On the Complexity of Universal Leader Election

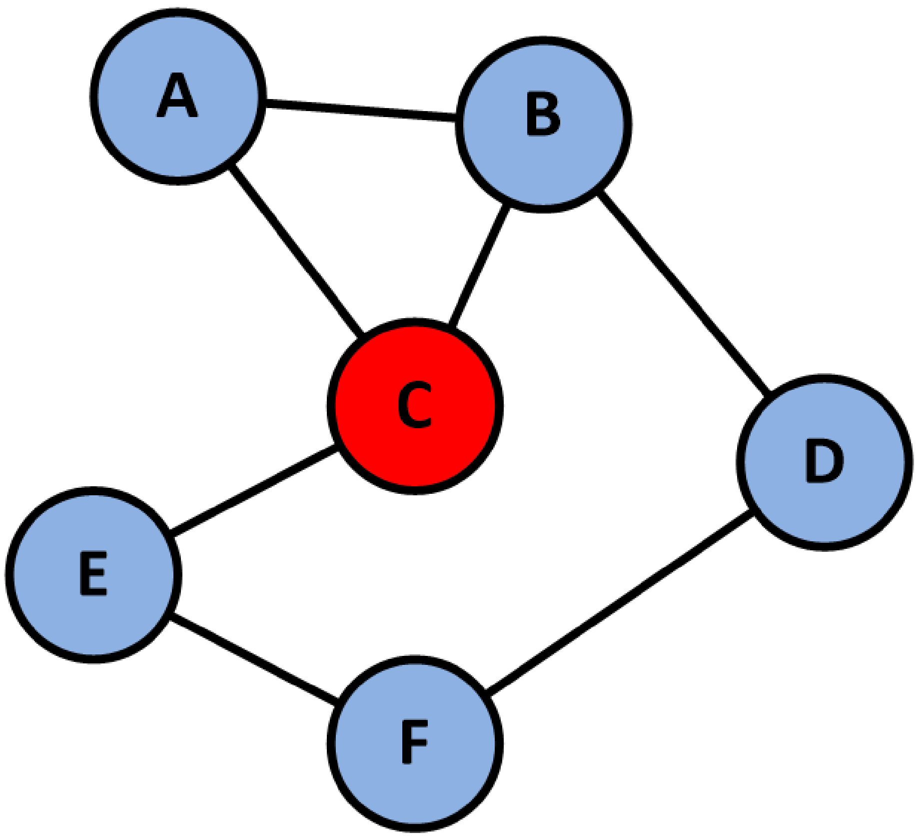
Paper by S. Kutten, G. Pandurangan, D. Peleg, P. Robinson and A. Trehan

Presentation by Adrian-Philipp Leuenberger

Overview

- Introduction and definitions
- Proofing the complexities
- Example algorithm

Leader Election – What is it?



Leader Election – What is it?

- Network nodes elect *unique* leader among themselves
- Implicit: Only leader knows that he is the leader
- Explicit: All nodes know the leader
  - Not focus of paper
- Important for resource-constrained networks
  - Peer-to-peer networks
  - Ad-hoc networks
  - Sensor networks

Definitions

<u>Monte Carlo algorithm</u>

- Randomized algorithm
- Delivers correct result with probability $P = 1 - \varepsilon$, $\varepsilon > 0$

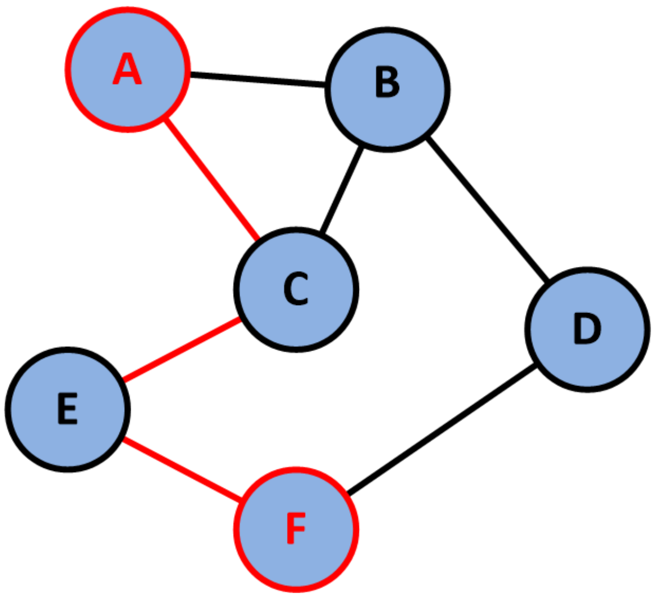
<u>Universal leader election algorithm</u>

- Take any n and m
- Algorithm succeeds on any graph with n nodes and m edges
- With success probability 1 – ε

Definitions

Network Diameter D

- Longest shortest path between any two nodes



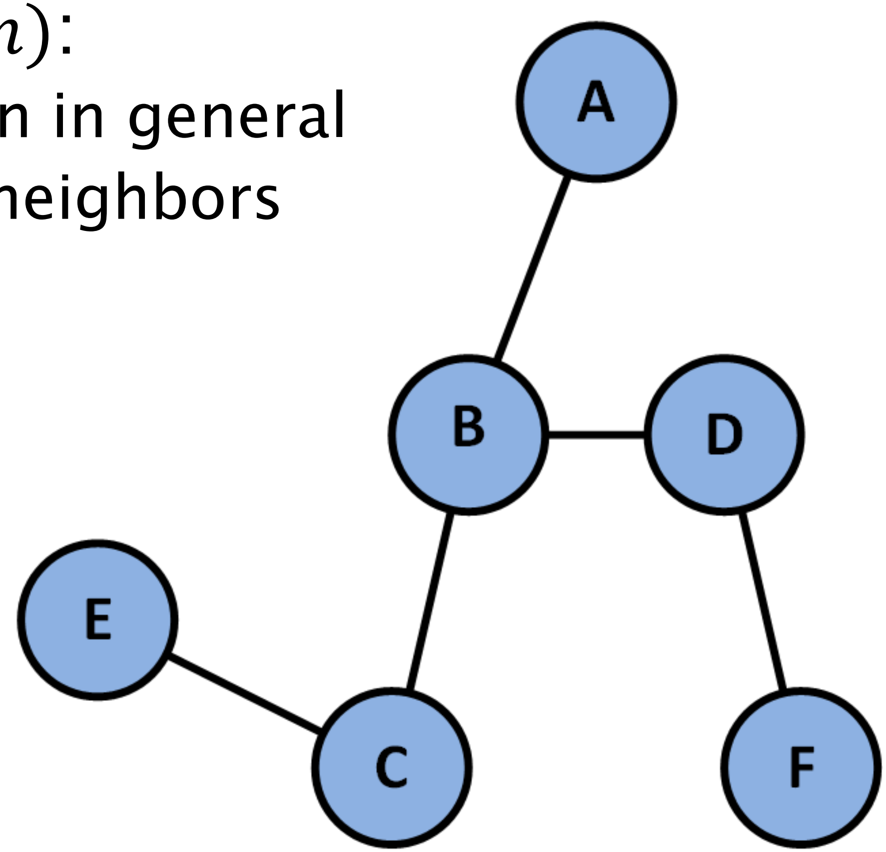
- Here, $D = 3$

So what's the paper about?

- Focus on *universal* LE algorithms
- Worst case analysis for message and time complexity
- Lower bounds:
  - Time complexity $\Omega(D)$
    - Network diameter D
  - Message complexity $\Omega(m)$
    - m edges
- Algorithms that meet the lower bounds

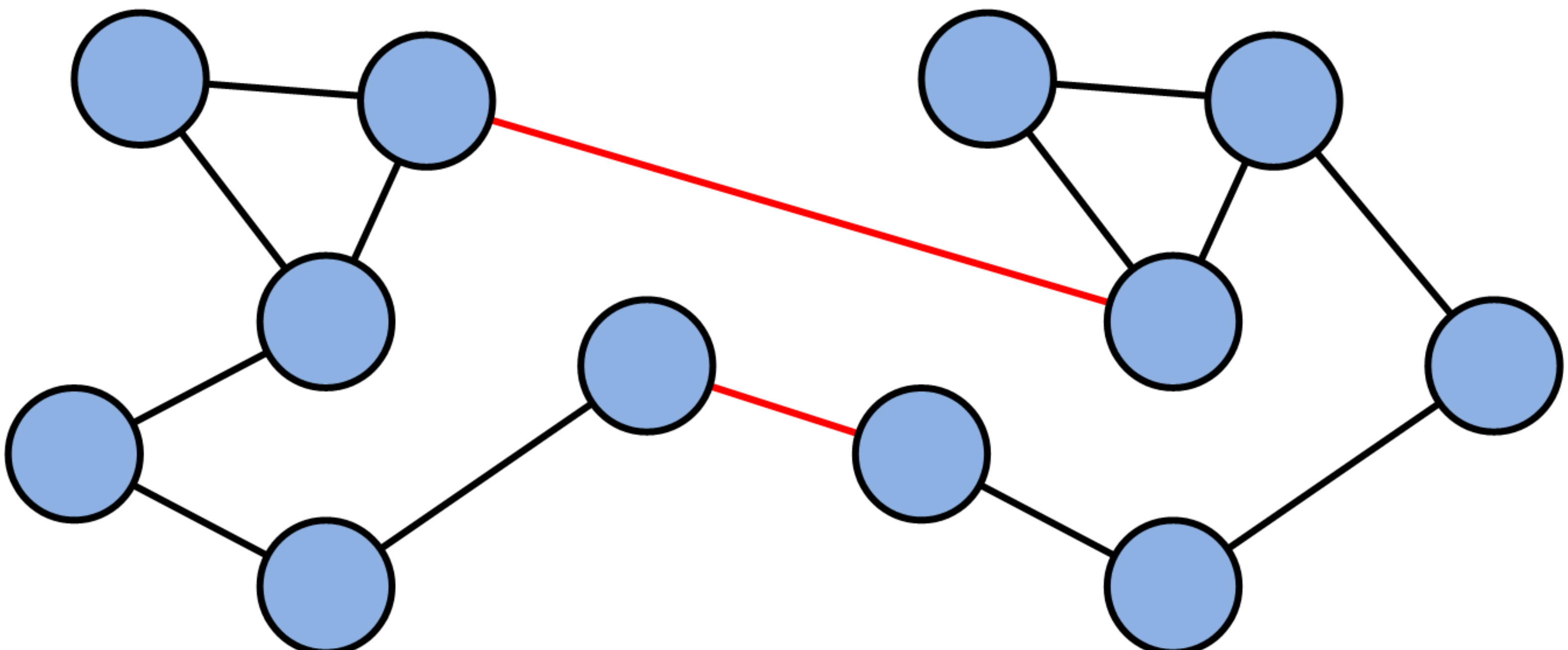
But why *those* lower bounds?

- Time complexity $\Omega(D)$:
  - Worst case: Send message on longest shortest path
- Message complexity $\Omega(m)$:
  - Network topology unknown in general
  - Must send message to all neighbors



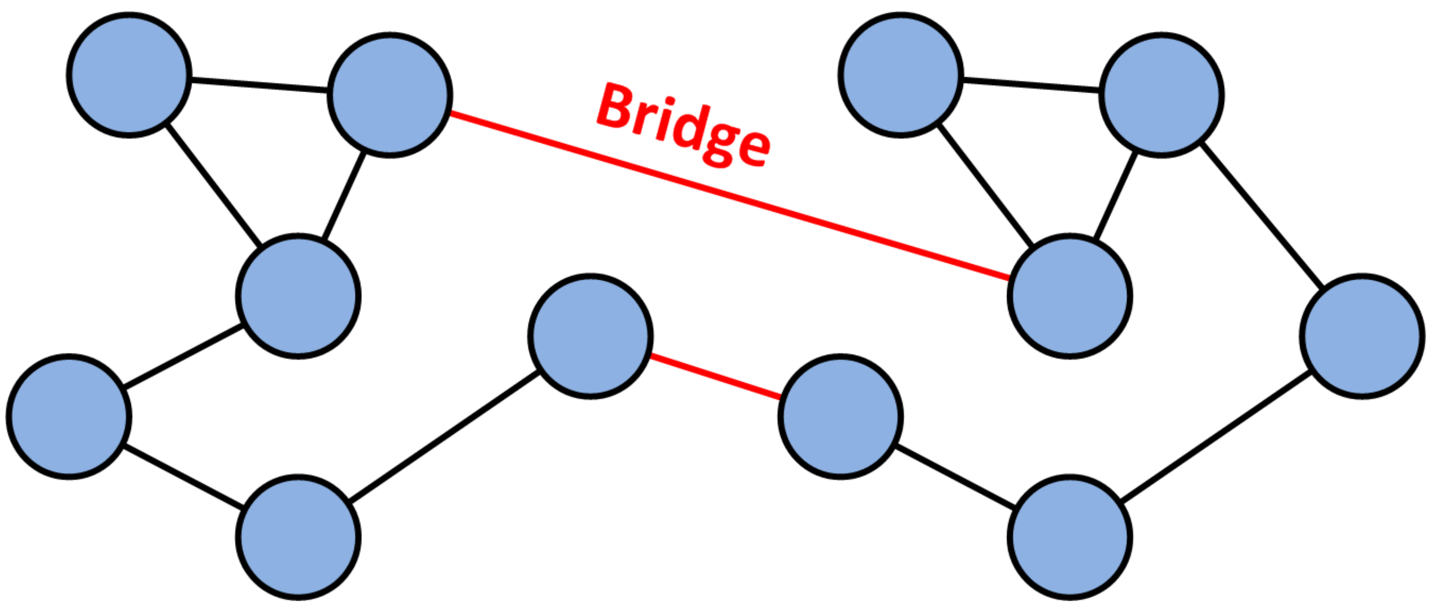
Dumbbell graphs

- Take a 2-connected graph G
  - n nodes, m edges
- m edges ➔ $2m^2$ possible dumbbell graphs
- $\boldsymbol{I}$: collection of all dumbbell graphs for G



Bridge crossing

- Algorithm B solves BC iff a message is sent over a bridge



Proving $\Omega(m)$ for LE – Proof idea

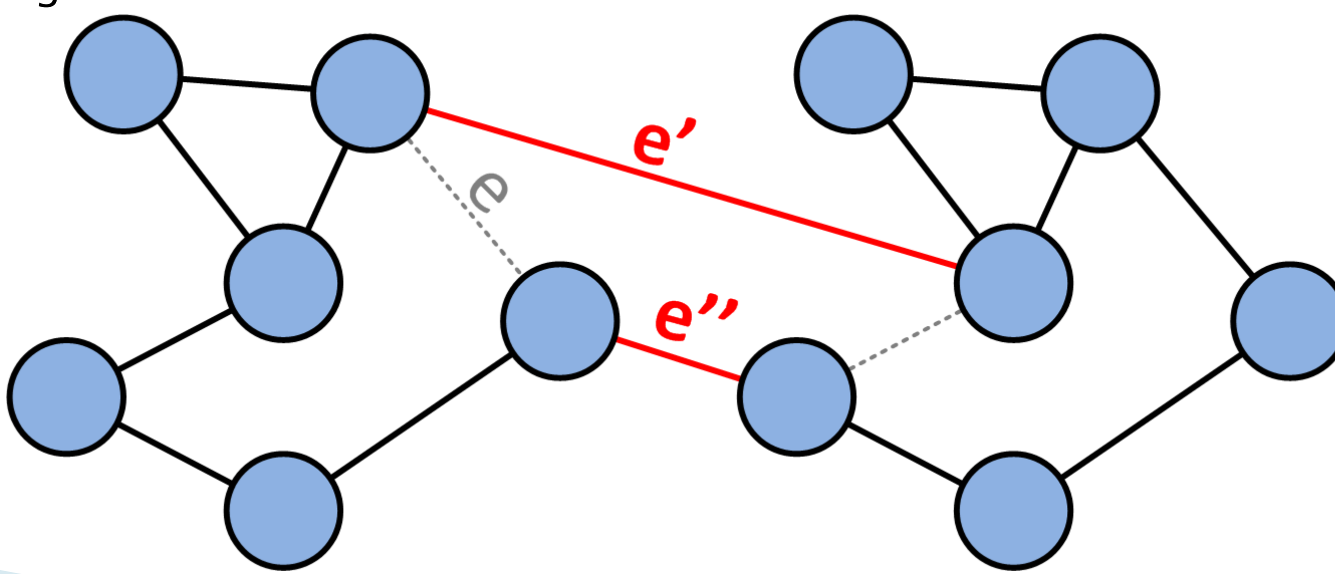
- Reduce *Bridge Crossing* to *Leader Election*
  - Show $\Omega(m)$ lower bound for *Bridge Crossing*
    - ➔ Imply $\Omega(m)$ lower bound for *Leader Election*
- Proof lower bound $\Omega(m)$ for message complexity for *Bridge Crossing*
- Use *Dumbbell* graphs for the proof

Proving $\Omega(m)$ for BC – High level proof idea

- Take any *deterministic* BC algorithm B
- $T(e)$: First round a message passes edge e in disconnected graph

- After T rounds:
  - At least T messages

- Two cases:
  - $T(e) = T(e')$
  - $T(e) = T(e'')$



Proving $\Omega(m)$ for LE – Step 1

- Assumption:
  - Universal LE algorithm R
    - Success probability $1 - \beta$
  - Deterministic LE algorithm A
    - Solves LE on at least a $1 - 2\beta$ fraction of $\boldsymbol{I}$
- Lemma 1:
  - ε and $\delta \geq$ ¼ positive constants with $7\varepsilon + \delta \leq 1$
  - A solves LE on at least a $1 - \varepsilon$ fraction of $\boldsymbol{I}$

  ➔ A solves BC on at least a δ fraction of $\boldsymbol{I}$
- Therefore, with $\varepsilon = 2\beta$:
  - LE algorithm A achieves BC on $\delta \geq$ ¼ of all graphs in $\boldsymbol{I}$.

Proving $\Omega(m)$ for LE – Step 2

- Assumption:
  - Universal LE algorithm R
    - Success probability $1 - \beta$
  - Deterministic LE algorithm A
    - Solves LE on at least a $1 - 2\beta$ fraction of $\boldsymbol{I}$
- We know:
  - A achieves BC on at least ¼ of all graphs in $\boldsymbol{I}$.
- Lemma 2:
  - If A solves BC on at least ¼ of all graphs in $\boldsymbol{I}$
  - Then expected message complexity is $\Omega(m)$
- Therefore:
  - Algorithm A has an expected message complexity of $\Omega(m)$.

Proving $\Omega(m)$ for LE – Step 3

- Assumption:
  - Universal LE algorithm R
    - Success probability $1 - \beta$
  - Deterministic LE algorithm A
    - Solves LE on at least a $1 - 2\beta$ fraction of $\boldsymbol{I}$
- We know:
  - A achieves BC on at least ¼ of all graphs in $\boldsymbol{I}$.
  - A has an expected message complexity of $\Omega(m)$.

- Lemma 3 (Yao's Minmax Principle):
  - If A has cost X and success rate at least $1 - 2\beta$ on $\boldsymbol{I}$
  - Then R has worst case cost of at least $X/2$ and success probability $1 - \beta$ on $\boldsymbol{I}$
- Therefore:
  - If A succeeds on at least $1 - 2\beta$ fraction of $\boldsymbol{I}$ with $\Omega(m)$ messages
  - Then R must succeed with probability $1 - \beta$ and $\Omega(m/2) = \Omega(m)$ messages.

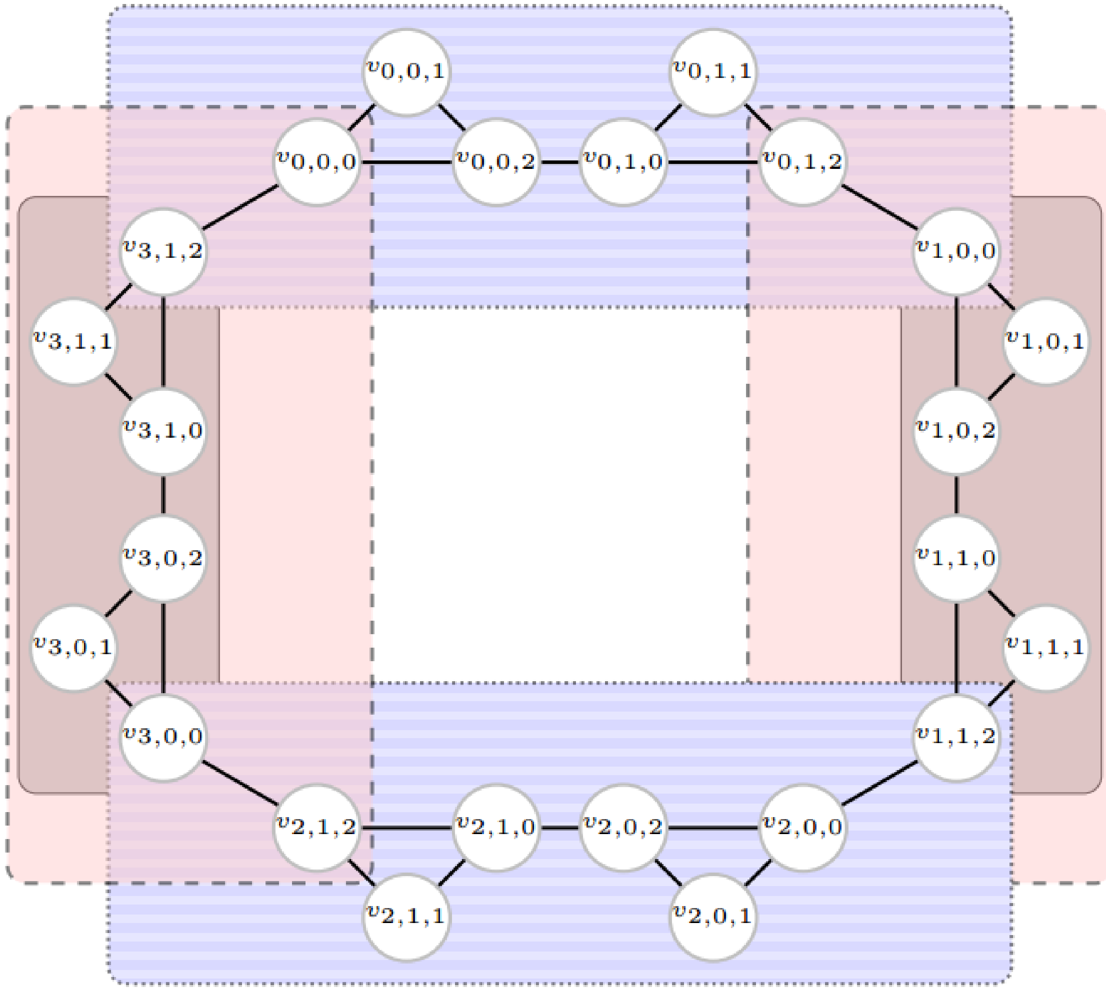
Proving $\Omega(m)$ – What just happened

1. Deterministic LE algorithm A likely solves bridge crossing
2. Bridge crossing: $\Omega(m)$ messages in expectation
3. LE algorithm A must have expected message complexity $\Omega(m)$
4. Cost of A implies lower bound for randomized algorithm R ➔ $\Omega(m)$ messages expected for any R

Proving Ω(D) – The idea

- Take any n and D
  - $D' = 4\lceil ^{D}/_{4} \rceil$ cliques
  - $\gamma(n) * D' \geq n$ nodes per clique
  - 4 neighborhoods or *arcs*
  - Execution time T

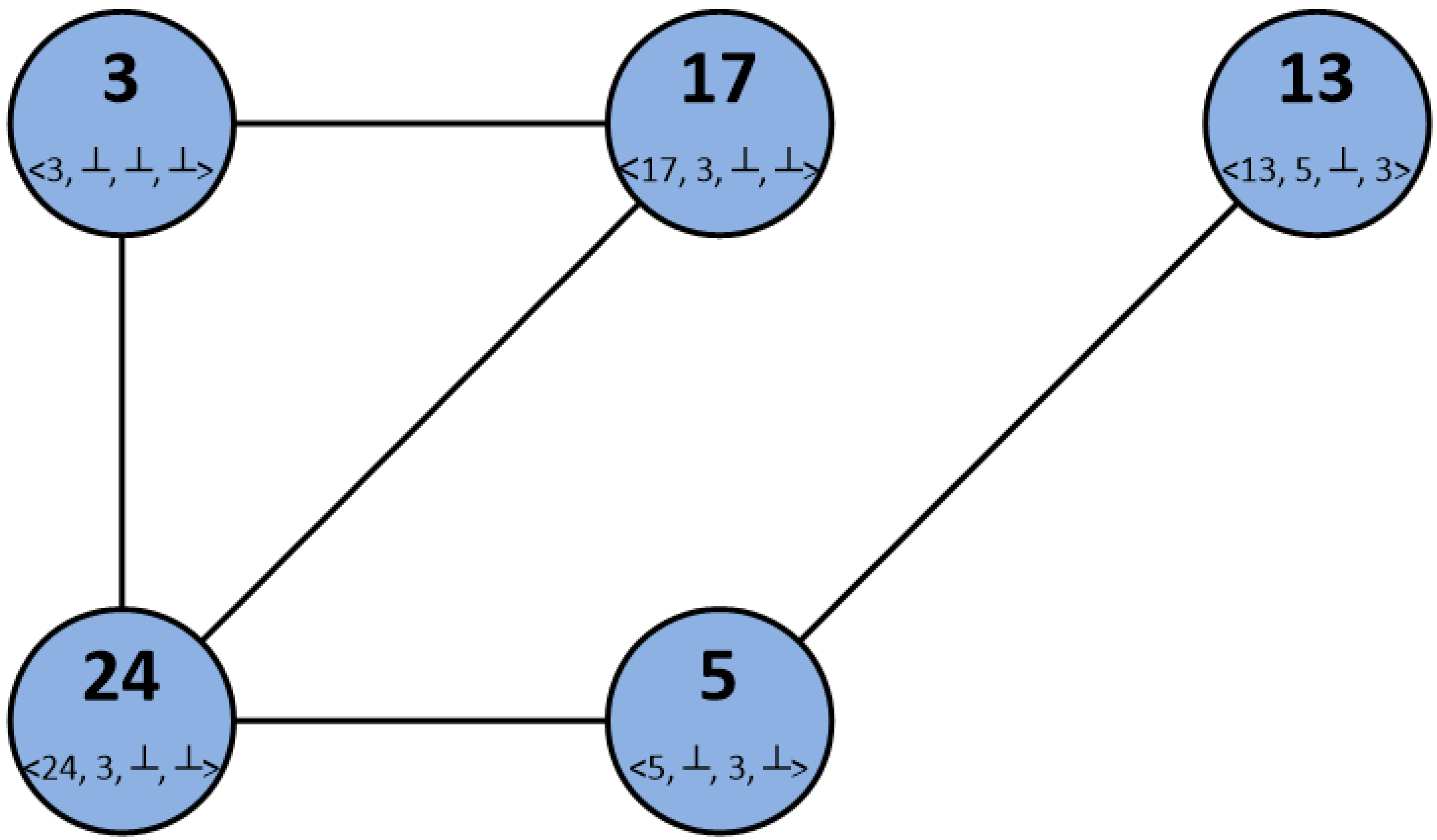
- Two cases:
  - $T \in o(D)$ with $p = \delta$
  - $T \in \Omega(D)$ with $p = 1 - \delta$



Example algorithm: The *basic* Least Element algorithm

- Each node n keeps track of its local state
  - Rank $\rho(n) \epsilon [1, n^4]$
  - List of all least ranks of its neighbors
- Nodes choose their rank $\rho(n)$ randomly
- Succeeds if there is only one node with least rank

The *basic* Least Element algorithm



The *basic* Least Element algorithm

- Observations
  - In each round
    - Node n forwards at most one message to neighbors
    - At most $2m$ rank messages in total
- Time complexity is $O(D)$
  - At most D time units to forward on longest shortest path
- *Expected* message complexity is $O(m \log n)$
  - $O(m)$ messages sent per round
  - $O(\log n)$ messages stored and forwarded per node

The *improved* Least Element algorithm

- Try to achieve $O(m)$ message complexity instead of $O(m \log n)$

- Take any function $f(n) \leq n$
- A nodes becomes candidates with probability ${}^{f(n)}/_{n}$
- Candidates
  - Choose rank rank from $[1, n^4]$
  - Forward own rank
- Non-candidates
  - Choose rank $n^4 + 1$
  - Only update list and forward received ranks

- Algorithm succeeds if
  - At least one node chooses to be a candidate
  - There is only one node with least rank

The *improved* Least Element algorithm (cont'd)

- Time complexity of improved version is still $O(D)$
- Message complexity is $O(m * \min(\log f(n), D))$
- Success probability is $1 - 1/e^{\Theta(f(n))}$

- Choose $f(n) = 4\log(1/\varepsilon)$ for some constant $\varepsilon > 0$, then
  - Success probability at least $1 - \varepsilon^{\Theta(1)}$
  - Message complexity is $O(m * \min(\log\log(1/\varepsilon), D)) = O(m)$

What was shown

- Worst case lower bounds for universal LE algorithms:
  - $\Omega(D)$ time complexity
  - $\Omega(m)$ messages
- Algorithm that also matches the bounds

References

- On the Complexity of Universal Leader Election
  - Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson and Amitabh Trehan, PODC ´13
- Efficient Distributed Approximation Algorithms via Probabilistic Tree Embeddings
  - Maleq Khan et. al., PODC ´08

Any questions?