CSE 120, UCSD Winter 2019

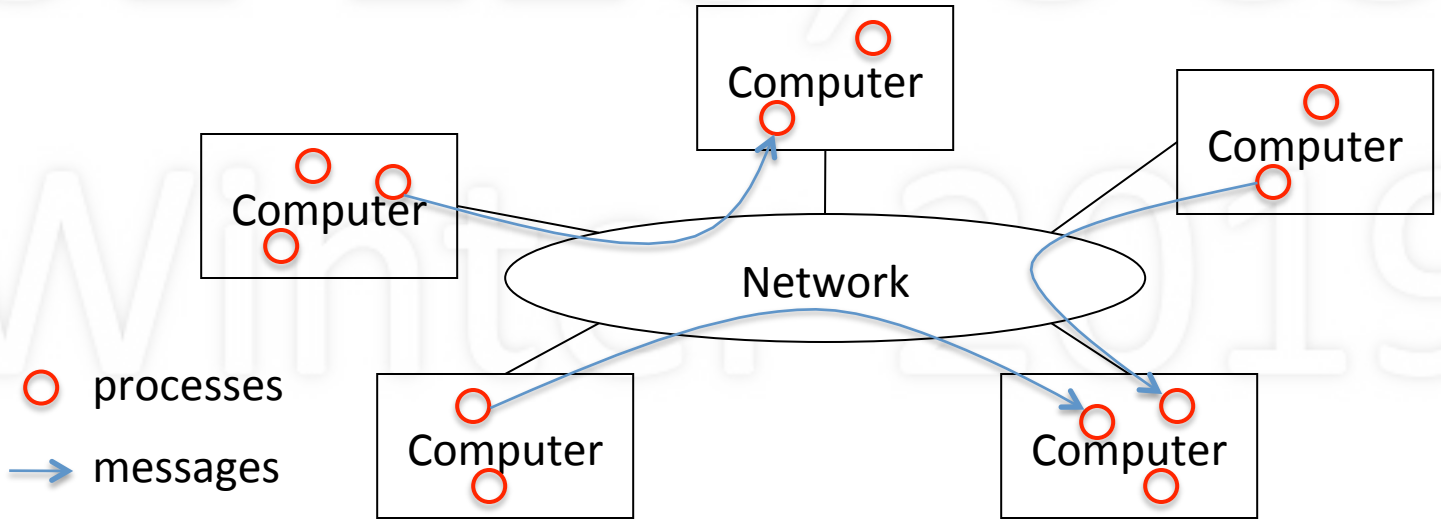
CSE 120: Principles of Operating Systems
Lecture 16: Distributed Systems

Prof. Joseph Pasquale
University of California, San Diego

March 13, 2019

© 2019 by Joseph Pasquale

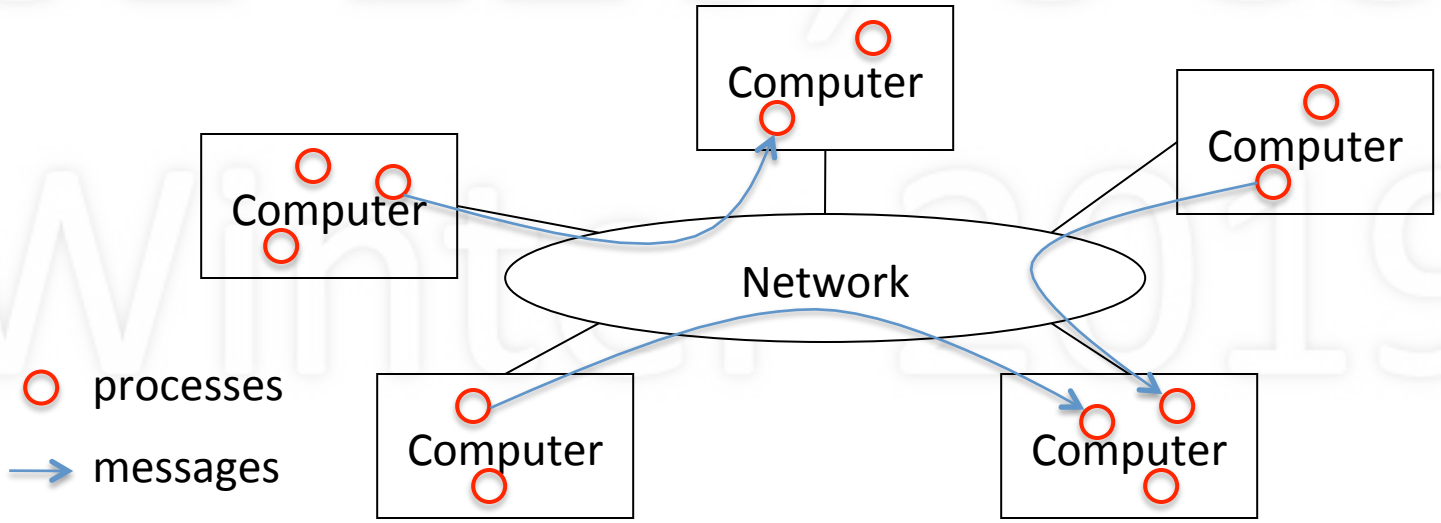
What is a Distributed System?



- Cooperating processes in a computer network
- Degree of integration
  - Loose: Internet applications, email, web browsing
  - Medium: remote execution, remote file systems
  - Tight: process migration, distributed file systems

© 2019 by Joseph Pasquale

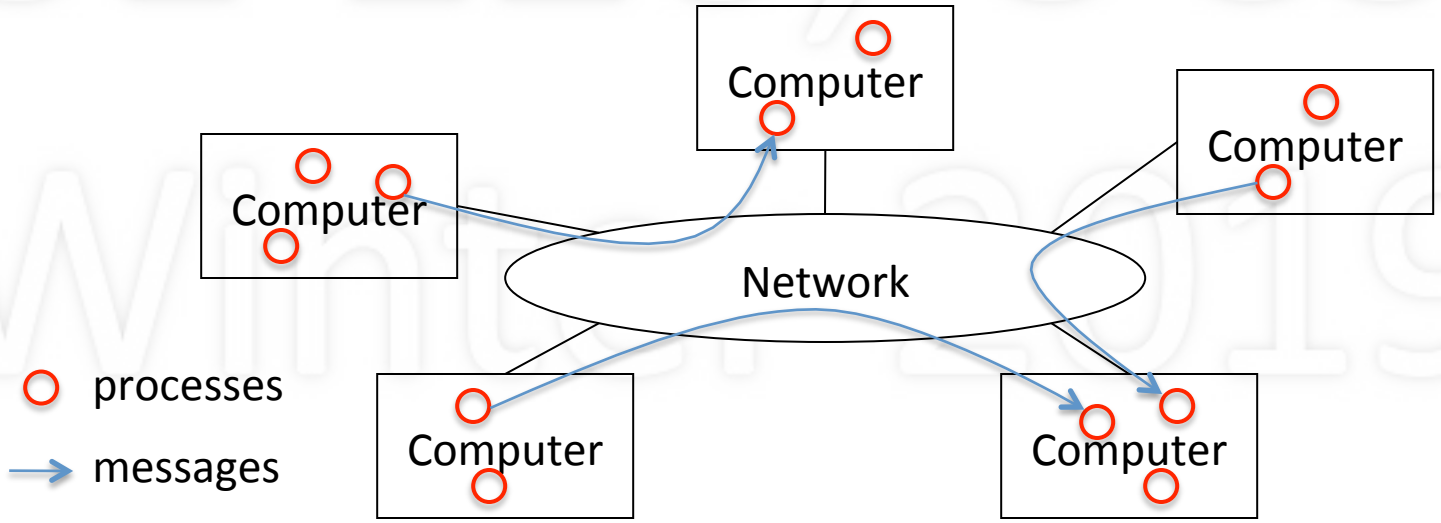
Advantages



- Speed: parallelism, less contention
- Reliability: redundancy, fault tolerance, "NSPF"
- Scalability: incremental growth, economy of scale
- Geographic distribution: low latency, reliability

© 2019 by Joseph Pasquale

Disadvantages

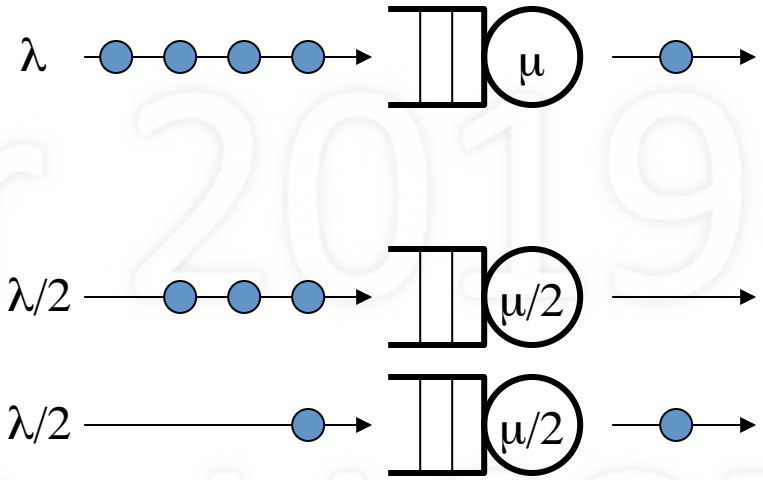


- Fundamental problems of decentralized control
  - State uncertainty: no shared memory or clock
  - Action uncertainty: mutually conflicting decisions
- Distributed algorithms are complex

© 2019 by Joseph Pasquale

Is Distribution Better?

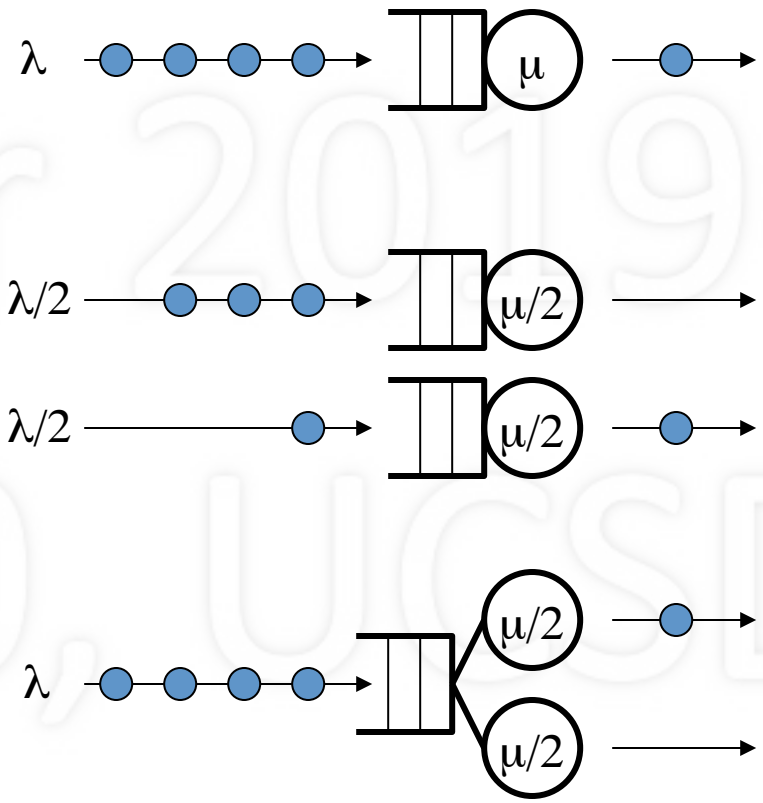
- Single fast server with single queue
- Multiple slower servers with separate queues



© 2019 by Joseph Pasquale

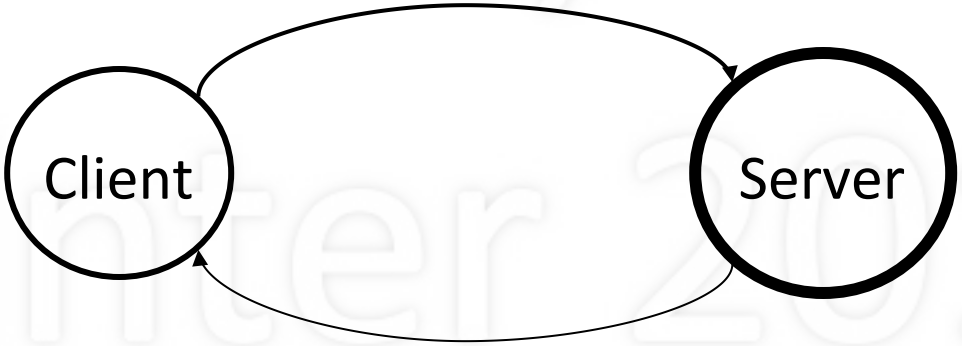
Is Distribution Better?

- Single fast server with single queue
- Multiple slower servers with separate queues
- Multiple slower servers, single queue
- Little's Law: $N = \lambda W$



© 2019 by Joseph Pasquale

The Client/Server Model



- Client
  - Short-lived process that makes requests
  - "User-side" of application
- Server
  - Exports well-defined requests/response interface
  - Long-lived process that waits for requests
  - Upon receiving request, carries it out (may spawn)

© 2019 by Joseph Pasquale

Peer-to-Peer

- A peer talks directly with another peer
  - No intermediary (e.g., central server) involved
  - Symmetric (unlike asymmetric client/server)
- In actuality, may be dynamic client/server
  - A requests file from B; A acts as client, B as server
  - C can now request file from A; A acts as server

© 2019 by Joseph Pasquale

Distributed Algorithms

- Event ordering
- Leader election
- Mutual exclusion

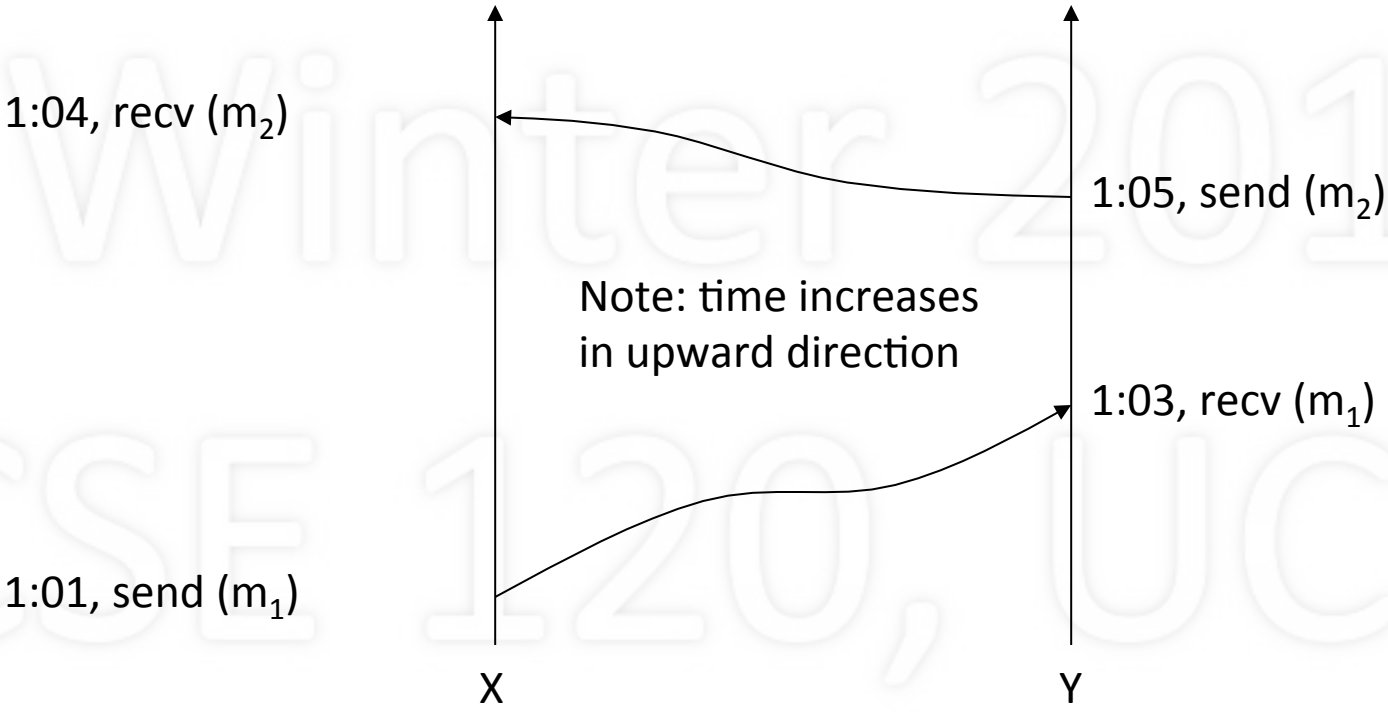
CSE 120, UCSD Winter 2019
CSE 120, UCSD Winter 2019
CSE 120, UCSD Winter 2019

Event Ordering

- Order events given no shared clock/memory
- Happened-before relation: →
  - A, B events in same process and A before B: A → B
  - A is a send event, B is a receive event: A → B
  - If A → B and B → C, then A → C
- Implementation
  - Timestamp all events based on local clock
  - Upon receiving a message, advance local clock
  - Resolve ties by ordering machines

© 2019 by Joseph Pasquale

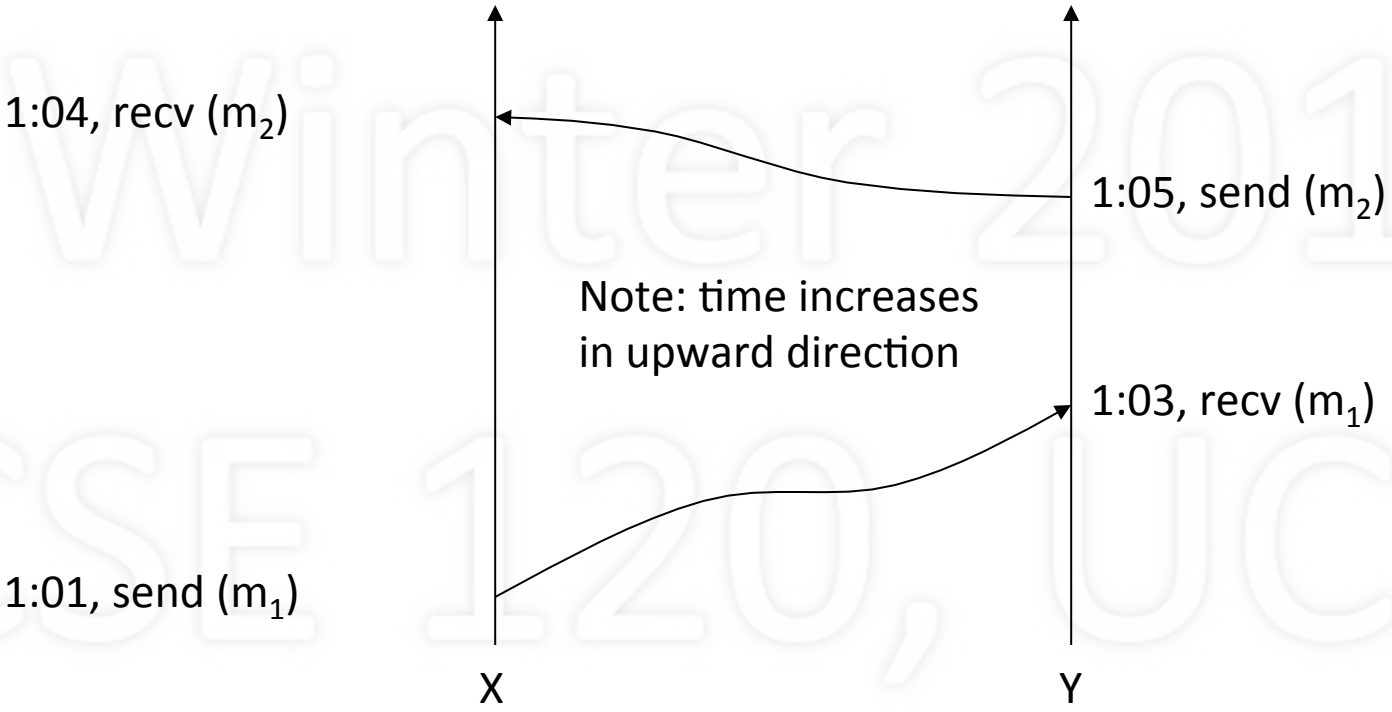
Event Ordering: Example



1:01 send (m_1); 1:03 recv (m_1); 1:05 send (m_2); 1:04 recv (m_2);

© 2019 by Joseph Pasquale

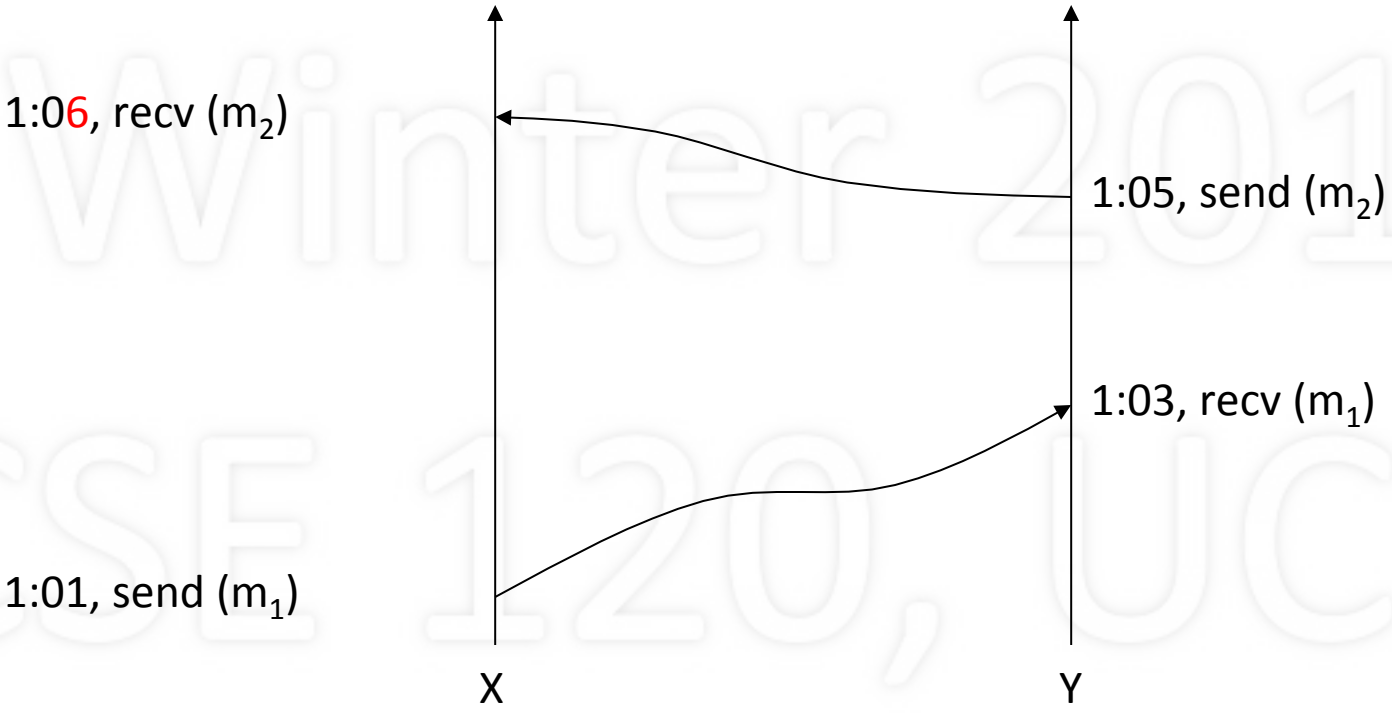
Event Ordering: Example



1:01 send (m_1); 1:03 recv (m_1); 1:05 send (m_2); 1:04 recv (m_2);

© 2019 by Joseph Pasquale

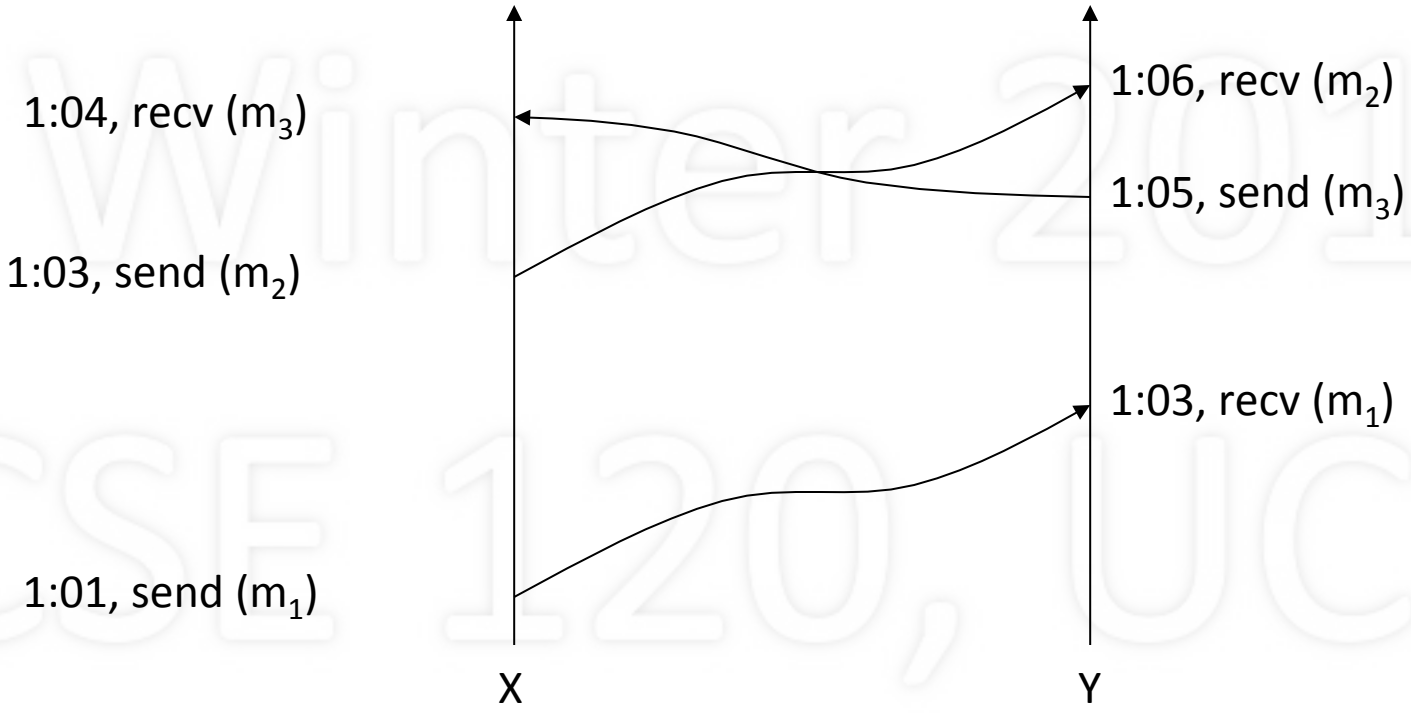
Event Ordering: Example



1:01 send (m_1); 1:03 recv (m_1); 1:05 send (m_2); 1:06 recv (m_2)

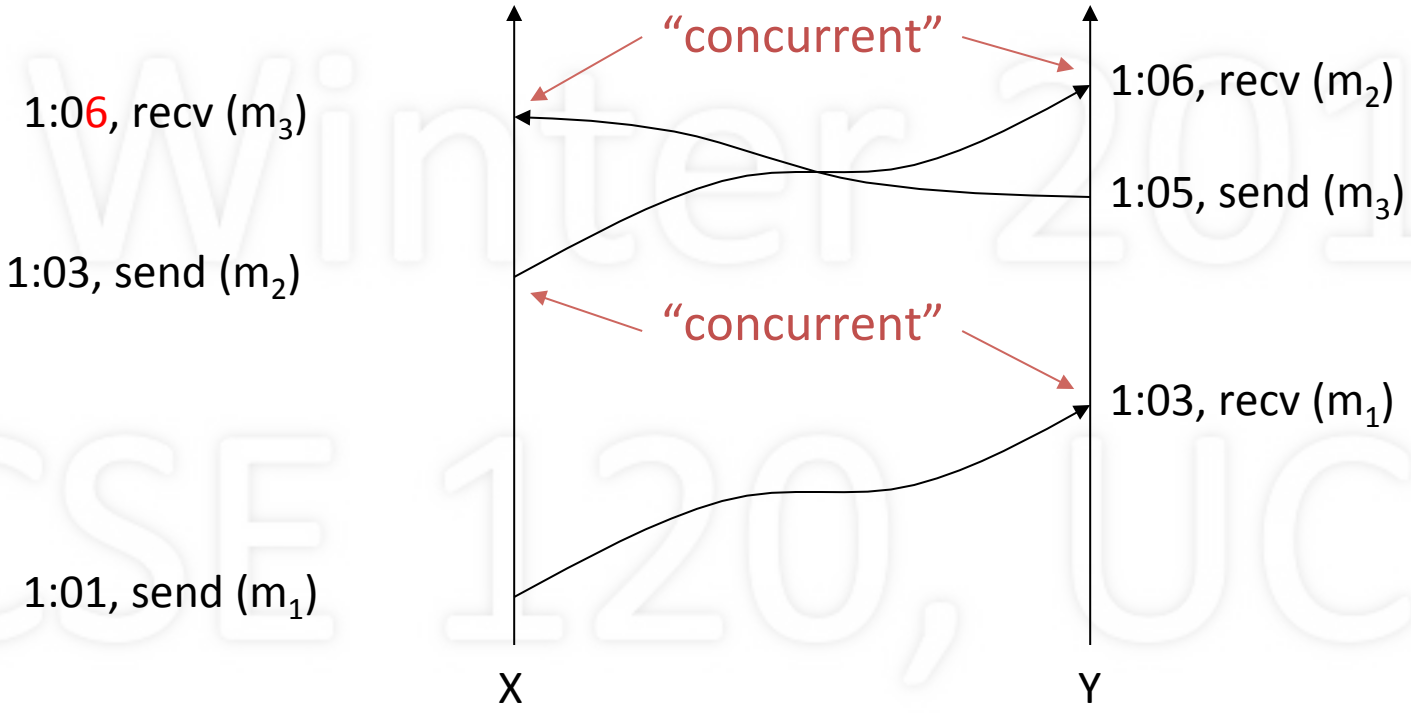
© 2019 by Joseph Pasquale

Event Ordering: Another Example



© 2019 by Joseph Pasquale

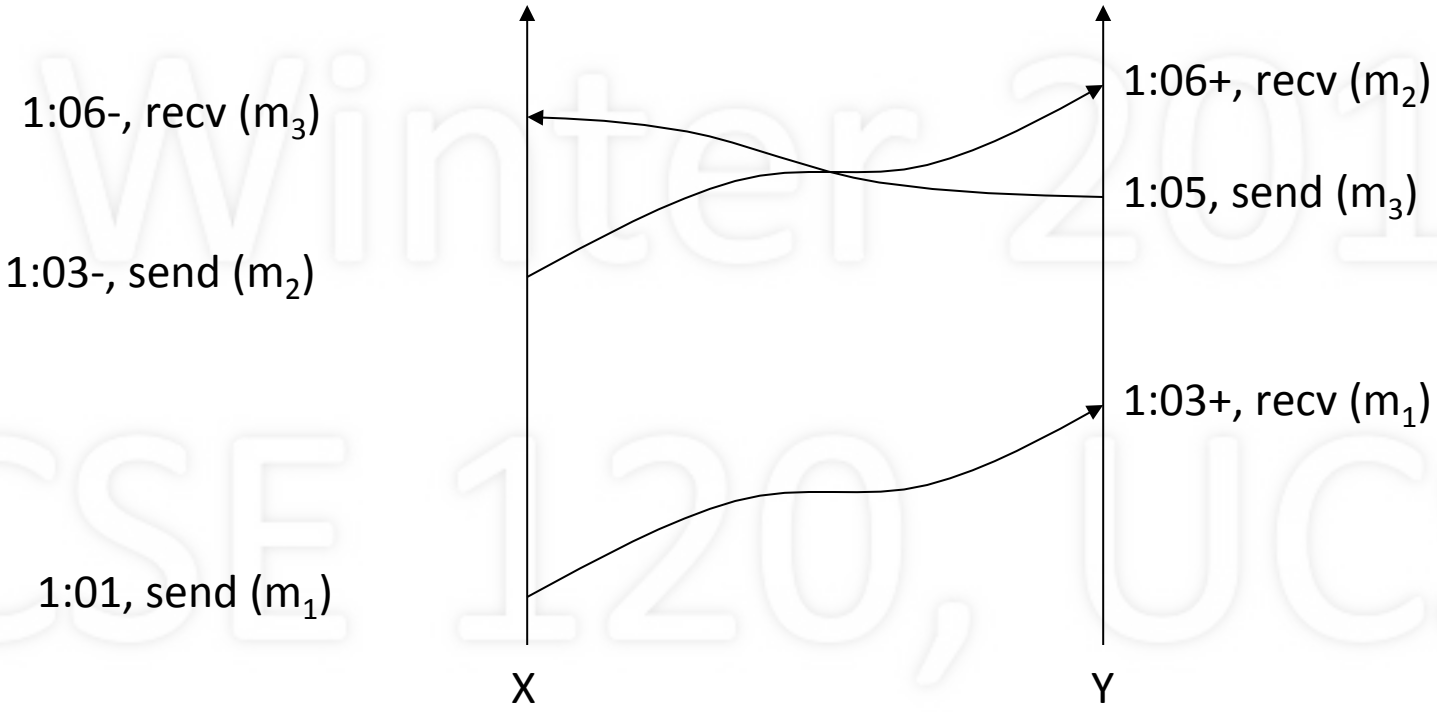
Event Ordering: Another Example



1:01 send (m_1); 1:03 recv (m_1); 1:03 send (m_2); 1:05 send (m_3); 1:06 recv (m_3); 1:06 recv (m_2)

© 2019 by Joseph Pasquale

Event Ordering: Another Example



1:01 send (m_1); 1:03- send (m_2); 1:03+ recv (m_1); 1:05 send (m_3); 1:06- recv (m_3); 1:06+ recv (m_2)

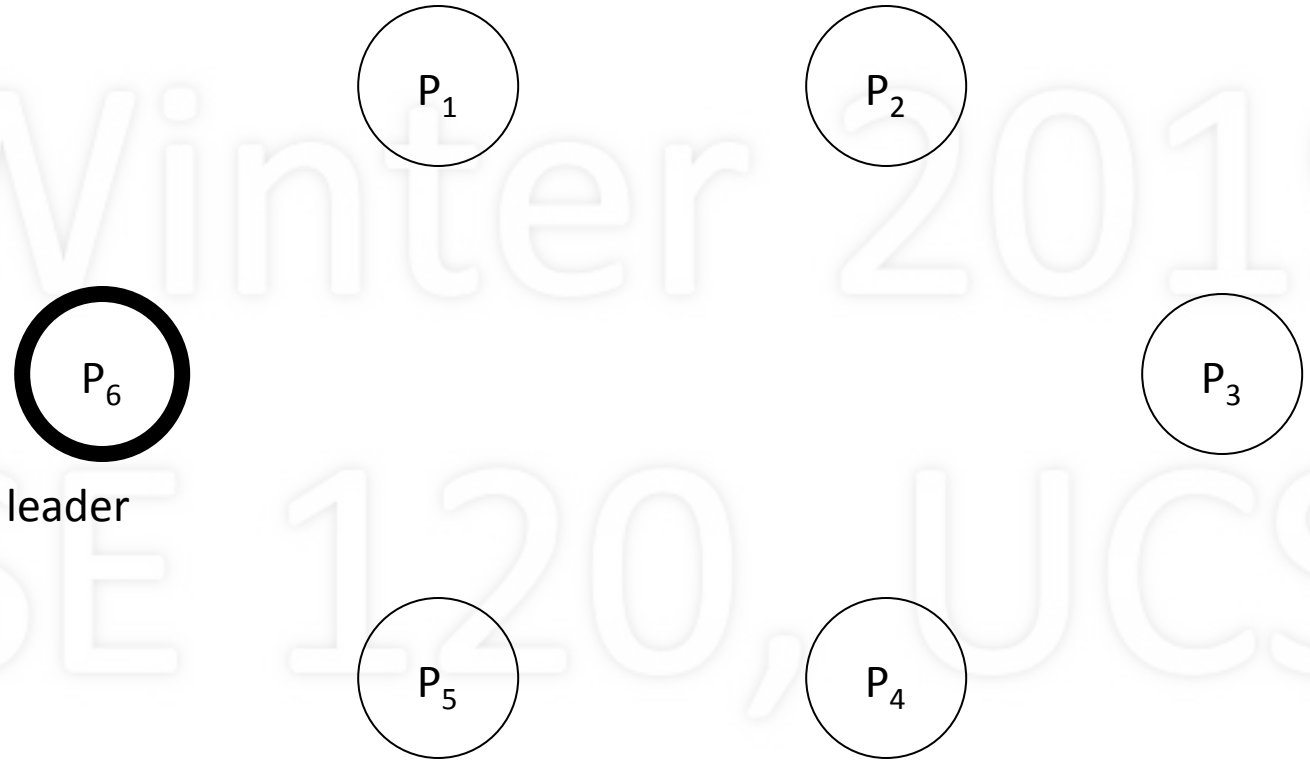
© 2019 by Joseph Pasquale

Leader Election

- Many distributed algorithms rely on leader
- Need to determine if leader exists; if not, elect
- Bully algorithm (elects leader L)
  - Every process is numbered (priority): P_1, P_2, …
  - P_j sends request to L, no reply; tries to elect itself
  - P_j sends "Can I be leader?" to all $P_{k>j}$
  - No replies, P_j sends to all $P_{i<j}$, "I am leader", done
  - If some $P_{k>j}$ replies, P_j lets P_k try to elect itself
  - If no message from P_k, P_j tries to elect itself again

© 2019 by Joseph Pasquale

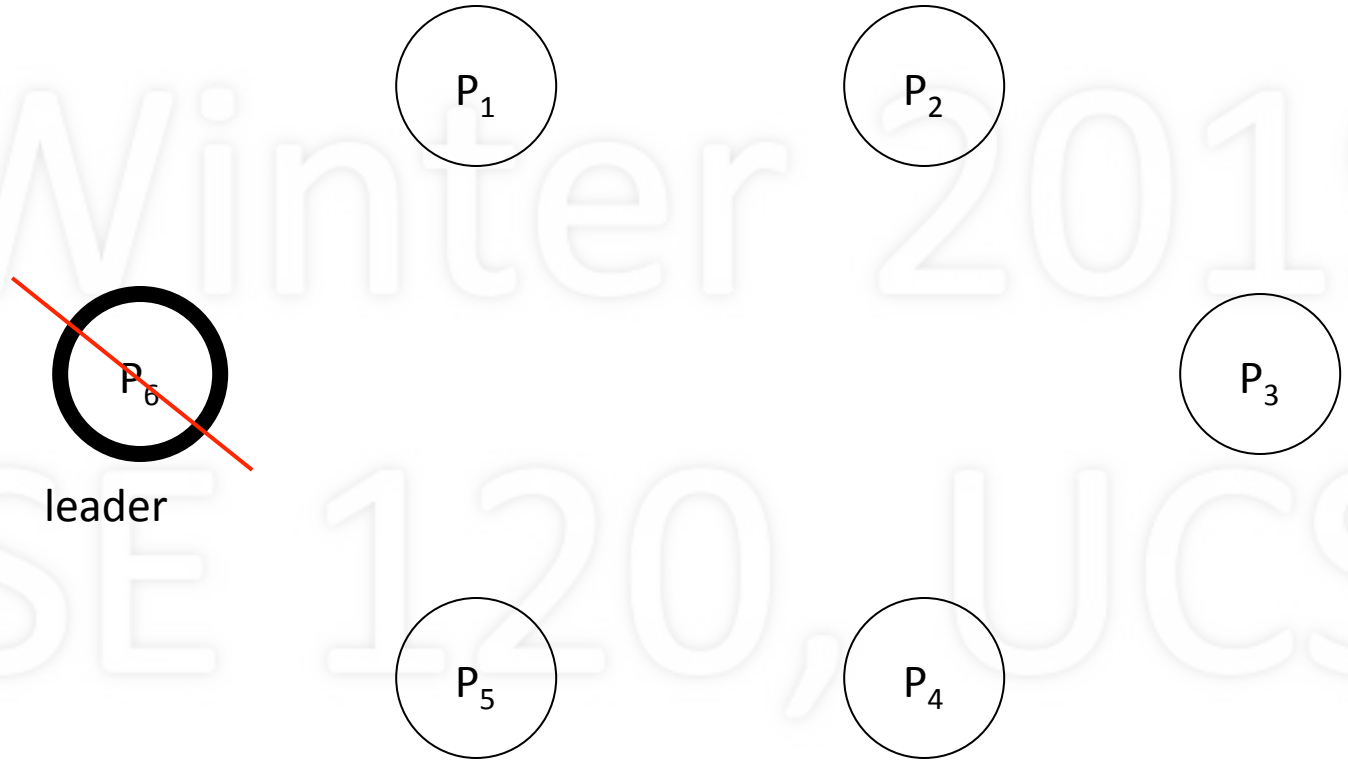
Leader Election



Assume P_6 is leader (highest number)

© 2019 by Joseph Pasquale

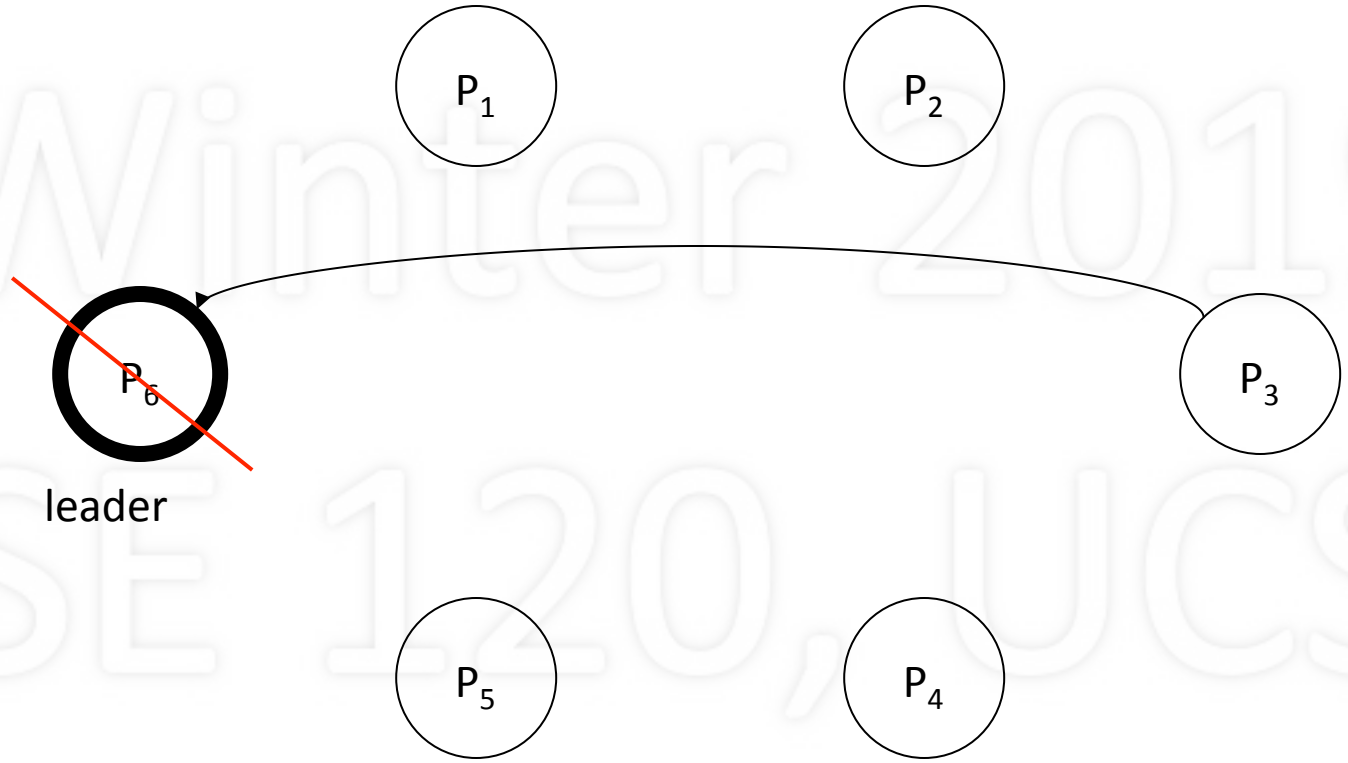
Leader Election



P_6 fails

© 2019 by Joseph Pasquale

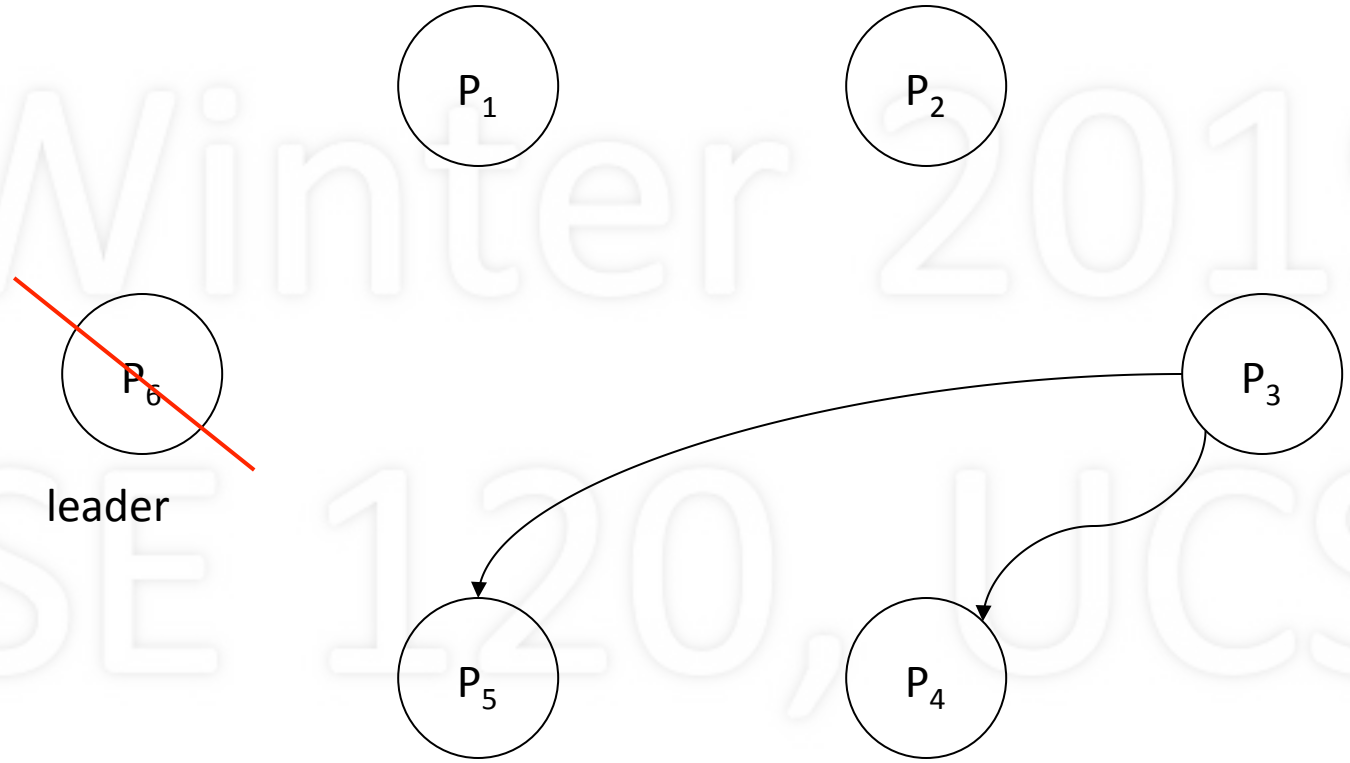
Leader Election



P_3 tries to contact leader; no reply

© 2019 by Joseph Pasquale

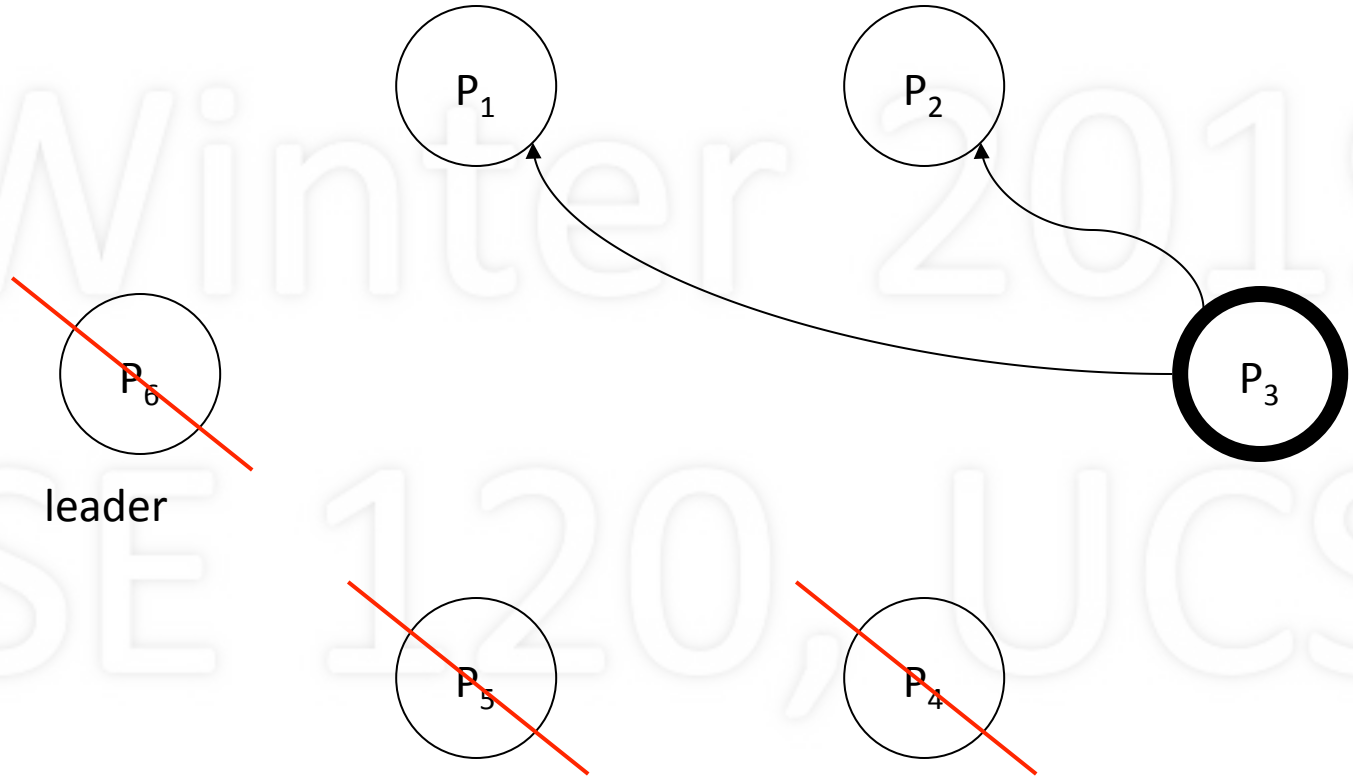
Leader Election



P_3 sends election message to P_4 and P_5

© 2019 by Joseph Pasquale

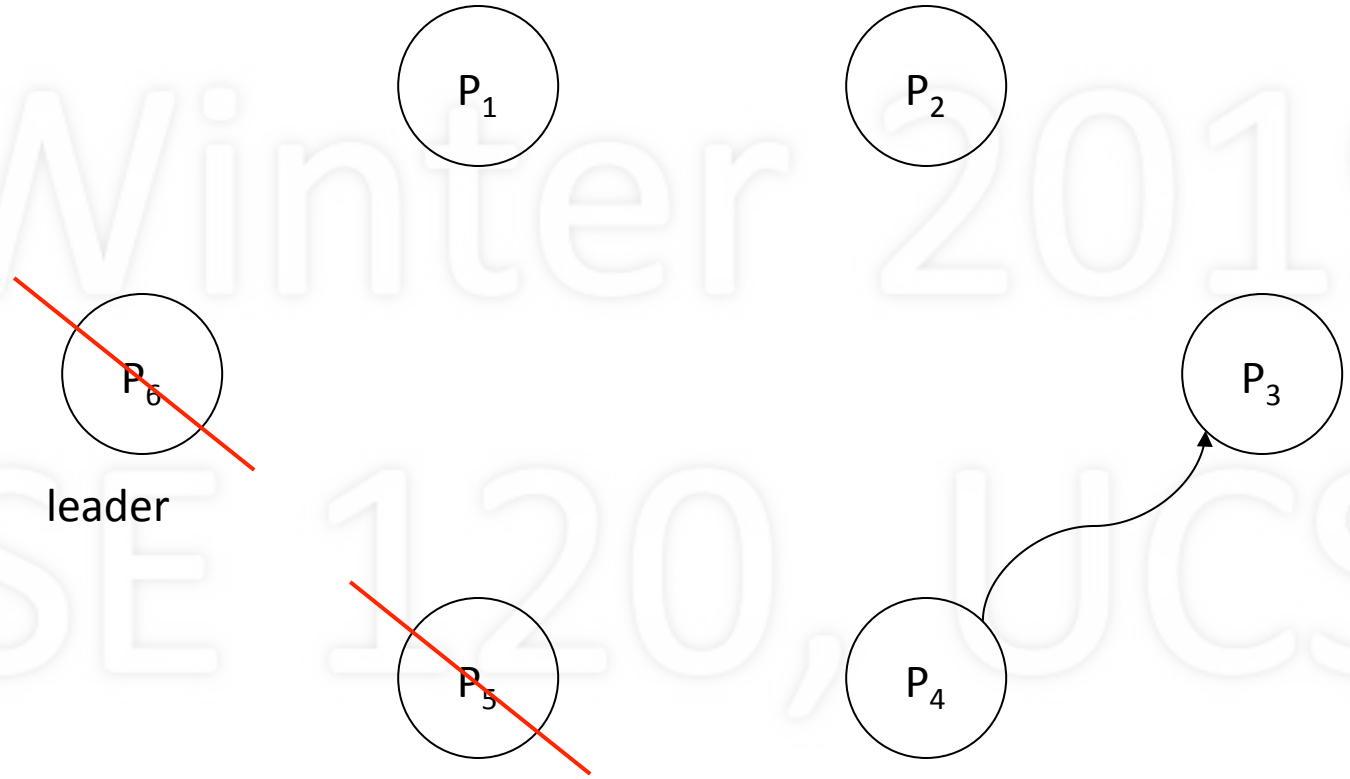
Leader Election



If P_3 were to not hear from P_4 and P_5, it would tell P_1 and P_2 "I'm the leader"

© 2019 by Joseph Pasquale

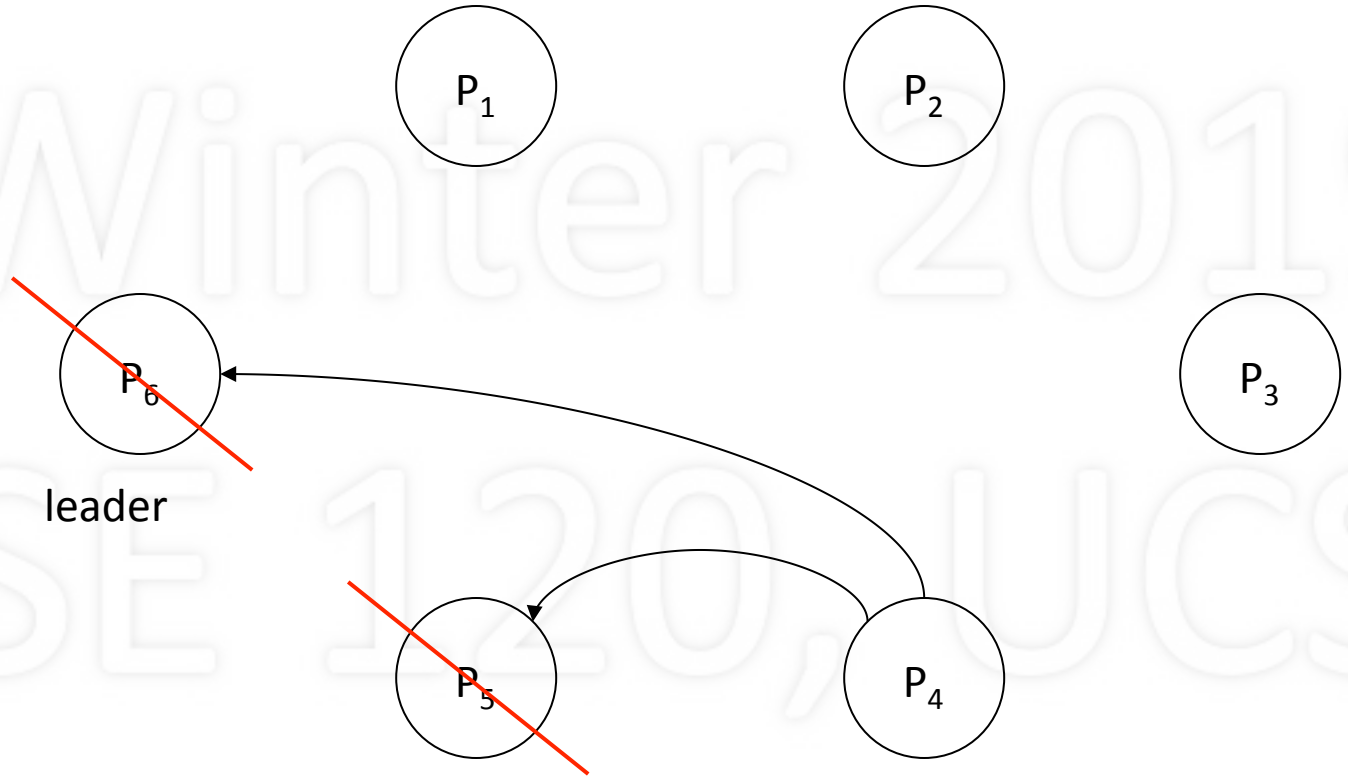
Leader Election



If only P_4 is alive, it replies to P_3

© 2019 by Joseph Pasquale

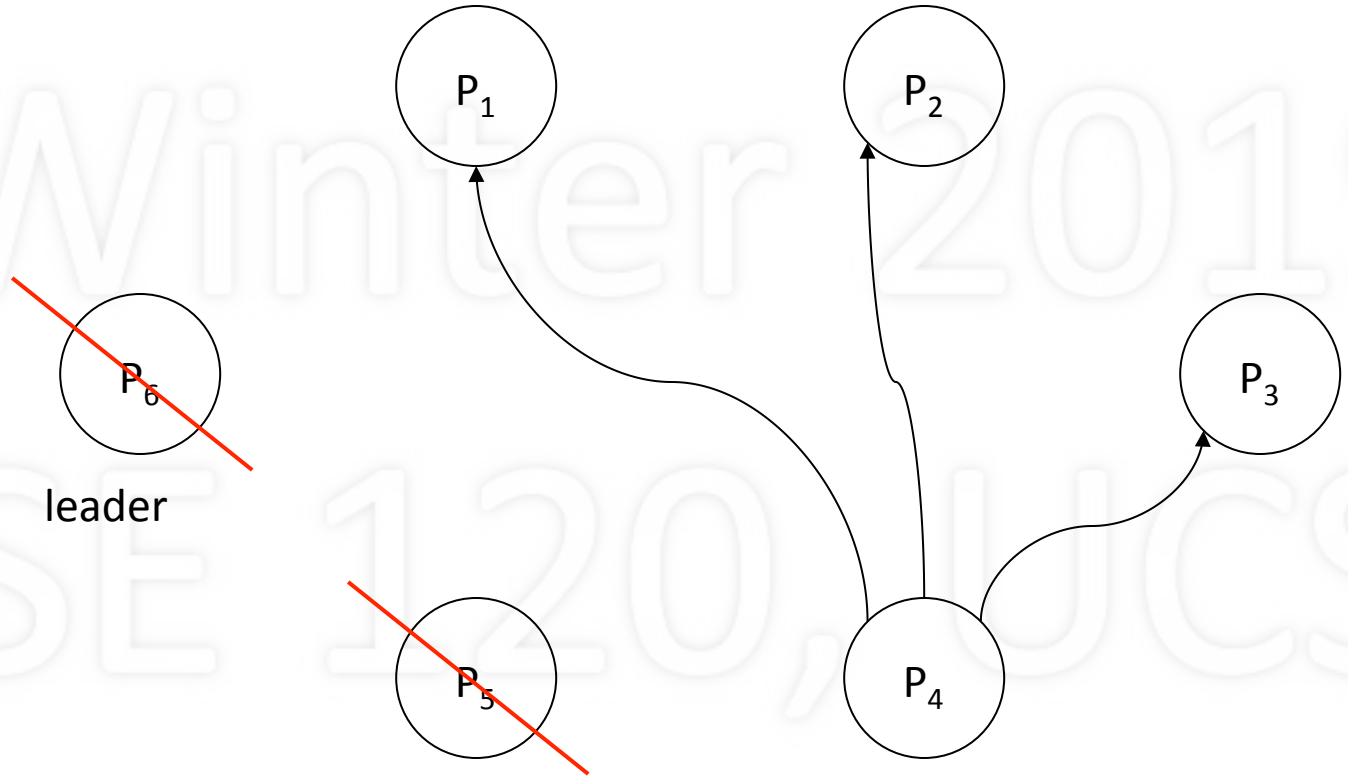
Leader Election



P_4 tries to elect itself as leader, sending messages to P_5 and P_6; no replies

© 2019 by Joseph Pasquale

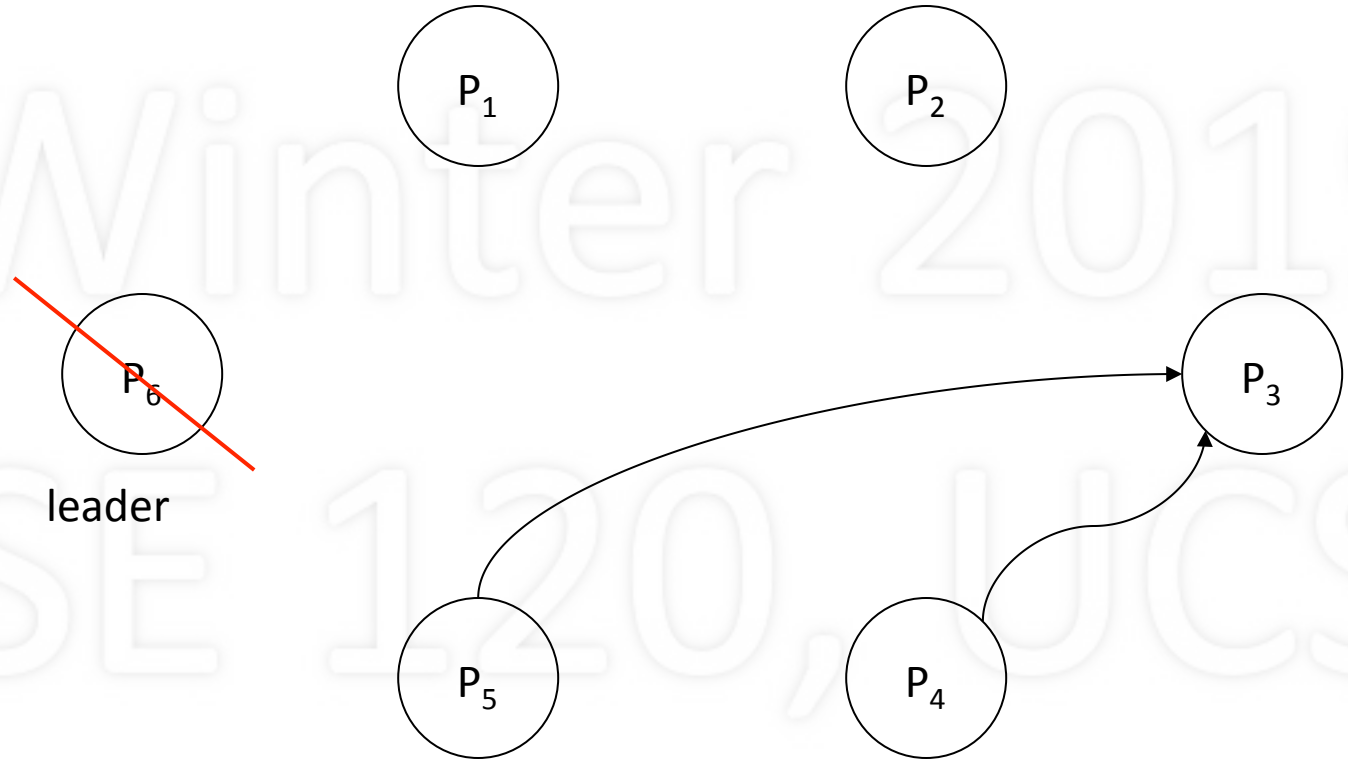
Leader Election



P_4 declares itself as leader, telling P_1, P_2, and P_3

© 2019 by Joseph Pasquale

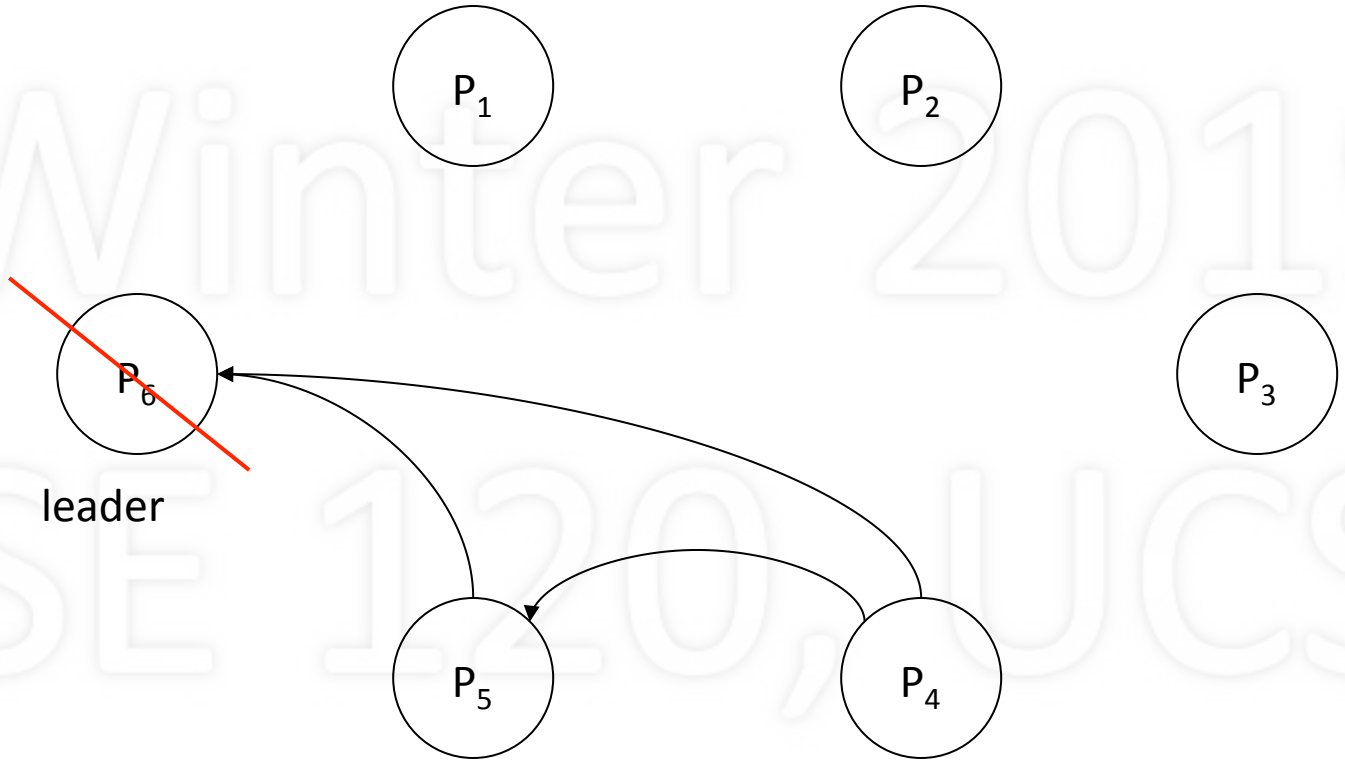
Leader Election



If both P_4 and P_5 are alive, they reply to P_3

© 2019 by Joseph Pasquale

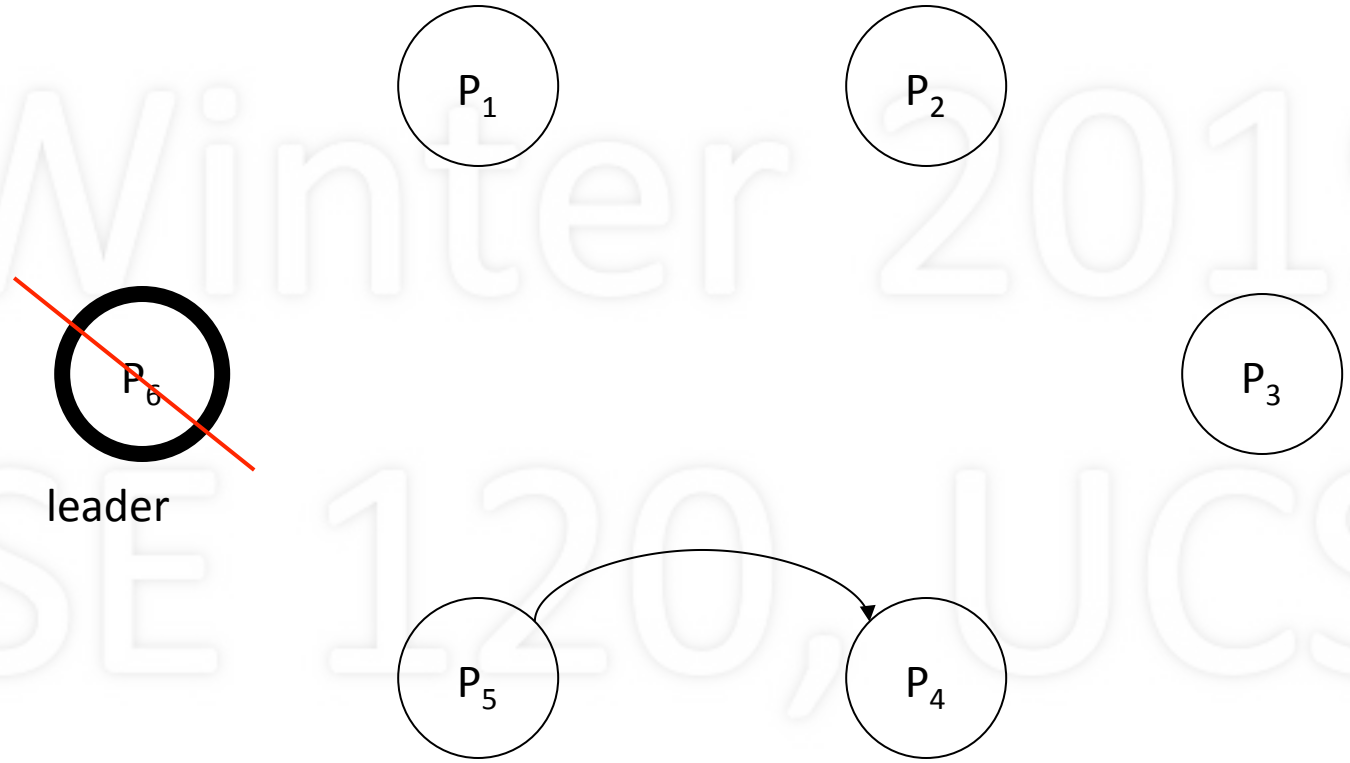
Leader Election



Both P_4 and P_5 try to elect themselves

© 2019 by Joseph Pasquale

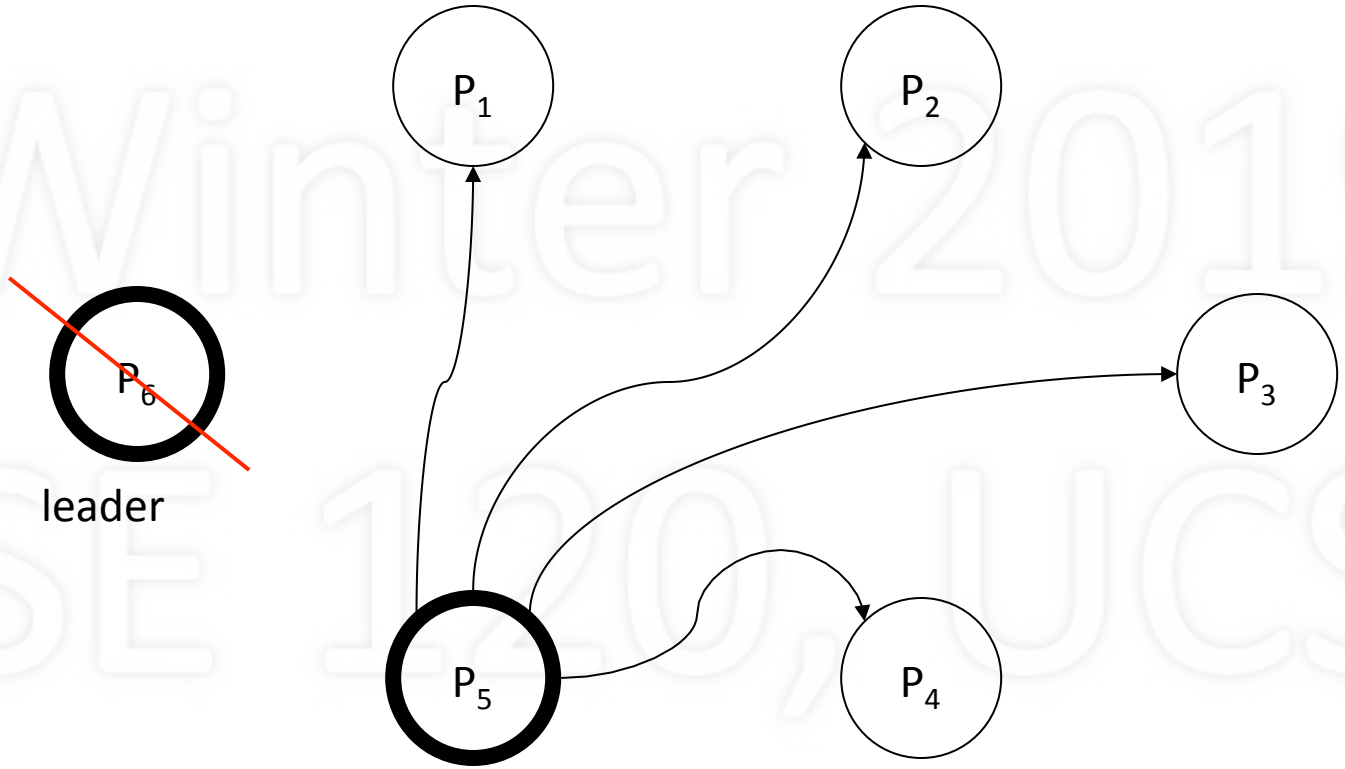
Leader Election



P_4 gets reply from P_5, but no reply from P_6 to P_5

© 2019 by Joseph Pasquale

Leader Election



P_5 tells everyone below it, "I'm the leader"

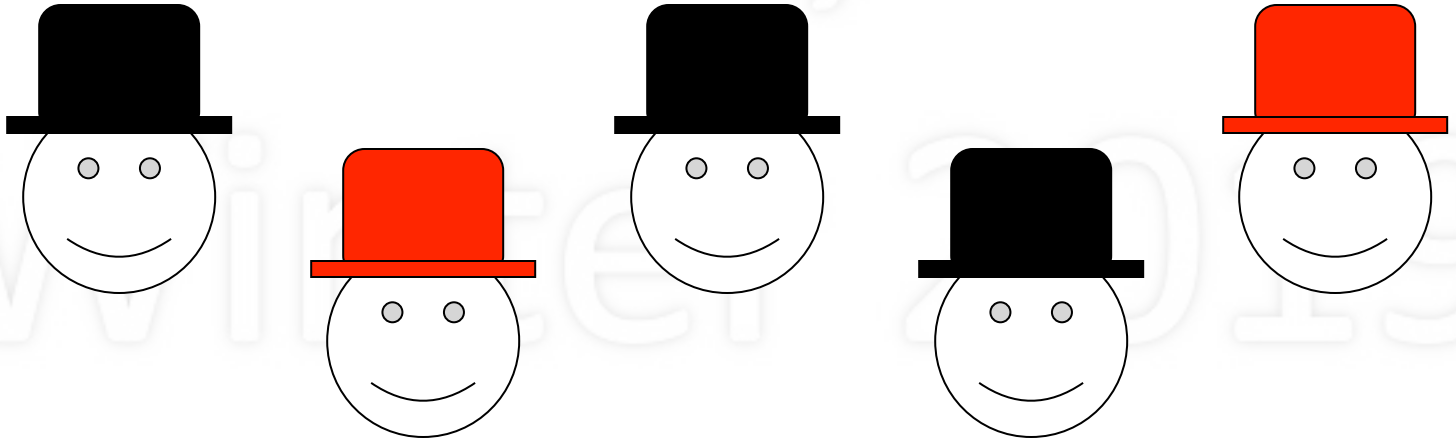
© 2019 by Joseph Pasquale

Mutual Exclusion

- Centralized approach
  - Single process acts as coordinator server
  - Request, reply (to allow entrance), release
- Distributed approach
  - Process sends time-stamped request to all others
  - Waits until it receives replies from all (ok to enter)
  - Enter critical section (may get requests, defers)
  - Upon exiting, responds (to release) to all deferred
  - Use timestamps to order "simultaneous" requests

© 2019 by Joseph Pasquale

Common Knowledge Problem

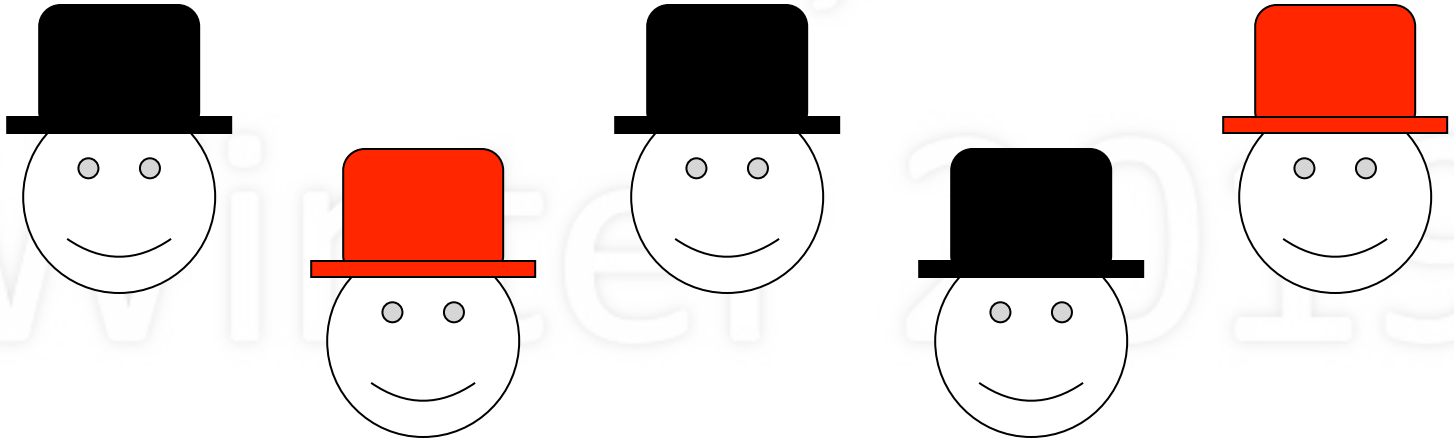


- n people, each wearing red or black hat
- Each can see the others' hats, but not own hat
- Hint: "At least one of you has a black hat"
  - Told publicly: all hear it, all know that all hear it, …

CSE 120, UCSD Winter 2019

© 2019 by Joseph Pasquale

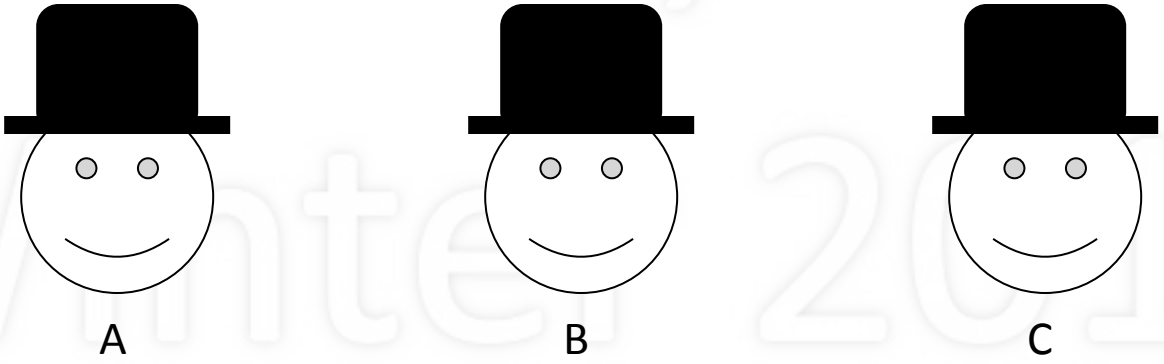
Common Knowledge Problem



- They are asked (publicly): “Is your hat black?”
- Possible replies (made simultaneously)
  - “Yes”
  - “No”
  - “I don’t know”

CSE 120, UCSD Winter 2019

Example: 3 Wearing Black Hats



All told: “At least one of you has a black hat”

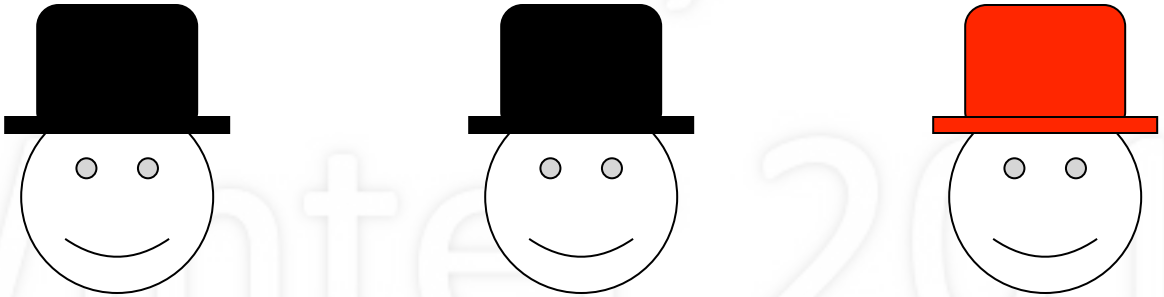
Q1. “Is your hat black?” All: “I don’t know”

Q2. “Is your hat black?” All: “I don’t know”

Q3. “Is your hat black?” All: “Yes!”

© 2019 by Joseph Pasquale

Questions

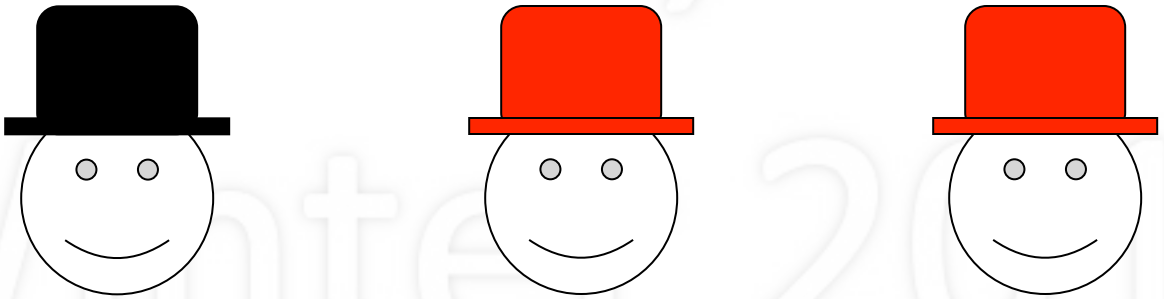


- How do each figure out they have black hat?
- Why is the hint necessary?
  - After all, each can see someone with a black hat
  - Does it have to be public?
- What is relevance to distributed systems?

CSE 120, UCSD Winter 2019

© 2019 by Joseph Pasquale

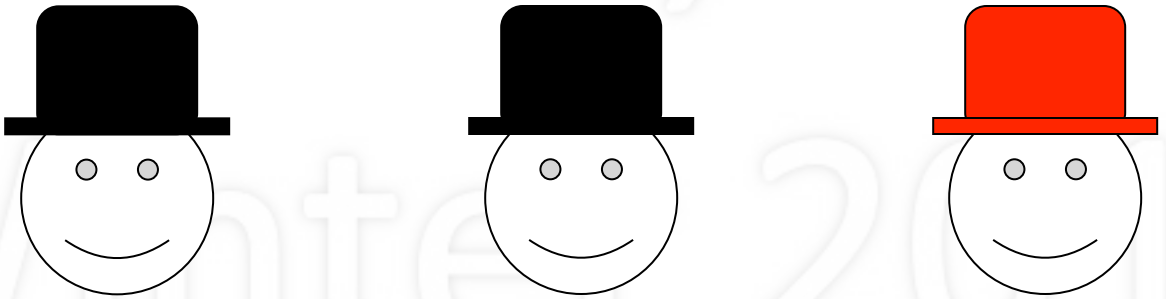
What If One Black Hat?



- A sees 2 red hats
- Can answer "Yes" after 1st question
  - Was told "at least one of you has a black hat"

CSE 120, UCSD Winter 2019

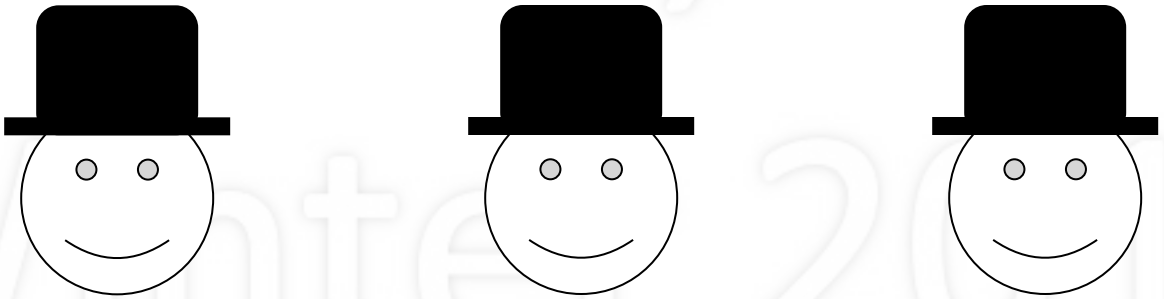
What If Two Black Hats?



- A sees 1 red hat and 1 black hat
- All answer "I don't know" to the first question
- A: If I had red, B would have said "Yes"
- So, after 2nd question, A (and B) can say "Yes"

© 2019 by Joseph Pasquale

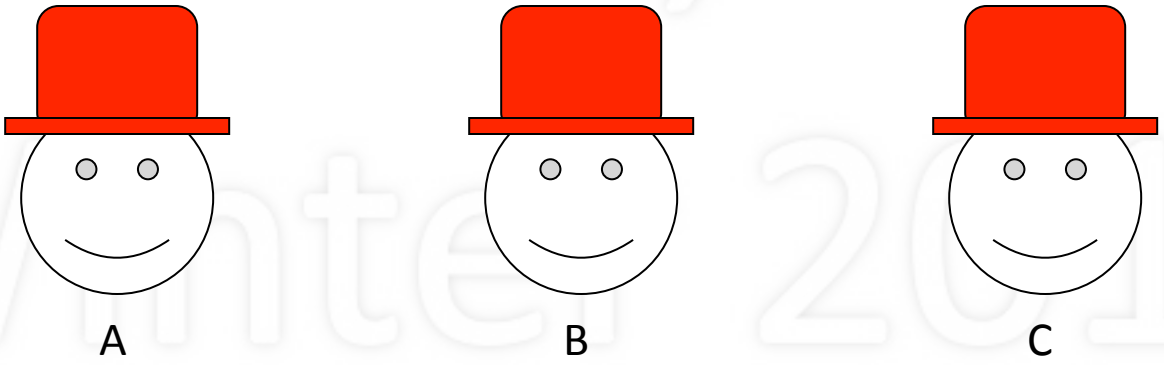
What If Three Black Hats?



- All see 2 black hats
- All answer "I don't know" to 1st, 2nd questions
- After 3rd question, each can say "Yes"
  - If I had red, others will have said "Yes"
  - Therefore, I must have black

© 2019 by Joseph Pasquale

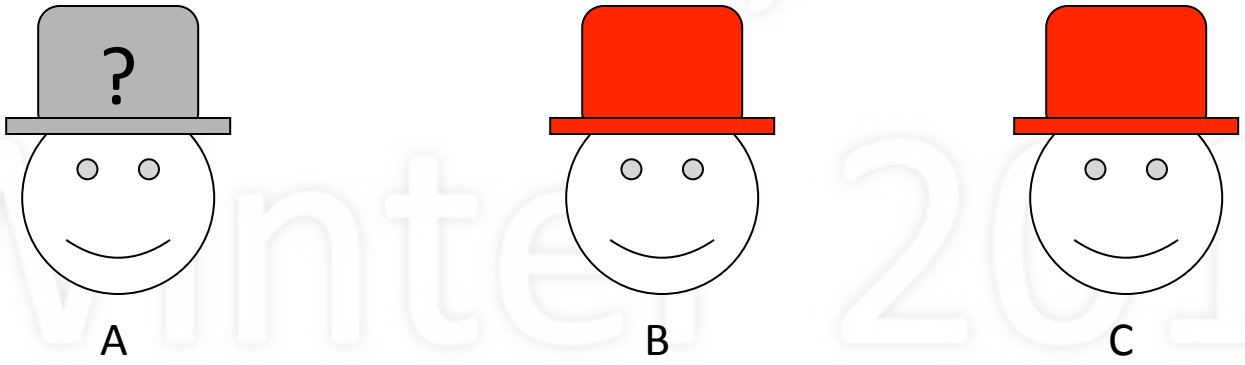
Why is the Hint Necessary?



- Hint allows each to tell difference between
  - all red hats
  - one black hat
- Provides information about *what others know*

© 2019 by Joseph Pasquale

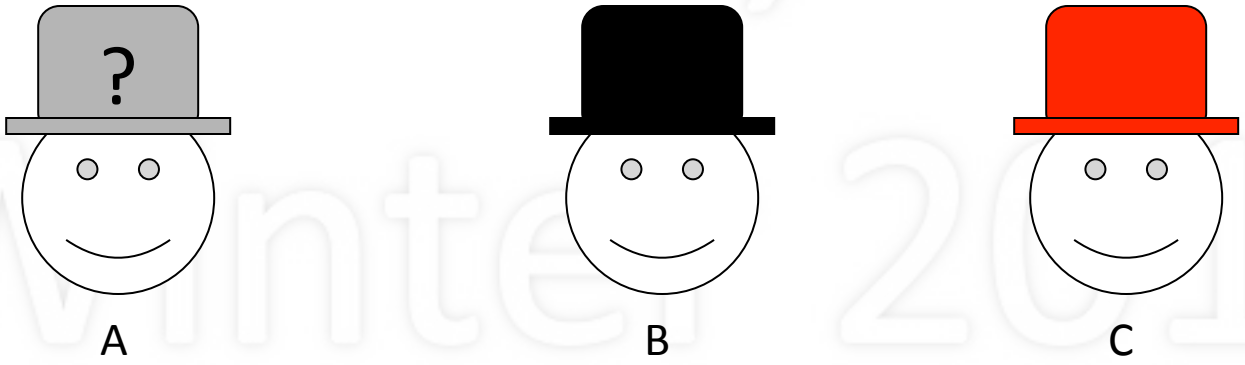
1st Time: What Does A Know?



- That B and C have red hats (by observation)
- That there is at least one black hat (the hint)
  - since everyone knows hint
- Deduction: A deduces it must have black hat

© 2019 by Joseph Pasquale

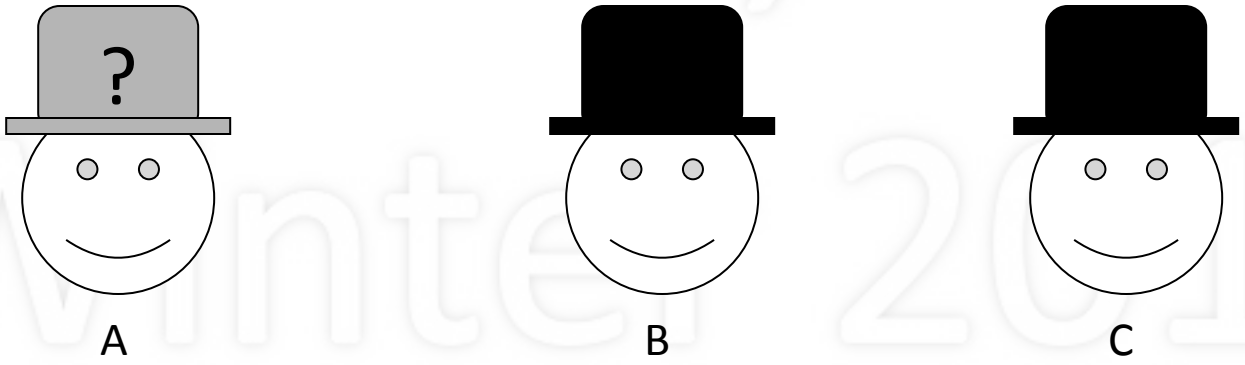
2nd Time: What Does A Know?



- That B has black, C has red hat (by observation)
- That there is at least one black hat (the hint)
- That B and C know the hint
  - since everyone knows that everyone knows hint
  - Thus, if A had red, B would have said "yes" after 1st Q
- Deduction: A deduces it must have black hat

© 2019 by Joseph Pasquale

3rd Time: What Does A Know?



- That B and C have black hats (by observation)
- That there is at least one black hat (the hint)
- That B, C know hint, and that B, C know A knows
  - since everyone knows that everyone knows that everyone knows hint
- Deduction: A deduces it must have black hat!

© 2019 by Joseph Pasquale

Common Knowledge

- Knowledge of x
  - K(x) = Everyone knows x
  - $K^2(x)$: Everyone knows that everyone knows x
- Common knowledge of x
  - K(x) and $K^2(x)$ and … $K^n(x)$ as n → infinity
- Hint is common knowledge
  - Because it is announced publicly
  - Each must assume all heard it and all are smart

© 2019 by Joseph Pasquale

Textbook

- Chapter 19
  - Lecture-related: 19.1, 19.4-19.10
  - Unfortunately, very little material in textbook
  - Try to get a copy of 8th edition, or check wikipedia
  - For 8th edition (not 9th, which also has too little)
    - Lecture-related: 16.1-16.2, 17.1-17.2, Chapter 18 (all)
    - Recommended: 17.3-17.5

CSE 120, UCSD Winter 2019

Review & Research

- What is a distributed system?
- What is meant by “degree of integration”?*
- What are the advantages of a distributed system?
- What is meant by “NSPF” (i.e., no single point of failure)

CSE 120, UCSD Winter 2019

R&R

- What are the disadvantages of distributed systems?
- What is meant by “state uncertainty”?
- What is meant by “action uncertainty”?
- Given the advantages and disadvantages, why have distributed systems become popular?**

CSE 120, UCSD Winter 2019

R&R

- How do the queueing systems shown in slides 5-6 demonstrate that distribution is better?**
- What is Little's Law (explain the meaning of the parameters)?***
- What is the Client/Server Model?*
- What is the Peer-to-Peer Model?*
- How do the Client/Server and Peer-to-Peer Models differ?**

CSE 120, UCSD Winter 2019

R&R

- What is the purpose of Event Ordering, and how does it work?**
- What is the purpose of Leader Election, and how does the Bully algorithm work?**
- How is mutual exclusion implemented in a distributed system, using the centralized approach vs. the distributed approach?**

CSE 120, UCSD Winter 2019

R&R

- What is the Red Hat/Black Hat problem?**
- If there are 3 black hats, why does it take 3 questions before all can say they have a black hat?**
- If there are n black hats, how many questions are needed before all can say they have a black hat?***

CSE 120, UCSD Winter 2019

R&R

- Say there are 2 black hats and 1 red hat: how do each figure out they have a black hat?**
- Why is the hint necessary?***
- What would happen if the hint were whispered in each person's ear, rather than being told publically?***

CSE 120, UCSD Winter 2019

R&R

- What is meant by Common Knowledge?**
- What is the relevance of Common Knowledge to distributed systems?***

CSE 120, UCSD Winter 2019

© 2019 by Joseph Pasquale