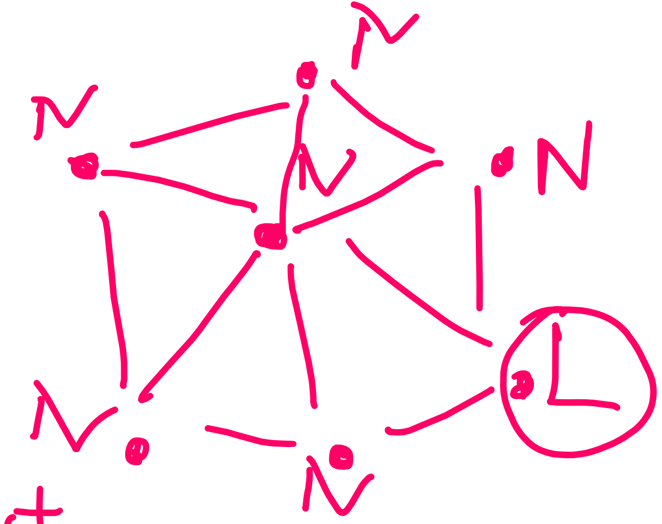
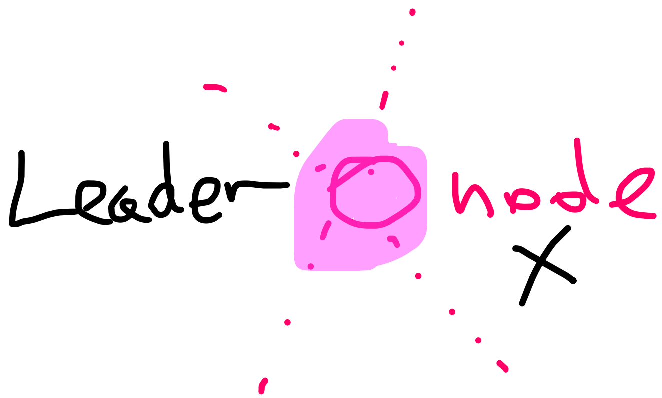
Leader election



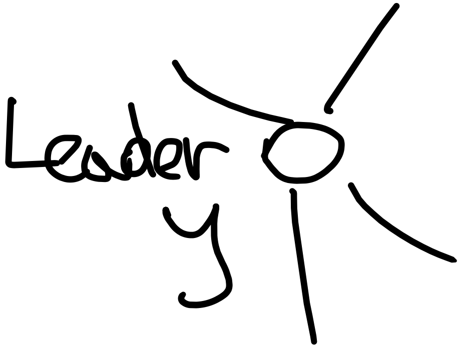
Problem 3.1 (Leader Election). Each node eventually decides whether it is a leader or not, subject to the constraint that there is exactly one leader.

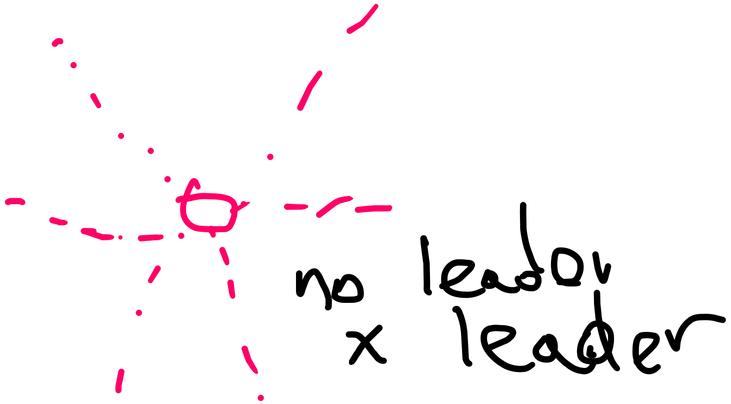
no ID's		ID assigned, distinct
1) anonymous	/	non-anonymous
2) uniform	/	non-uniform
$n = ?$		$n \approx \ldots$

Google-Apple ~~infection~~ exposure notification with smartphones, and BLE


Leader
node
X


no leader
x - leader

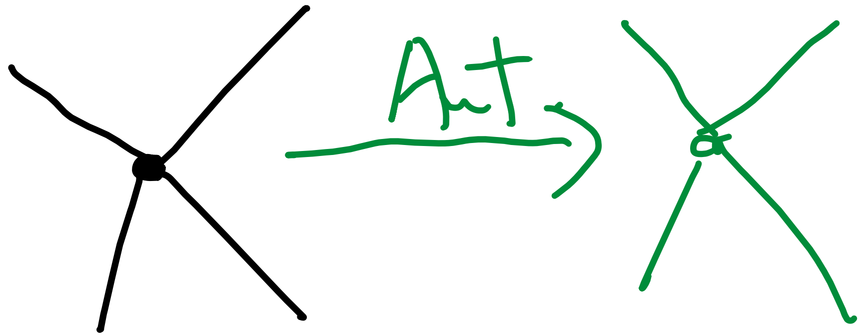

Leader
y


no leader
x leader

~~Deterministic~~, symmetric topologies

$\boxed{\text{same state property}}$

examples of topologies:



every node runs P, with the same parameters

$\Downarrow$

$\forall t$, after t steps all nodes are in the same state

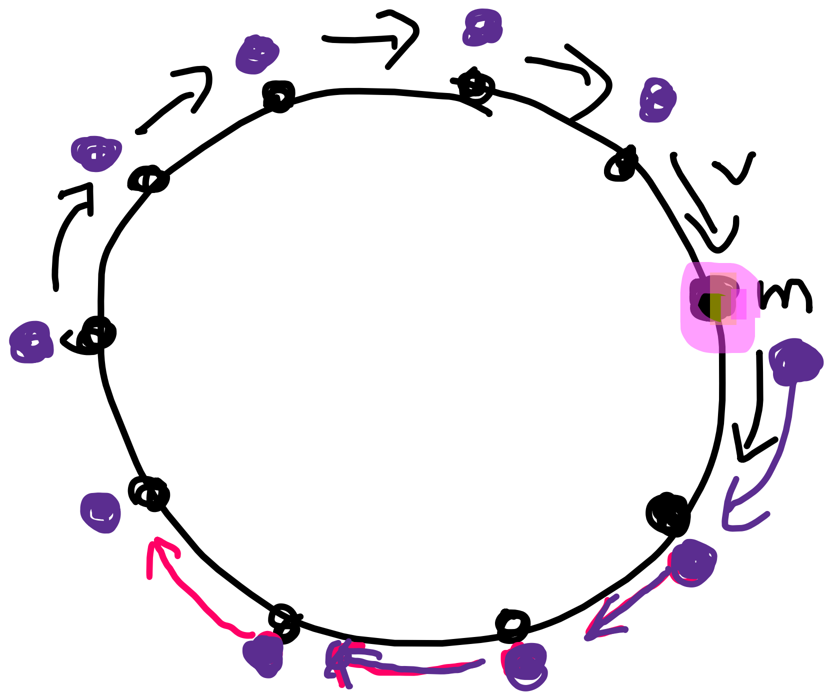
Algorithm 3.6 Clockwise Leader Election

```
Each node v executes the following code:
v sends a message with its identifier (for simplicity also v) to its
clockwise neighbor.
v sets m := v if the largest identifier seen so far
if v receives a message w with w > m then
v forwards message w to its clockwise neighbor and sets m := w
v decides not to be the leader, if it has not done so already.
else if v receives its own identifier v then
v decides to be the leader
end if
```

$m := \max(m, v)$

time $= n$

message $= ?$

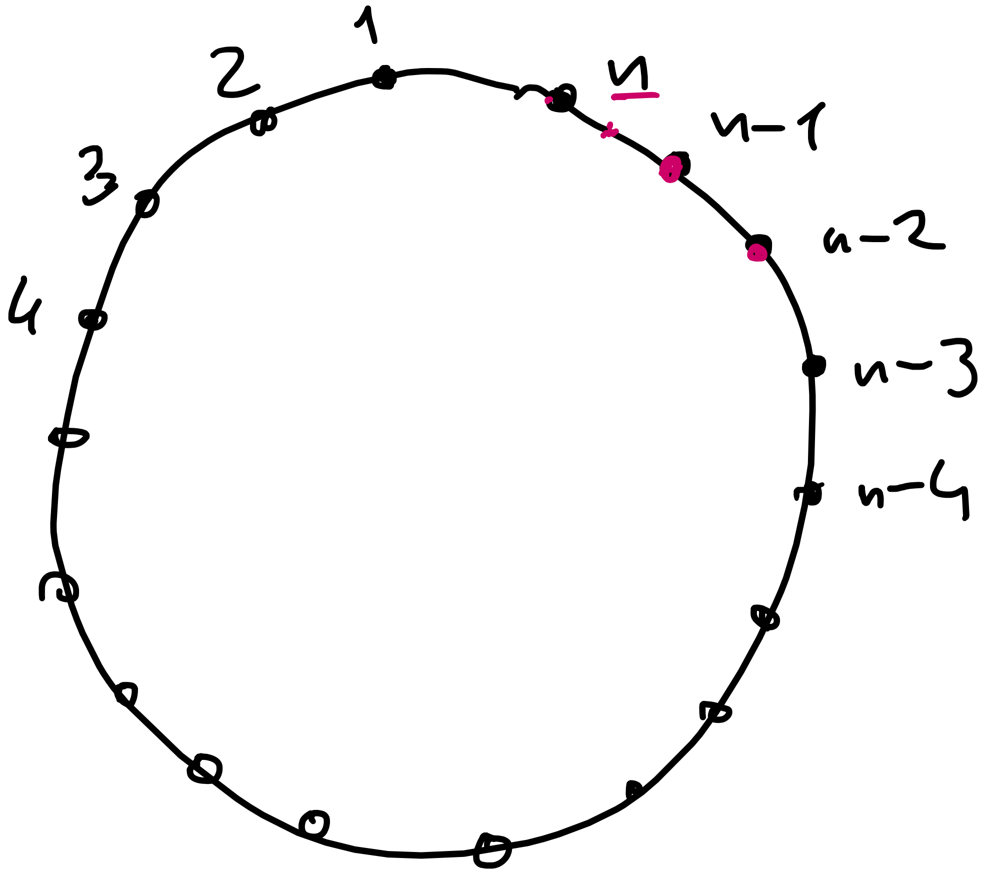


$m \rightsquigarrow \max(m, v)$

time complexity: $O(n)$ (obvious)

message complexity: $O(n^2)$ (worst case)

worst case; initial setting:



$n \to n$ messages

$n-1 \to n-1$ -"-

$n-2 \to n-2$ -"-

$\vdots$

$$\sum_{i=1}^{n} i \approx \frac{n^2}{2}$$

Algorithm 3.8 Radius Growth

1: Each node v does the following:

2: Initially all nodes are active. all nodes may still become leaders

3: Whenever a node v sees a message w with w > v, then v decides to not be a leader and becomes passive.

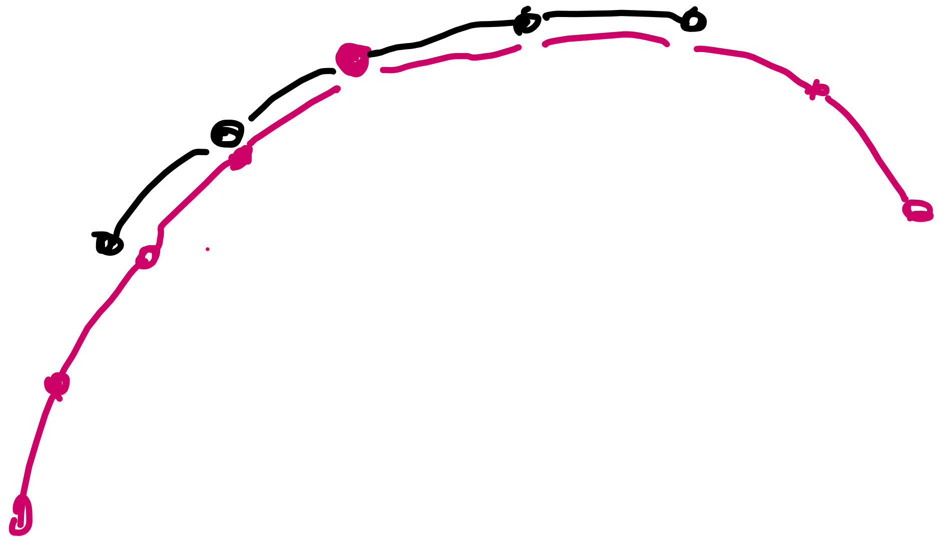
4: Active nodes search in an exponentially growing neighborhood (clockwise and counterclockwise) for nodes with higher identiers, by sending out probe messages. A probe message includes the ID of the original sender, a bit whether the sender can still become a leader, and a time-to-live number (TTL). The rst probe message sent by node v includes a TTL of 1.

5: Nodes (active or passive) receiving a probe message decrement the TTL and forward the message to the next neighbor; if their ID is larger than the one in the message, they set the leader bit to zero, as the probing node does not have the maximum ID.
If the TTL is zero, probe messages are returned to the sender using a reply message. The reply message contains the ID of the receiver (the original sender of the probe message) and the leader-bit.
Reply messages are forwarded by all nodes until they reach the receiver.

6: Upon receiving the reply message: If there was no node with higher ID in the search area (indicated by the bit in the reply message), the TTL is doubled and two new probe messages are sent (again to the two neighbors).
If there was a better candidate in the search area, then the node becomes passive.

7: If a node v receives its own probe message (not a reply) v decides to be the leader.

distance 2^i



message complexity:

1) after phase with kingdoms of size 2^i

there are $\leq n/2^i$ kings

$1 \sim 2^i$

$1 \sim 2^i$

$\sum \quad \frac{n}{2^i} \quad n$

round i : $n/2^i \longrightarrow 2 \cdot 2 \cdot 2^i = 4 \cdot 2^i$ per hing

totally: $4 \cdot 2^i \cdot \frac{n}{2^i} = 4 \cdot n$

round $i+1$: $n/2^{i+1} \longrightarrow 2 \cdot 2 \cdot 2^{i+1} = 4 \cdot 2^{i+1}$

total: $4 \cdot 2^{i+1} \cdot \frac{n}{2^{i+1}} = 4n$

total: $4n \cdot \#\text{rounds} =$

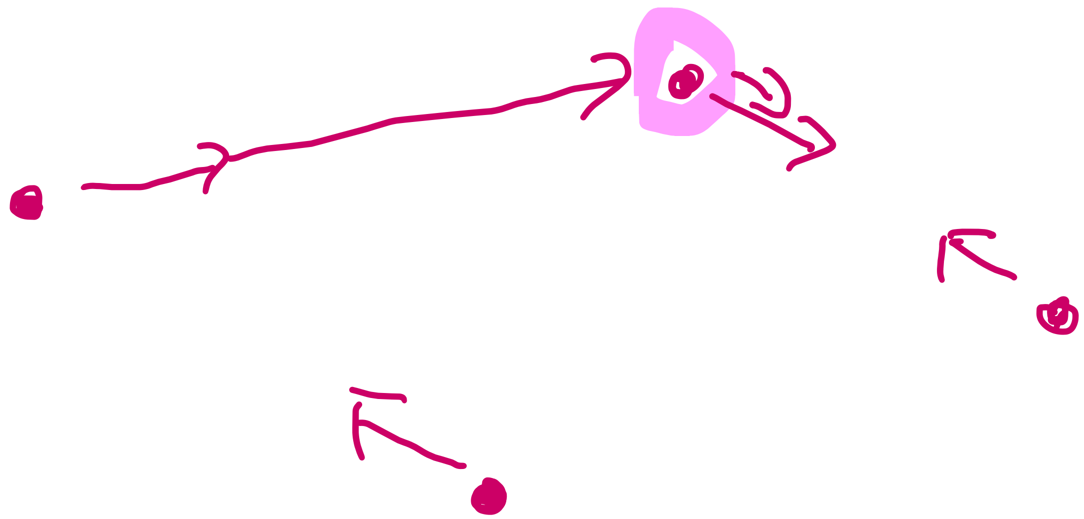
$= 4n \cdot \lceil \log n \rceil$

time: $O(n)$

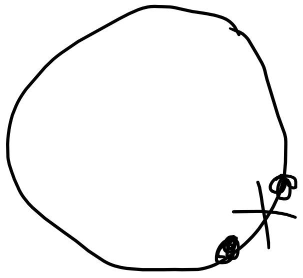
message complexity: $O(n \log n)$

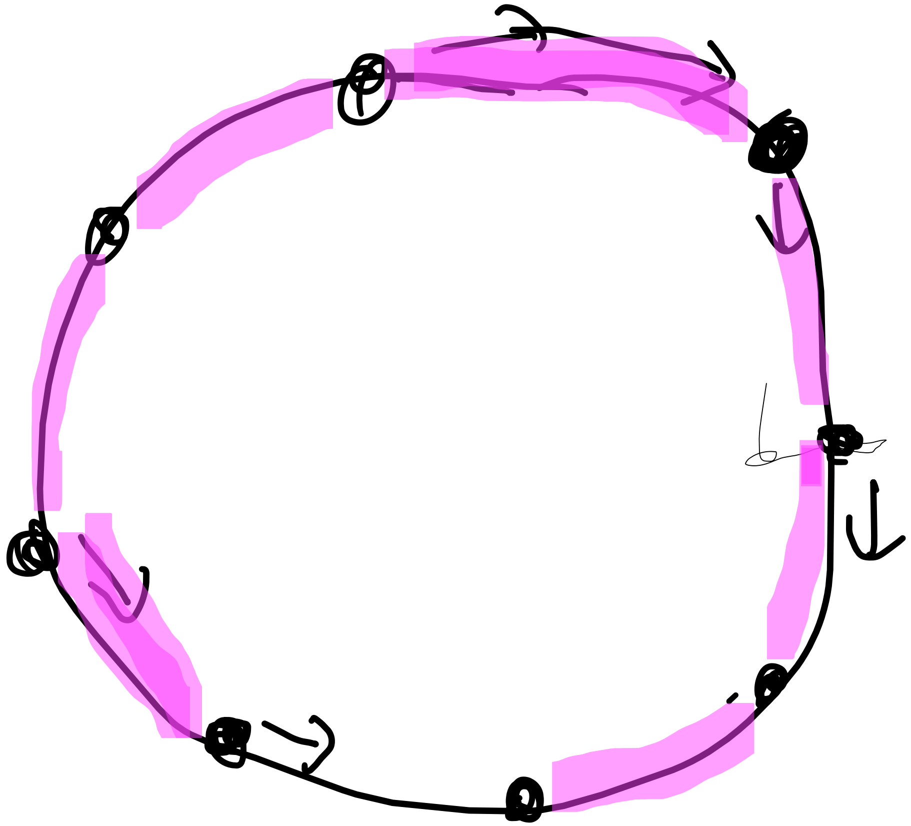
Lower bounds

model: adversary chooses the message in transit to deliver ("playing god")



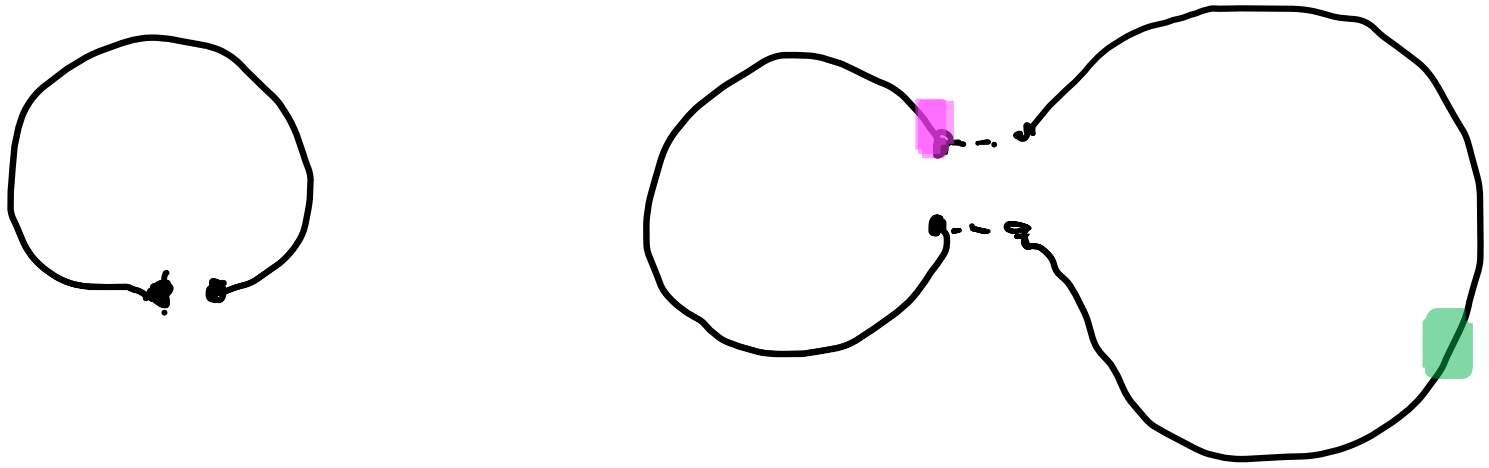
Open ring: schedule with one connection never used

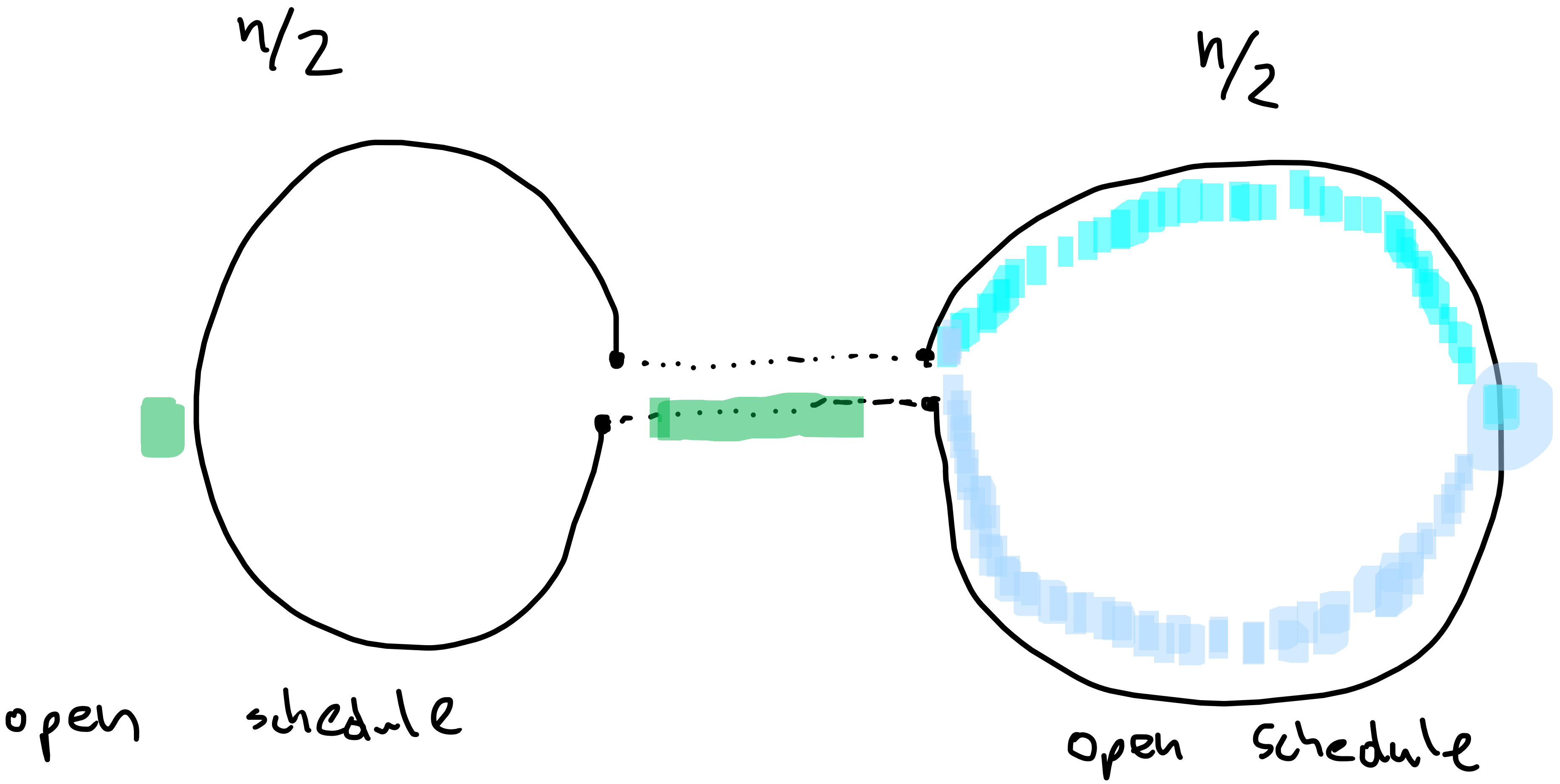




Lemma 3.15. Any uniform leader election algorithm for asynchronous rings has at least message complexity $M(n) = (n/4)(\log n + 1)$.

merging open rings




n/2
n/2
open
schedule
open
schedule

$$
\begin{aligned}
M(n) &\;\cancel{\geq}\; = \; 2 \cdot M\left(\frac{n}{2}\right) + \frac{n}{4} \\
&\geq 2 \cdot \left(\frac{n}{8}\left(\log \frac{n}{2} + 1\right)\right) + \frac{n}{4} \\
&= \frac{n}{4} \log n + \frac{n}{4} = \frac{n}{4}\left(\log n + 1\right).
\end{aligned}
$$

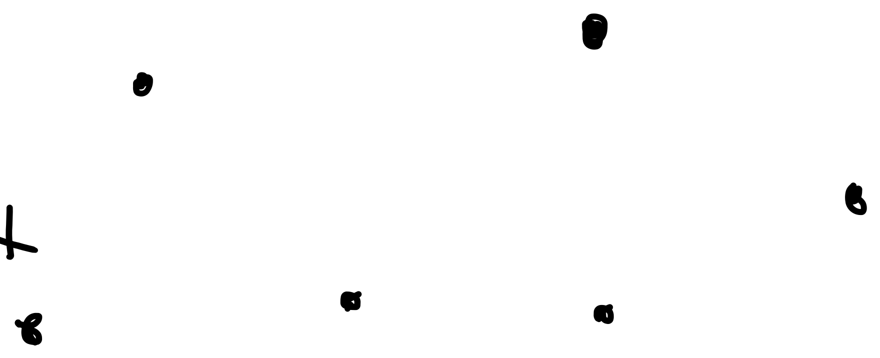
$$\frac{n}{4}(\log n + 1) = 2 \cdot \frac{n/2}{4}\left(\log \frac{n}{2} + 1\right) + \frac{n}{4}$$

$$\cancel{\frac{1}{4}} \log n + \cancel{\frac{1}{4}} = \cancel{\frac{1}{4}}\left(\log n \cancel{-1+1}\right) + \cancel{\frac{1}{4}}$$

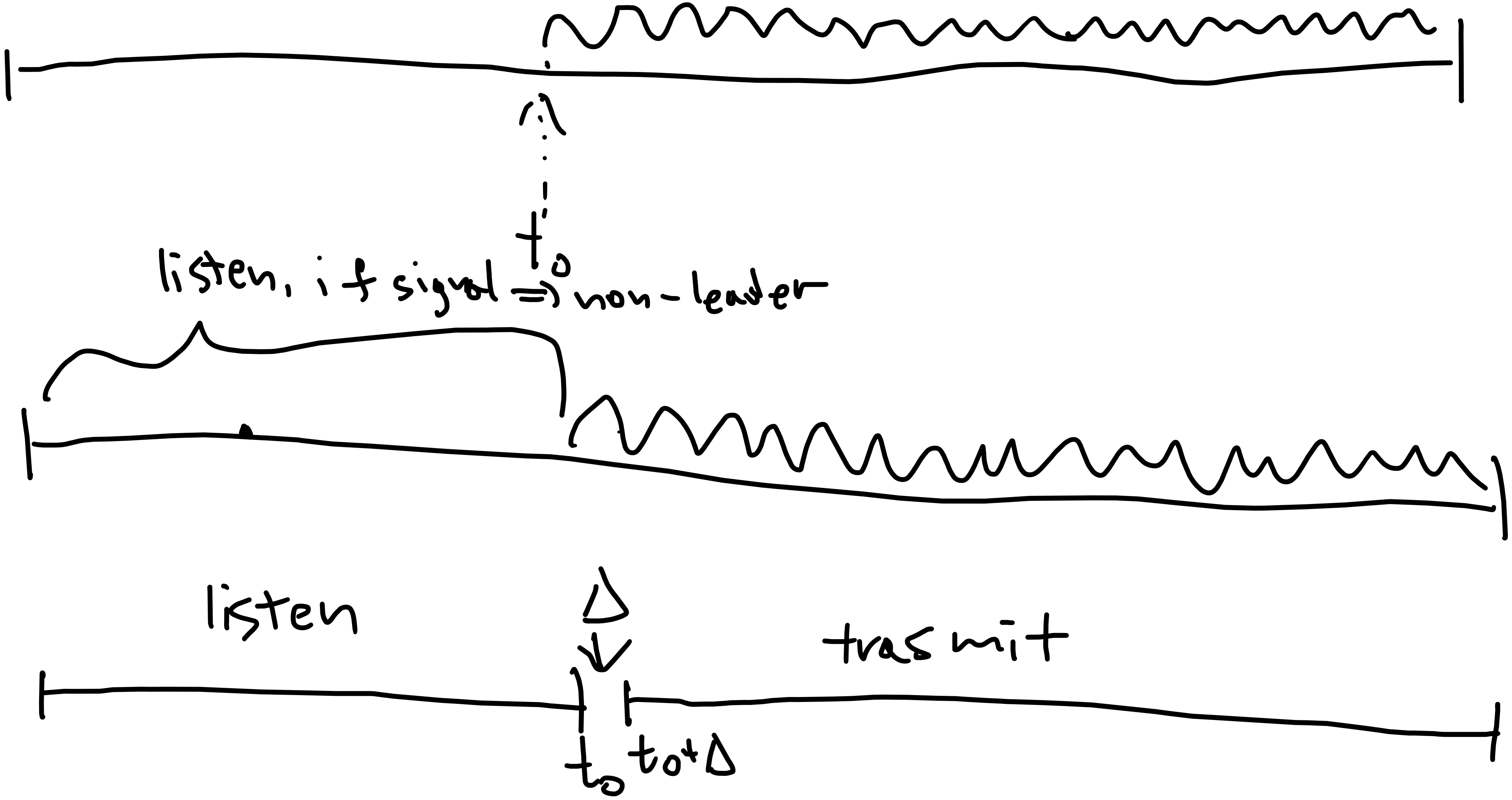
Algorithm 3.17 Synchronous Leader Election

```
Each node v concurrently executes the following code:
The algorithm operates in synchronous phases. Each phase consists of n
time steps. Node v counts phases, starting with 0.
if phase = v and v did not yet receive a message then
    v decides to be the leader
    v sends the message "v is leader" around the ring
end if
```

time $= O(n^2)$
overage $\approx n$
message: 1 broadcast



ID $\in [0, n^2]$
$t = 0$
$t = 1$:



t0
listen, if signal ⇒ non-leader
listen
Δ
trasmit
t0
t0+Δ