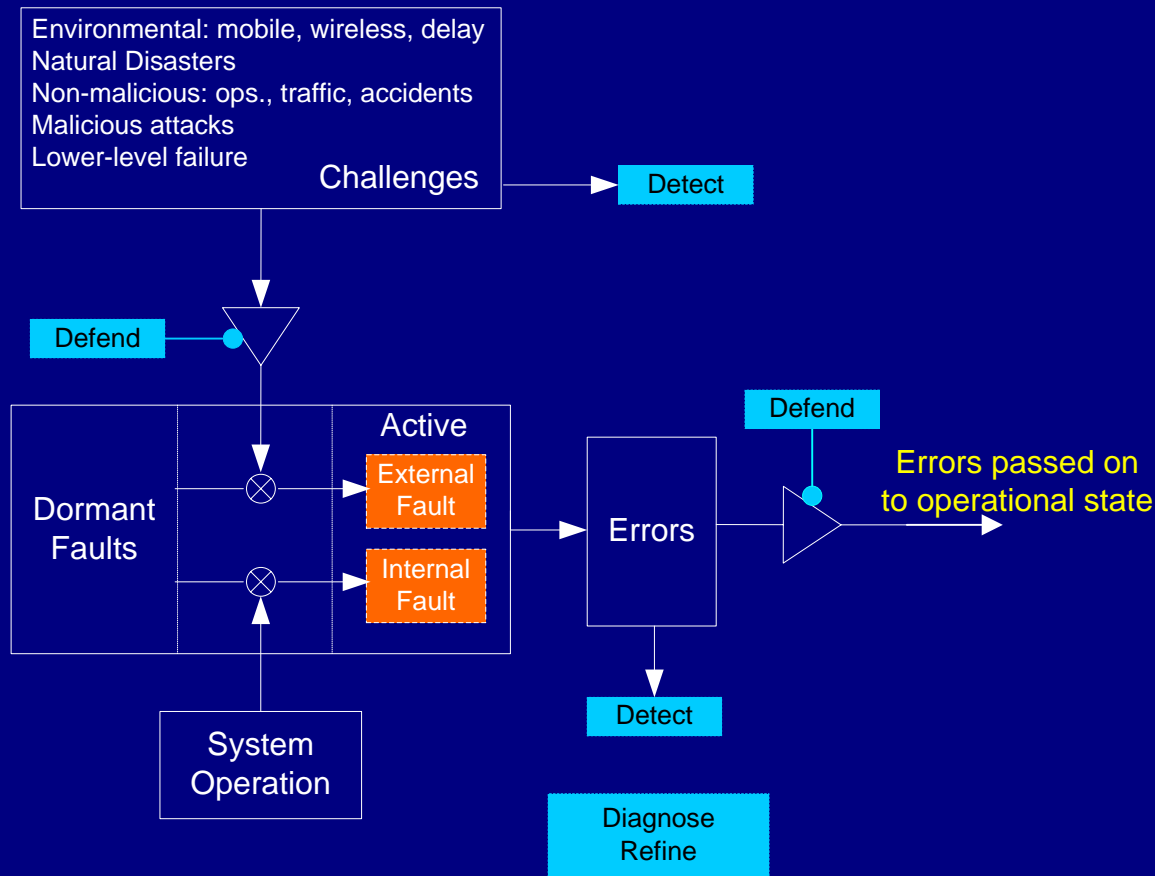
Failures

- *Fault* → *Error* →
- 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

[Laprie-1994], [Avizienis-Laprie-Randell-Landwehr-2004TR]

Fault → Error → Failure Chain

Challenges and Faults



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

Dimension

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

Types of Faults: Elementary Classes

Objective

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

Intent

- *Deliberate faults*
 - result of a harmful decision
- *Non-deliberate faults*
 - introduced without awareness
 - include mistakes

Types of Faults: Elementary Classes

Capability

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

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

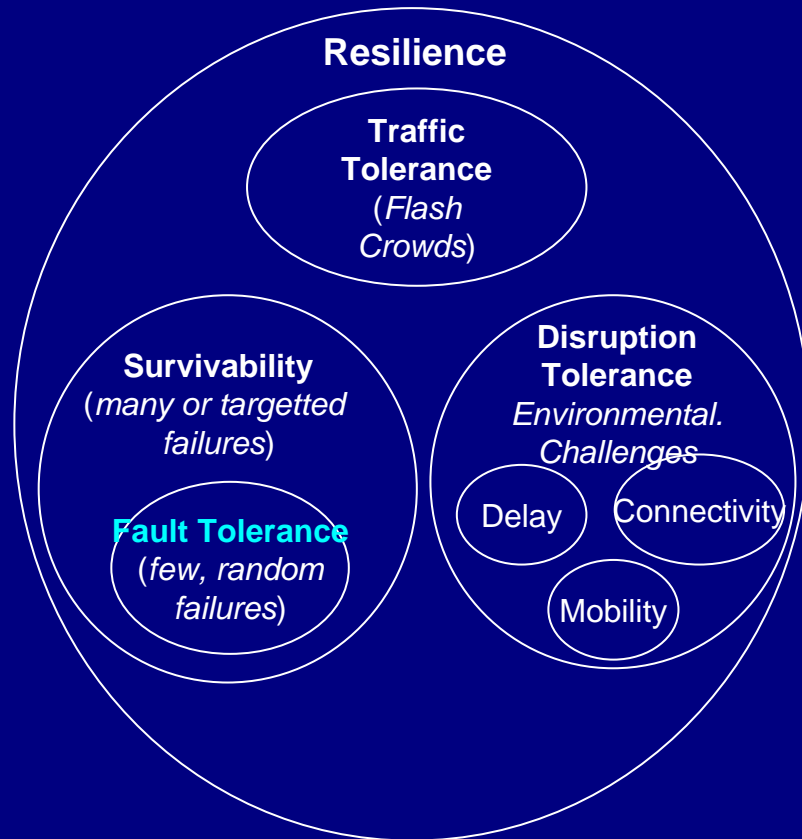
Type of Discipline

- Fault-tolerance is related to challenges
 - subset of survivability
 - peer to disruption-tolerance and traffic-tolerance
 - *needs further consideration*
- Fault tolerance measured by other disciplines
 - performance
 - dependability: availability, reliability, etc.
 - security

Fault Tolerance

Relationship to Other Disciplines

Challenge Based Concepts



Measurement and Metrics



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

[Lamport-Shostak-Pease-1982]

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 a 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

M -of- N Redundancy

- M -of- N redundancy
 - N modules in system
 - M modules needed to maintain operational state
 - $(N - M)$ 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?

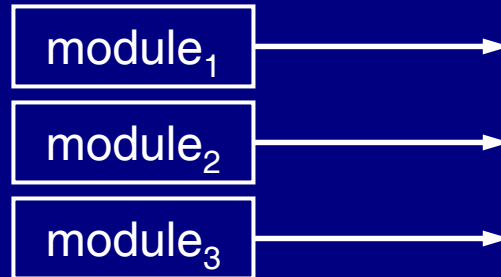
Redundancy

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

Triple Modular Redundancy

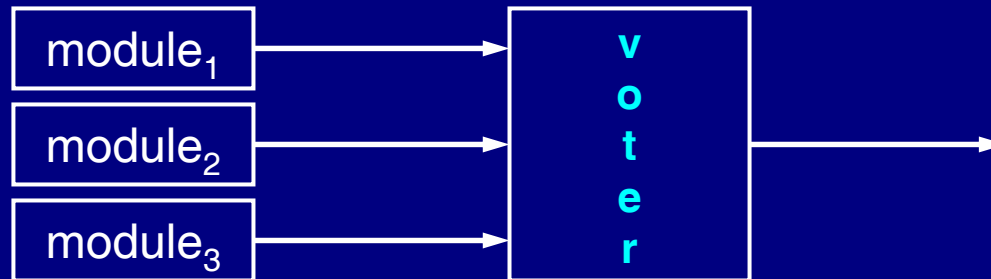


- Triple modular redundancy (TMR)
 - TMR is special case 2-of-3 triplex redundancy
 - three modules perform same task
 - typically identical modules in lock step or loose synchronisation
 - modules may not be identical (form of diversity)

what is missing?

Redundancy Techniques

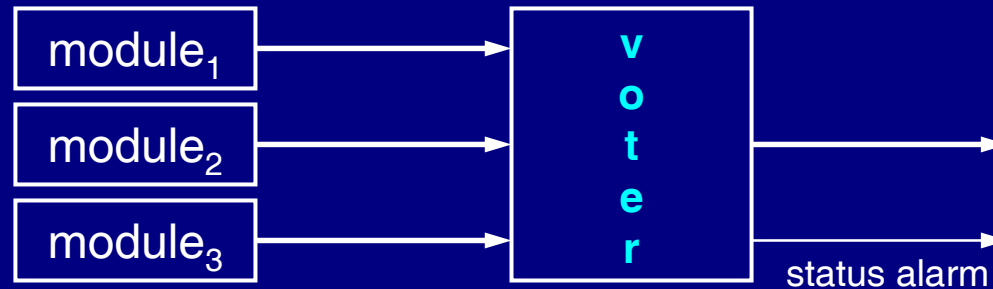
Triple Modular Redundancy



- 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

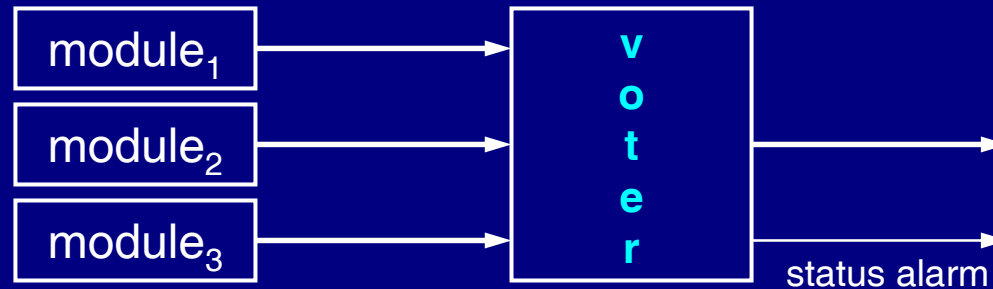


- Triple modular redundancy (TMR)
 - TMR is special case 2-of-3 triplex redundancy
 - three modules perform same task
 - *voter* compares result
 - identical result: system operational
 - 2 to 1 vote: fault masked; alarm raised
 - 3 different votes: unrecoverable fault; system fails

issues and assumptions?

Redundancy Techniques

TMR Issues and Assumptions

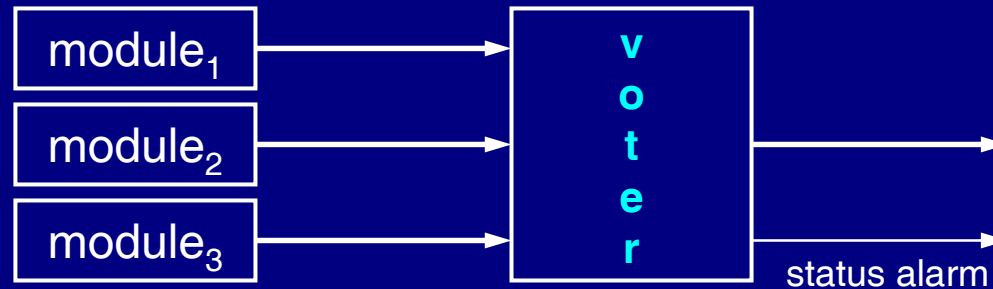


- Triple modular redundancy (TMR)
 - failures must be independent and uncorrelated

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

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
- Link layer
- Network layer
- 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/Fault_Tolerance

End of Foils