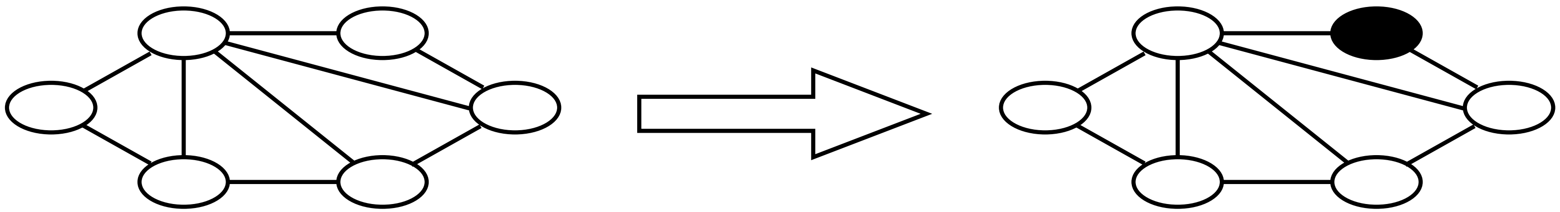
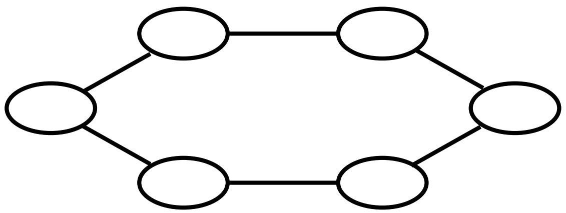
Election



Theorem *[Angluin 80]*

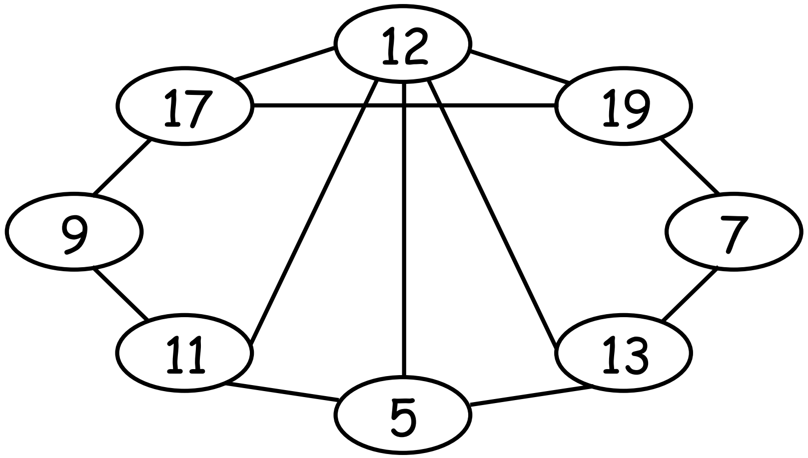
The election problem cannot be solved if entities do not have different identities.

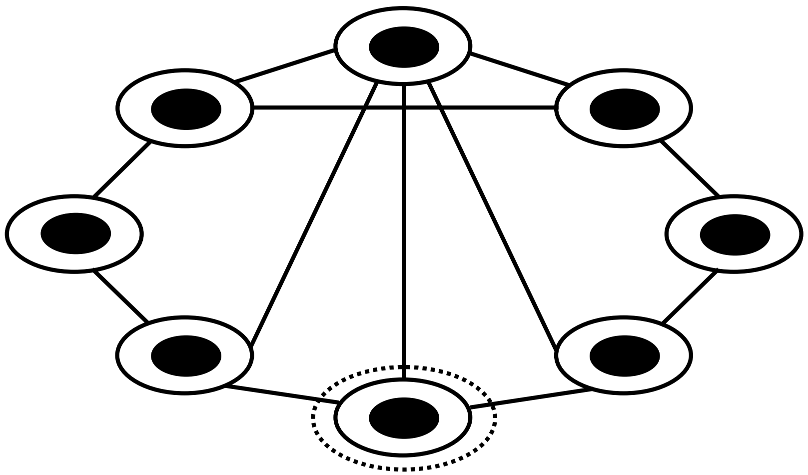


- Unique entities
- Same state
- Anonymous
- Synchronous

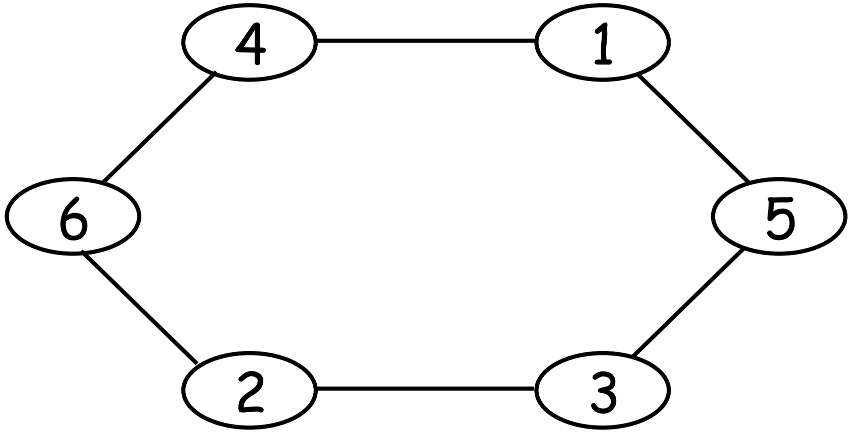
At each moment, they are doing the same thing.

Minimum Finding



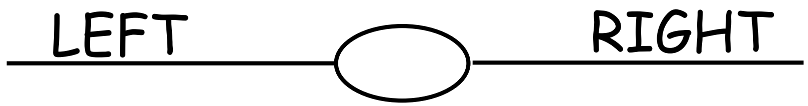


Ring



- n entities
- m= n links

- # entities = # links
- Symmetrical
- Each entity has two neighbors



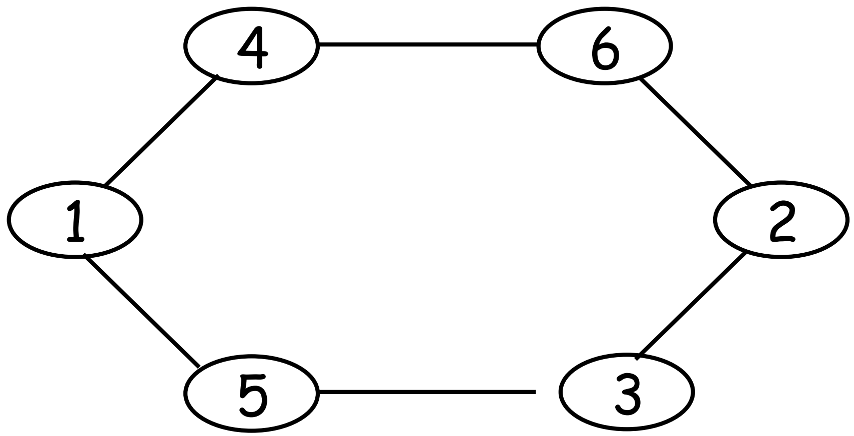
Sense of direction

Election Algorithms in Rings

- All the way
- As far as it can
- Controlled distance
- Electoral stages
  --- bidirectional version
- Alternating steps

All the way

Basic Idea: Each entity sees all identities.



ASSUMPTIONS

- Unidirectional or bidirectional ring.
- Local orientation.
- Distinct identities.

- All entities can be initiators
- bidirectional version

Ringsize???

FIFO message ordering ???

States: S={ASLEEP, AWAKE, FOLLOWER, LEADER}
S INIT={ASLEEP};
S_TERM={FOLLOWER, LEADER}.

ASLEEP

Spontaneously
INITIALIZE
become AWAKE

Receiving(``Election'', value, counter)
INITIALIZE;
send(``Election'', value, counter+1) **to other**
min:= Min{min, value}
count:= count+1
become AWAKE

<u>INITIALIZE</u>
count:= 0
size:= 1
known:= false
send("Election",id(x),size) **to right**;
min:= id(x)

AWAKE

Receiving ("Election", value, counter)

```
If value ≠ id(x) then
   send ("Election",value,counter+1) to other
   min:= MIN{min,value}
   count:= count+1
   if known = true then
              CHECK
   endif
else
       ringsize:= counter
       known:= true
       CHECK
endif
```

```
CHECK
if count = ringsize then
      if min = id(x) then
            become LEADER
      else
            become FOLLOWER
      endif
endif
```

Complexity

Each identity crosses each link --> n^2

The size of each message is log(*id*)

$O(n^2)$ messages
$O(n^2 \log(id_M))$ bits

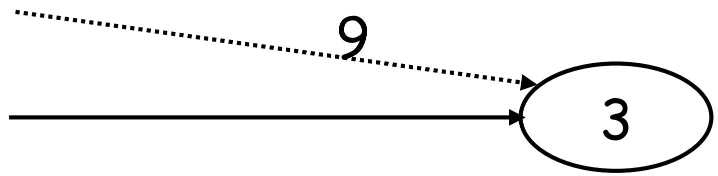
Observations:

1. The algorithm also solves the data collection problem.

2. It also works when the ring is unidirectional.

As far as it can

Basic Idea: It is not necessary to send and receive messages with larger *ids* than the *ids* that have already been seen.



ASSUMPTIONS

- Unidirectional or bidirectional ring
- Different *ids*
- Local orientation

States: S={ASLEEP, AWAKE, FOLLOWER, LEADER}
S_INIT={ASLEEP}
S_TERM={FOLLOWER, LEADER}

--- unidirectional version

ASLEEP

Spontaneously

```
send("Election",id(x)) to right
min:= id(x)
become AWAKE
```

Receiving("Election", value)

```
send("Election",id(x)) to right
min:= id(x)
If  value < min  then
        send("Election", value) to other
        min:= value
endif
become AWAKE
```

/* this could be avoided if id(x)>value

AWAKE

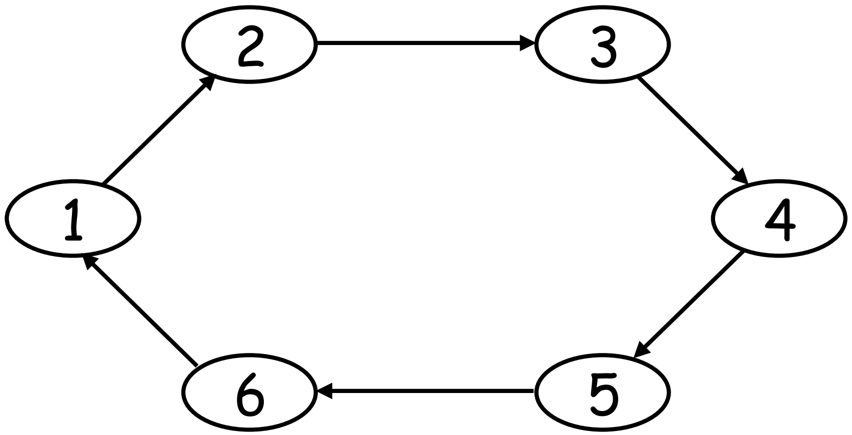
```
Receiving("Election'", value)
        if  value < min then
                send("Election", value)  to other
                min:= value
        else
           If value= min then NOTIFY endif
        endif

Receiving(Notify)
        send(Notify)  to other
        become  FOLLOWER
```

```
NOTIFY
        send(Notify) to  right
        become LEADER
```

Observations

- Notification is necessary !
- Bidirectional version

Worst-Case Complexity (Unidirectional Version)



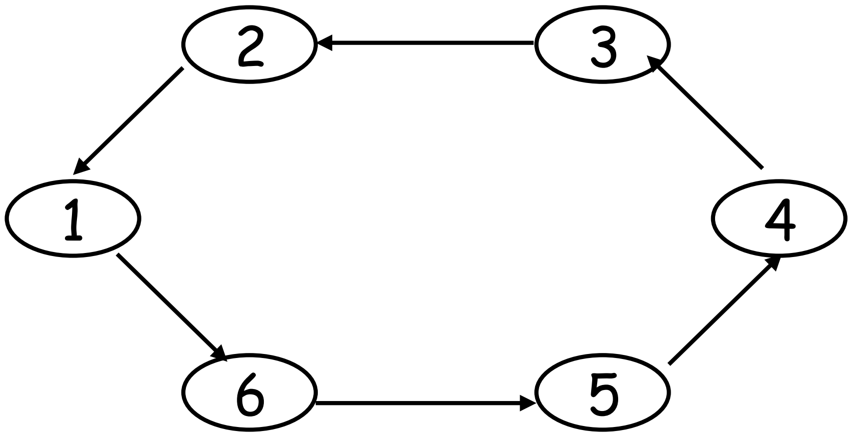
1 ---> n		links
2 ---> n - 1		links
3 ---> n - 2		links
…	…	…
n ---> 1		link

$$n + (n - 1) + (n - 2) + \ldots + 1 = \sum_{i=1}^{n} i = (n+1)\,(n\,)\,/\,2$$

Total: $n\,(n+1)\,/\,2 + n = O(n^2)$

Last n: notification

Best-Case Complexity (Unidirectional Version)

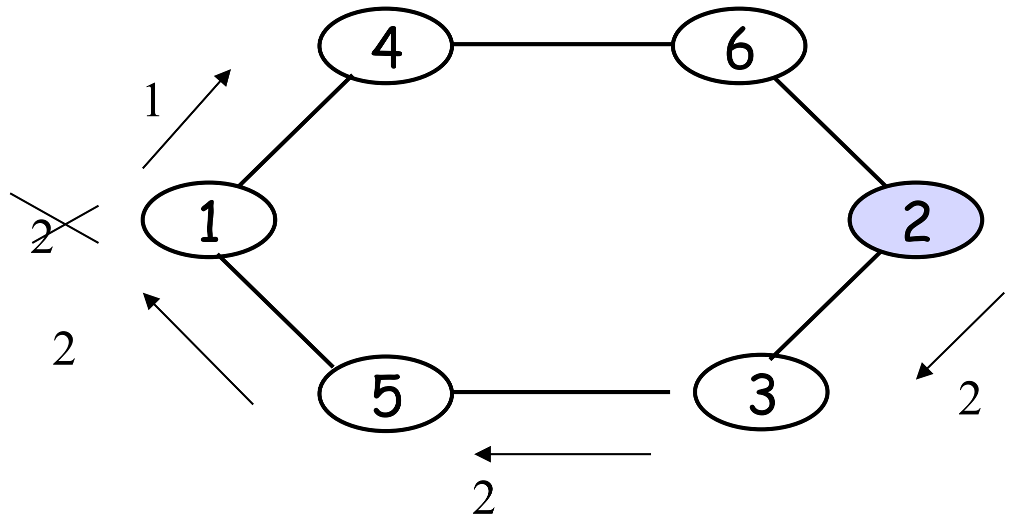


1 ---> n links
for all i ≠ 1 ---> 1 link (--> total = n - 1)

Total: n + (n - 1) + n = O(n)
Last n: notification

“ As far as it can “ algorithm

Let’s trace node 2’ value



Average-Case Complexity

Entities are ordered in an equiprobable manner.

Jth smallest id - crosses (n / J) links

$\sum_{J=1}^{n} (n / J)$ = n * Hn

Harmonic series of n numbers

(approx. 0.69 log n)

Total: n * Hn + n = 0.69 n log n + O(n) = O(n log n)

Controlled Distance

Basic idea: Operate in stages. An entity maintains control on its own message.

ASSUMPTIONS

- Bidirectional ring (with sense of direction)
- Different *ids*
- Local orientation

1) Limited distance (to avoid big to travel too much)

Ex: stage i: distance 2^{i-1}

2) Return messages (if seen something smaller does not continue)

3) Check both sides

4) smallest always win (regardless of stage number)

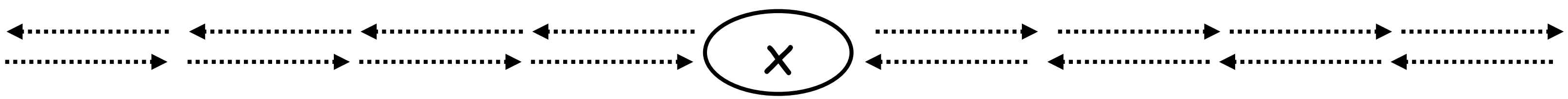
Candidate entities begin the algorithm.

Stage i:

- Each *candidate* entity sends a message with its own *id* in both directions

- the msg will travel until it encounters a smaller Id or reaches a certain distance

- If a msg does not encounters a smaller Id, it will return back to the originator



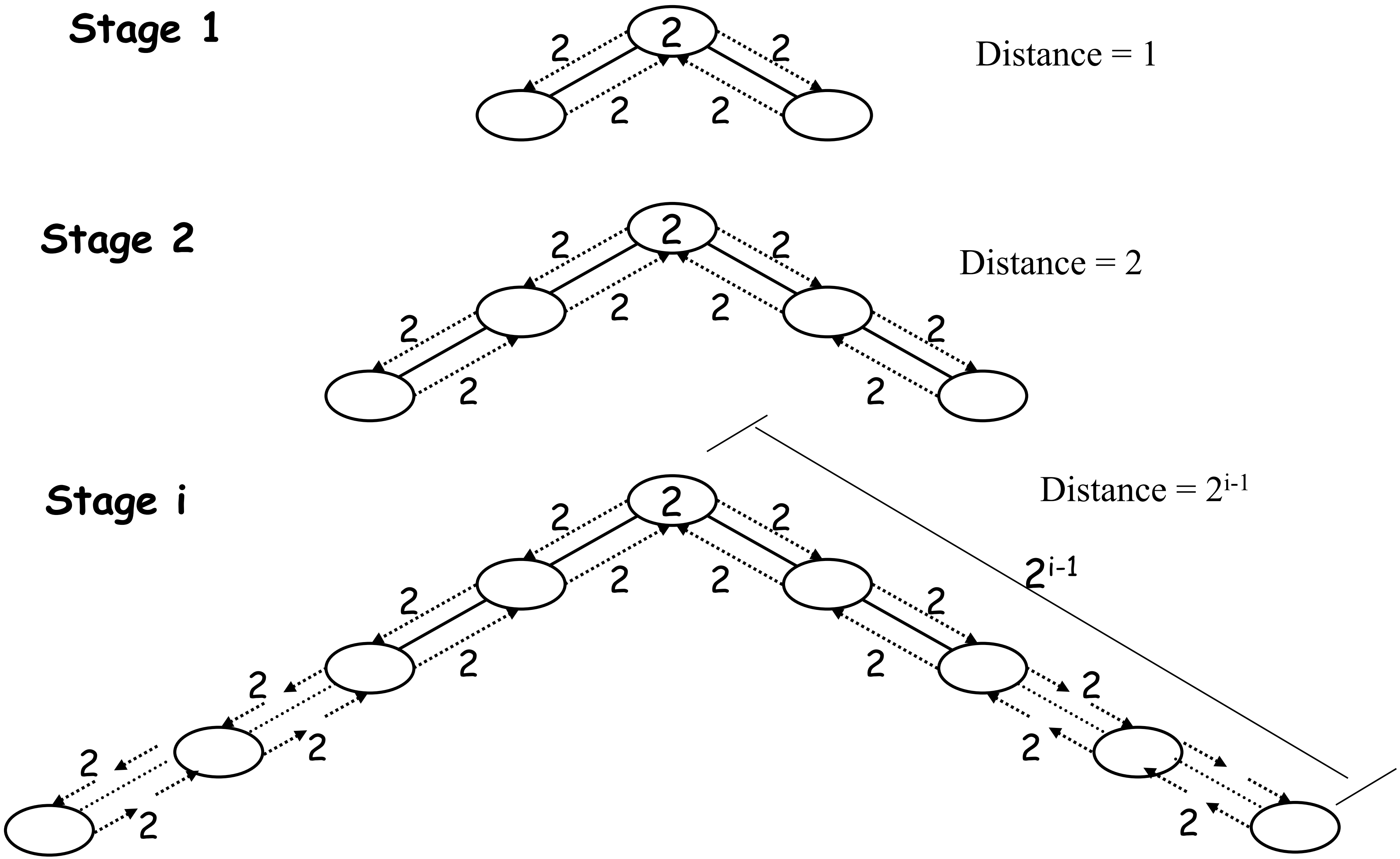
- A candidate receiving its own msg back from both direction survives and start the next stage

Entities encountered along the path read the message and:

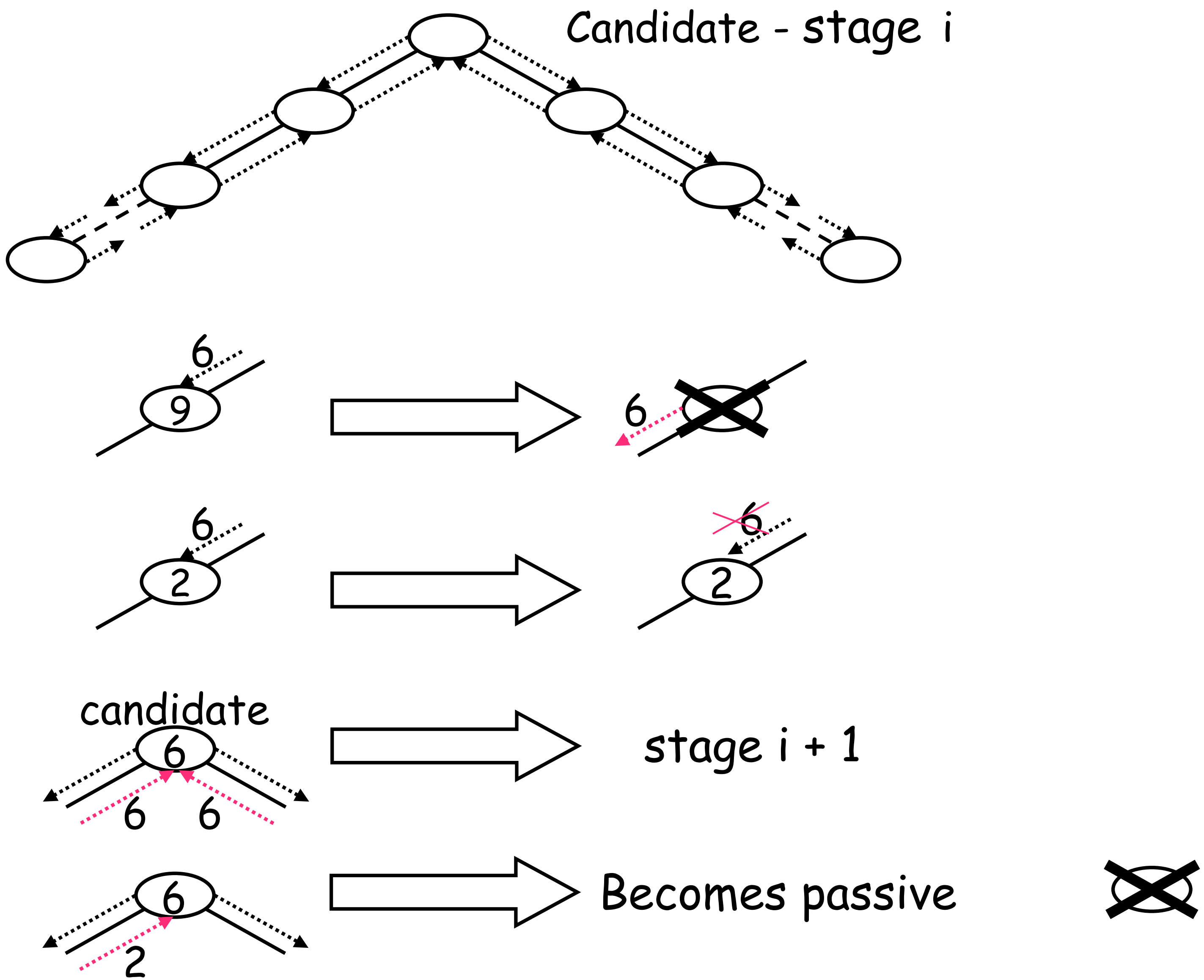
- Each entity *i* with a greater identity Id_i become defeated (passive).

- A defeated entity forwards the messages originating from other entities, if the message is a notification of termination, it terminates

More...


Stage 1
Distance = 1
Stage 2
Distance = 2
Stage i
Distance = 2^{i-1}
2^{i-1}
2

More...


Candidate - stage i
6
9
6
6
2
6
2
candidate
6
6
6
stage i + 1
6
2
Becomes passive

TERMINATION ?

If a candidate receives its message from the opposite side it sent it,
it becomes the leader and notifies

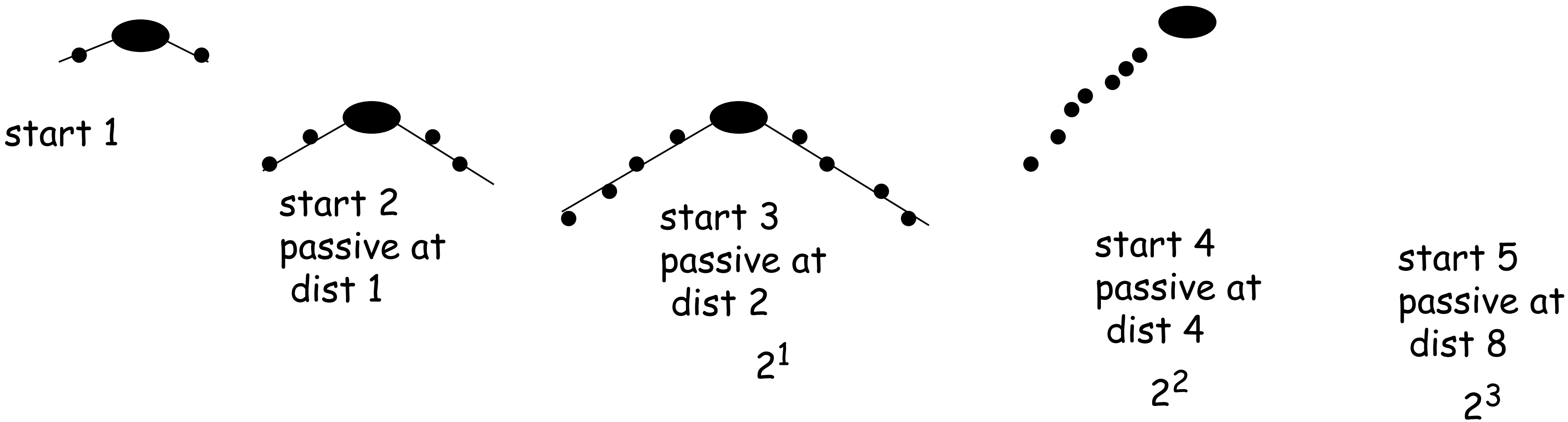
Complexity

When the distance is doubled at each stage
i.e., dis(i) = 2^{i-1}:

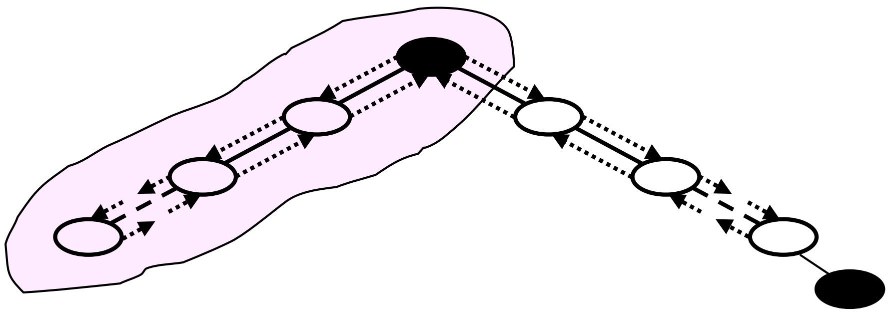
Notion of Logical Stage

n_i entities start stage i

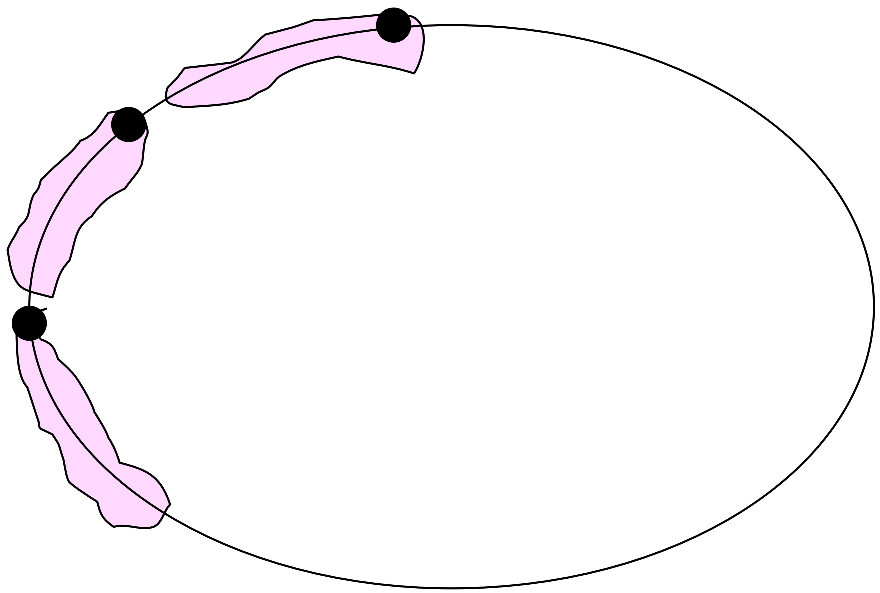
If x starts stage i the Id of x must be smaller than the Ids of the neighbours at distance up to 2^{i-2} on each side



Within any group of $2^{i-2}+1$ consecutive entities at most one can survive stage i-1.



$$n_i \le n/\,(2^{i-2}+1)$$



Starting stage i:

"Forth" messages:

each will travel at most 2^{i-1}

Tot: $2\ n_i\ \ 2^{i-1}$

"Back" messages:

each survivor will receive one from each side

$2\ n_{i+1}\ \ 2^{i-1}$

each entity that started the stage but did not survive will receive none or one

$(n_i - n_{i+1})\ 2^{i-1}$

Tot: $2\ n_{i+1}\ \ 2^{i-1} + (n_{i+1} - n_i)\ 2^{i-1}$

stage i:

Tot: $2\, n_i \;\; 2^{i-1} + 2\, n_{i+1} \;\; 2^{i-1} + (n_{i+1} - n_i)\, 2^{i-1}$

$= (3\, n_i + n_{i+1})\, 2^{i-1}$

$n_i \leq n / (2^{i-2} + 1)$

$\leq (3 \lfloor n / (2^{i-2} + 1) \rfloor + \lfloor n / (2^{i-1} + 1) \rfloor)\, 2^{i-1}$

$$< \frac{3\, n\, 2^{i-1}}{(2^{i-2} + 1)} + \frac{n\, 2^{i-1}}{(2^{i-1} + 1)}$$

$$< \frac{3\, n\, 2^{i-1}}{2^{i-2}} + \frac{n\, 2^{i-1}}{2^{i-1}}$$

$= 3\, n\, 2 + n = 7\, n$

The first stage is a bit different:

If everybody starts:

the survivors $4\ n_2\ 2^0$

the others $3\ (n - n_2)\ 2^0$

2 “forth”, 1 “back”

$n_2 \leq n / (2^0+1)$

$4\ n_2 + 3\ n - 3n_2 = n_2 + 3\ n$

$= n/2 + 3\ n \quad < 4n$

Total Number of Stages

As soon as 2^{i-1} is greater than or equal to n

$$2^{i-1} \geq n$$

WHEN $i = \log n + 1$

$\log n + 1$ STAGES

first stage

$$TOT \leq \sum_{i=1}^{\log n} 7n \quad + O(n)$$

$$= n \sum_{i=2}^{\log n} 7 = 7\, n \log n + O(n)$$

$$O(n \log n)$$

Conjecture:

In unidirectional rings,
the worst case complexity is $\Omega(n^2)$;
to have a complexity of $O(n \log n)$ messages,
bidirectionality is necessary.

NOT TRUE !

Electoral Stages

Basic idea:

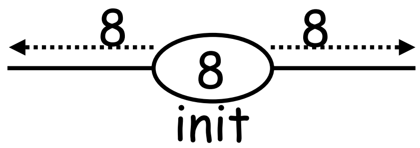
A message will travel until it reaches another candidate

A candidate will receive a message from both sides

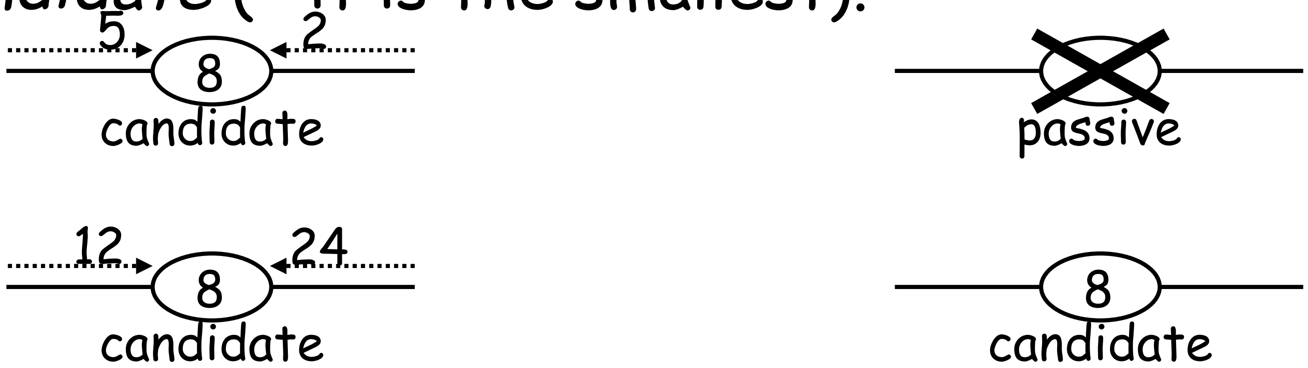
ASSUMPTIONS

- Different *ids*
- Bidirectional ring (+ unidirectional version)
- Local orientation
- Ordering of messages

Each *candidate* sends its own *Id* in both directions.

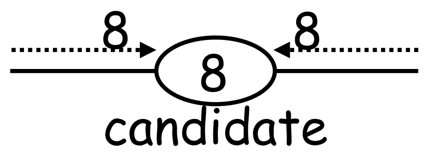


When a *candidate i* receives two messages Id_j (from the right) and Id_k (from the left), it determines if it becomes *passive* (= it is not the smallest), or if it remains *candidate* (= it is the smallest).



The minimal entity will never cease to send messages.

When an entity knows that it is the *leader*

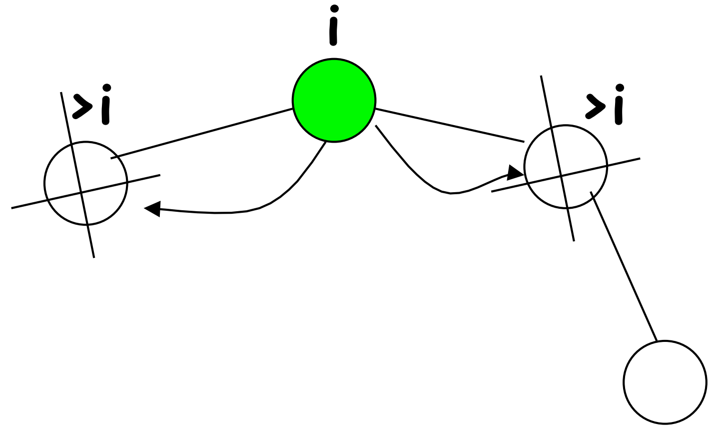


it sends a *notification* message which travels around the ring.

Complexity - Worst Case

At each step: At least half the entities became passive.

$$n_{i+1} \le \frac{n_i}{2}$$



$n_0 = n$

$n_1 \le n/2$

$n_i \le n/2^i$

⇒ $n/2^k \le 1$ when $k \ge \log n$

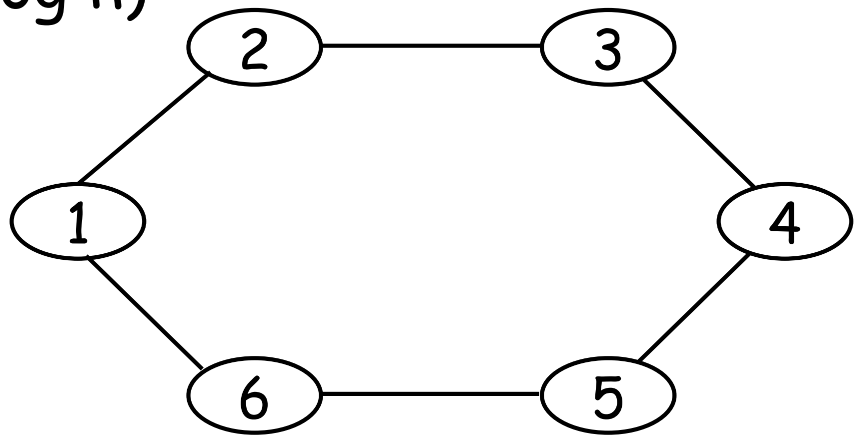
 # steps: At most $\lfloor (\log n) \rfloor$

Each entity sends or resends 2 messages.
 # messages: 2n
 # bits: $2n * \lfloor (\log n) \rfloor$

Last entity: 2n messages to understand that it is the last active entity, then n notification messages.

Total: $2n * \lfloor (\log n) \rfloor + 3n = O(n \log n)$

Best Case ?



Unidirectional version

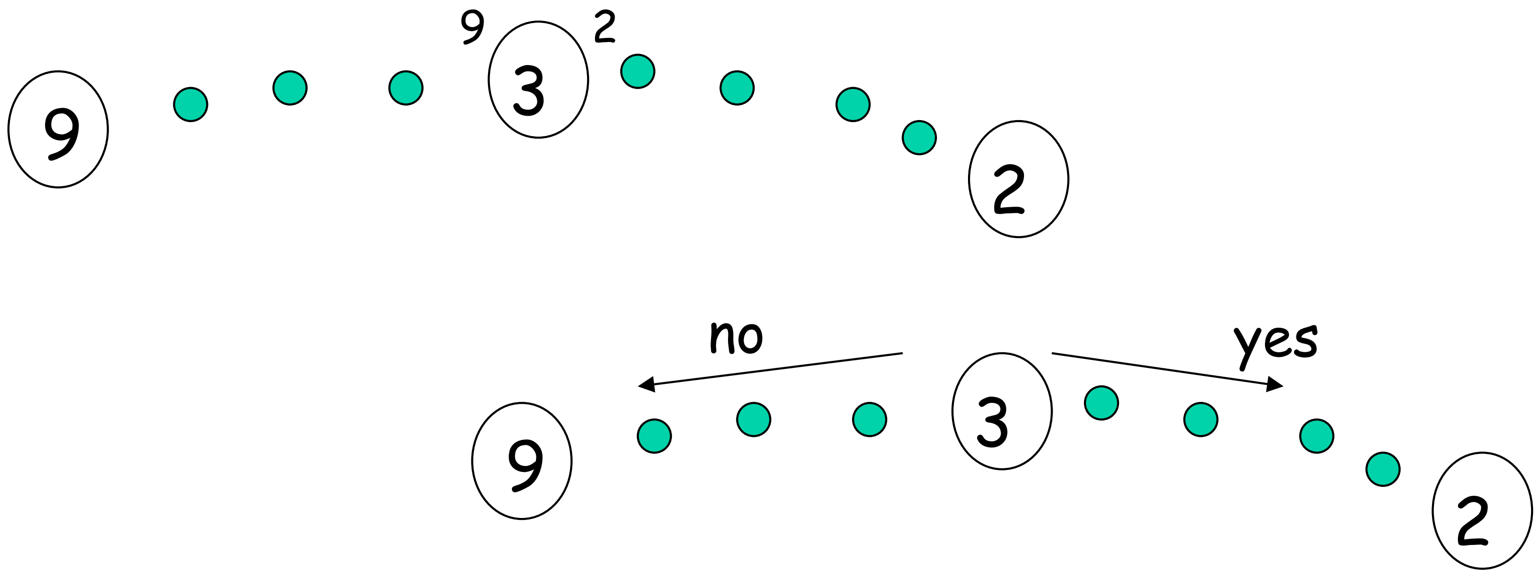
Simulation of the bidirectional algorithm with the same complexity.

Examples ....

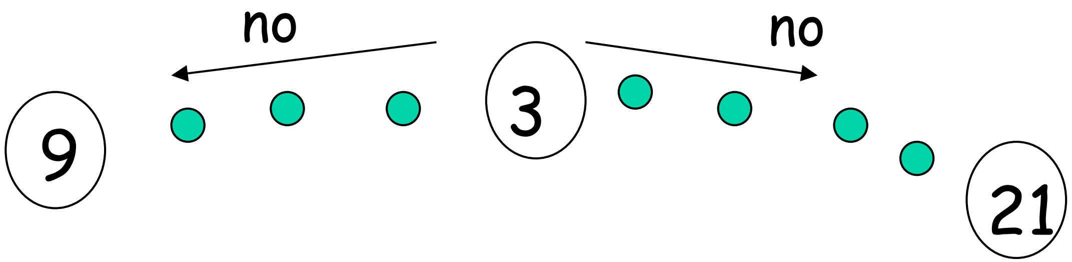
Hirschberg and Sinclair's Conjecture is false.

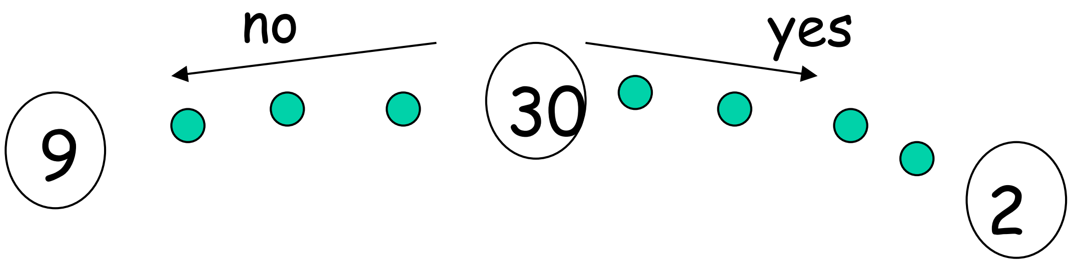
Stages with Feedback

A feedback is sent back to the originator of the message



send YES to the smallest of the two IF it is smaller than me
(otherwise send NO)
send NO to the other





Examples ... Analysis ...

Alternating Directions

Basic idea: Alternating directions.

- Different *ids*.
- Bidirectional ring and sense of direction.
- Local orientation.
- Message ordering.

send-left
begin-to-kill (if possible)
send-right

Algorithm:

1. **Each entity** sends a message **to its right**. This message contains the entity's own *id*.

2. Each entity compares the *id* it received from its left to its own *id*.

3. If its own *id* is greater than the received *id*, the entity becomes passive.

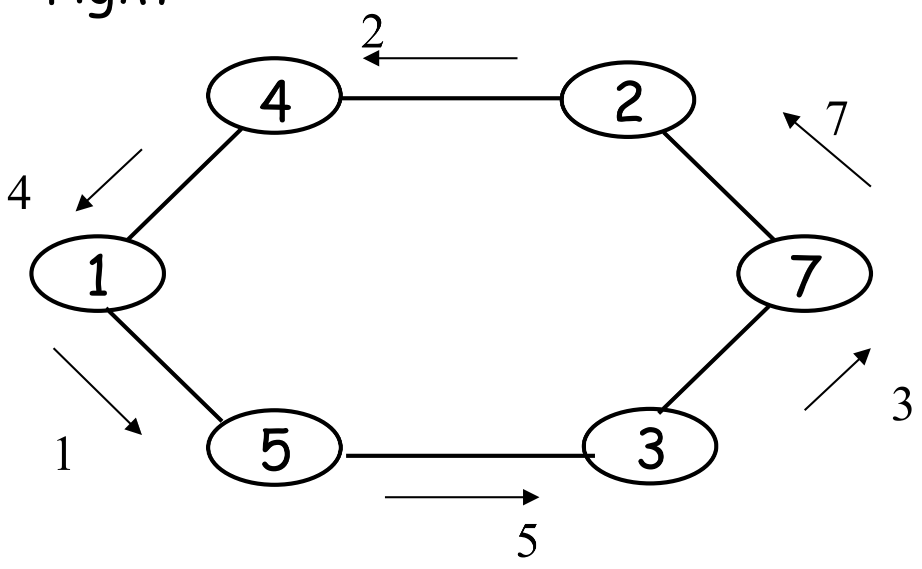
4. All entities that remained active (surviving) send their *id*s **to their left**.

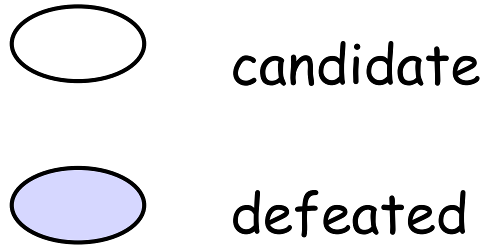
5. A surviving entity compares the *id* it received from its right with its own *id*.

6. If its own *id* is greater than the *id* it received, it becomes passive.

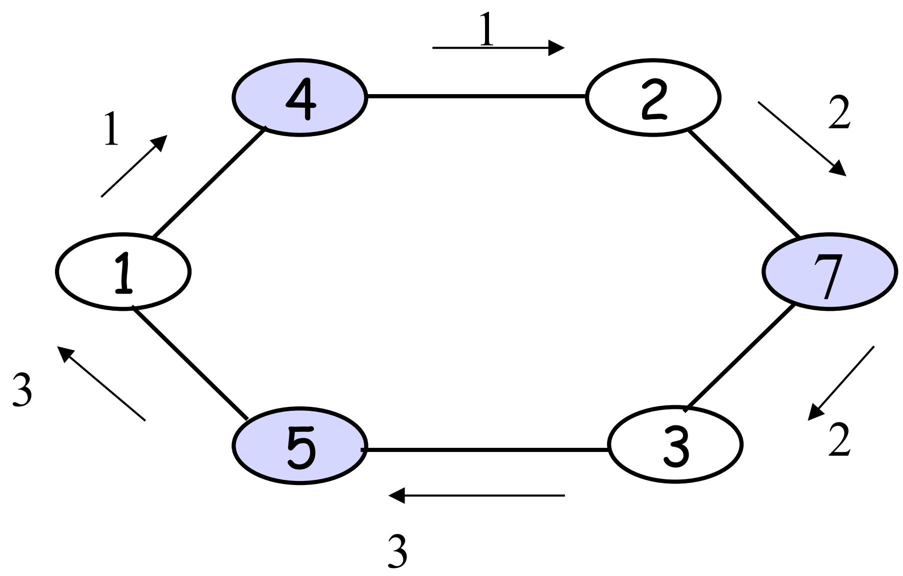
7. Go back to step 1 and repeat until an entity receives its own *id* and becomes *leader*.

Step 1: right



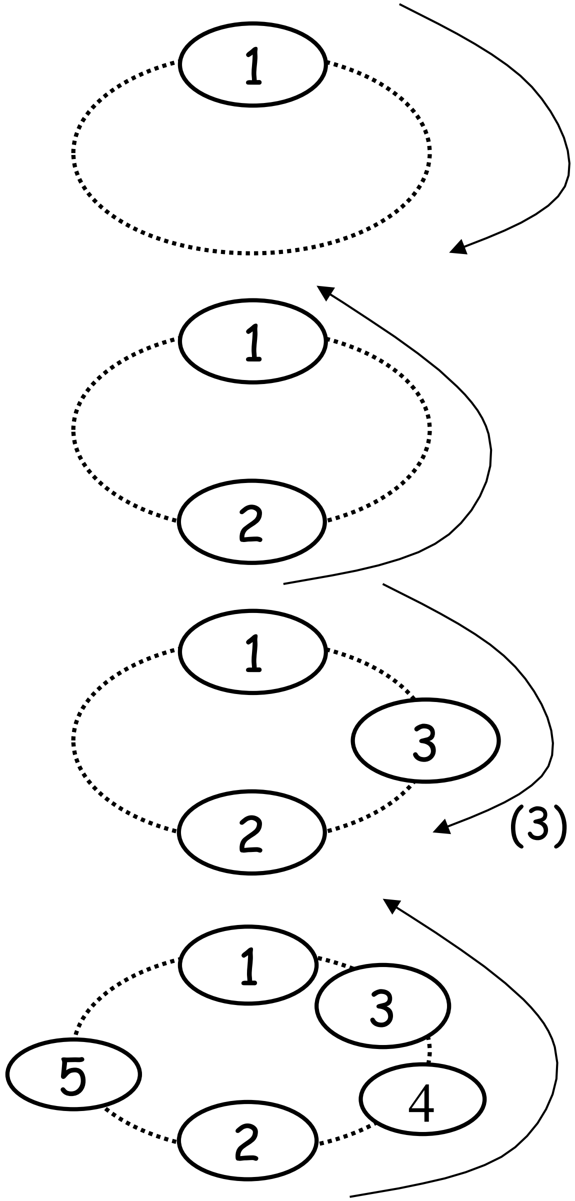


Step 2: left



Complexity

Analyze # of steps in worst case:



Last phase k — 1 active entity

Phase k - 1 — at least 2 active entities

(2) will become passive at the next step.

Phase k - 2 — at least 3 active entities

(3) must be there; otherwise, (2) would be killed.

Phase k - 2 — at least 5 active entities

1 2 3 5 8 13 21 ....

More...

 # steps =

index of the lowest Fibonacci number >= n

$F_1 = 1$

$F_2 = 2$

$F_3 = 3$

$F_4 = 5$

$F_5 = 8$

. . .

$F_k = i = ?$

$= \text{approx. } 1.45 \log_2 n$

 # Messages = n for each step

Total = approx. $1.45\, n \log_2 n$

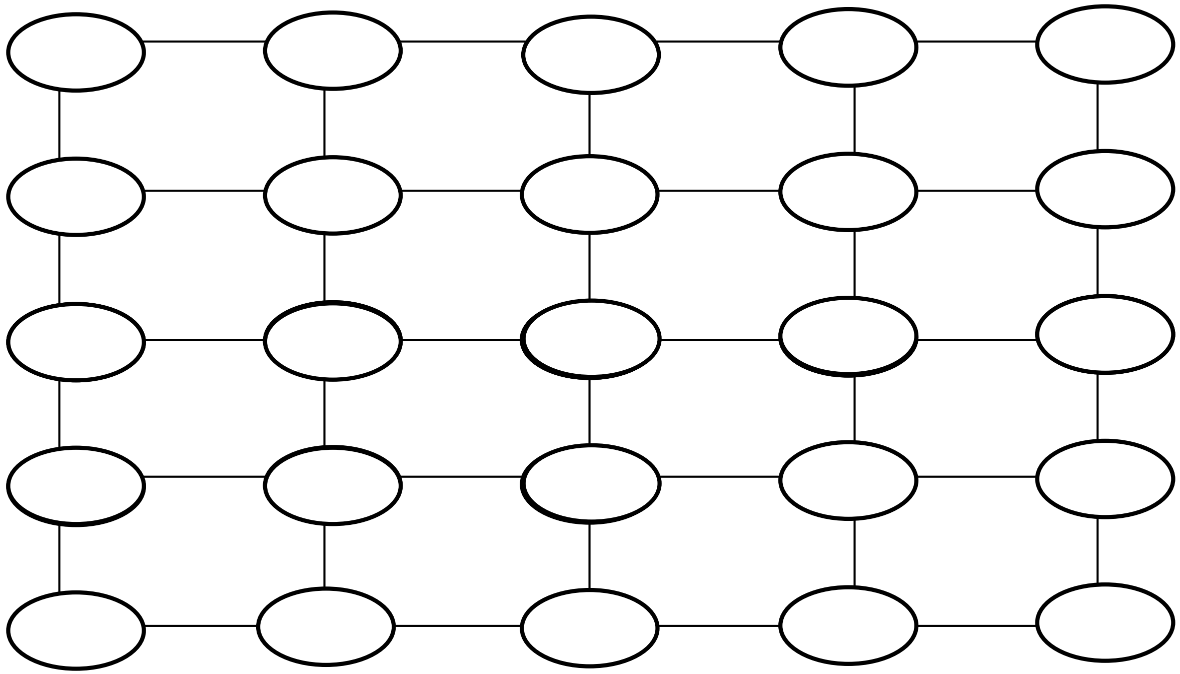
upper bounds

Bidirectional		Unidirectional	
LeLann (1977) "All the way"	n^2	LeLann (1977) Unidirectional simulation	n^2
Chang & Roberts (1979) "As far as you can" average case n log n	n^2	Chang & Roberts	n^2
Hirshberg & Sinclair (1980) stages message control	$7n \log n$		
Franklin (1982) stages	$2n \log n$	Dolev, Klawe & Rodeh Unidirectional simulation	$2n \log n$
Peterson (1982) Alternate	$1.44n \log n$	Peterson 1982 Unidirectional simulation	$1.44n \log n$
		Dolev, Klawe & Rodeh (1982)	$1.36n \log n$
		Higham, Przytycka (1984)	$1.22n \log n$

lower bounds

Burns	$0.5n \log n$
Pachl, Korach Rotem (1984)	$0.69n \log$

Mesh



If it is square mesh: n nodes = $n^{\frac{1}{2}} \times n^{\frac{1}{2}}$

m = O(n)

Asymmetric topology

corners

border

internal

Idea: Elect as a leader one of the four corners

Three phases:

1) Wake up

2) Election (on the border) among the corners

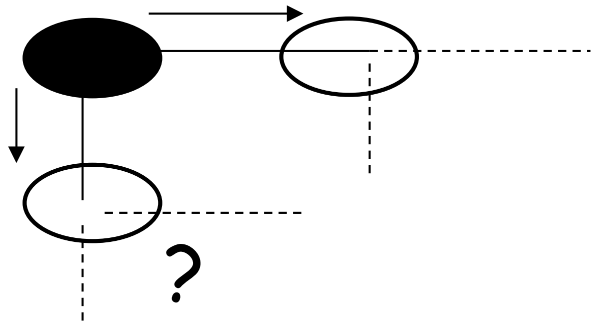
3) Notification

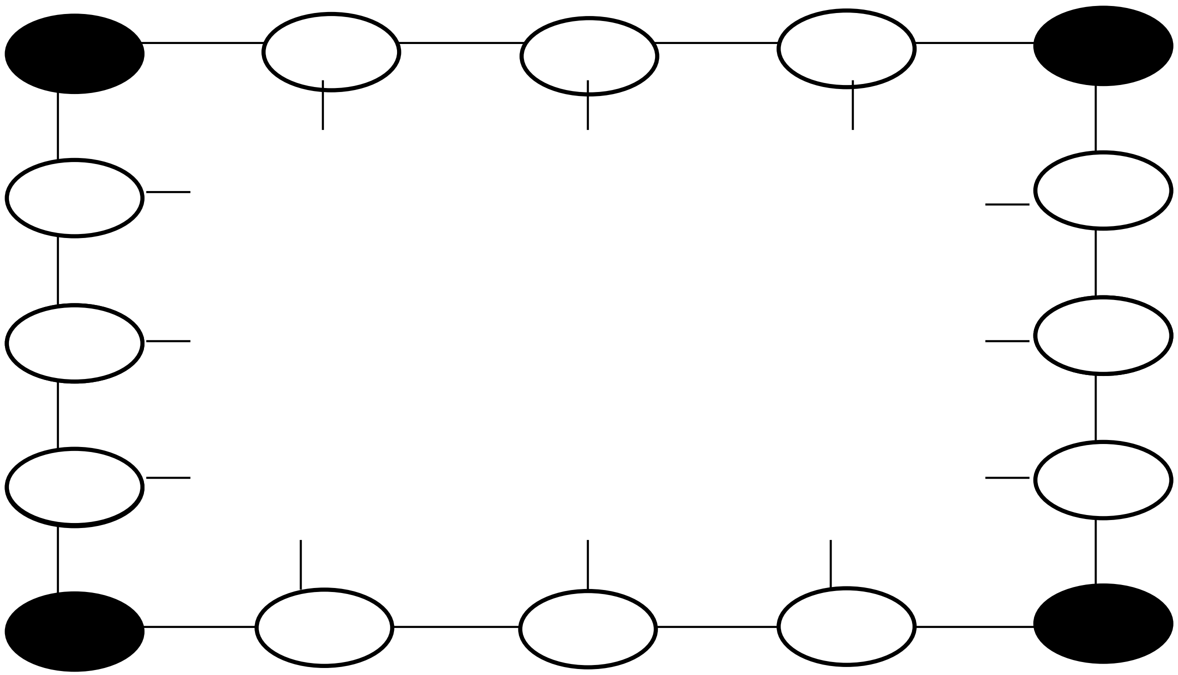
1) Wake up

- Each initiator send a wake-up to its neighbours
- A non-initiator receiving a wake up, sends it to
  its other neighbours

$$O(m) = O(n)$$

2) Election on the border started by the corners





$O(\sqrt{n})$

3) Notification

by flooding

$O(m) = O(n)$

TOT: $O(n)$

Torus