Reading List

- Tanenbaum Section 7.2
Group Membership Problem

- Agreement on the membership of a group of cooperating processes in a distributed system

- Consistent system-wide view of the operational members in the presence of
  - processor (or process) failure
  - processor (or process) join
  - processor (or process) departures
  - communication failure

- Group Membership Service maintains membership of a distributed system on behalf of processes that compose it
GMP - Informal Definition

- All operational members see the same sequence of view transitions

  Linear order on system view changes

- See Figure 1

- Several research papers formally define the problem - beyond the scope
Figure 1 - GM View Changes

P1 → P1,P4 → P1,P4,P2 → P1,P4,P2,P3 → P4,P2,P1
What is difficult about this problem?

- **Main Challenge in Asynchronous Systems:**
  - It is difficult to distinguish between a process that has crashed and a process that is very slow.
  - *Perceived failure* of processors due to message loss or communication delay.
  - Timeouts? it is impossible to determine with absolute certainty whether a processor has crashed in an asynchronous distributed system.

- **Related issues:**
  - Initial system startup - bootstrap problem
  - Multiple concurrent failures
  - Coordinator failure/partition handling
  - Precise meaning of a consistent view
  - Heartbeat/hardware multicast support
Coordinator-based Approach

- Used in many group communication systems including ISIS, Horus, RTCAST, Amoeba, ...
- Unique identifier for each member, e.g., IP addr + processid
- Linear ordering of member ids
- Designate a coordinator (or manager) for maintaining and disseminating membership information
- Two cases:
  - member (non-coordinator) failure: 2-phase protocol
  - coordinator failure: 3-phase protocol
Case 1: Non-coordinator Failure

failure

Commit(Q)

remove(Q)

coordinator

Phase I

Phase II
Case 2: Coordinator Failure

Q

coordinator

X

interrogation

propose

commit

Y

Phase I

Phase II

Phase III

Z

failure
Coordinator Failure

Heartbeat
Partition Handling

- Primary partition group: e.g. ISIS system
  - majority partition continues - membership of subsequent group should overlap with the membership of current group
  - minority group suspends
  - potential for singleton groups and lack of progress

- Allow partitions and re-merge: e.g. Transis system

- Allow non-overlapping simultaneous groups
Service Model

Node 1

Node 2

Node 3

Membership Daemon

Membership Daemon

Membership Daemon
Group Membership and Distributed Consensus

Why doesn’t impossibility results of distributed consensus apply to group membership problem?

What do if timeout does not reliably detect a process failure?

- Distributed consensus is strictly stronger: termination, agreement, validity and integrity properties.

- Group membership allows “operational” processes to force out another “presumed failed” process.