Polynomial Counting in Anonymous Dynamic Networks

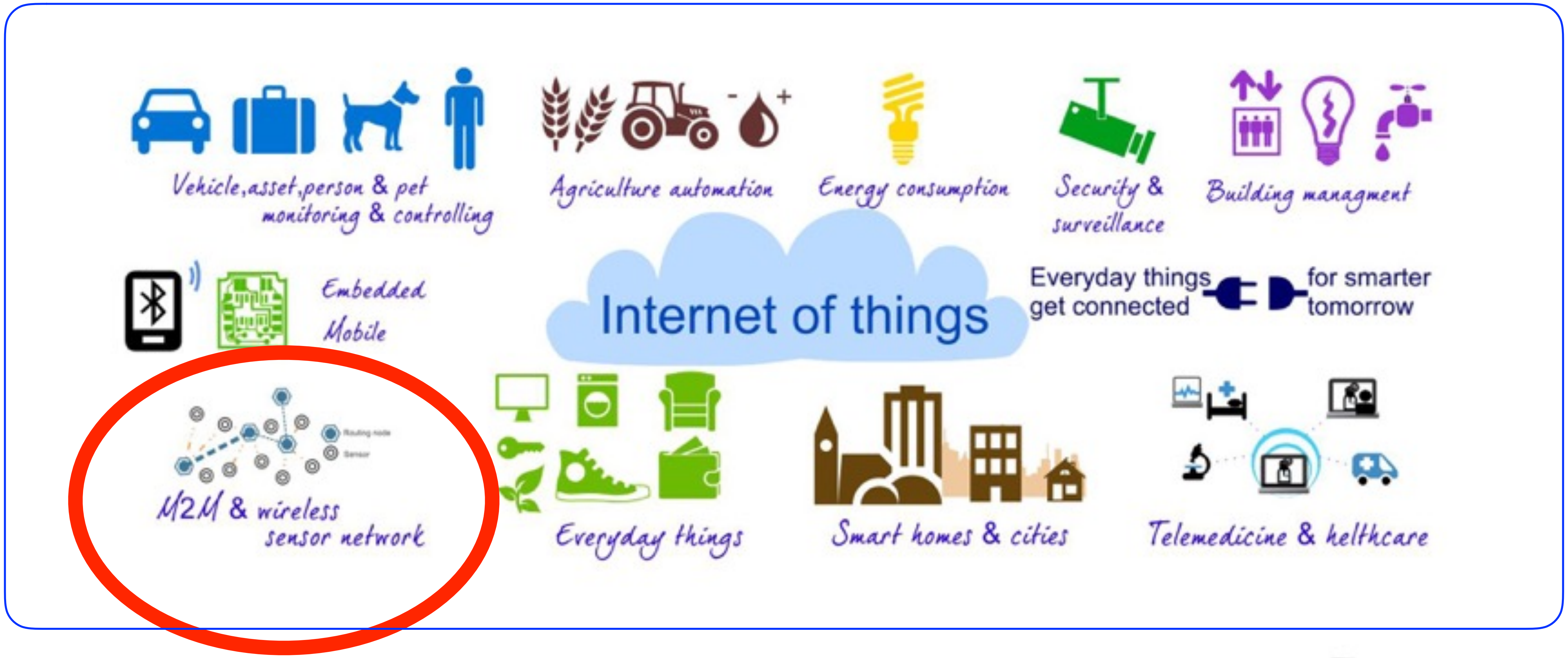
with Applications to
Anonymous Dynamic Algebraic Computations

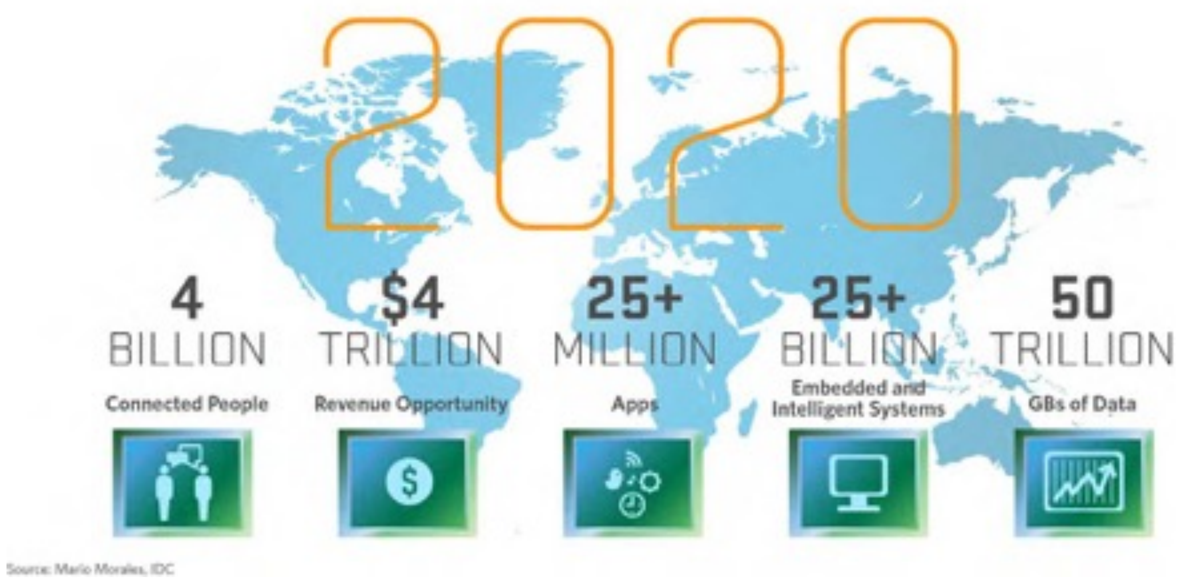
Dariusz R. Kowalski
U. Liverpool (UK)

Miguel A. Mosteiro
Pace Univ. (USA)

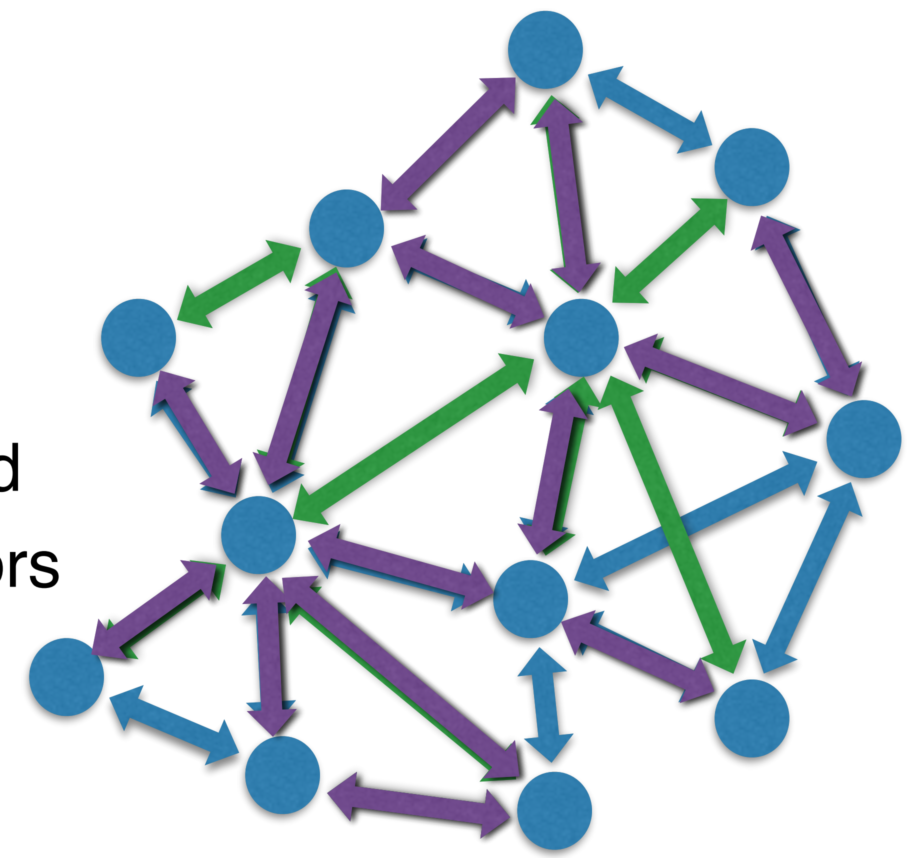
ICALP 2018

The Internet of Things





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

[1] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Naming and counting in anonymous unknown dynamic networks. SSS 2013.
[2] F. Kuhn, N. A. Lynch, R. Oshman. Distributed computation in dynamic networks. STOC 2010.

The Counting Problem

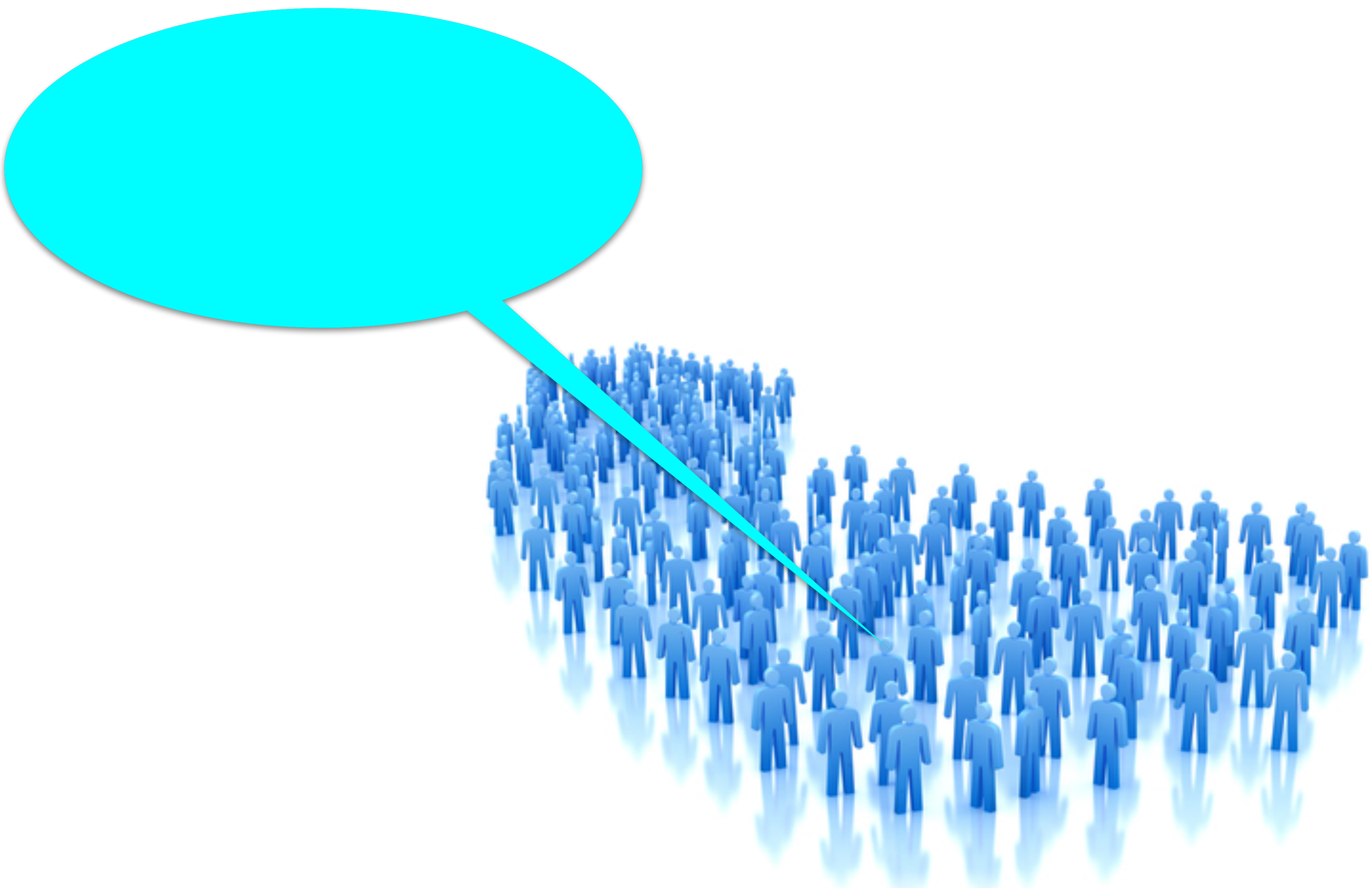
How do you count the size of your group,

if the members are all identical and move?

The Counting Problem

How do you count the size of your group,

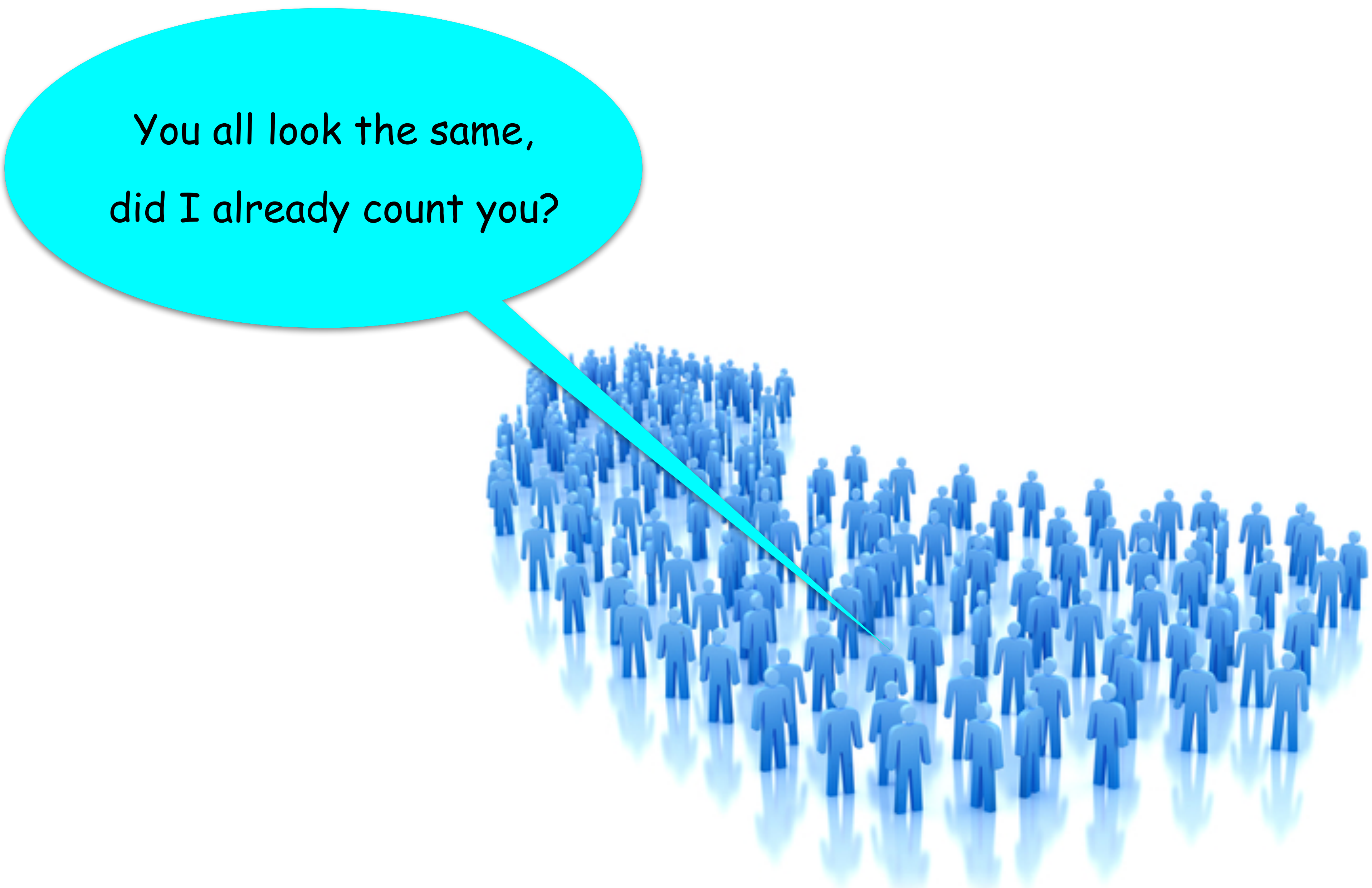
if the members are all identical and move?



The Counting Problem

How do you count the size of your group,

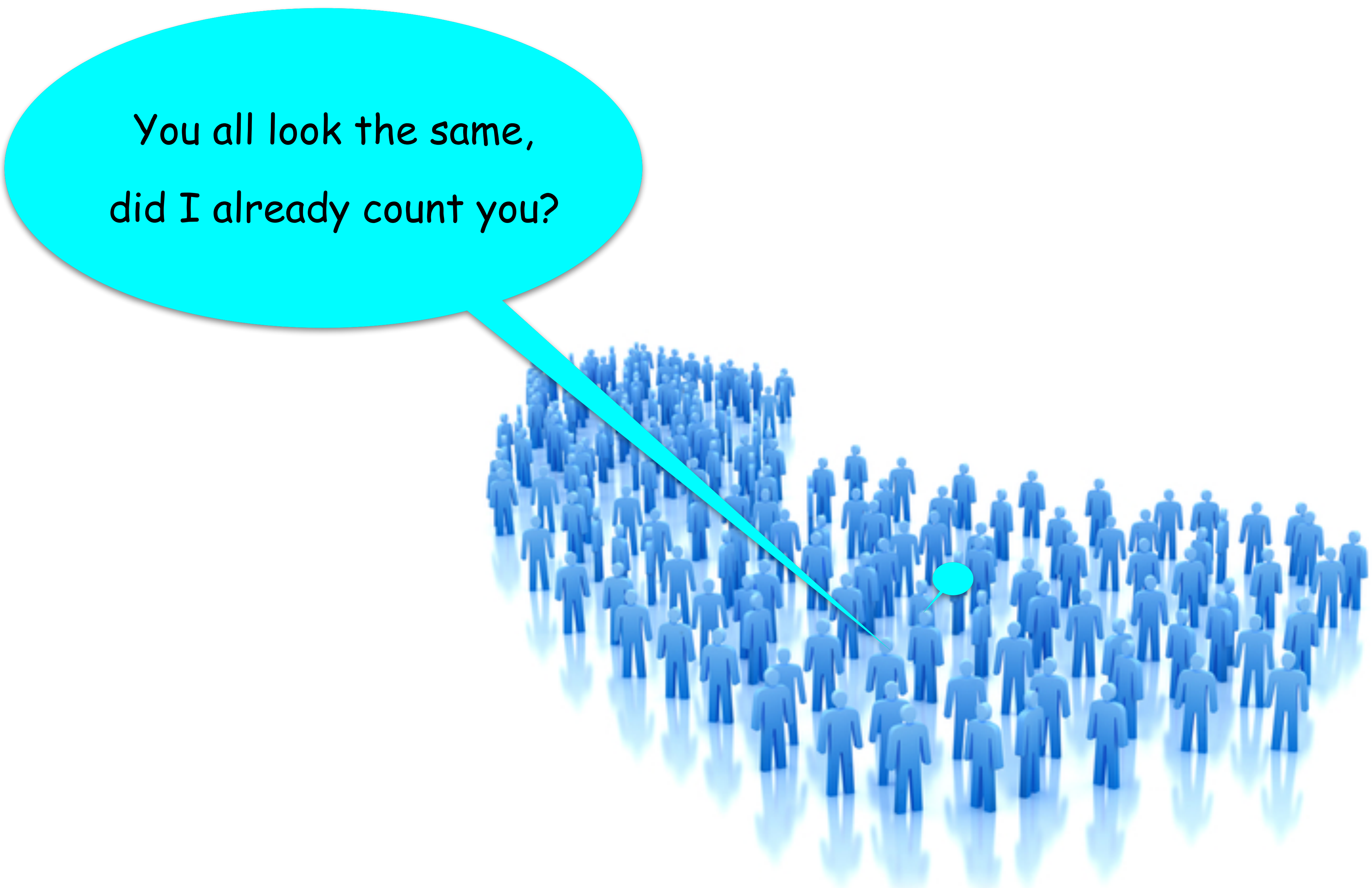
if the members are all identical and move?



The Counting Problem

How do you count the size of your group,

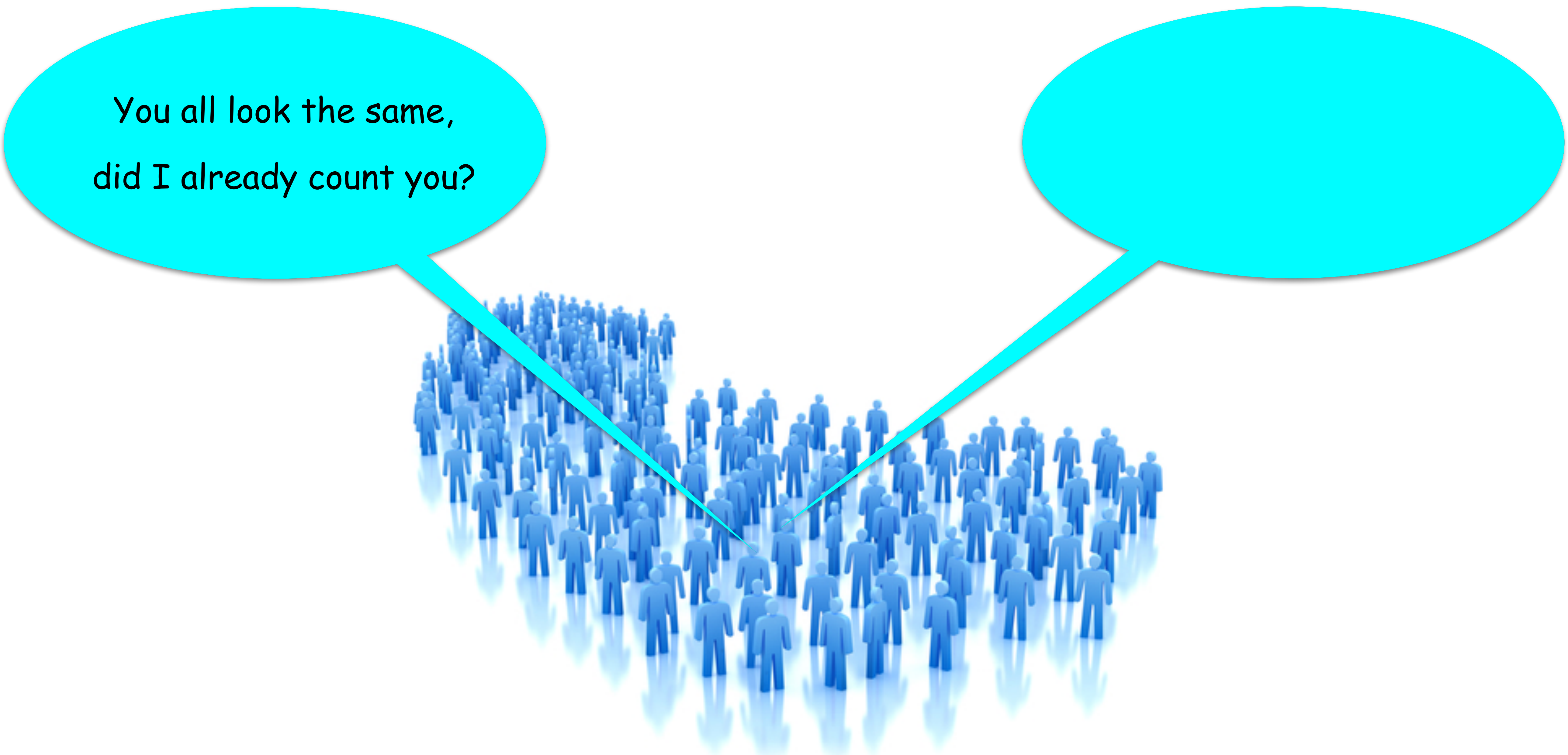
if the members are all identical and move?



The Counting Problem

How do you count the size of your group,

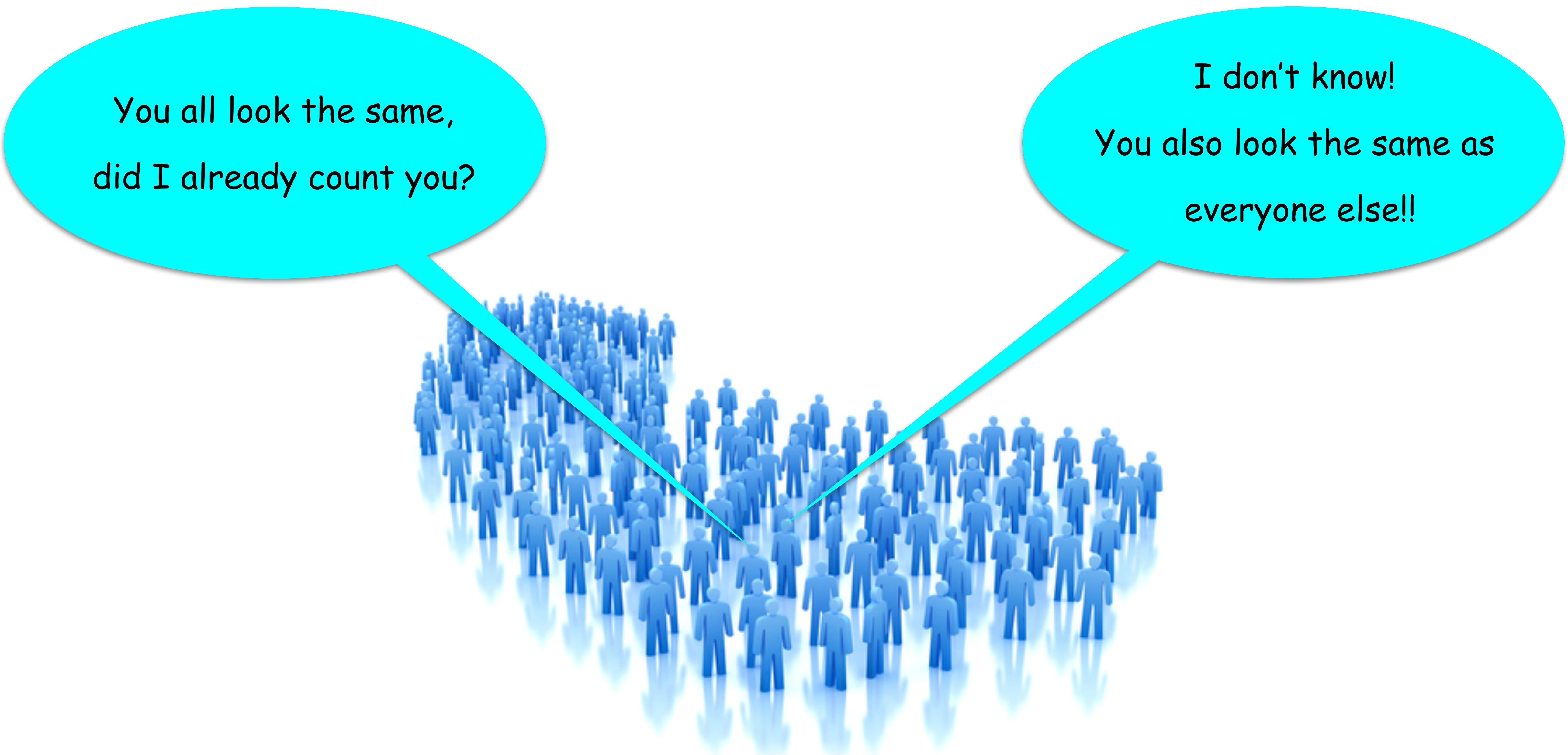
if the members are all identical and move?



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 algorithms

need the number of processors to decide termination.

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

all nodes eventually terminate knowing *n* »

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].
  - Recently, exact-size exponential time Counting with termination:
    - [5] Incremental Counting (IC): needs dyn. max degree d_{max}, poly space.
    - [6] EXT Counting: no d_{max}, but exponential space.

Exponential speedup, but still not practical

- **Lower bound on the time complexity**
  - Ω(D) where D is the dyn. diameter.
  - Ω(log n) even if D is constant [4].

Huge gap

- **Experimental work**
  - IC on trees, paths, stars, G(n,p) [7].

Polynomial in practice

[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.

[7] M. Chakraborty, A. Milani and M. A. Mosteiro. Counting in practical anonymous dynamic networks is polynomial. NETYS 2016.

Previous work

restrictions/ shortcomings

algorithm	needs		computes	stops?	complexity	
	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [20]		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [10]	✓	✓	n	✓	$O(e^{N^2}N^3) \Rightarrow O(e^{d_{\max}^{2n}}d_{\max}^{3n})$ using [20]	
Unconscious [10]			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [11]		Oracle for each node	n	Eventually	Unknown	
EXT [9]			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [21]		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\,\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [This work]			n	✓	$O(n^5 \ln^2 n)$	PSPACE

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

[10] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Conscious and unconscious counting on anonymous dynamic networks. ICDCN 2014.

[11] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Counting in anonymous dynamic networks under worst-case adversary. ICDCS 2014.

[21] A. Milani and M. A. Mosteiro. A faster counting protocol for anonymous dynamic networks. OPODIS 2015.

[9] R. Baldoni and G. A. Di Luna. Non trivial computations in anonymous dynamic networks. OPODIS 2015.

Contributions

- Methodical Counting (MC) algorithm:
  - no knowledge of network characteristics
  - computes the exact size of the network
  - all nodes obtain n and terminate
  - **first polynomial time** guarantees
  - **exponentially faster** than best previous work
- Design of control mechanisms:
  - for mass-distribution-based computations,
    to detect wrong convergence-time estimation
- Novel approach opens path:
  - to study more complex computations using same techniques
- Extensions to algebraic and other computations:
  - sum, average, max, min, multiple Boolean functions, others

MC Key Ingredients

Key idea:

- distribute a potential value iteratively (resembling previous works),
- but let the leader participate in the process as any other node,
- **leader removes potential but only after it has accumulated enough!**

MC Key Ingredients

Our approach allows to leverage previous work on

lazy random walks in evolving graphs [1].

But, **not a simple de-randomization,**

so, we use lazy random walks bounds,

but only when parameters are temporarily fixed,

and the number of received messages is not invalid.

- In ADNs, **neighbors cannot be distinguished**.
- Even **number of neighbors unknown** at transmission time,

  only after receiving but may change for next round.

- **Unknown network parameters =>**

  potential received could be bigger than 1.

- **Mixing and cover time** of lazy random walks depend on n =>

  cannot be used for termination.

[1] C. Avin, M. Koucky, and Z. Lotker. How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). ICALP 2008.

MC Structure

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

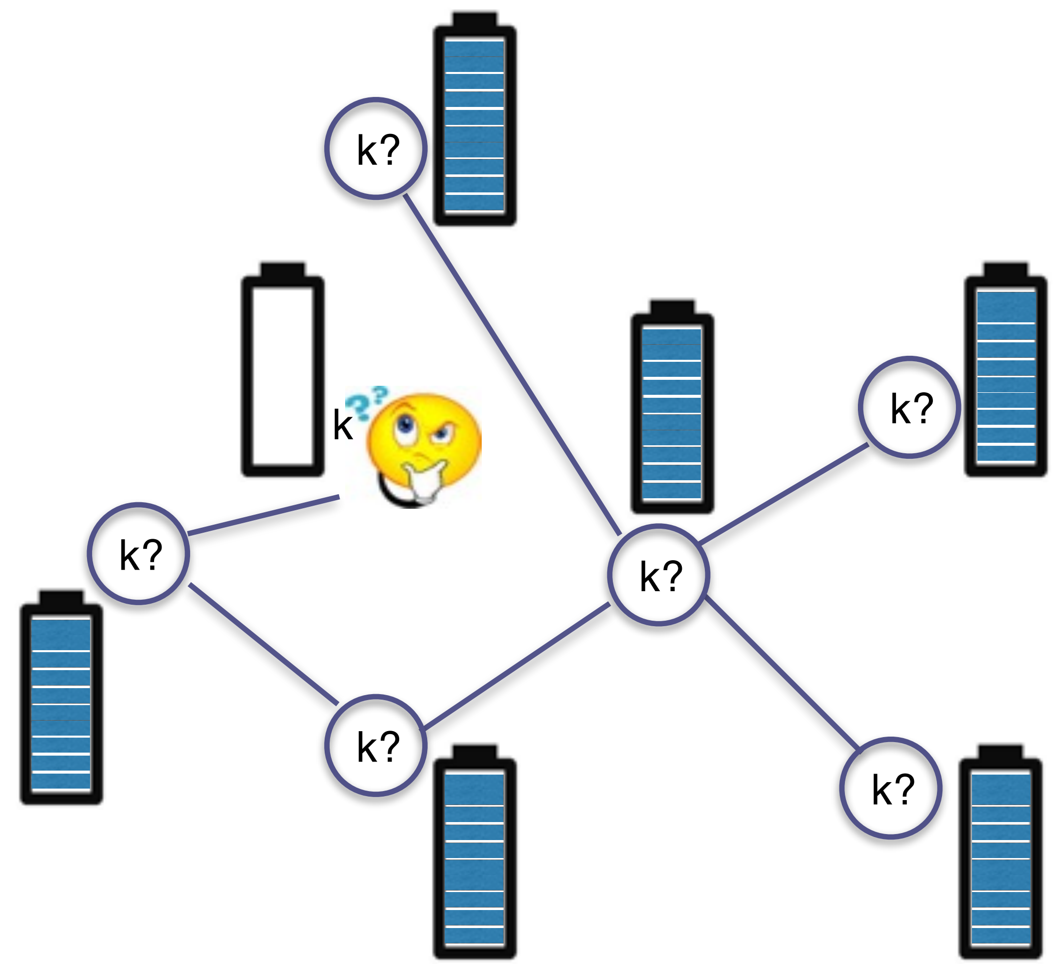
p(k) phases:

(to let the leader remove "enough" potential ρ)

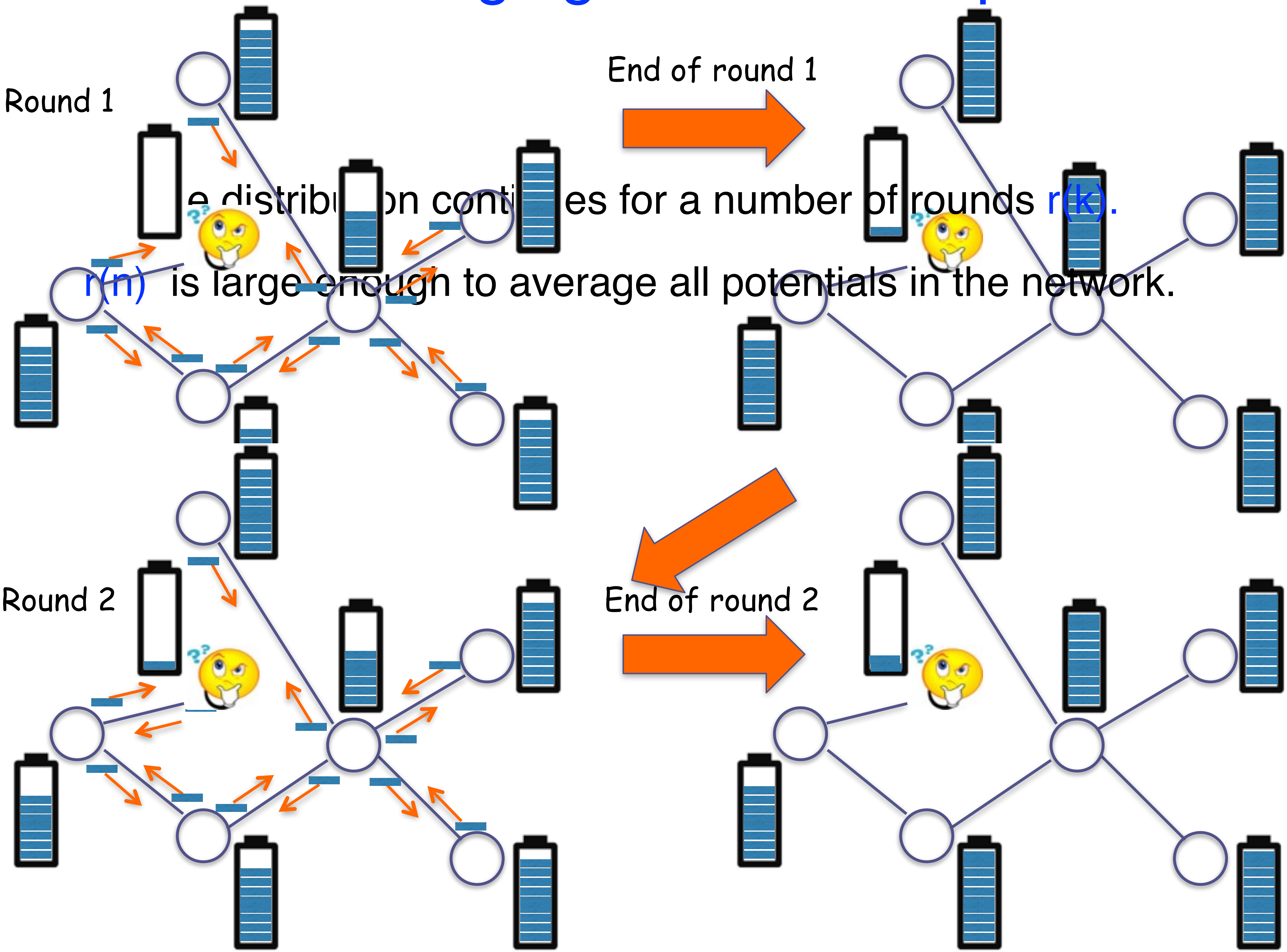
r(k) rounds:

(to "average" the current potentials Φ)

let's see how…



MC Averaging Phase Example



e distrib on conti es for a number of rounds r(k).

r(n) is large enough to average all potentials in the network.

MC Epoch Example

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

p(k) phases:

(to let the leader remove "enough" potential ρ)

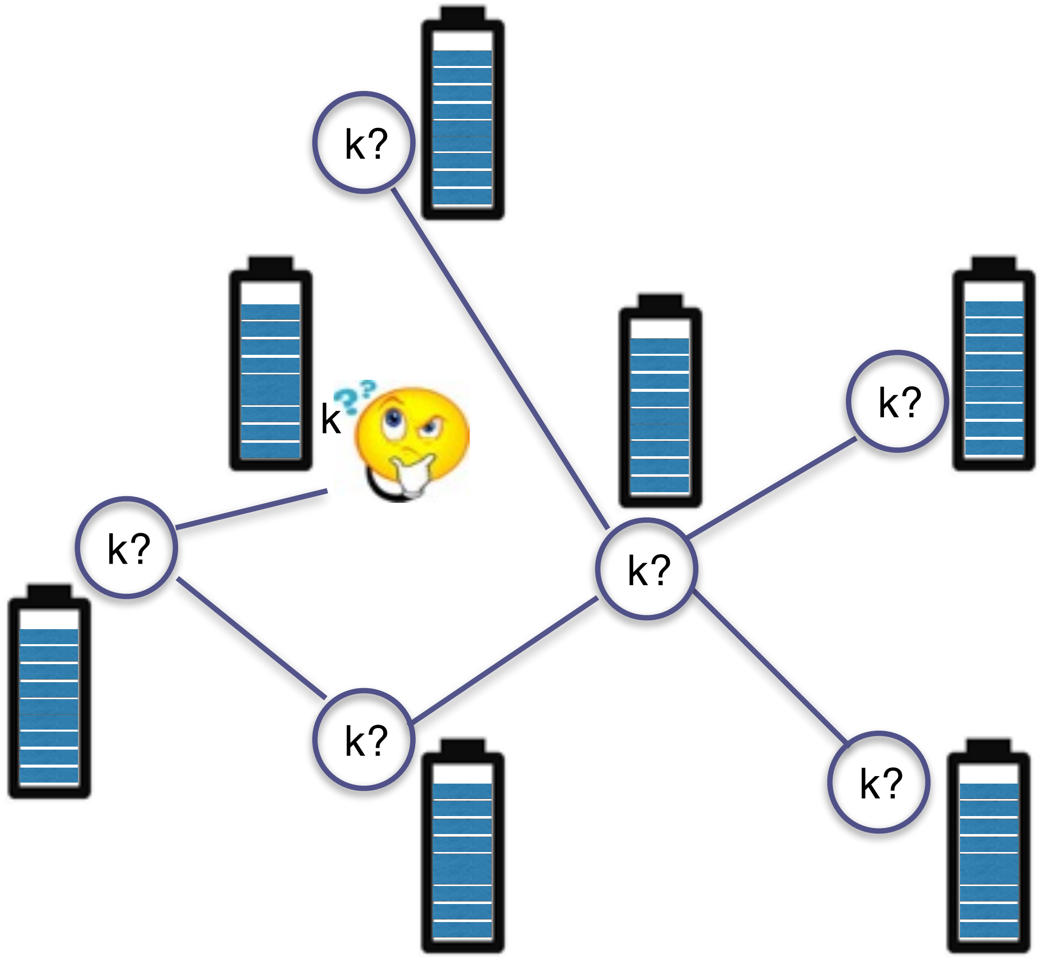
r(k) rounds:

(to "average" the current potentials Φ)

mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i \in N} \Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$



$\rho=$

MC Epoch Example

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

p(k) phases:

(to let the leader remove "enough" potential ρ)

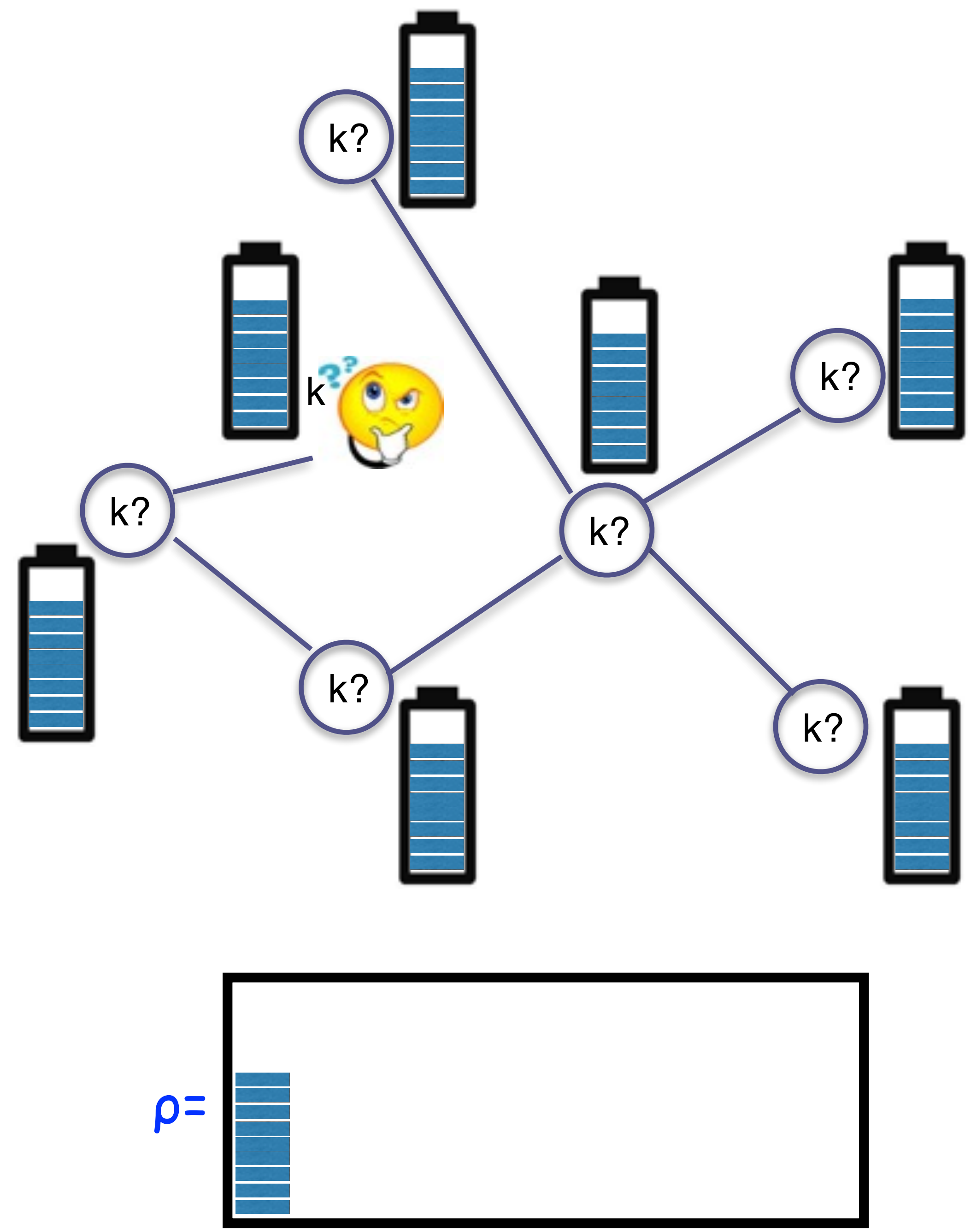
r(k) rounds:

(to "average" the current potentials Φ)

mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i\in N}\, \Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$



MC Epoch Example

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

$p(k)$ phases:

(to let the leader remove "enough" potential ρ)

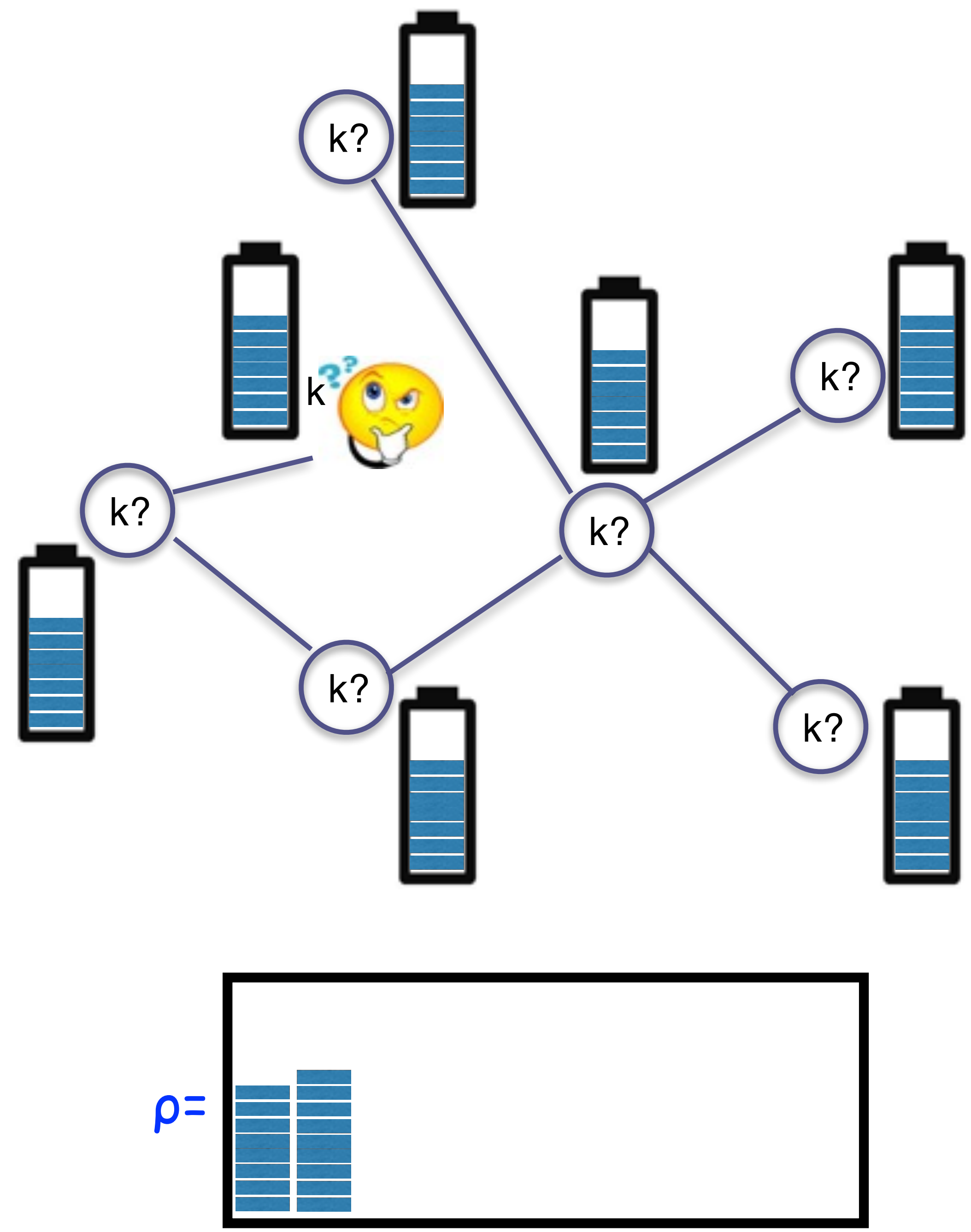
$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i\in N}\, \Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$



MC Epoch Example

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

$p(k)$ phases:

(to let the leader remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)

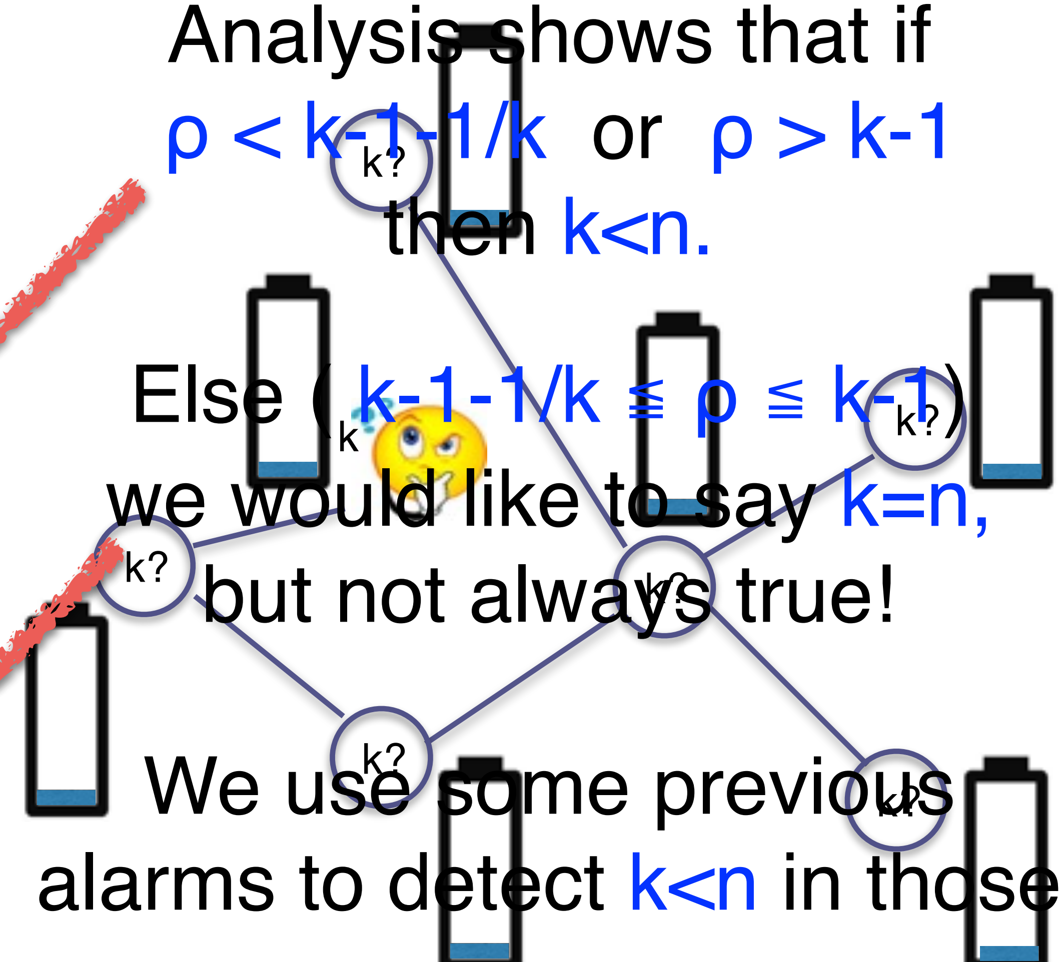
mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i\in N}\,\Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$

- if $\rho < k\text{-}1\text{-}1/k$ or $\rho > k\text{-}1$
  - try next k
- else notify all nodes that $k=n$

After $p(k)$ phases…



Analysis shows that if

$\rho < k\text{-}1\text{-}1/k$ or $\rho > k\text{-}1$

then $k<n$.

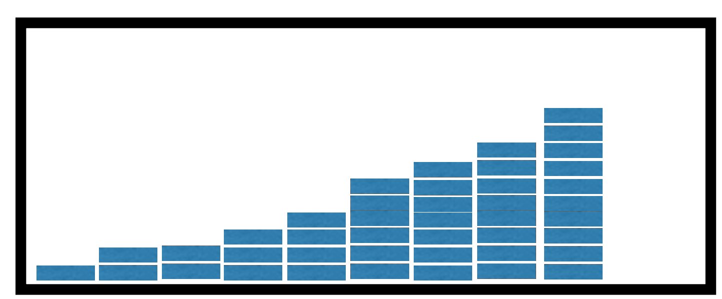
Else ($k\text{-}1\text{-}1/k \leqq \rho \leqq k\text{-}1$)

we would like to say $k=n$,

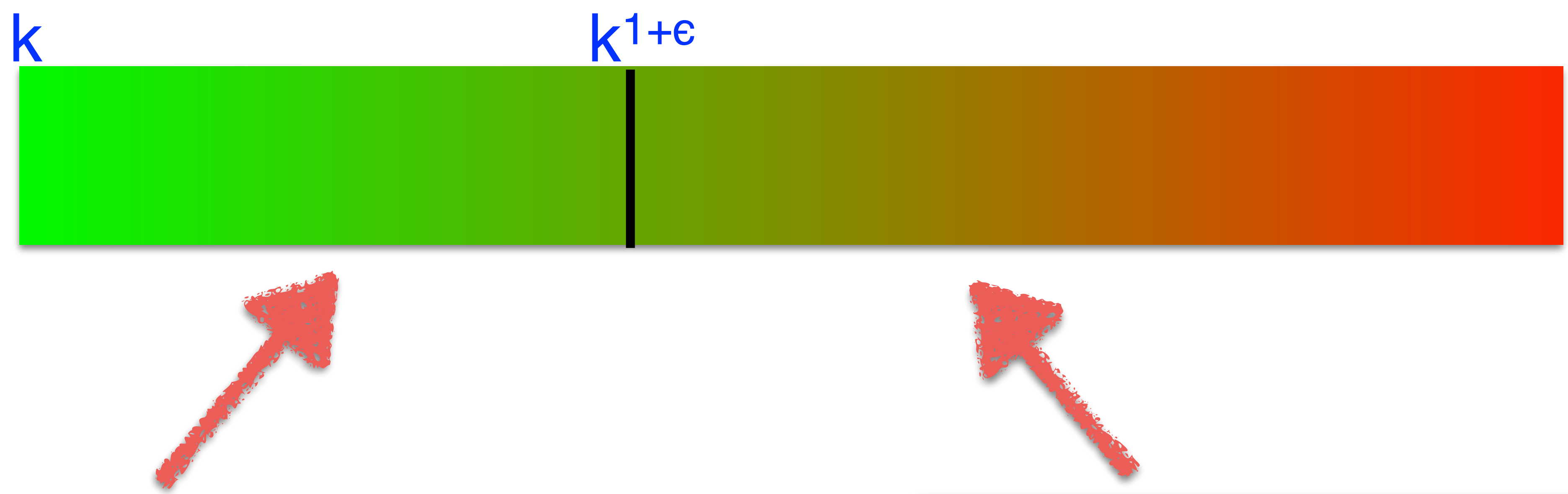
but not always true!

We use some previous alarms to detect $k<n$ in those cases…

$\rho=$



MC Alarms (for k<n)



If n is “close” to k then
leader removes
“too much” potential.

If n is “far” from k then
not “many” nodes
have “low” potential,

so, leader receives alarm from
nodes with “high” potential
“soon” after first phase.

MC Epoch Example

epochs:

- one for each estimate $k=2,3,\dots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

$p(k)$ phases:

(to let the leader remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i\in N} \Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$

- if $k-1-1/k <= \rho <= k-1$ and status = normal
  - notify all nodes that $k=n$
- else try next k

We use some previous alarms to detect $k<n$ in those cases.

And now the leader can notify $k=n$ when ρ is in that range.

MC Epoch Example

epochs:

- one for each estimate $k=2,3,\ldots,n$
- initially, "potential" value: $\Phi_{non\text{-}leader}=1$, $\Phi_{leader}=0$

$p(k)$ phases:

(to let the leader remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

- broadcast Φ and receive neighbors' Φ_i
- $\Phi = \Phi + \Sigma_{i \in N} \Phi_i/d(k) - |N|\Phi/d(k)$

- leader "removes" its potential: $\rho=\rho+\Phi$, $\Phi=0$

- if $k-1-1/k <= \rho <= k-1$ and status = normal
  - notify all nodes that $k=n$
- else try next k

We use some previous alarms to detect $k<n$ in those cases,

and now the leader can notify $k=n$ when ρ is in that range.

Main Theorem

THEOREM 6.2. *Given an Anonymous Dynamic Network with n nodes, after running* METHODICAL COUNTING *for each estimate $k = 2, 3, \ldots, n$ with parameters*

$$d = k^{1+\epsilon},$$
$$p = \left\lceil \frac{(2+\epsilon)k^{1+\epsilon}}{1 - 1/k} \ln k \right\rceil,$$
$$r = \left\lceil \left(4 + 2\epsilon + \max\left\{0, -\frac{2\ln(k^{\epsilon} - 1)}{\ln k}\right\}\right) dk^{2+2\epsilon} \ln k \right\rceil,$$
$$\tau = 1 - 1/k^{1+\epsilon},$$

where $\epsilon > 0$, all nodes stop after $\sum_{k=2}^{n}(pr + k)$ rounds of communication and output n.

COROLLARY 6.1. *The time complexity of* METHODICAL COUNTING *is $O(n^5 \log^2 n)$.*

MC Extensions

SUM: assume each node i stores a value v_i, and we need to compute the exact sum.

Compute n and SUM simultaneously:

For each node i

- Append to potential Φ_i the bit representation of value v_i as a sequence of values (initially in $\{0,1\}$ but later averaged iteratively and independently).

$$\langle \phi_i, v_{i0}, v_{i1}, v_{i2}, \ldots \rangle$$

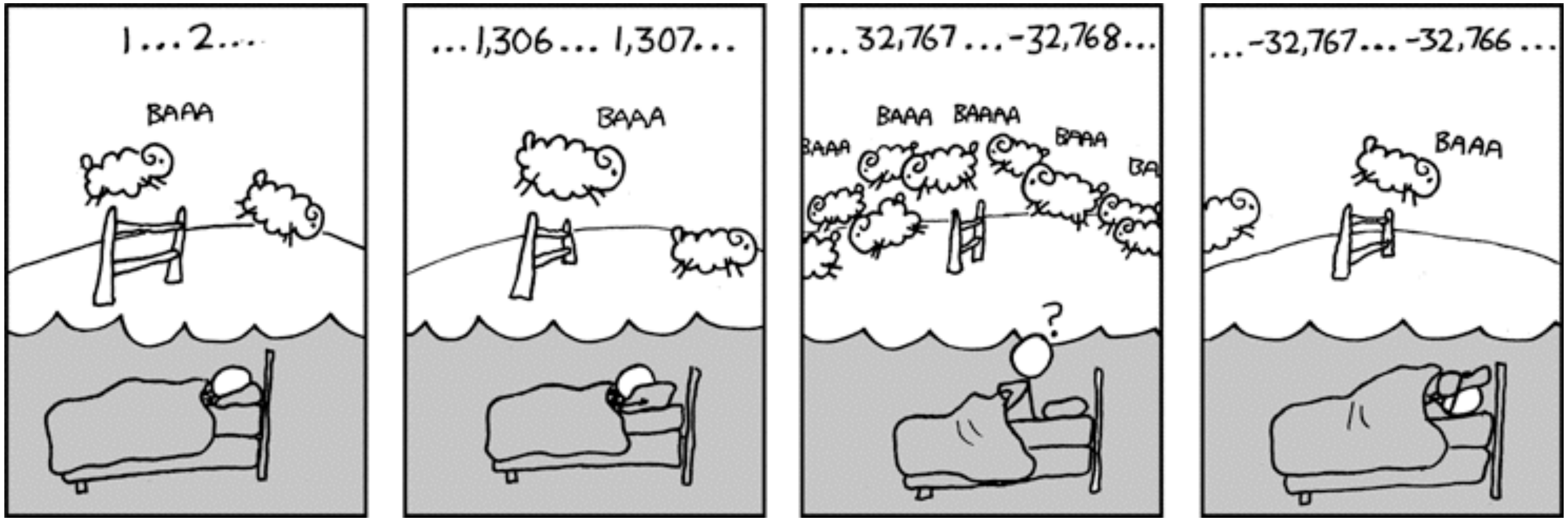
- Apply same algorithm to each v_{ij} independently, as well as to the potential.
- Store the v_{ij}'s at the end of each first phase, call them v'_{ij}.
- At the end of each epoch while $k<n$, reset to the original v_{ij}'s.
- At the end of last epoch ($k=n$), compute $\sum_j \lceil n v'_{ij} \rceil 2^j$.

Others: AVG, Boolean (AND, OR, XOR, etc.), some database queries.

Future and Ongoing Work

- Many leaders.
- Improve upper and lower bounds.
- Other computations in ADNs.
- Asynchronous protocol.

Thank you!



1...2...
BAAA
...1,306... 1,307...
BAAA
...32,767...-32,768...
BAAA
BAAA
BAAAA
BAAA
?
...-32,767...-32,766...
BAAA