11/1/10

Midterm #2
Wednesday
in OHE 122
during class

Last Time: Fully Distributed Mutual Exclusion algorithm

3 Requirements

1. Reliable messaging
2. Globally unique ID
3. Total ordering of events
   - Requires messages, from a single group member, are delivered in timestamp order

Mutual Exclusion

Message Processing Algorithm

1. A group member sends a Request msg to all group members

   Msg contains:
   - identifier of C.R.
   - timestamp
   - ID of requesting member

2. A member processing a Request

msg can be in 1 of 3 states

a. Member is not in that C.R. & has no pending request for that C.R.

b. Member is in that C.R.

c. Member has a pending request for that C.R.

3. How / when to respond

a. Send OK msg to requestor

b. Queue the request, when I exit the C.R., send OK msgs to all queued requests for that C.R.

c. Compare timestamps (& global IDs if timestamps are the same)

- If my timestamp is earlier, like 'b'
- If my timestamp is later, like 'a'

C.R. Entry Rule: A member can enter a C.R. when OK msgs have been received from all other members

Will it work?

Worst Case: All members send requests for a particular C.R. "at the same time"

# 4 members		Msg Queue
A send OK; receives	3 OK	B, C, D
B send 1 ok; "	2 OK +1	C, D
C send 2 OK; "	1 OK +1	D
D send 3 OK; "	Ø OK +1	

$O(n^2)$

\+ Correct
\+ Scales somewhat
− N points of failure

Variation #1 - Always send a reply to a Request

Member will send an OK, or a NOTOK msg upon receiving a request

+ Requesting members can use the lack of a response to start a process to determine if a member is dead.

- Still have a problem in 2 cases
  - A member dies after sending a NOTOK, before exiting the C.R.
  - A member in the C.R. dies

Variation #2 - Majority Rules

New Rules:

① A member can enter a C.R. when $N/2 + 1$ OK msgs are received

② Members only send 1 OK msg when they receive a Release msg (someone exits the C.R.)

+ $\frac{N}{2} - 1$ members can die without affecting the algorithm

- Single point of failure - member in the C.R. dies (no sending of Release msg)

Another Approach to Dist M.E: Token Ring algorithm

For each C.R., there is a Token msg
- it contains the C.R. identifier

To enter a C.R., the member must have the token msg
- the member keeps the token until they exit the C.R.

Group organization is a "logical" ring
- Members know who is in the group & the order of members

If a member does not want to enter a C.R., when receiving that token msg, they forward the msg to their neighbor

\+ Correct

— Token msgs consume network resources even with no C.R. activity

When can Token Ring Algorithm Fail?

Failure mode: Token msg lost

Where this can happen:

- network failure → reliable messaging
- my neighbor is dead → have neighbor ack
  if no ack, send to next neighbor
- * a member can die holding token
  - critical failure mode

⇓

One solution for distributed algorithms is Election Algorithms