ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Midterm Review

CS 539 / ECE 526

Distributed Algorithms

Overview

- Models of distributed computing
- Fundamental problems and algorithms
  - Correctness proofs and efficiency
- Negative results

Models of Distributed Computing

- **Message passing** vs. shared memory
- Generic graph vs. complete graph
- Lockstep, synchrony, asynchrony, partial sync
- No fault vs. crash fault vs. Byzantine fault
- Deterministic vs. randomized
- Cryptography (signatures) vs. not

Algorithms Covered

- Basic graph algorithms
  - Flooding broadcast, broadcast/convergecast using a spanning tree, building a spanning tree, BFS, DFS*
- Clock synchronization
  - 2 procs, *n* procs using reference, or using averaging*
- Synchronizers: local (2), global, hybrid*
- Logical clocks: Lamport, vector
- Consensus:
  - Flooding broadcast, Dolev-Strong, transformations
  - Reliable/consistent bcast, graded agreement, Ben-Or
  - Paxos, ~~PBFT~~

Remark

- Broadcast is an overloaded term in this class
  - Spanning tree broadcast
  - Flooding broadcast (without faults)
  - Flooding broadcast (with crash)
  - Dolev-Strong broadcast
  - Reliable broadcast, Bracha broadcast
  - Consistent broadcast
  - Graded broadcast
- Do *not* say "broadcasts x" if you mean to say "sends x to all"

For Each Algorithm

- What (combination of) models does it assume?
- Why is it correct?
- What is the efficiency?

- *What purpose does each step serve?
- *Is it optimal in terms of …

Impossibilities Covered

- Clock synchronization skew bound
- Synchronizer fault tolerance
- Two general impossibility
- Consensus round and communication bounds
- Consensus fault bounds (many)

For Each Impossibility

- What (combination of) models does it require?

  I.e., When does it apply?

- When does it *not* apply?

- *Is it known to be tight? Due to which algo?

- *How is it proved? What is the intuition?

Fault Bounds Summary

- Async deterministic: $f = 0$
  - Broadcast, agreement, total-order bcast, replication
- Psync or randomized async
  - Broadcast: $f = 0$
  - Agreement, total-order broadcast, or replication: crash: $f < n/2$, Byzantine: $f < n/3$
- Sync
  - Crash: $f < n$ for all four problems
  - Byzantine no signature: $f < n/3$ for all four problems
  - Byzantine with signature
    - $f < n$ for broadcast and total-order broadcast
    - $f < n/2$ for agreement and replication

Fault Bounds Better Summary

- Byzantine agreement: $f < n/2$
- Byzantine replication: $f < n/2$
- Byzantine bcast/agreement w/o sig: $f < n/3$
- Async deterministic agreement: $f = 0$
- Psync broadcast: $f = 0$
- Psync crash agreement: $f < n/2$
- Psync Byzantine agreement: $f < n/3$

Psync Agreement Fault Bound

- Crash: $f < n/2$
  - Proof: Two groups $|P| \le f$ and $|Q| \le f$
  - Scenario I: P non-faulty & receive v, Q crash
    - P eventually commit v due to validity
  - Scenario II: Q non-faulty & receive v', P crash
    - Q eventually commit v' due to validity
  - Scenario III: Both non-faulty, P receive v, Q receive v'
    GST sufficiently large → Both think the other crashed
    - P commit v, Q commit v'

Psync Agreement Fault Bound

- Byzantine: $f < n/3$
  - Proof: Three groups $|P| \le f$, $|Q| \le f$, $|R| \le f$
  - Scenario I: P/R non-faulty & receive v, Q crash
    - P eventually commit v due to validity
  - Scenario II: Q/R non-faulty & receive v', P crash
    - Q eventually commit v' due to validity
  - Scenario III: P non-faulty & receive v, Q non-faulty & receive v', R Byzantine behave towards P like in I and towards Q like in II. GST sufficiently large.
    - P cannot distinguish from I, commit v
    - Q cannot distinguish from II, commit v'