ES623 Networked Embedded Systems

Fault Tolerance

5th April 2013

Fault-Tolerant Unit (FTU)

- Purpose of FTU is to mask the failures of a node
- If a node implements the fail-silent abstraction, then the duplication of nodes is sufficient to tolerate a single node failure.
- If the node can exhibit value errors at the host/network interface CNI, then triple-modular redundancy (TMR), must be implemented.
- Assuming that the behavior of the nodes is replica determinate, and do not exhibit babbling idiot timing failures in bus systems.
- a node can exhibit Byzantine failures also

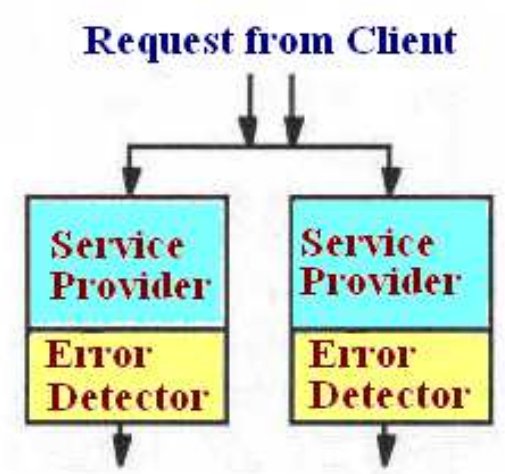
Fail-Silent Node

- Produces correct results or does not produce any results at all
- In time-triggered architecture, an FTU that consists of two fail-silent nodes produces either zero, one, or two correct result messages.
- If it produces no message, it has failed.
- If it produces one or two messages, it is operational.
- The receiver must discard redundant result messages.

Fail-Silent Node

- In a bus-based system, an FTU can comprise a *shadow node* in addition to the two active nodes.
- The shadow node acts as a *standby:*
  - it reads all messages from the bus, and is fully synchronized with the active nodes, but does not produce any output messages as long as it is in the "shadow" state.
- As soon as one of the active nodes fails, the "shadow" node acquires the output bus slots of the failed node, and thereby becomes an active node.
- If the failed node is repaired, it reintegrates itself as a shadow node.

Fail-Silent Node



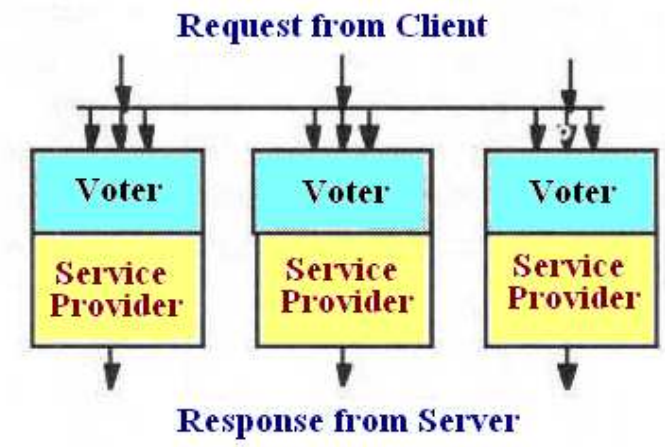
Fail-Silent Node

- Advantages
  - Whenever an active node fails, the redundancy within the FTU is reestablished within a short time interval.
  - During normal operation the shadow node does not consume any bandwidth of the communication system.
  - During repair of the failed node, the redundancy within the FTU is maintained.

Triple-Modular Redundancy

- If a node can exhibit value failures at the CNI with a probability that cannot be tolerated in the given application domain,
  - fault-tolerant unit must consist of three nodes and a *voter*.
- *Voter detects and masks errors in one step by comparing* three independently computed results, and selecting result that has been computed by majority, i.e., by two out of three

Triple-Modular Redundancy



- Two different kinds of voting strategies:
  - exact voting and inexact voting

Triple-Modular Redundancy

- Exact voting: a bit-by-bit comparison of data fields in the result messages of the three nodes is performed
- Inexact voting: two messages are assumed to contain the same result if the results are within some *application-specific interval*.
  - Used if replica determinism cannot be guaranteed.
  - selection of an appropriate interval is a delicate task
  - If interval is too large, erroneous values will be accepted as correct
  - If interval is too small, correct values will be rejected as erroneous

Byzantine Resilient Fault-Tolerant Unit

- If no assumption about the failure mode of a node, then, four nodes are needed to form a FTU that can tolerate a single Byzantine (or malicious) fault.
- Byzantine agreement protocols to tolerate the Byzantine failures of *k nodes :*
  - An FTU must consist of at least *3k+1 nodes.*
  - Each node must be connected to all other nodes of the FTU by *k+1 disjoint* communication paths.
  - To detect malicious nodes, *k+1* rounds of communication must be executed. A round of communication requires every node to send a message to all the other nodes.
  - Nodes must be synchronized to with a known precision.