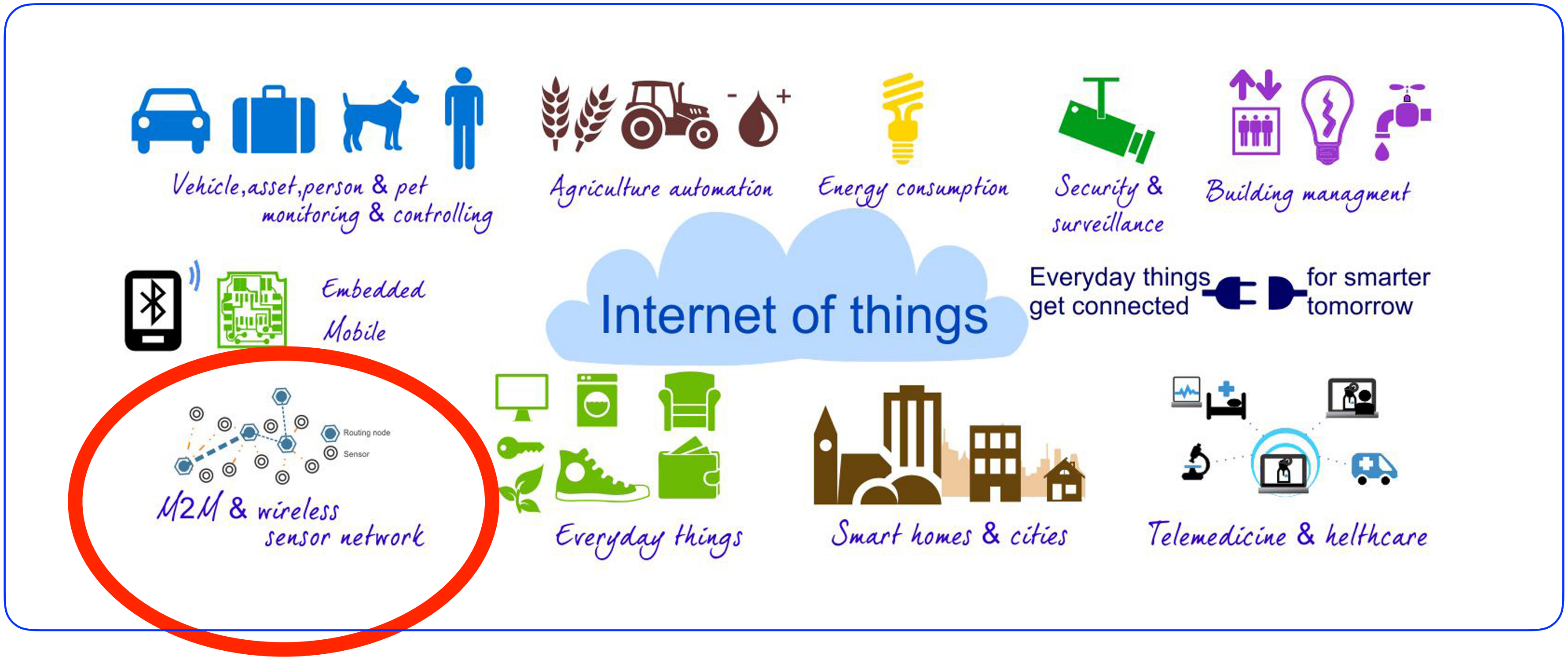
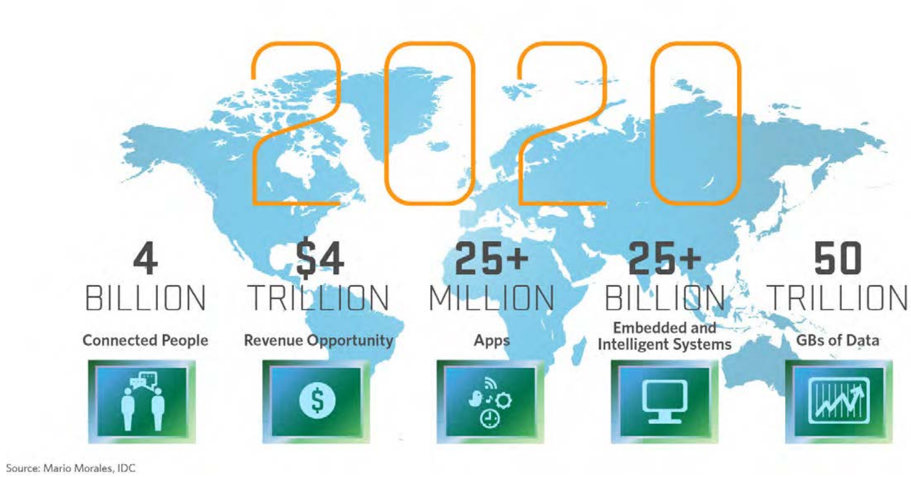
Counting in Practical Anonymous Dynamic Networks is Polynomial

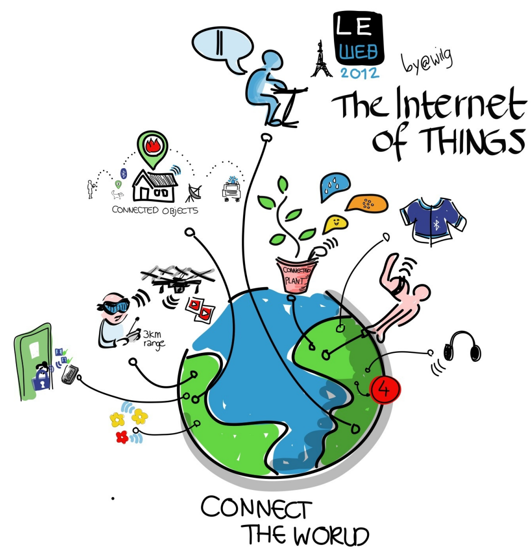
Maitri Chakraborty, Alessia Milani, and
Miguel A. Mosteiro

NETyS 2016

The Internet of Things



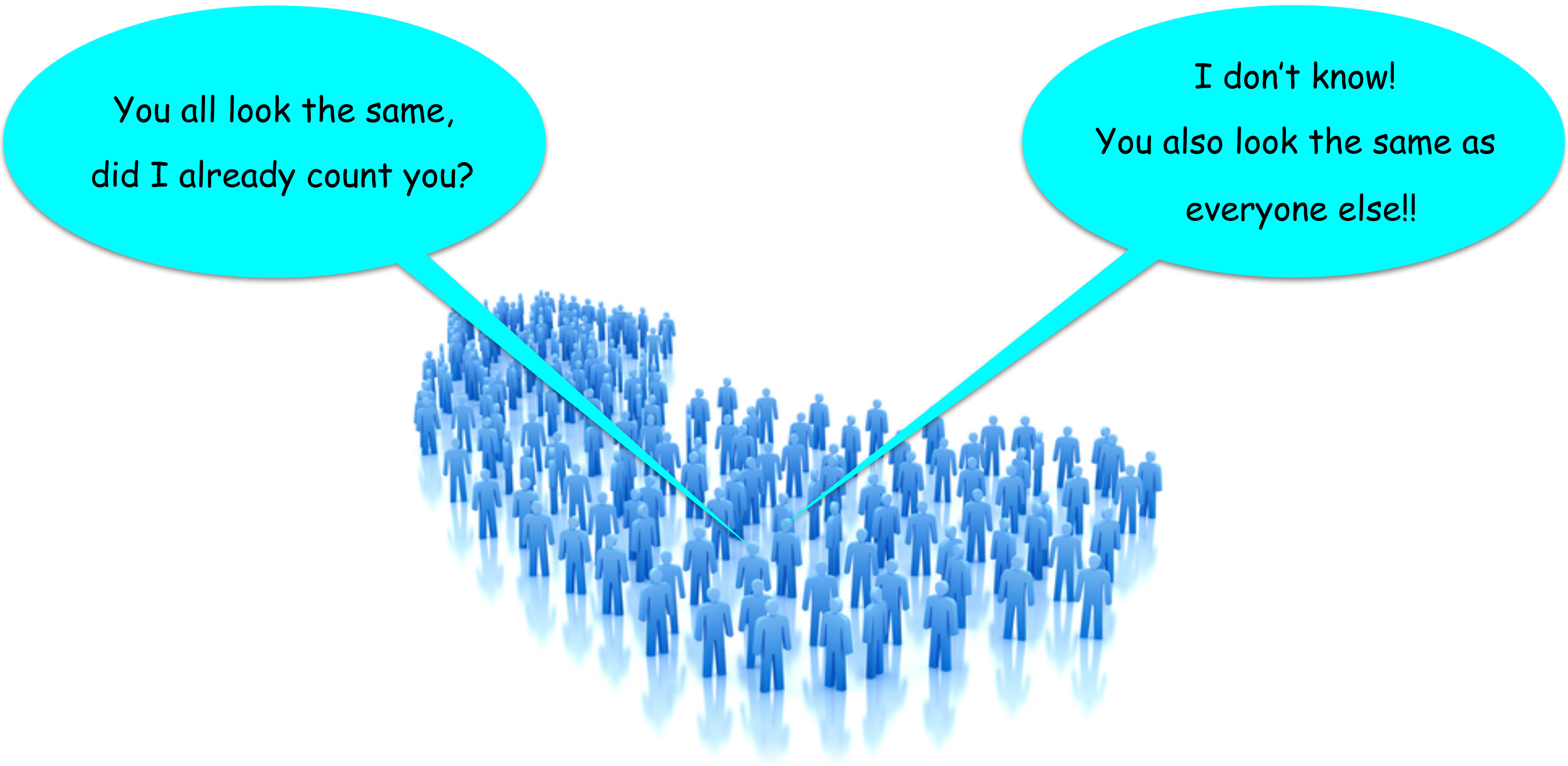




The Counting Problem

How do you count the size of your group,

if the members are all identical and move?



Why do we care?

The problem is clean, but why do we care?

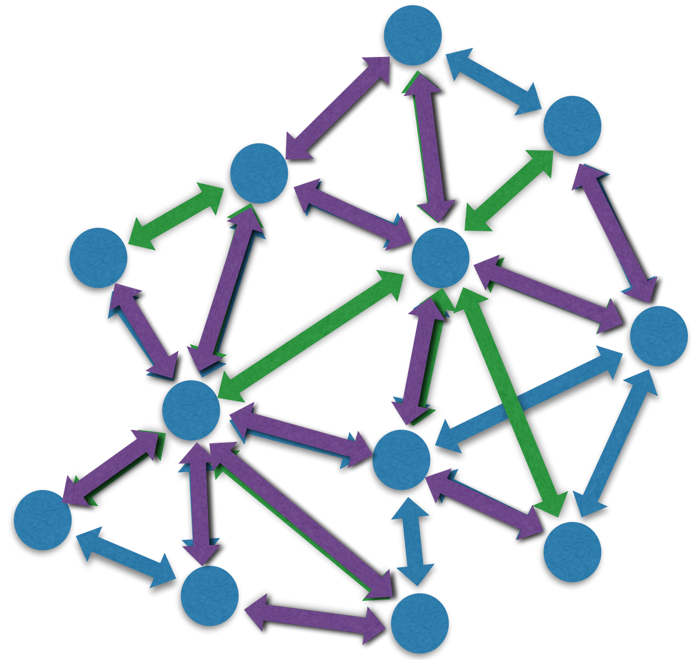
Distributed protocols

need the number of processors to decide termination.

We need a protocol: « Given a system of n nodes,

all nodes eventually terminate knowing n »

Anonymous Dynamic Networks



- **Fixed set of n nodes**
  - No identifiers or labels
  - A special node, called the leader [1]
- **Synchronous communication** : At each round
  - a node broadcasts a message to its neighbors
  - receives the messages of its neighbors
  - executes some local computation
- **1-interval connectivity** [2]
  - communication links may change from round to round, but
  - at each round the network is connected
- **An upper bound Δ** on the maximum degree is known by all nodes

[1] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. SSS 2013

[2] Fabian Kuhn, Nancy A. Lynch, Rotem Oshman. Distributed computation in dynamic networks. STOC 2010

Previous work

- **Previous Counting Protocols**
  - Guarantee only an exponential upper bound on the network size [1] or
  - They guarantee the exact size but
    - Take double-exponential number of rounds [2] or
    - Take exponential number of rounds, but do not terminate [2] or
    - Terminate but no running-time guarantees [3]
  - (very) recently, exact-size exponential time Counting with termination:
    - [5] Incremental Counting (IC): poly space.
    - [6] EXT Counting: no Δ but exponential space.

Exponential speedup, but still not practical

- **Lower bound on the time complexity**
  - Ω(D) where D is the dynamic diameter and
  - Ω(logn) even if D is constant [4]

Huge gap

[1] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. SSS 2013
[2] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Conscious and unconscious counting on anonymous dynamic networks. ICDCN 2014
[3] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Counting in anonymous dynamic networks under worst-case adversary. ICDCS 2014
[4] G. A. Di Luna and R.Baldoni. Investigating the cost of anonymity on dynamic networks. 2015.
[5] A. Milani and M. A. Mosteiro. A faster counting protocol for anonymous dynamic networks. OPODIS 2015.
[6] R.Baldoni and G. A. Di Luna. Non Trivial Computations in Anonymous Dynamic Networks. OPODIS 2015.

Contributions

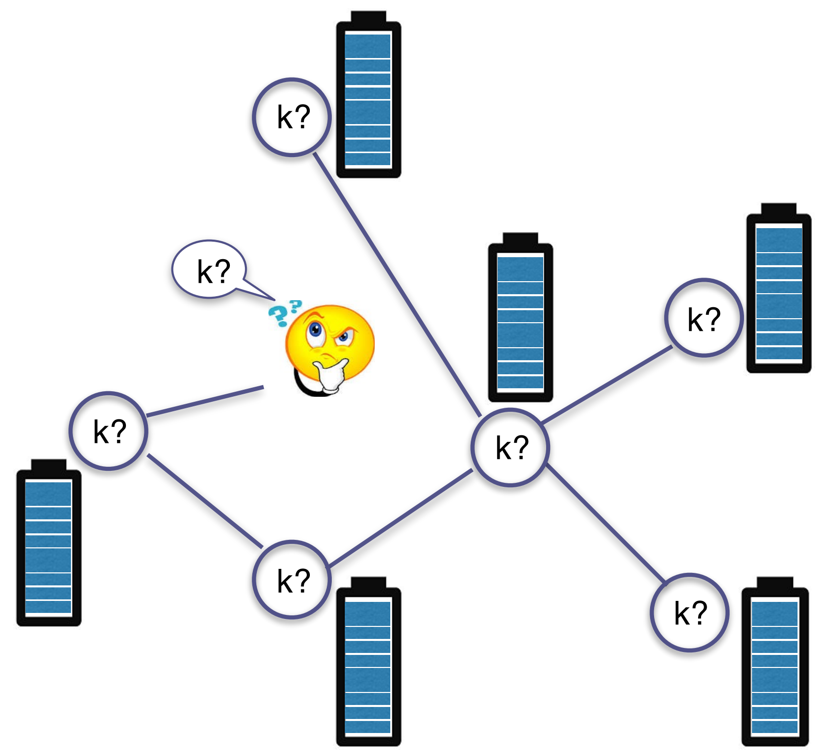
- Experimental evaluation of Incremental Counting:
  - Incremental Counting is polynomial (and practical)
  - variety of input network topologies that may appear in practice
  - insight on network dynamics impact on dissemination

Incremental Counting

Initially, each non-leader has "energy" 1

1. Guess a size k of the system
   - start from k=2

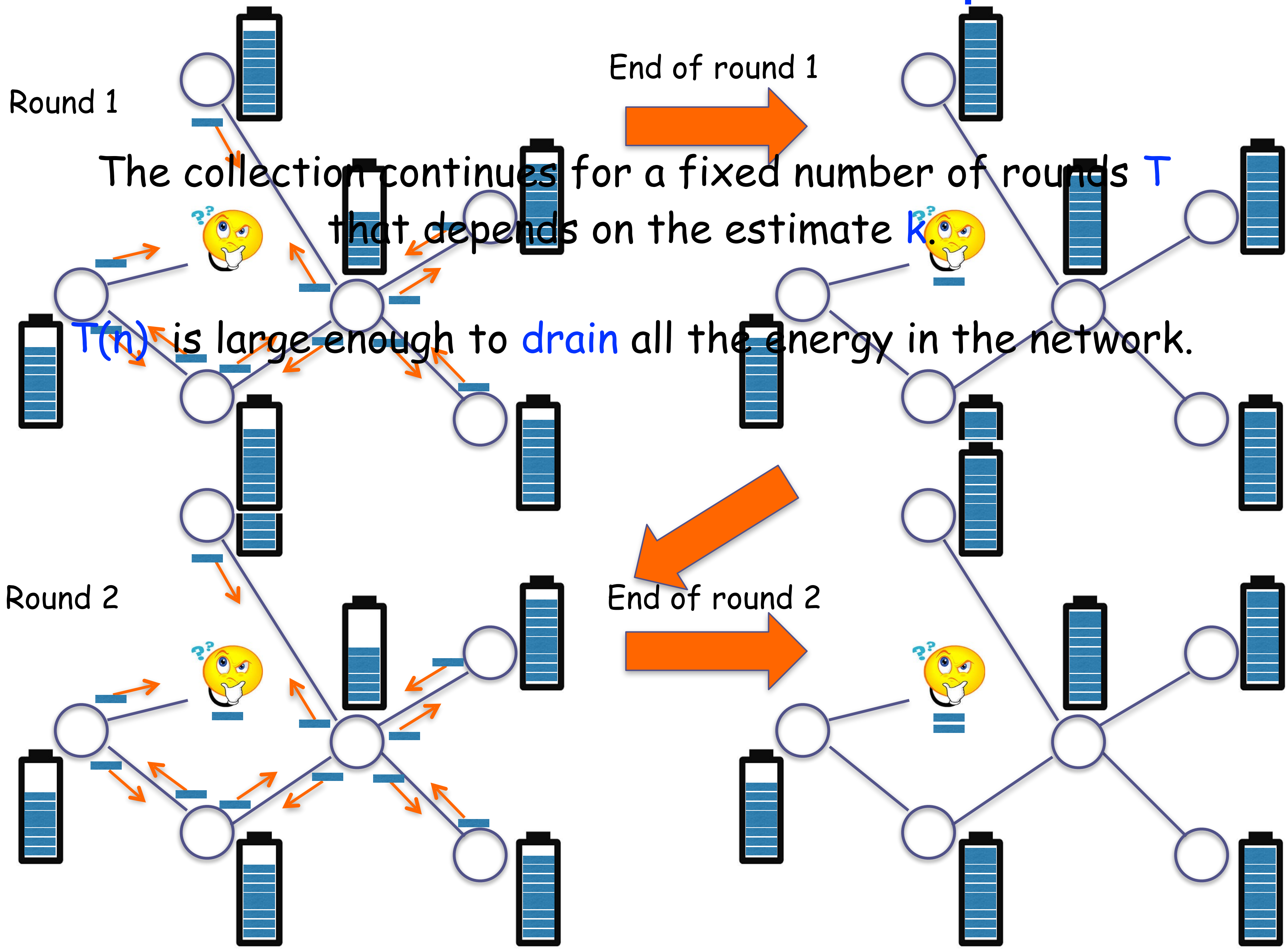
2. **Collection** phase:
   - to let the leader collect "enough" energy



IC Collection Phase example

The collection continues for a fixed number of rounds T that depends on the estimate k.

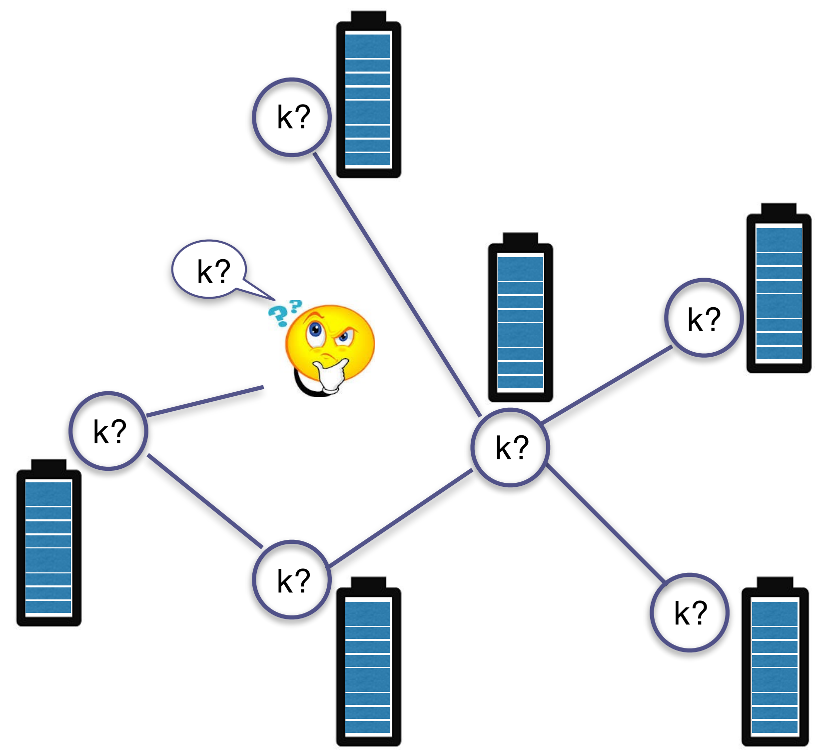
T(n) is large enough to drain all the energy in the network.



Incremental Counting

Initially, each non-leader has "energy" 1

1. Guess a size k of the system
   – start from k=2
2. **Collection** phase:
   – to let the leader collect "enough" energy
3. **Verification** phase:
   – to check whether the guess k is correct



Challenges for Verification

<u>IC Verification Phase</u>:

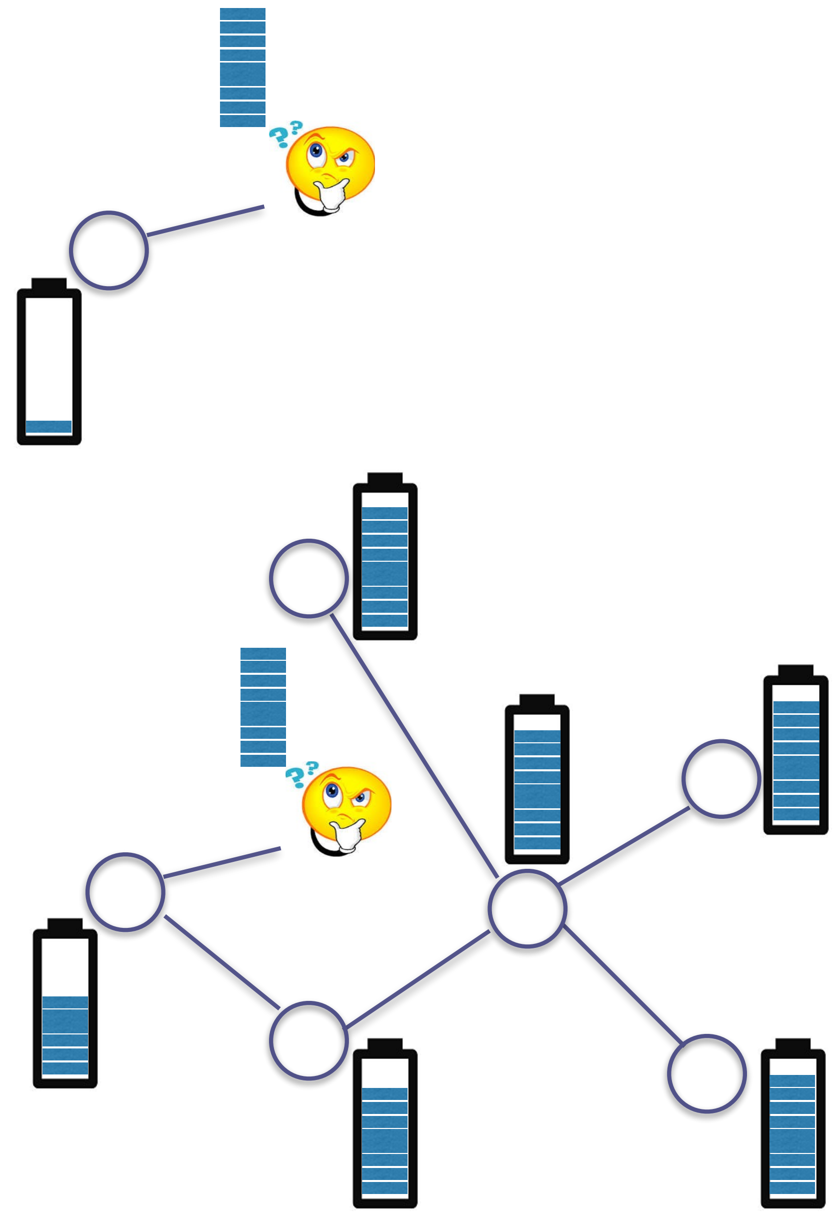
If $e_{leader} > k-1 \Rightarrow k < n$

But,

if $k-1-1/k^c \leq e_{leader} \leq k-1$???

$\Downarrow$

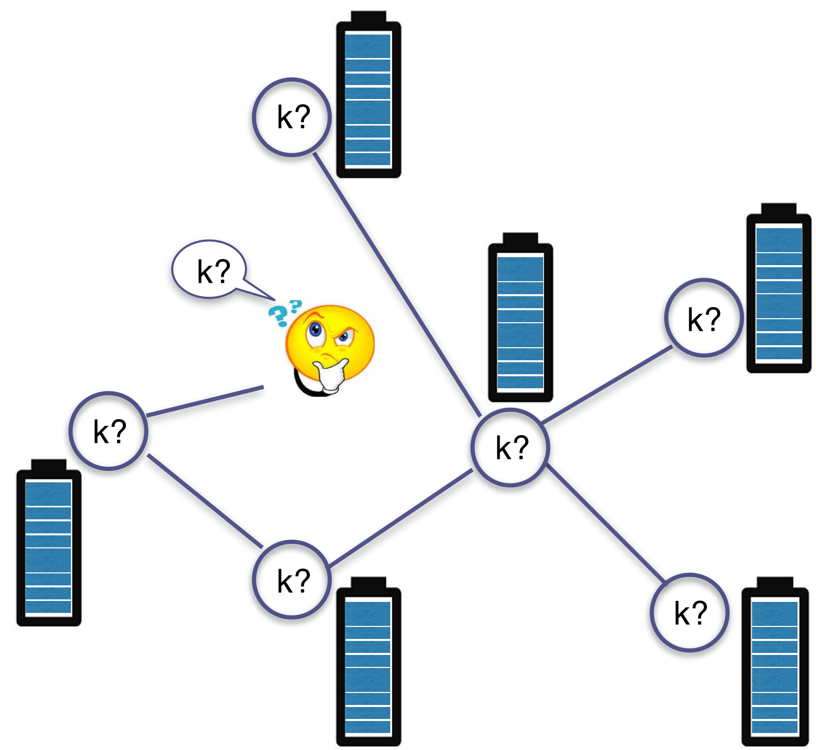
Need to check if some non-leader has more than $1/k^c$ energy left.



Incremental Counting

Initially, each non-leader has “energy” 1

1. Guess a size k of the system
   - start from k=2
2. **Collection** phase:
   - to let the leader collect “enough” energy
3. **Verification** phase:
   - to check whether the guess k is correct
4. **Notification** phase:
   - k=n : let all nodes know that k is the size
   - k<n : wait and go to step 2, guessing k+1



Incremental Counting

Initially, each non-leader has "energy" 1

1. Guess a size k of the system
   - start from k=2
2. **Collection** phase:
   - to let the leader collect "enough" energy
3. **Verification** phase:
   - to check whether the guess k is correct
4. **Notification** phase:
   - k=n : let all nodes know that k is the size
   - k<n : wait and go to step 2, guessing k+1

Worst-case analysis:

IC computes exact size

in less than

$(2\Delta)^{n+1}(n+1)\ \ (n+1)/\ln(2\Delta)$

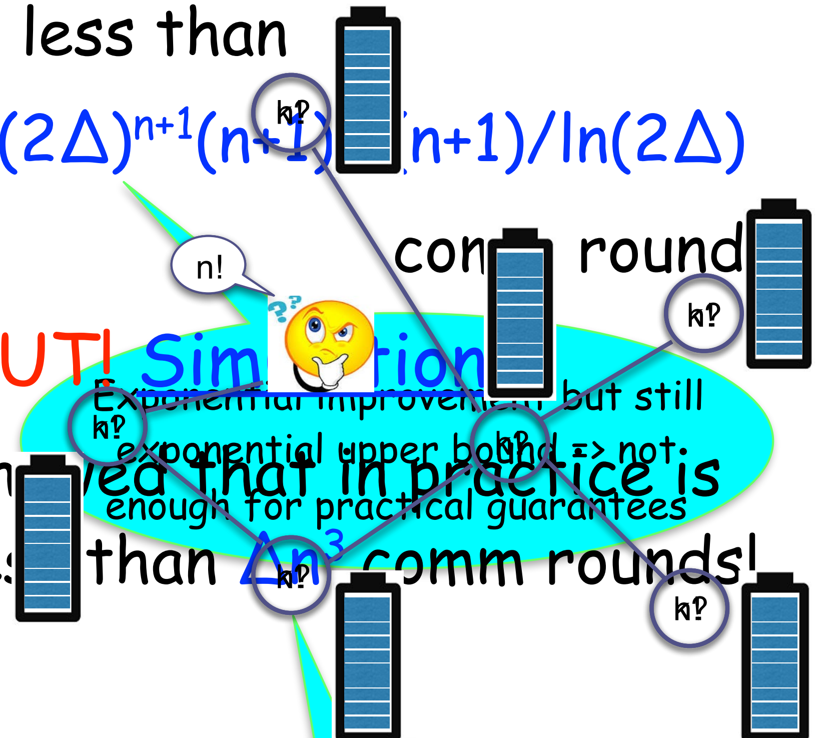
com round

BUT! Simulation

sh ed that in practice is

les than Δn^3 comm rounds!

Exponential improvement but still exponential upper bound => not enough for practical guarantees



Practical for IoT subnetworks!!

IC Simulations: inputs

- Extremal cases:
  - path with the leader in one end
  - star centered at the leader
- Random graph (Erdos-Renyi)
  - need to handle disconnections
- Random tree rooted at the leader:

Uniform from equivalence classes defined by isomorphisms

Pruned down to max degree Δ

Algorithm 2: Random tree generator algorithm. Auxiliary functions in Algorithm 3.

Function GENTREE(n, Δ)
  $t \leftarrow$ SIZES(n) // Compute number of unlabeled rooted trees of size $1, 2, \ldots, n$.
  $p \leftarrow$ DISTRIB(t, n) // Compute distributions on subtrees for each n.
  tree $\leftarrow$ RANRUT(p, n) // Choose an unlabeled rooted tree uniformly at random.
  PRUNE (tree, Δ) // Move subtrees downwards until max degree of tree is Δ.
  return tree

Centralized simulator

Algorithm 1: INCREMENTAL COUNTING algorithm for the leader node.

$k \leftarrow 1$
$halt \leftarrow false$
while $\neg halt$ **do**
  $k \leftarrow k + 1$
  $IsCorrect \leftarrow true$
  $e_\ell \leftarrow 0$
  `// Collection Phase`
  for *each of* $\tau(k)$ *communication rounds* **do**
    receive $e_1, e_2, \ldots e_s$ from neighbors, where $1 \leq s \leq \Delta$
    $e_\ell \leftarrow e_\ell + e_1 + e_2 + \ldots + e_s$
  `// Verification Phase`
  for *each of* $1 + \left\lceil \frac{k}{1-1/k^c} \right\rceil$ *communication rounds* **do**
    receive $e_1, e_2, \ldots e_s$ from neig...ors, where $1 \leq s \leq \Delta$
    if $k - 1 - 1/k^c \leq e_\ell \leq k - 1$ **then**
      for $j := 1 \ldots s$ **do**
        if $e_j > 1/k^c$ **then**
          $IsCorrect \leftarrow false$
    else
      $IsCorrect \leftarrow false$
  `// Notification Phase`
  for *each of* k *communication rounds* **do**
    if $IsCorrect$ **then**
      broadcast $\langle Halt \rangle$
      $halt \leftarrow true$
    else
      do nothing
output k

fixed # rounds for the leader to collect at least k-1-1/k^c energy

fixed # rounds

$k \leftarrow 1$, $IsCorrect \leftarrow$ false, $r \leftarrow 1$, $E \leftarrow$ new se...
while $\neg IsCorrect$ **do**
  $k \leftarrow k + 1$, $IsCorrect \leftarrow$ true
  Collection Phase:
    $(e_1, e_2, e_3, \ldots, e_n) \leftarrow (0, 1, 1, \ldots$
    while $e_1 < k - 1 - 1/k^{1.01}$ **do**
      $(e_1, e_2, \ldots, e_n) \leftarrow \mathbf{F}(E) \cdot (e_1, e_2, \ldots, e_n)^T$ `// broadcast simulation`
      if $r \equiv 0 \mod T$ **then** $E \leftarrow$ new set of links
      $r \leftarrow r + 1$
  Verification Phase:
    if $e_1 > k - 1$ **then** $IsCorrect$ ...
    $(e'_1, e'_2, e'_3, \ldots, e'_n) \leftarrow (0, e_2, e_3 \ldots, e_n$ ...
    for $1 + \lceil k/(1 - 1/k^{1.01}) \rceil$ *iterations* **do**
      for *each* i **do** $e''_i \leftarrow \max_{j \in N(i,E) \cup \{i\}} e'_j$ `// broadcast simulation`
      $(e'_1, e'_2, \ldots, e'_n) \leftarrow (e''_1, e''_2, \ldots, e''_n)$
      if $r \equiv 0 \mod T$ **then** $E \leftarrow$ new set of links
      $r \leftarrow r + 1$
    if $e'_1 > 1/k^{1.01}$ **then** $IsCorrect \leftarrow$ false
  Notification Phase:
    $(h_1, h_2, h_3, \ldots, h_n) \leftarrow (IsCorrect$, ...se, fals...
    halt flags
    for k *iterations* **do**
      for *each* i **do** $h'_i \leftarrow \bigvee_{j \in N(i,E)} h_j$ //
      $(h_1, h_2, \ldots, h_n) \leftarrow (h'_1, h'_2, \ldots, h'_n)$
      if $r \equiv 0 \mod T$ **then** $E \leftarrow$ new set of l...ks
      $r \leftarrow r + 1$
output k

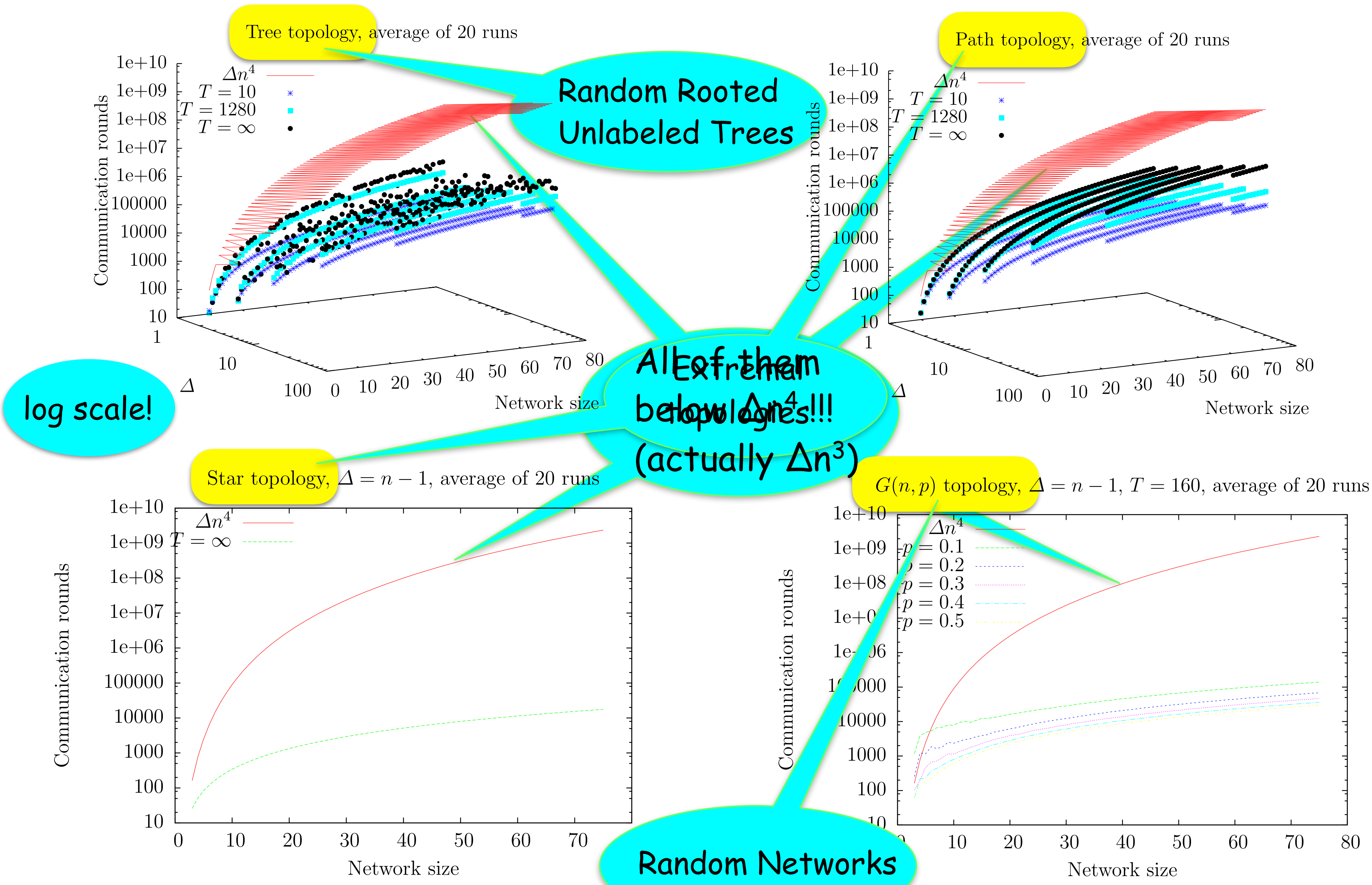
collect until leader has at least k-1-1/k^c energy

parametric Δ

parametric dynamics

to handle disconnection in G(n,p)

IC Simulations



Centralized simulator provides upper bound for distributed implementation.

Future and Ongoing Work

- Improve lower bound.
- Distributed implementation that does not require synchronization.
- Remove the knowledge of Δ using $o(2^n)$ space.
- Other computations in ADNs.

Thank you!