8-27-2019

LCR Algorithm / Unidir rings

Leader = process with max id

at each process:

initially
- leader = unknown
- "send" = my id
- to my neighbor ~~s~~ transmit the message "send"

At each round:
- if a message containing an id v is received:
  - if $v >$ my id
    - "send" = v;
    - ~~send~~ transmit message send to neighbor
  - else if v = my id
    - leader = v;
  - else ~~d~~ discard received message

termination:

Leader sends a message with leader's id to all.

.
.
.

~~less~~

Correctness:

① terminator
after $2n-1$ rounds, all processes terminate

② Proof that max id is the leader:

⋮

Complexity:

time: ⌐

messages: $O(n^2)$ in the worst case

$n + (n-1) + (n-2) + \cdots$

↑ largest id goes on n hops

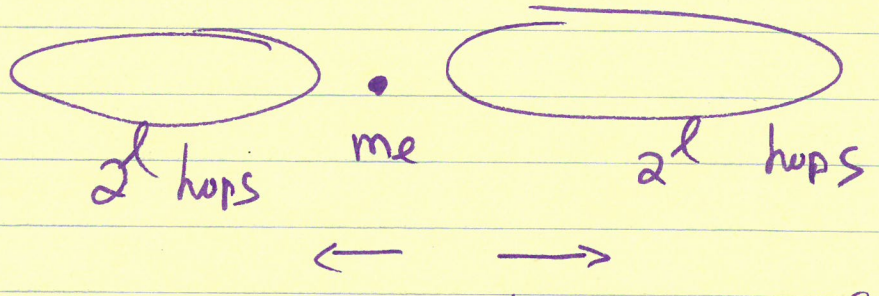
↑ 2nd largest id goes on $(n-1)$ hops . . .

Better message Complexity?

Bidirectional rings. (All links are bidirectional)

HS Algorithm

Phases: in phase ℓ ($\ell = 0, 1, 2 \ldots$)
explore 2^ℓ hops in both directions.



token in forward (out) dir

initially,
$\ell = 0$,
each process participates in the run for the leader.

During phase ℓ: (for process i):

if i is still running:
{ assemble a token (i, 2^ℓ, out)
and send to both neighbors
⋮
if both tokens come back after $2 \cdot 2^\ell$ rounds,
{ ℓ++; "i" is still
} running.

# of hops

Process k receives a token (j, x, dir)

id of process who started

```
if ((k == j) && (dir == out)) { I am the leader .. }
else if (j > k) {

    if (x == 1) { // return the token;
        Send token(j, -, in) back to the neighbor who sent this;
    }
    else { send token(j, x-1, out) to (next hop) neighbor
        // forward the token
    }
}
else // k > j
    { discard token

    }

if ((k == j) && (dir == out)) { // I'm back }

//
```

Process k receives a token $(\oplus j, -, in)$:

if $(k \neq j)$ { // forward further.

}

else { // it came back to me.

if (both tokens came back during this round)

{ ~~phase~~ // go to next phase.

ℓ++;

initiate token for new phase

:

}

}

Termination?

Leader: send message to both neighbors announcing the result

Non-leaders: on receiving this announcement: forward the message, store leader id & terminate

# of phases?

$0, 1, \ldots \lceil \log_2 n \rceil$;

$= \lceil \log_2 n \rceil + 1$

# of rounds for phase ℓ:

$2^\ell + 2^\ell$ rounds or $2^{\ell+1}$

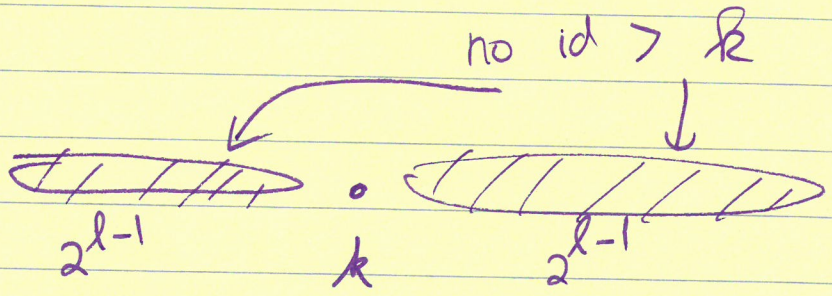
Total # of rounds $= \sum_{\ell=0}^{\lceil \log_2 n \rceil} 2^{\ell+1} = O(n)$

Message Complexity:

Phase 0: $\leq 4n$ messages.

phase ℓ; $\ell > 0$:

# of Processes initiating tokens in phase ℓ = # of Processes getting both tokens back in phase $\ell-1$



For k to initiate a token in phase ℓ, id of k is larger than any id in both shaded regions

Max # of such "winners" in phase $\ell-1$

=

(7)

$$= \frac{n}{1+2 \cdot 2^{l-1}} \quad ? \text{ or}$$

$$= \frac{n}{1+2^{l-1}} \quad ? \checkmark$$