CONSENSUS AND FAILURE DETECTION

FLP: Impossibility of Consensus

- Consensus is impossible
  - … in asynchronous system
  - … with crash-stop failures
- Adversary argument:
  - Any protocol cannot block
  - Delay delivery of critical message
  - Force system to reconfigure
  - Deliver message now it's no longer critical
  - Continue ad infinitum

FLP: Impossibility of Consensus

- Consensus is impossible
  - … in asynchronous system
  - … with crash-stop failures
- Adversary argument:
  - Relies on only one failure (message loss)
    - …which never actually happens!
  - *Key point: protocol cannot distinguish failure from delay*

FLP: Impossibility of Consensus

- Suppose we knew *exactly* one failure
- If N processes, then every process broadcasts its input (true or false) to every other process
- Each process: Make decision after receiving N-1 broadcasts

Properties of Failure Detectors

- Completeness: detection of every crash
  - **Strong completeness:** Eventually, every process that crashes is permanently suspected by every correct process
  - **Weak completeness:** Eventually, every process that crashes is permanently suspected by some correct process

51

51

Properties of Failure Detectors

- Accuracy: does it make mistakes?
  - **Strong accuracy:** No process suspected before it crashes.
  - **Weak accuracy:** Some correct process is never suspected
  - **Eventual strong accuracy:** there is a time after which correct processes are not suspected by any correct process
  - **Eventual weak accuracy:** there is a time after which some correct process is not suspected by any correct process

52

52

A sampling of failure detectors

Completeness	Accuracy			
	Strong	Weak	Eventually Strong	Eventually Weak
Strong	*Perfect* $\mathcal{P}$	*Strong* $\mathcal{S}$	*Eventually Perfect* $\Diamond\mathcal{P}$	*Eventually Strong* $\Diamond\mathcal{S}$
Weak	$\mathcal{D}$	*Weak* $\mathcal{W}$	$\Diamond\mathcal{D}$	*Eventually Weak* $\Diamond\mathcal{W}$

Perfect Detector

- Named *Perfect,* written $\mathcal{P}$
- Strong completeness and strong accuracy
- Immediately detects all failures
- Never makes mistakes

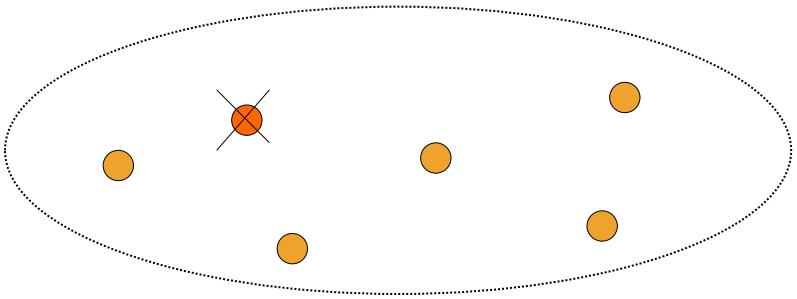
Eventually Weak Detector

- Eventually Weak: ◊W: "diamond-W"
- Weak Completeness: There is a time after which every process that crashes is suspected by *some* correct process
  - If it crashes, "we eventually, accurately detect the crash"
- Eventually Weak Accuracy: There is a time after which *some* correct process is never suspected by any correct process
  - Think: "we can eventually agree upon a leader."
  - Failure detectors are unreliable, but mistakes are recognized

55

55

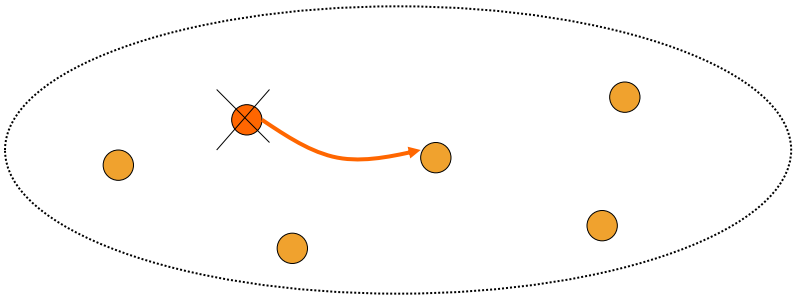
From Weak Completeness to Strong Completeness



56

56

From Weak Completeness to Strong Completeness

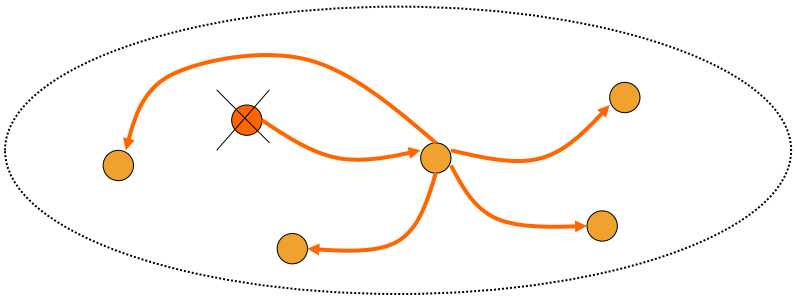


Weak Completeness: Failed node is detected by **some** correct process

57

57

From Weak Completeness to Strong Completeness

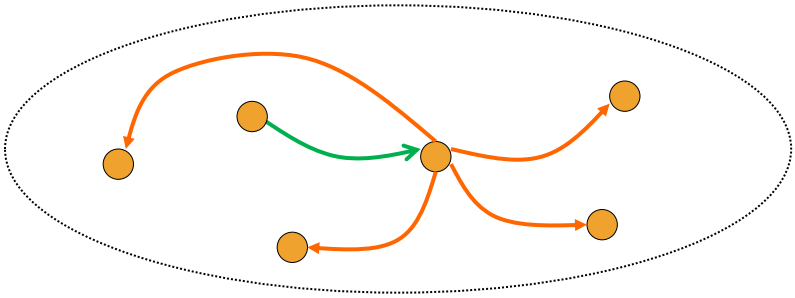


Strong Completeness: Initial detector notifies the other correct nodes

58

58

From Weak Completeness to Strong Completeness



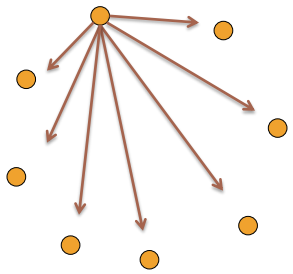
Accuracy: "Failed" node eventually notifies correct processes of their mistake

59

59

Consensus with Eventually Strong Detector

- Round i (repeat until final value):
  - Coordinator is process (i mod N)
  - Broadcast to all processes for their value
  - Wait for majority to respond (assume < N/2 fails)

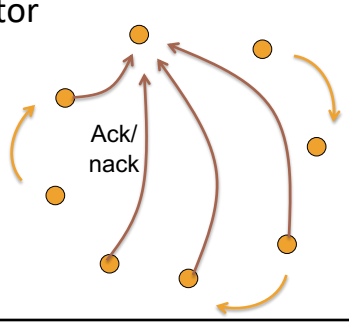


60

60

Consensus with Eventually Strong Detector

- Round i:
  - Each correct process may ack with its value…
  - …or believe coordinator has failed, i += 1
  - …must still send nack for termination of coordinator

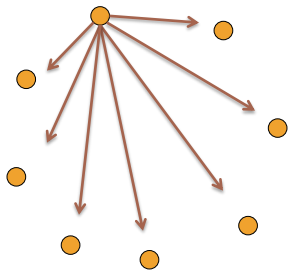


61

61

Consensus with Eventually Strong Detector

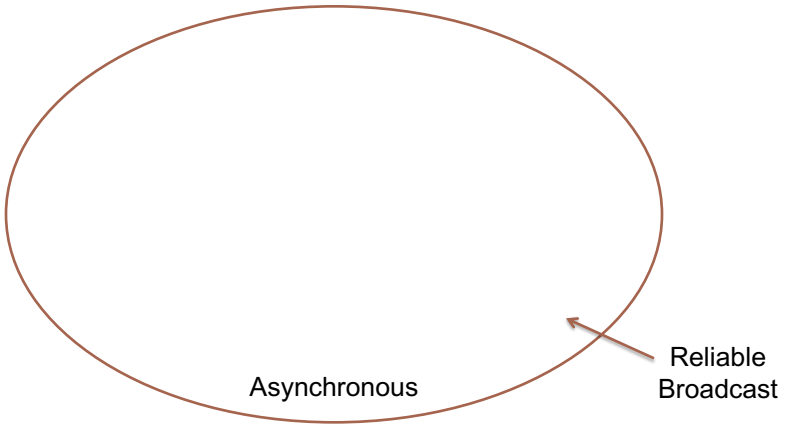
- Termination:
  - Eventual weak accuracy: *Some* coordinator will *eventually* be seen correct by all correct processes
  - With majority vote, broadcast final value



62

62

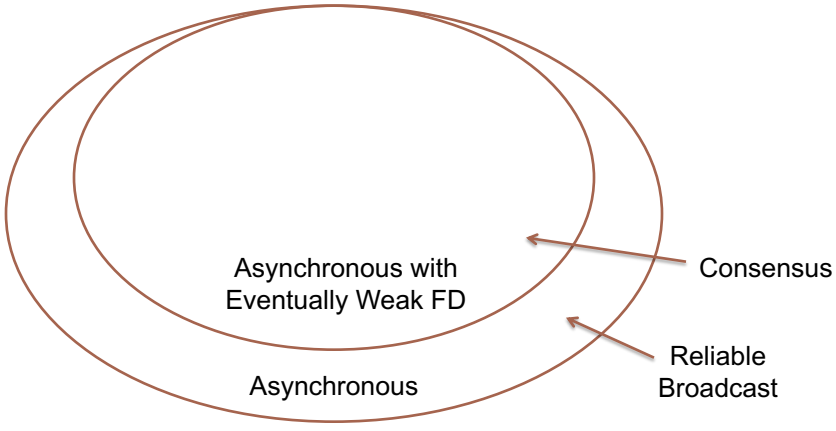
Power of Failure Detectors



63

63

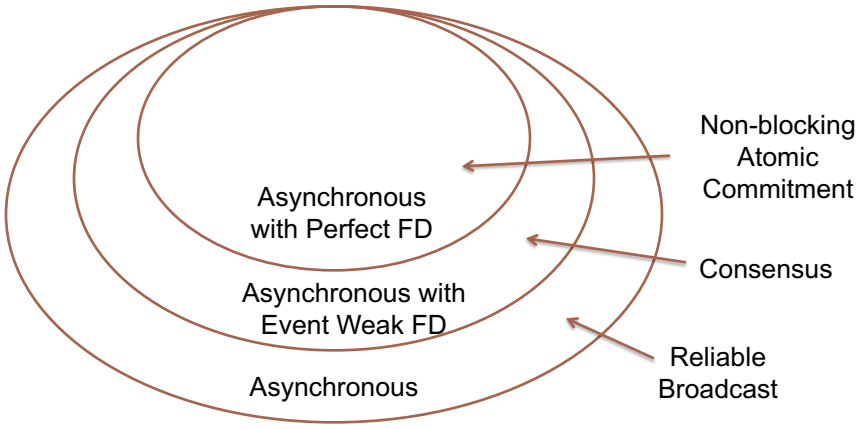
Power of Failure Detectors



64

64

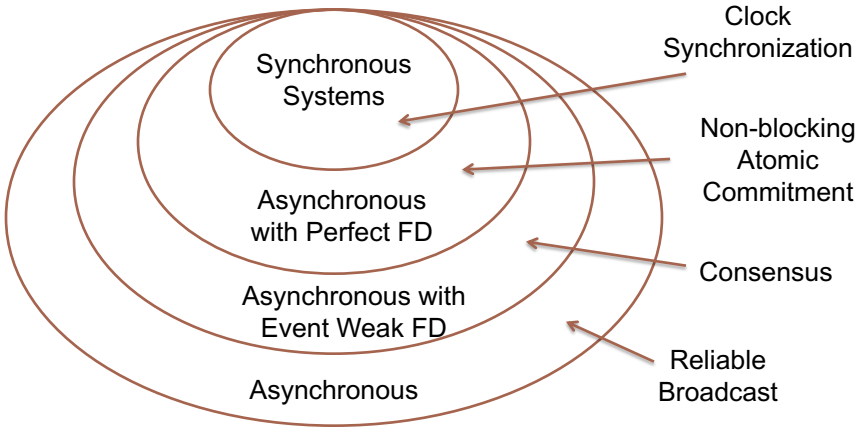
Power of Failure Detectors



65

65

Power of Failure Detectors



66

66

How to Proceed?

- Approximate ◊W with sufficiently long timeouts
  - Problem: latency
- Use probabilistic protocols
  - Solve consensus with high probability
- Change problem e.g. to group membership
  - Process group approach, false positives ok
- Accept consensus protocol that terminates with high probability
  - Paxos algorithm

67