Insider Threats to Information Systems
CS-590T
Lecture 2: Failure Detectors

Department of Computer Sciences
Purdue University
Failure Detectors

- Impossibility result: it is impossible to design an asynchronous fault-tolerant consensus algorithm, even when only one process can crash. (FLP85)
- Proof Idea: It is shown how an infinite sequence of events can be constructed such that the algorithm never terminates (stays indecisive forever).
- The impossibility comes from the fact that in an asynchronous system, it is impossible to distinguish between a faulty-process and a slow process.
Circumventing the Impossibly Result

- Design asynchronous consensus algorithms that are always correct, but do not always terminate.
- Use randomization, protocol always terminate, but is not deterministic.
- Use quorum systems: collection of subset of servers. Each pair of which intersect, the quorum operates on behalf of the system and consistency is preserved by the intersection property.
Failure Detectors Properties

• **Accuracy**: the failure detector makes no mistakes when labeling processes as faulty.

• **Completeness**: the failure detector eventually suspects every process that actually crashes.

• Based on their properties failure detectors can be classified in several classes.

• A failure detector $D$ is reducible to another one $D'$ if there is a distributed algorithm that can transform $D$ in $D'$.

• Two failure detectors are equivalent if they are reducible to each other.
Completeness

• **Strong Completeness**: There is a time after which every process that crashes is suspected by **EVERY** correct process.

• **Weak Completeness**: There is a time after which every process that crashes is permanently suspected by **SOME** correct process.
Accuracy

- **Strong Accuracy**: No process is suspected before it crashes.
- **Weak Accuracy**: Some correct process is never suspected. (at least one 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 never suspected by any correct process.
Failure Detection Implementation

• **Push**: processes keep sending heartbeats “I am alive” to the monitor. If no message is received for awhile from some process, that process is suspected as being dead.

• **Pull**: monitor asks the processes “Are you alive?”, and process will respond “Yes, I am alive”. If no answer is received from some process, the process is suspected as being dead.

• What are advantages and disadvantages of these two models?
Unreliable Failure Detectors

- A process is suspected that it was faulty, that can be true or false, if false the list of alive processes is modified.
- Failure detectors can add/remove processed from the list of suspects; different processes have different lists.
- The assumptions are that:
  - After a while the network becomes stable so the failure detector does not make mistakes anymore.
  - In the unstable period, the failure detector can make mistakes.
Metrics

- Detection time
- Mistake recurrence time
- Mistake duration
- Average mistake rate
- Query accuracy probability
- Good period duration
- Network load
Today’s Presentations

- Details on why it is impossible to solve consensus in an asynchronous model, when one process can crash.
- How to build scalable distributed failure detectors
Recommended Reading

- On scalable and efficient distributed failure detectors.
- I. Gupta, T.D. Chandra and German S. Goldszmidt.
- Impossibility of Distributed Consensus with One Faulty Process. M.J. Fischer, N.A. Lynch and M.S. Paterson