A biological solution to a fundamental distributed computing problem

\

Yehuda Afek, Noga Alon, Omer Barad, Eran Hornstein, Naama Barkai, Ziv Bar-Joseph.
Science 331, 183 (2011), pp. 183 - 185

kubuntu

Maximal Independent set

- Given graph G = (V, E)
- Maximal Independent Set
  - Subset A of the nodes such that:

    1) Each node is either in A or is adjacent to a node in A

    2) If u and v both belong to A → u and v are not adjacent
- Notation
  - n: upper bound on |V|
  - D: upper bound on number of active (see further) neighbours of any node (possibly n)

kubuntu

Distributed MIS

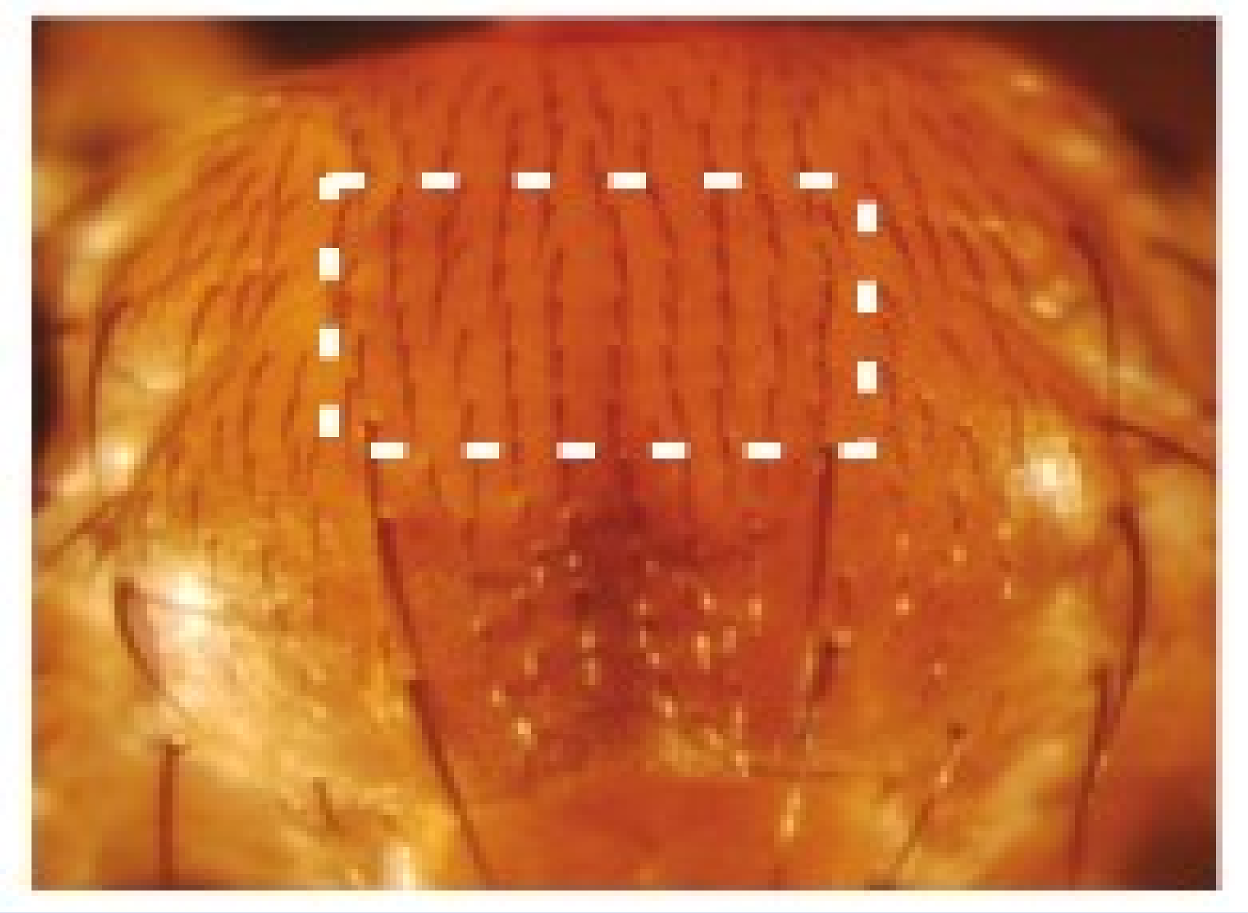
- Identical nodes
- Synchronous model
- In a round, a node can only tell whether or not it received a message
  - It cannot count the number of messages it received

kubuntu

Previous work

- Distributed MIS impossible using deterministic algorithms [Cohen et al. 1984]
- Polylogarithmic time probabilistic algorithms [Luby 1986, Alon et al. 1986]
  - Require knowledge of number of active neighbours (nodes who have not yet been assigned to A or V-A)
  - Require messages of size function of the number of nodes in the network

A biological perspective



- Sensory organ precursors of the fly's bristles
- SOPs form a MIS computed over a set of initially indifferentiated cells
- The MIS is computed using a biological process that roughly follows the beacon algorithm described in the next slide

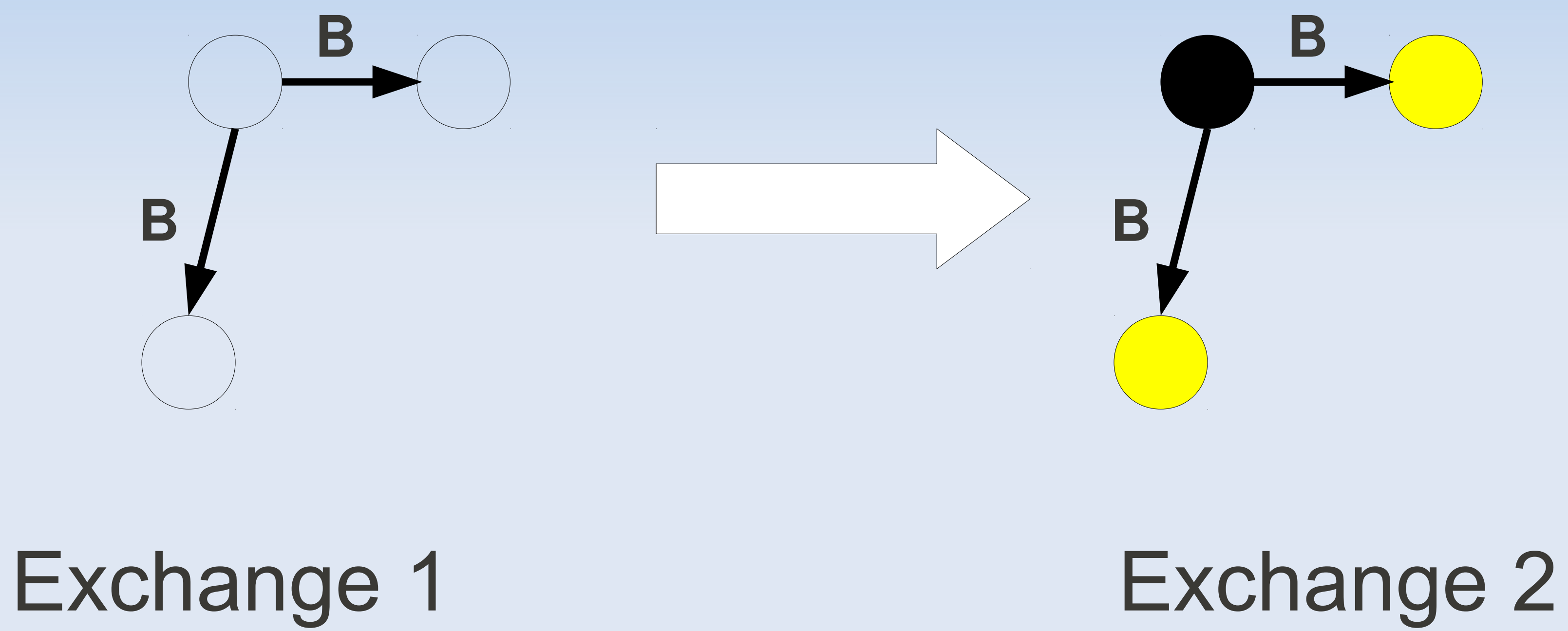
kubuntu

Beacon algorithm

1. Algorithm: MIS (n, D) at node u
2. For i = 0: log D
    3. For j = 0: M log n // M is constant derived below
        4. * exchange 1*
        5. $v = 0$
        6. With probability $\frac{1}{2^{\log D - i}}$ broadcast ***B*** to neighbors and set $v = 1$ // ***B*** is one bit
        7. If received message from neighbor, then $v = 0$
        8. * exchange 2 *
        9. If $v = 1$ then
            10. Broadcast B; join MIS; exit the algorithm
        11. Else
            12. If received message B in this exchange, then mark node u inactive; exit the algorithm
        13. End
    14. End
15. End

- Active node: a node for which a decision has not been made yet

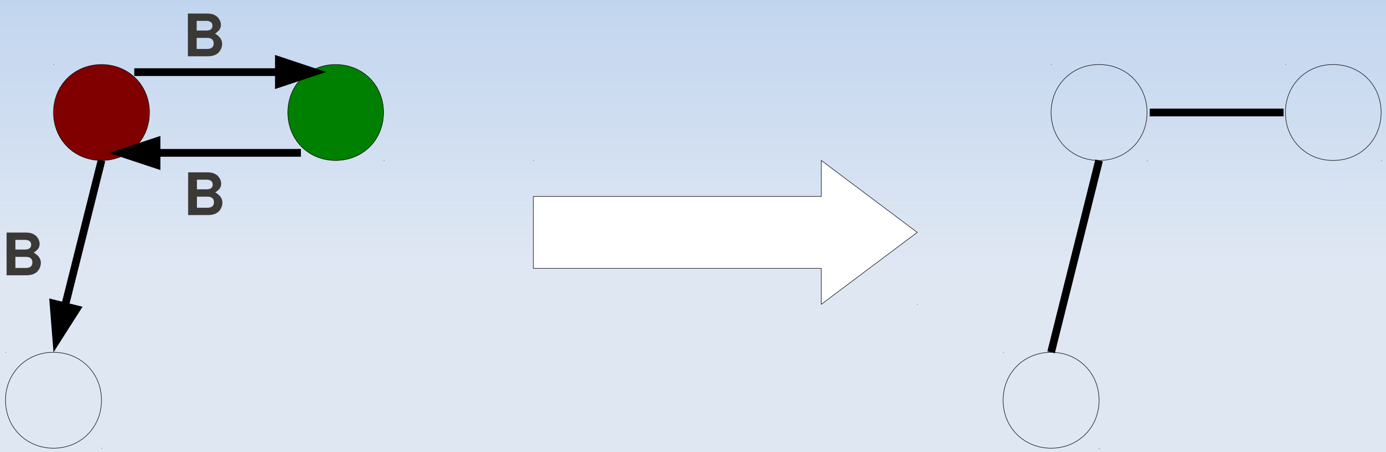
Generic round: first possibility



- Black node joins A, yellow nodes don't
- All three nodes become inactive

kubuntu

Generic round: second possibility

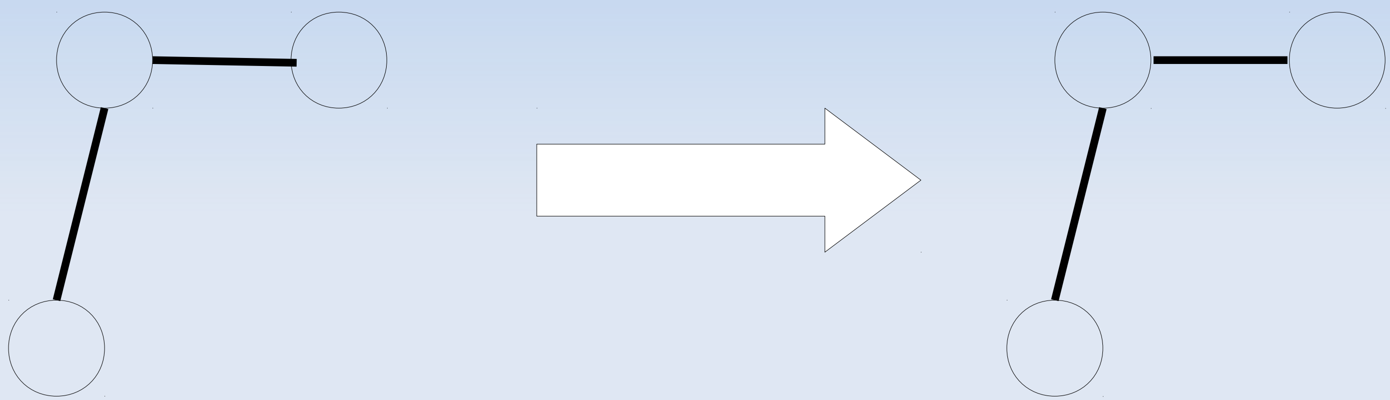


Exchange 1

Exchange 2

- Both red and green node send beacons
- All three nodes remain active

Generic round: third possibility



Exchange 1 Exchange 2

- No broadcast
- All three nodes remain active

Properties of the algorithm

- Lemma 1: No two nodes in A are connected to each other
- Lemma 2: if node w becomes inactive and it does not belong to A, then it is adjacent to a node in A
- Corollary 3: if run forever, the algorithm eventually produces a MIS for G
- Proofs of lemmas 1 and 2: see previous two slides

kubuntu

Running time

Theorem 4: with probability at least $1 - \frac{\log D}{n^2}$

all nodes are either in A or adjacent to a node in A by the end of the algorithm

Corollary 5 with high probability, the algorithm computes a MIS for G in $O(\log^2 n)$ rounds

kubuntu

Proof of thm. 4

1) Lemmas 1 and 2 ensure that the only reason why the algorithm does not compute a MIS for G is that there are still active nodes when it terminates

2) The proof follows from the following Lemma 6: with probability at least $1-\frac{i}{n^2}$ there are no nodes with degree $>\frac{D}{2^i}$ at the end of phase i

- Phase: an iteration of lines 3 – 14 of the algorithm
- Degree of u in a phase: # u's active neighbours + 1

kubuntu

Proof of Lemma 6/1

- By induction on i
- Trivial for i = 0
- Assume true for i – 1 and consider node v with > $D/2^i$ neighbours. Then:

$$P\left(v \vee neighbour\ of\ v\ broadcasts\right) \geq 1 - \frac{1}{e}$$

kubuntu

Proof of Lemma 6/2

- On the other hand, as a node broadcast a message:

$$P(No\,collisions\,occur) \geq \left(1 - \frac{1}{D/2^i}\right)^{2D/2^i} \approx \frac{1}{e^2}$$

- As a consequence, in any round of phase i:

$$P(v\ removed) \geq \left(1 - \frac{1}{e}\right)\frac{1}{e^2}$$

Proof of Lemma 6/3

- As a consequence:

$$P(v \text{ removed during } i-th \text{ phase}) \geq 1 - \frac{1}{n^3}$$

- Proof of lemma then follows since i) at most n nodes, ii) above bound and iii) induction hypotheses

Proof of Thm. 4/cont.

- From Lemma 6, all nodes left in the algorithm at the end of phase logD have degree 1 with probability at least $1-\frac{\log D}{n^2}$
- They have no active neighbours and thus they will insert themselves into A