Polynomial Anonymous Dynamic Distributed Computing without a Unique Leader

Dariusz R. Kowalski
U. Liverpool (UK)

Miguel A. Mosteiro
Pace Univ. (USA)

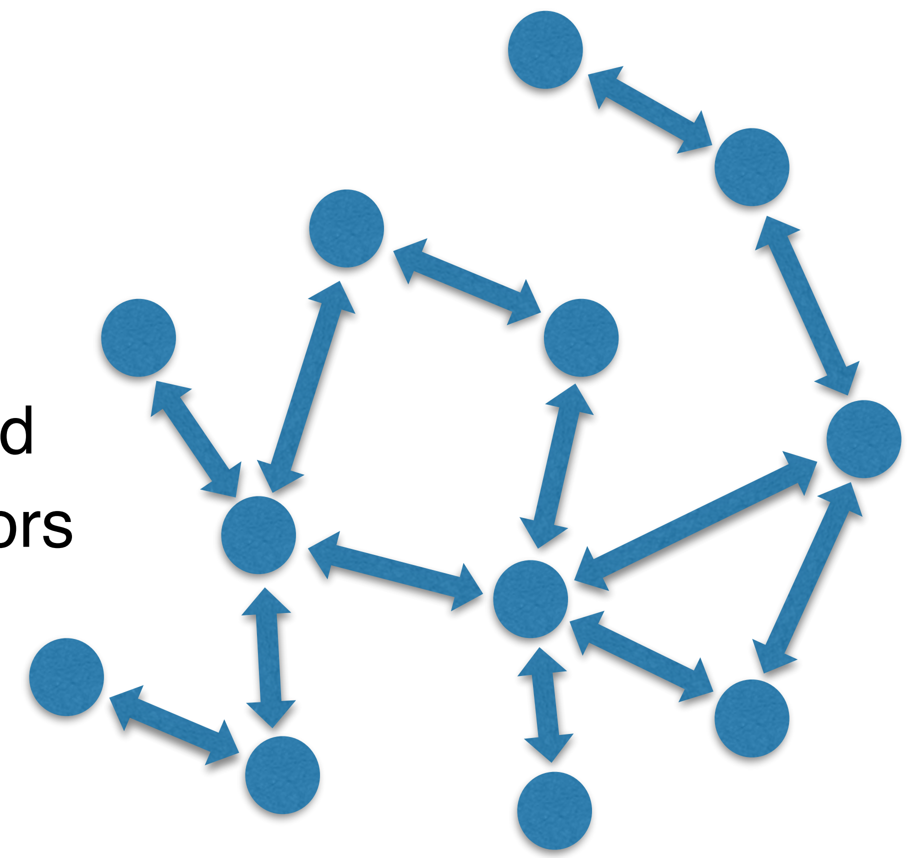
ICALP 2019

Anonymous Dynamic Networks

- **Fixed set of n nodes**
  - No identifiers or labels
- **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** [1]
  - communication links may change from round to round, but
  - at each round the network is connected

[1] F. Kuhn, N. A. Lynch, R. Oshman. Distributed computation in dynamic networks. STOC 2010.

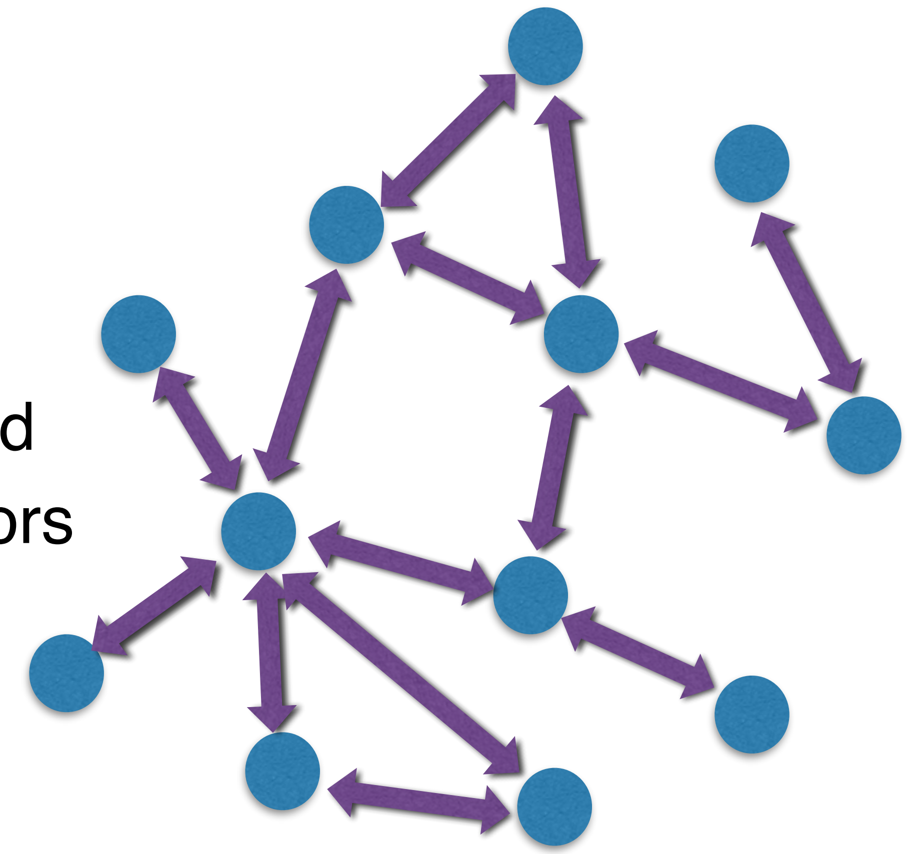
Anonymous Dynamic Networks



- **Fixed set of n nodes**
  - No identifiers or labels
- **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** [1]
  - communication links may change from round to round, but
  - at each round the network is connected

[1] F. Kuhn, N. A. Lynch, R. Oshman. Distributed computation in dynamic networks. STOC 2010.

Anonymous Dynamic Networks



- **Fixed set of n nodes**
  - No identifiers or labels
- **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** [1]
  - communication links may change from round to round, but
  - at each round the network is connected

[1] F. Kuhn, N. A. Lynch, R. Oshman. Distributed computation in dynamic networks. STOC 2010.

The Counting Problem

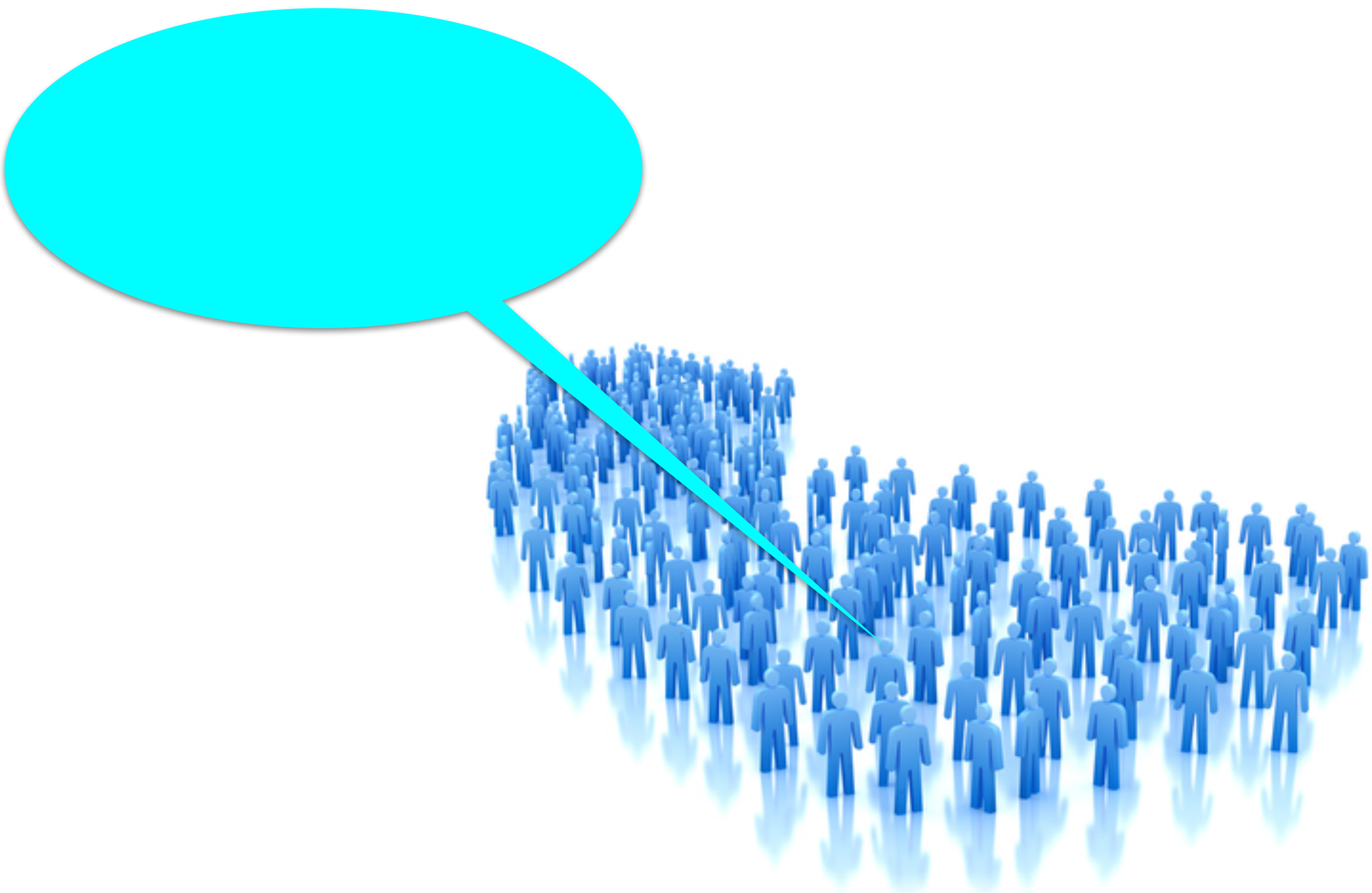
How do you count the size of your group,

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

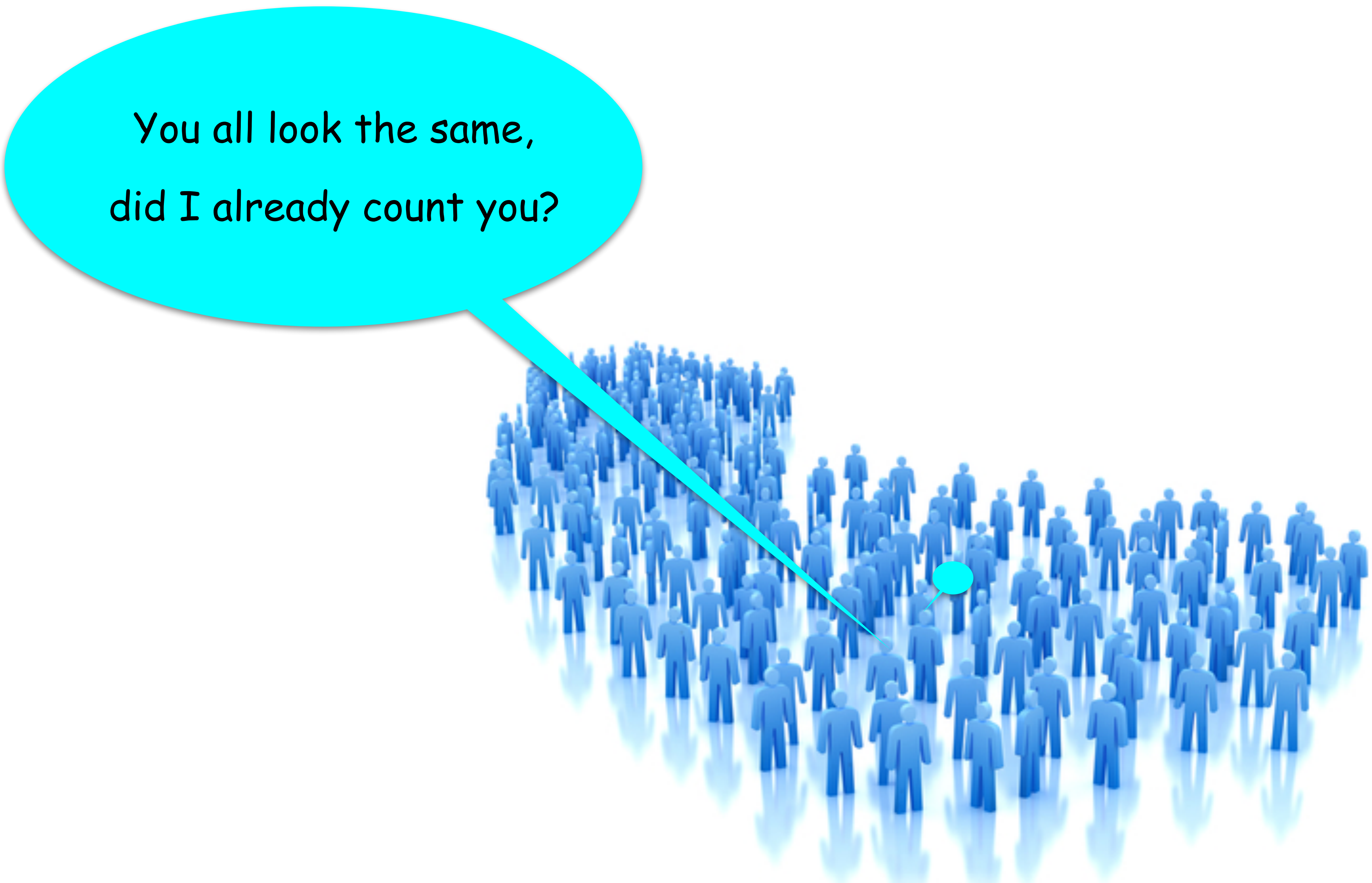
How do you count the size of your group,

if the members are all identical and move?

The Counting Problem

How do you count the size of your group,

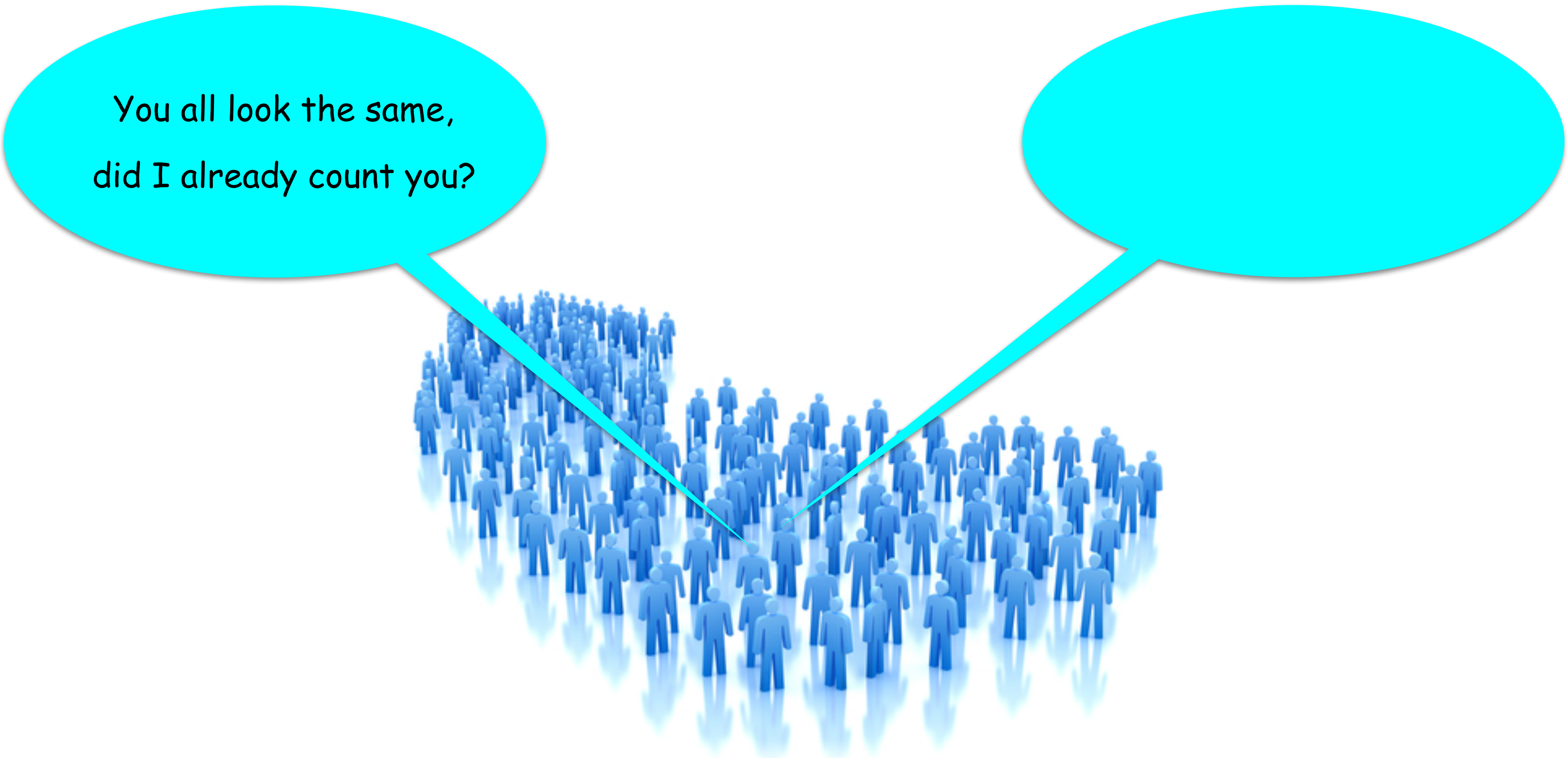
if the members are all identical and move?



The Counting Problem

How do you count the size of your group,

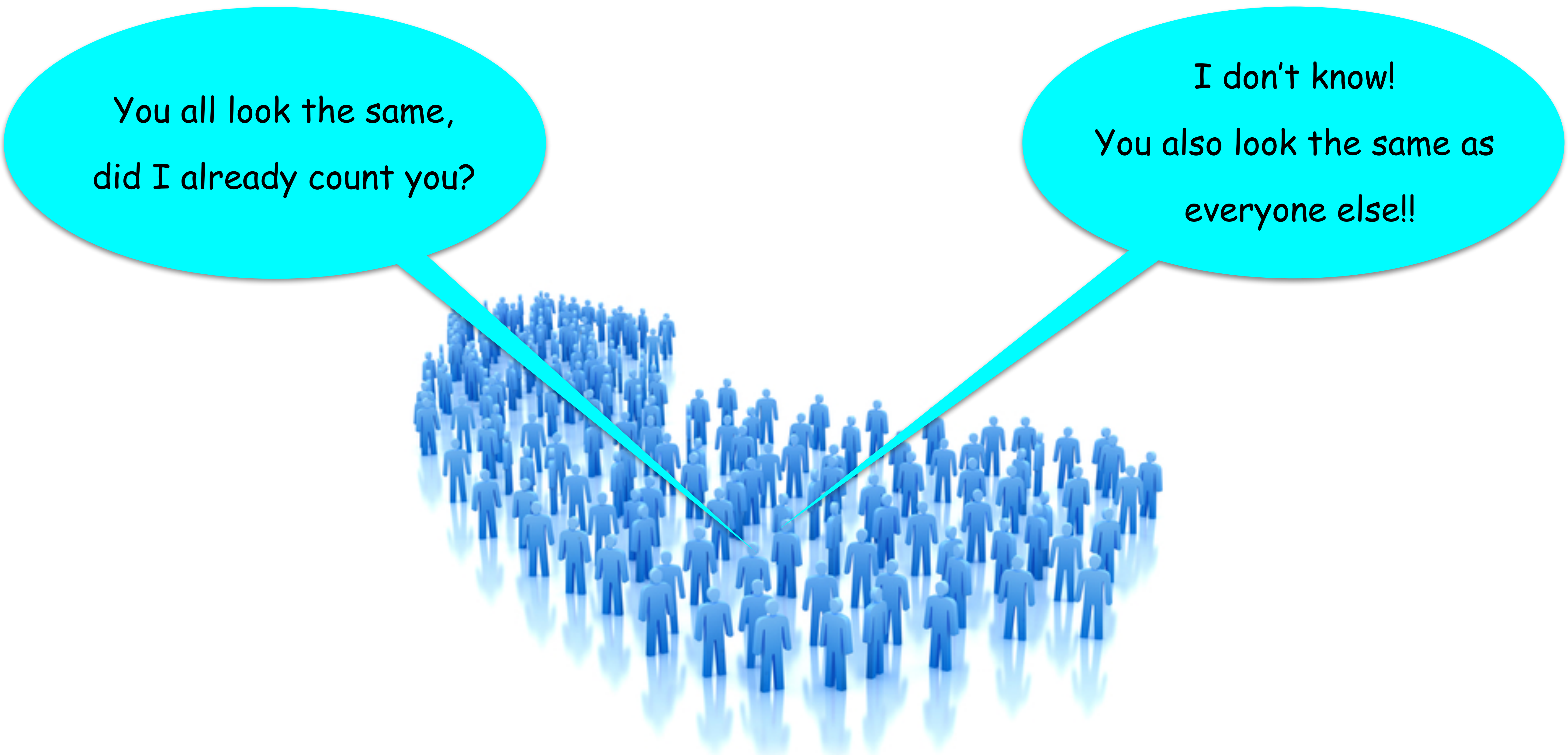
if the members are all identical and move?



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?

Why do we care?

The problem is clean, but why do we care?

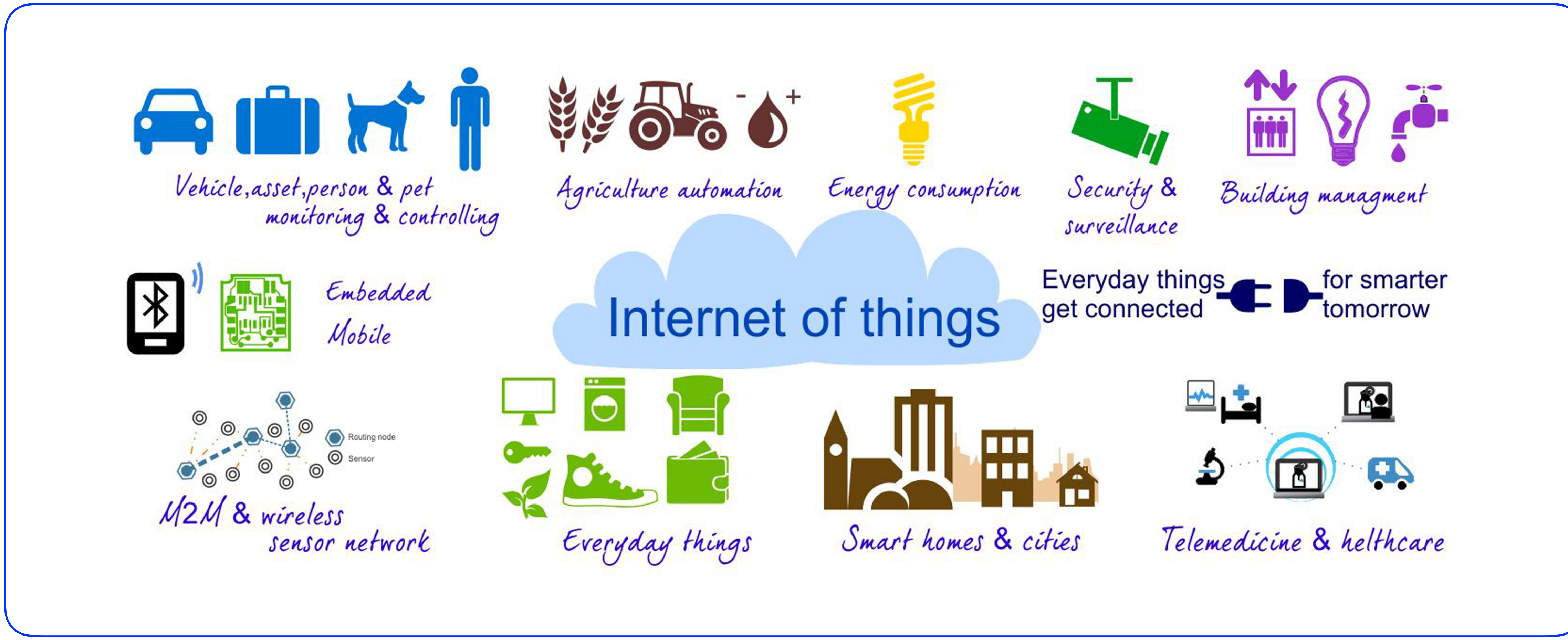
Distributed algorithms

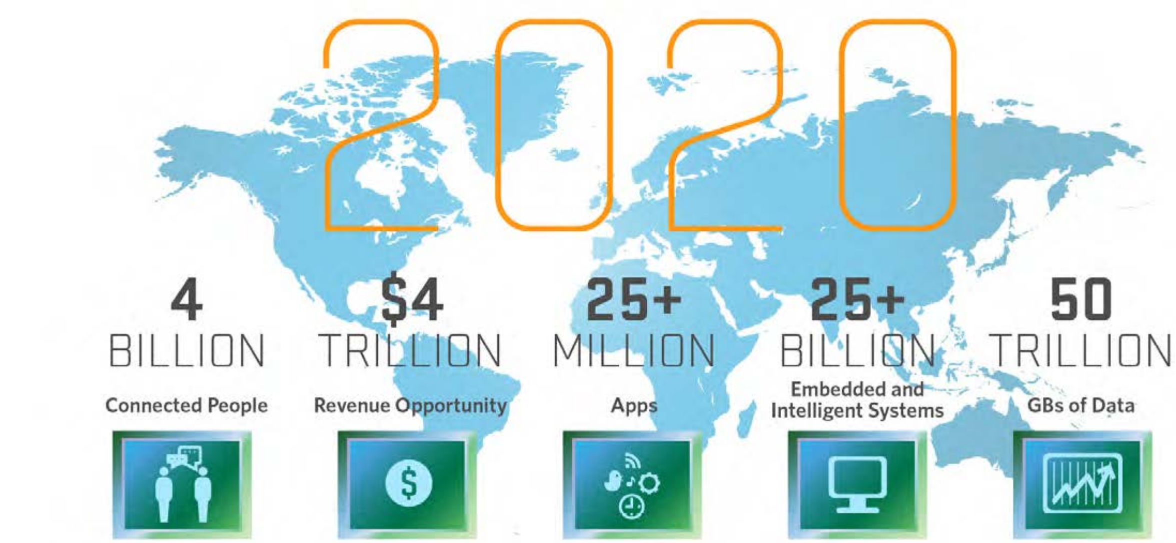
need the number of processors to decide termination.

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

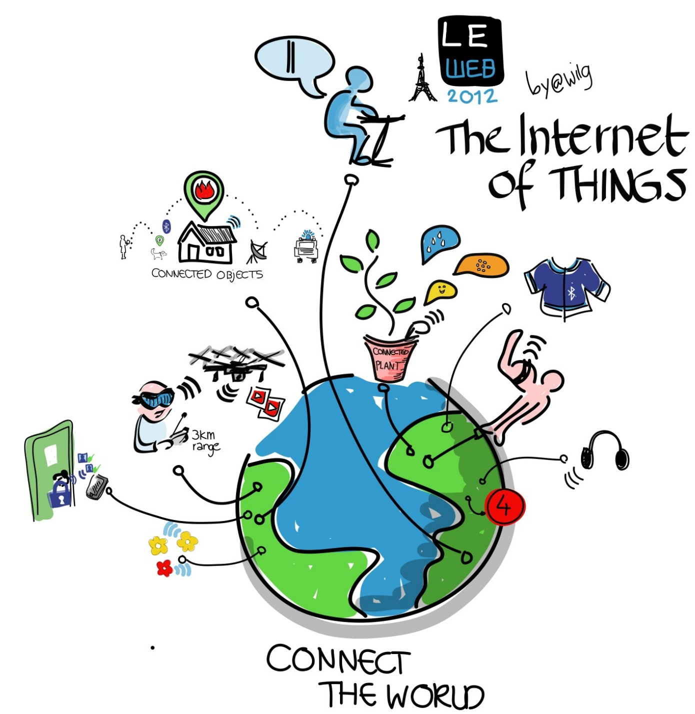
all nodes eventually terminate knowing n ».

Why do we care?

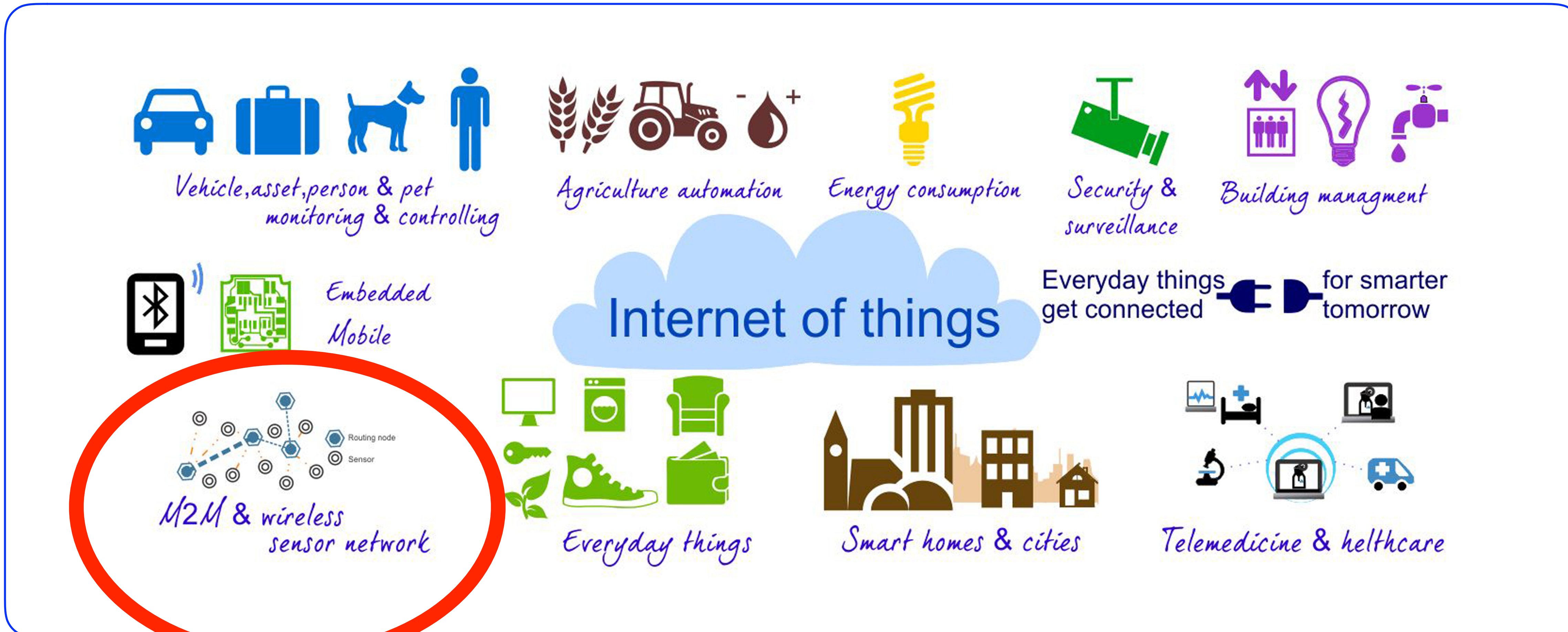


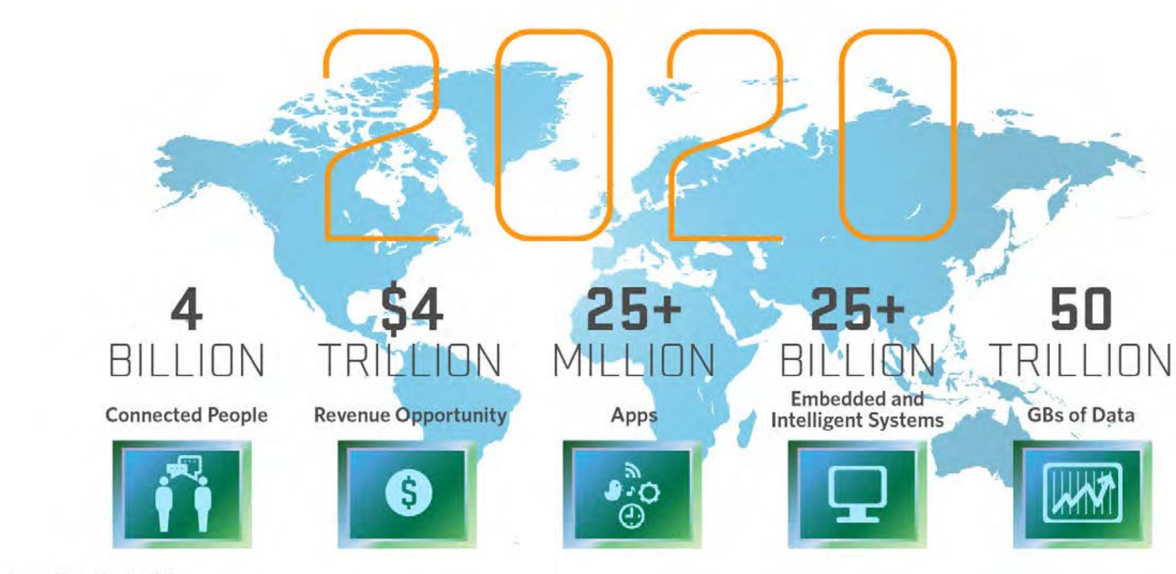


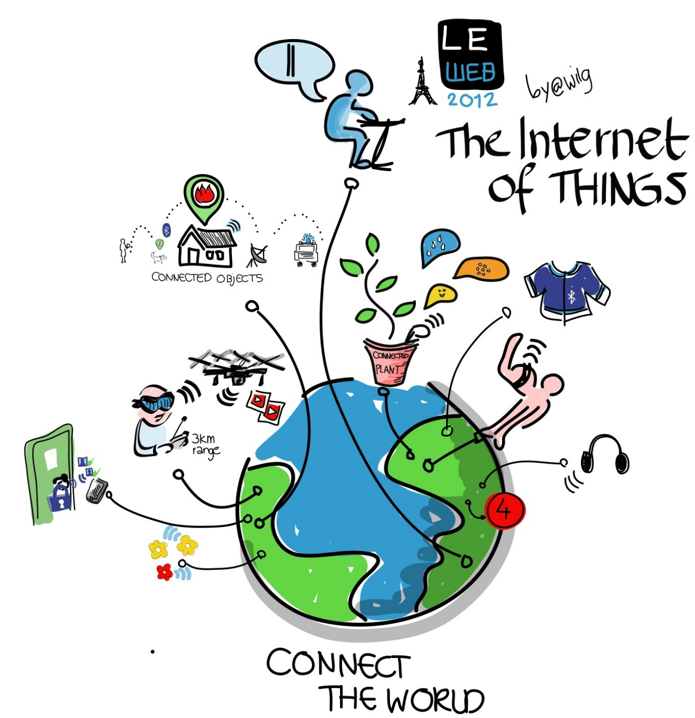
Source: Mario Morales, IDC



Why do we care?







Previous work

algorithm	needs			computes	stops?	complexity	
	distinguished nodes	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [5]	1		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [2]	1	✓	✓	n	✓	$O(e^{N^2}N^3) \Rightarrow O(e^{d_{\max}^{2n}} d_{\max}^{3n})$ using [5]	
Unconscious [2]	1			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [3]	1		Oracle for each node	n	Eventually	Unknown	
EXT [1]	1			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [6]	1		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\,\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [4]	1			n	✓	$O(n^5 \ln^2 n)$	PSPACE

restrictions/ shortcomings

[5] 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.

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

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

[4] D. Kowalski and M. A. Mosteiro. Polynomial counting in anonymous dynamic networks with applications to anonymous dynamic algebraic computations. ICALP 2018.

Previous work

algorithm	needs			computes	stops?	complexity	
	distinguished nodes	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [5]	1		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [2]	1	✓	✓	n	✓	$O(e^{N^2} N^3) \Rightarrow O(e^{d_{\max}^{2n}} d_{\max}^{3n})$ using [5]	
Unconscious [2]	1			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [3]	1		Oracle for each node	n	Eventually	Unknown	
EXT [1]	1			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [6]	1		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [4]	1			n	✓	$O(n^5 \ln^2 n)$	PSPACE

restrictions/ shortcomings

first polynomial

[5] 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.

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

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

[4] D. Kowalski and M. A. Mosteiro. Polynomial counting in anonymous dynamic networks with applications to anonymous dynamic algebraic computations. ICALP 2018.

Previous work

algorithm	needs			computes	stops?	complexity	
	distinguished nodes	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [5]	1		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [2]	1	✓	✓	n	✓	$O(e^{N^2}N^3) \Rightarrow O(e^{d_{\max}^{2n}} d_{\max}^{3n})$ using [5]	
Unconscious [2]	1			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [3]	1		Oracle for each node	n	Eventually	Unknown	
EXT [1]	1			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [6]	1		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [4]	1			n	✓	$O(n^5 \ln^2 n)$	PSPACE

restrictions/ shortcomings

needs at least one [5]

first polynomial

[5] 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.

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

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

[4] D. Kowalski and M. A. Mosteiro. Polynomial counting in anonymous dynamic networks with applications to anonymous dynamic algebraic computations. ICALP 2018.

Questions

- Can we count deterministically with more than one special node?
- What information about the special nodes is needed?
- How "special"? Can the nodes be identical and just have two different programs?
- Can we let the nodes choose program at *random* and make them all identical?
- Can we tighten previous bounds?

Questions

- Can we count deterministically with more than one special node? Yes!
- What information about the special nodes is needed? Count
- How "special"? Can the nodes be identical and just have two different programs? Yes!
- Can we let the nodes choose program at *random* and make them all identical? Yes!
- Can we tighten previous bounds? Yes!

Results

ℓ black nodes and n-ℓ white nodes:

- Impossibility:
  - Deterministic counting: not possible if ℓ is unknown
  - Randomized counting: if ℓ is unknown or zero, exist executions that do not stop

Results

ℓ black nodes and n-ℓ white nodes:

- Impossibility:
    - Deterministic counting: not possible if ℓ is unknown
    - Randomized counting: if ℓ is unknown or zero, exist executions that do not stop
- Methodical Multi-Counting (MMC) algorithm:
    - *all* nodes obtain n and terminate
    - no network info needed except ℓ
- Leader-less Methodical Counting (LLMC) algorithm:
    - first algorithm applicable to ADNs with all identical nodes

Results

algorithm	needs			computes	stops?	complexity	
	distinguished nodes	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [5]	1		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [2]	1	✓	✓	n	✓	$O(e^{N^2}N^3) \Rightarrow$ $O(e^{d_{\max}^{2n}}d_{\max}^{3n})$ using [5]	
Unconscious [2]	1			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [3]	1		Oracle for each node	n	Eventually	Unknown	
EXT [1]	1			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [6]	1		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1}\,\frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [4]	1			n	✓	$O(n^5 \ln^2 n)$	PSPACE
Methodical multi-Counting [This work]	$\ell \geq 1$			n	✓	$O((n^{4+\epsilon}/\ell)\log^3 n)$ for any $\epsilon > 0$	PSPACE
Leader-less Methodical-Counting [This work]	0			n prob. $\geq 1-\zeta$ for any $\zeta > 0$	✓	$O(\eta^{4+\epsilon}\log^3 \eta)$ for any $\epsilon > 0$ and $\eta = \max\{n, \lceil\lceil 12/\zeta\rceil\rceil\}$	PSPACE

Results

algorithm	needs			computes	stops?	complexity	
	distinguished nodes	size upper bound N	dynamic maximum degree u.b. $d_{\max}$			time	space
Degree Counting [5]	1		✓	$O(d_{\max}^n)$	✓	$O(n)$	
Conscious [2]	1	✓	✓	n	✓	$O(e^{N^2} N^3) \Rightarrow O(e^{d_{\max}^{2n}} d_{\max}^{3n})$ using [5]	
Unconscious [2]	1			n	No	No theoretical bounds	
$\mathcal{A}_{\mathcal{O}^P}$ [3]	1		Oracle for each node	n	Eventually	Unknown	
EXT [1]	1			n	✓	$O(n^{n+4})$	EXPSPACE
Incremental Counting [6]	1		✓	n	✓	$O\left(n\,(2d_{\max})^{n+1} \frac{\ln n}{\ln d_{\max}}\right)$	
Methodical Counting [4]	1			n	✓	$O(n^5 \ln^2 n)$	PSPACE
Methodical multi-Counting [This work]	$\ell \geq 1$			n	✓	$O((n^{4+\epsilon}/\ell)\log^3 n)$ for any $\epsilon > 0$	PSPACE
Leader-less Methodical-Counting [This work]	0			n prob. $\geq 1-\zeta$ for any $\zeta > 0$	✓	$O(\eta^{4+\epsilon}\log^3 \eta)$ for any $\epsilon > 0$ and $\eta = \max\{n, \lceil\lceil 12/\zeta\rceil\rceil\}$	PSPACE

first with $\ell \neq 1$

$\sim n\ell$ / log n speedup (for $\zeta \in \Omega(1/n)$),

faster than MC even for $\ell = 1$

Deterministic Counting

Even knowing ℓ, trivial application of ℓ instances of MC not clear:

- How the black nodes communicate?
- How do they compare/combine final results?
- Black nodes are all identical!

Deterministic Counting

Even knowing ℓ, trivial application of ℓ instances of MC not clear:

- How the black nodes communicate?
- How do they compare/combine final results?
- Black nodes are all identical!

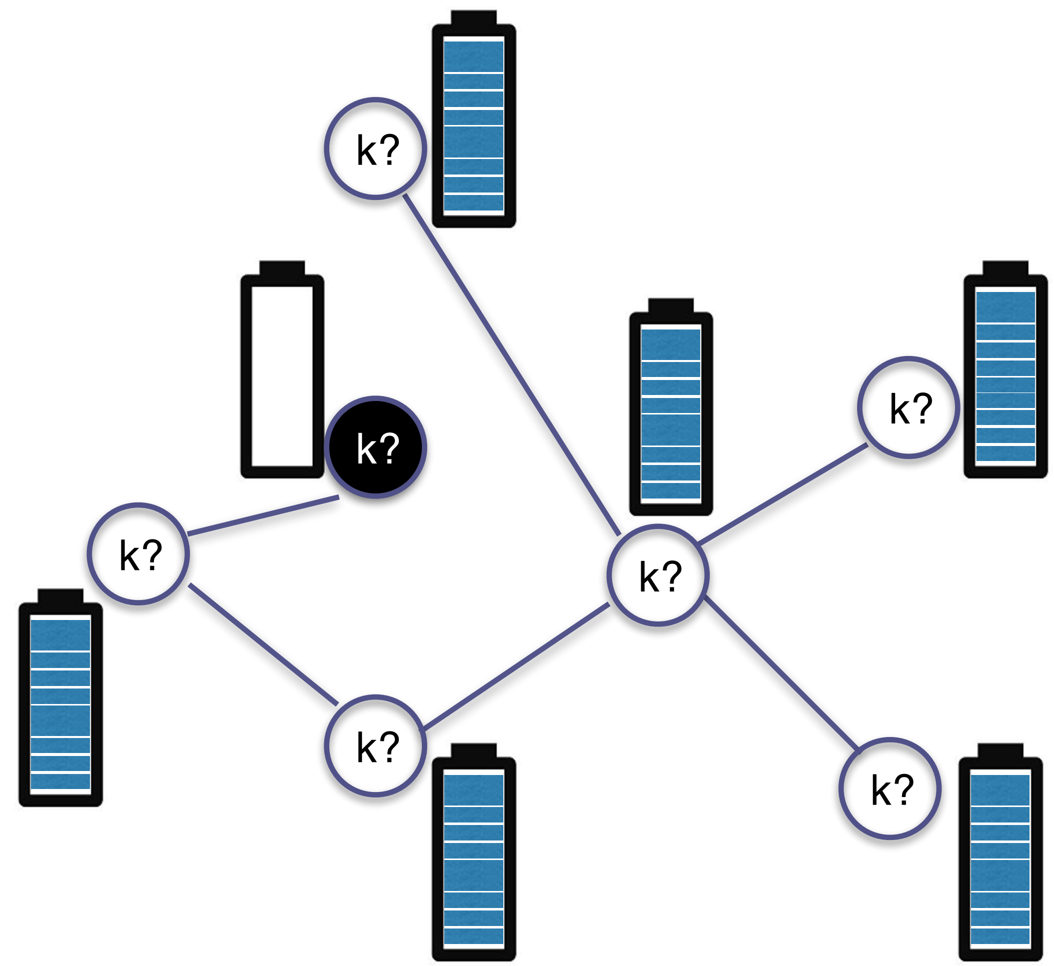
MMC Key Ingredients:

- try network size estimates $k = \ell+1, 2(\ell+1), 4(\ell+1), \dots$ binary search after estimate $k > n$
- share some potential values iteratively
- all nodes (black and white) share potential
- black nodes remove potential from the system every now and then
- carefully designed alarms allow to detect correct or wrong estimate

MMC Structure

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$



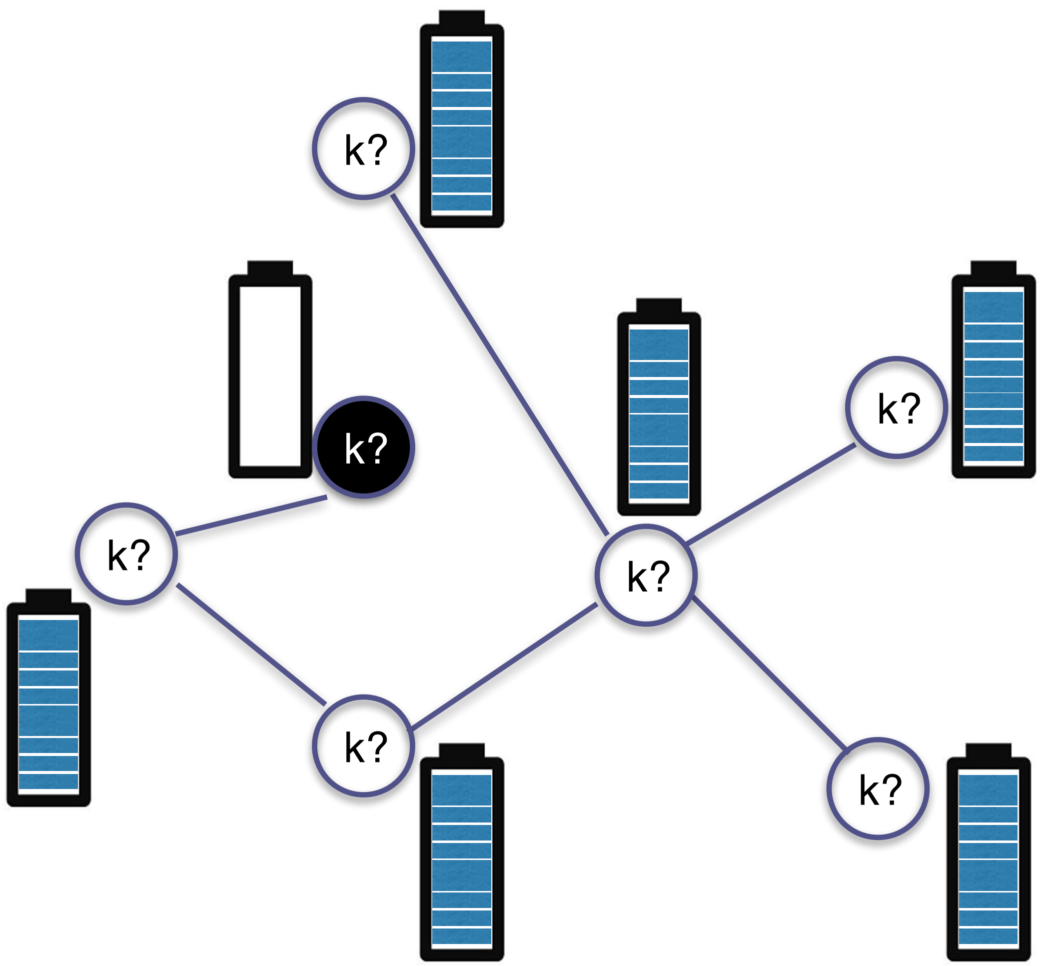
MMC Structure

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \dots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

(to let blacks remove "enough" potential ρ)



MMC Structure

epochs:

– one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$

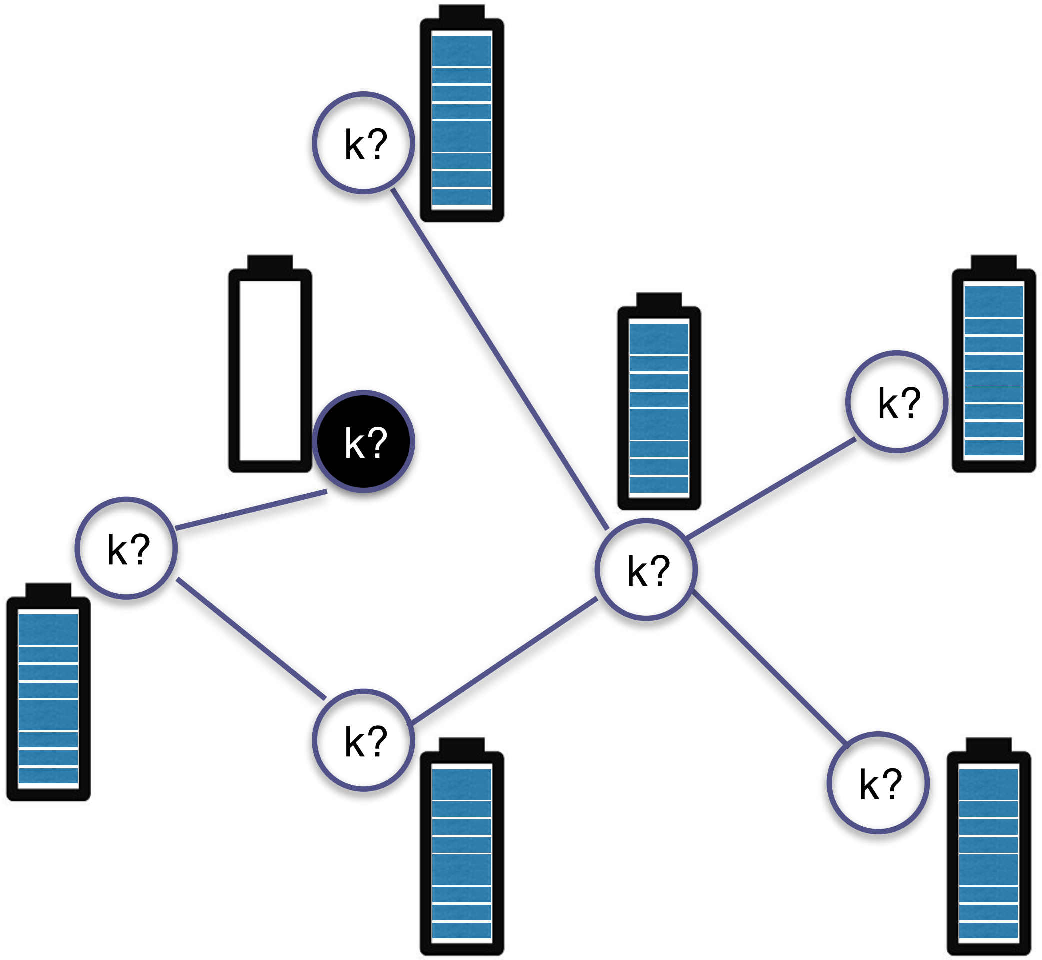
– initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

(to let blacks remove "enough" potential ρ)

r(k) rounds:

(to "average" the current potentials Φ)



epochs:

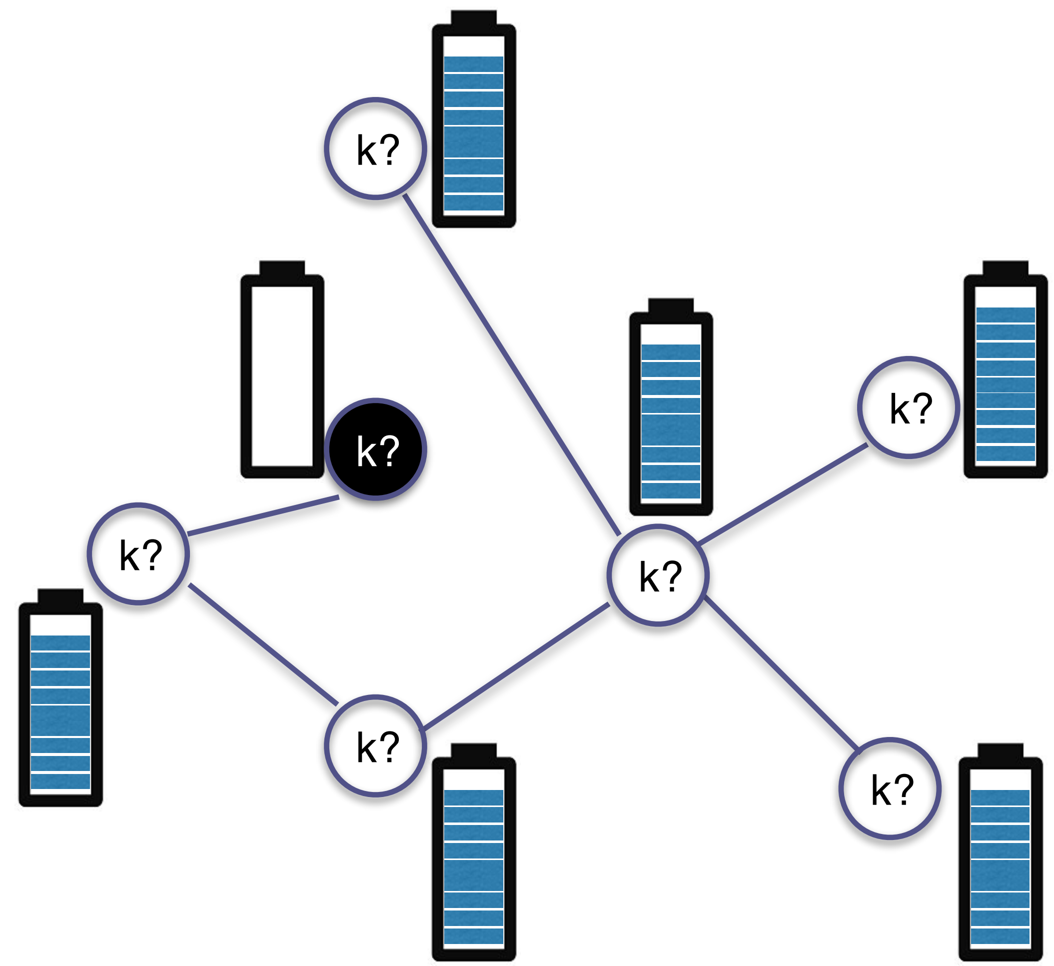
- one for each estimate $k = l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white} = l$, $\Phi_{black} = 0$

$p(k)$ phases:

(to let blacks remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)



$\rho =$

MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

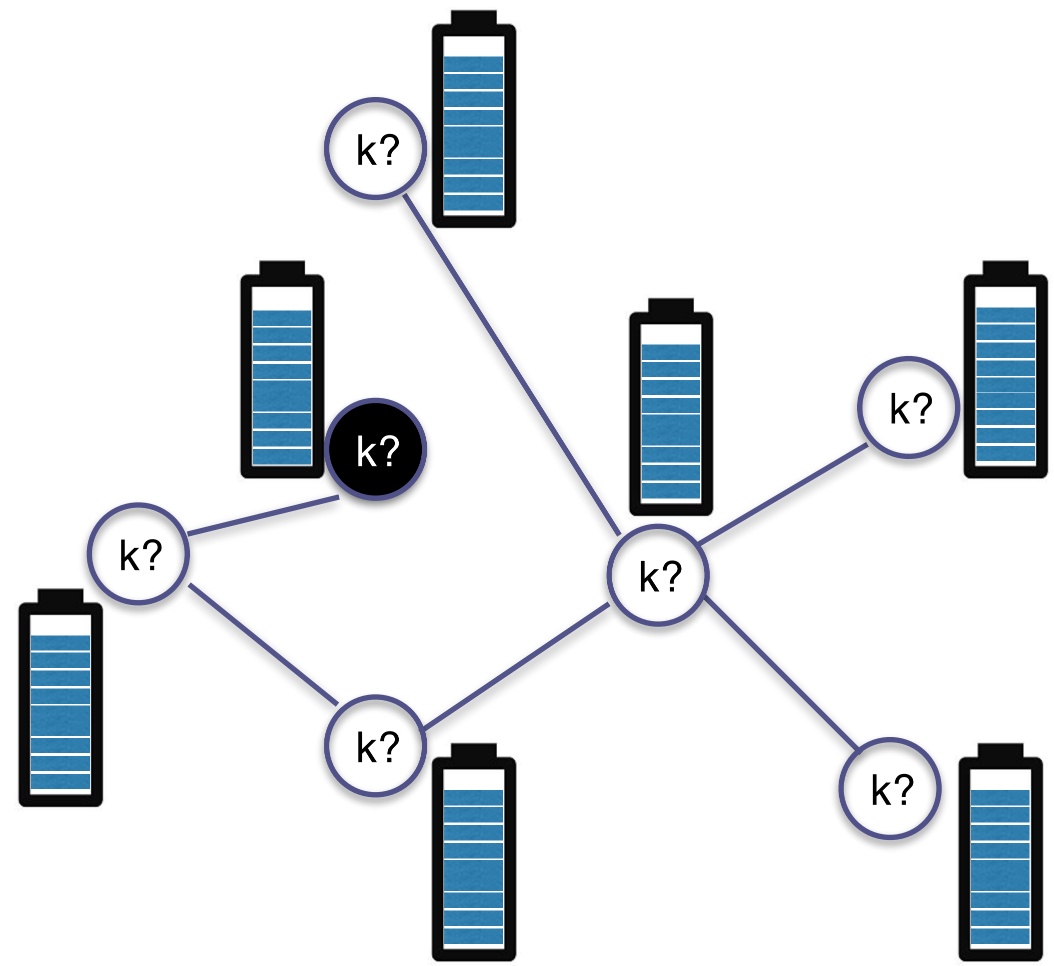
(to let blacks remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

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



$\rho=$

MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

(to let blacks remove "enough" potential ρ)

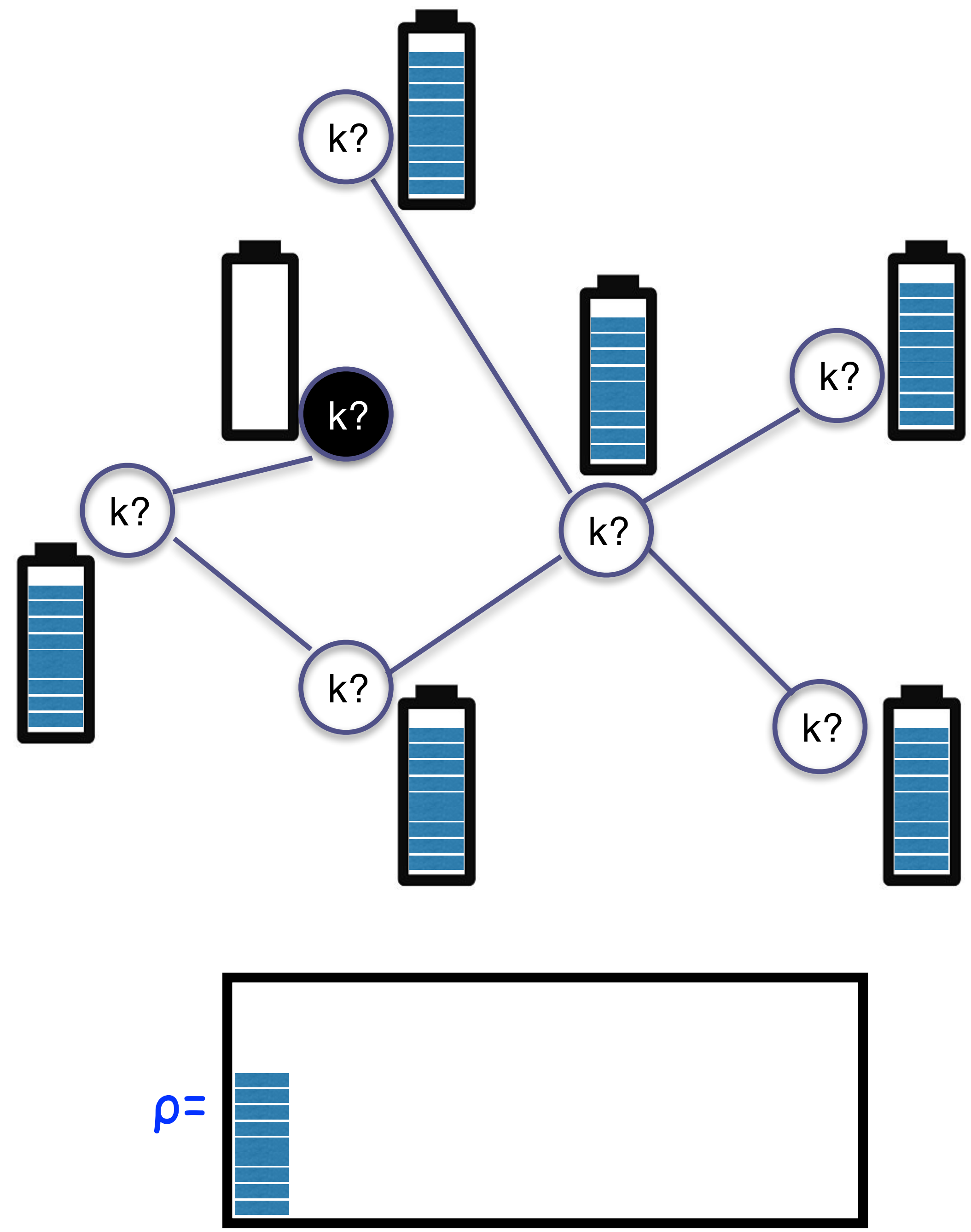
r(k) rounds:

(to "average" the current potentials Φ)

mass distribution:

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

- blacks "remove" their potential: $\rho=\rho+\Phi$, $\Phi=0$



MMC Epoch Example

epochs:

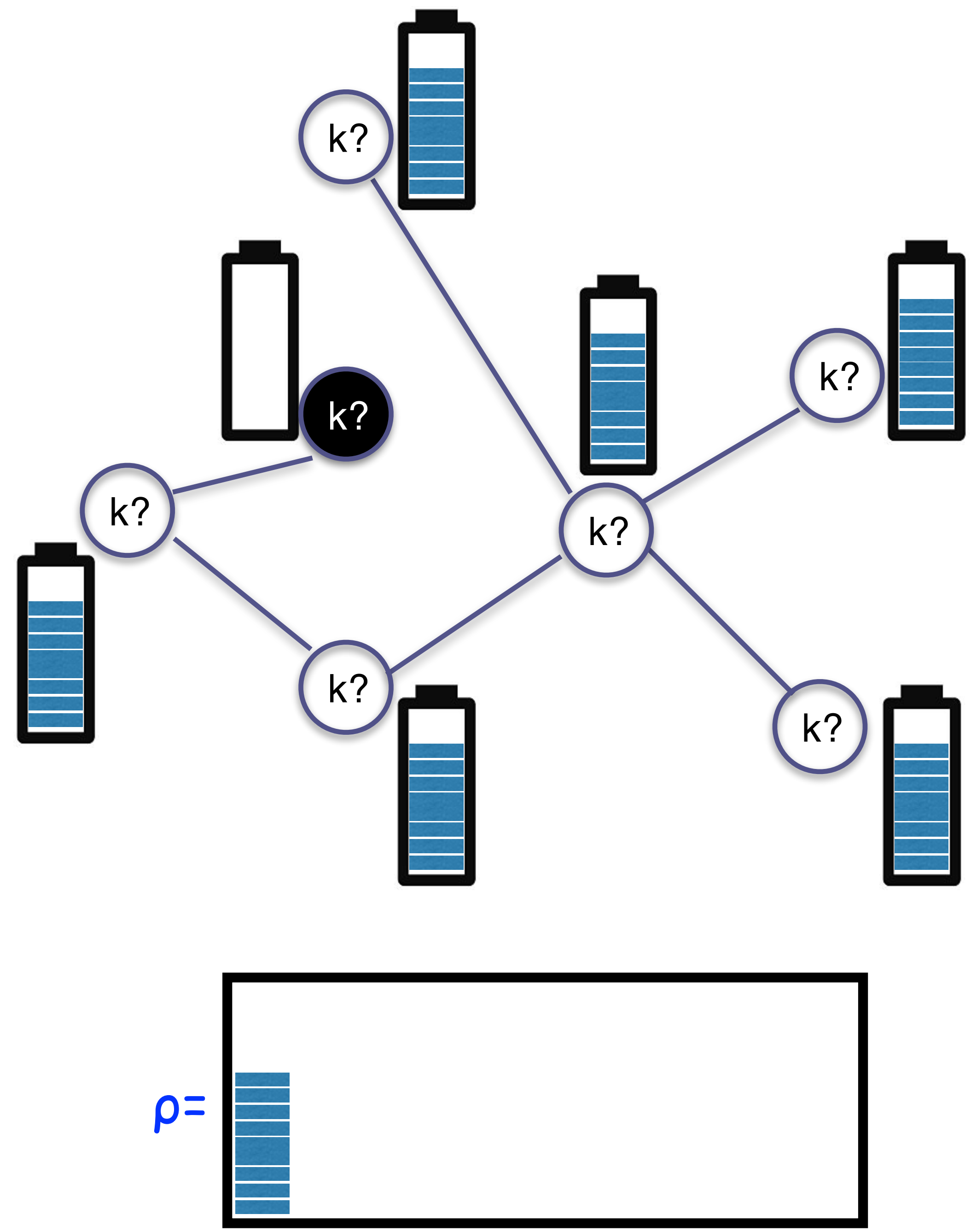
- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

(to let blacks remove "enough" potential ρ)

r(k) rounds:

(to "average" the current potentials Φ)



MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

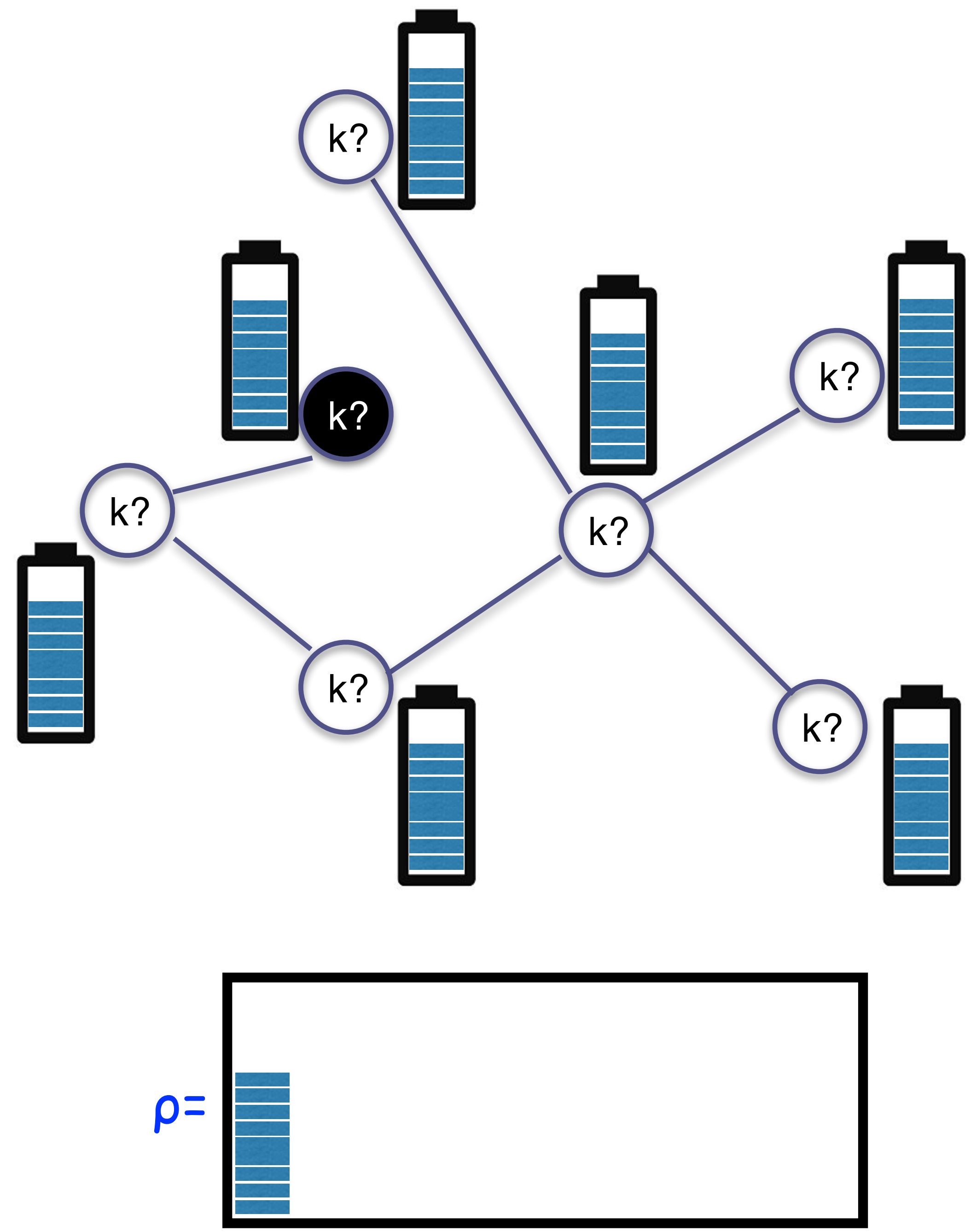
(to let blacks remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

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



MMC Epoch Example

epochs:

- one for each estimate $k=l+1,2(l+1),4(l+1),\ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

(to let blacks remove "enough" potential ρ)

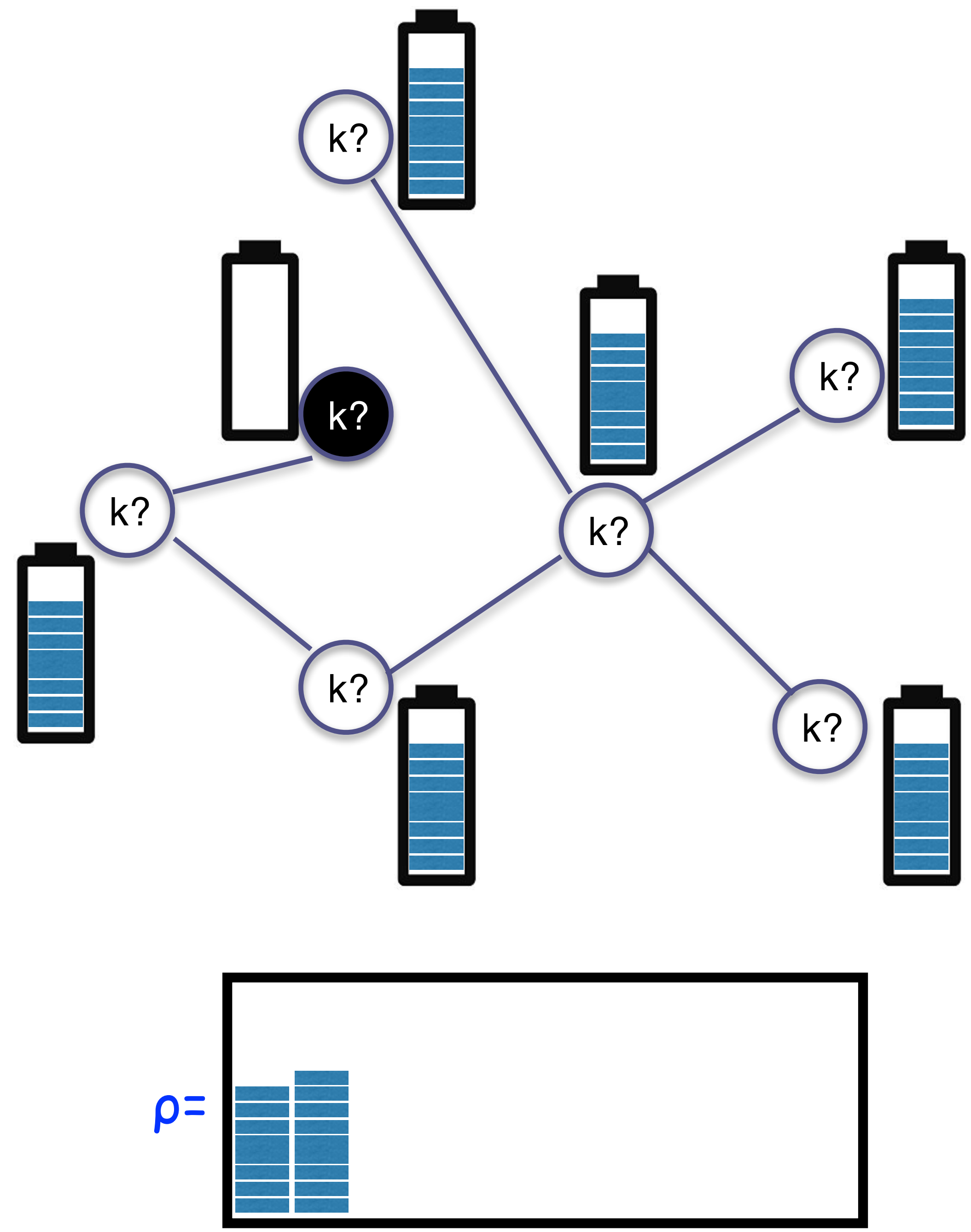
$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

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

- blacks "remove" their potential: $\rho=\rho+\Phi$, $\Phi=0$



MMC Epoch Example

epochs:

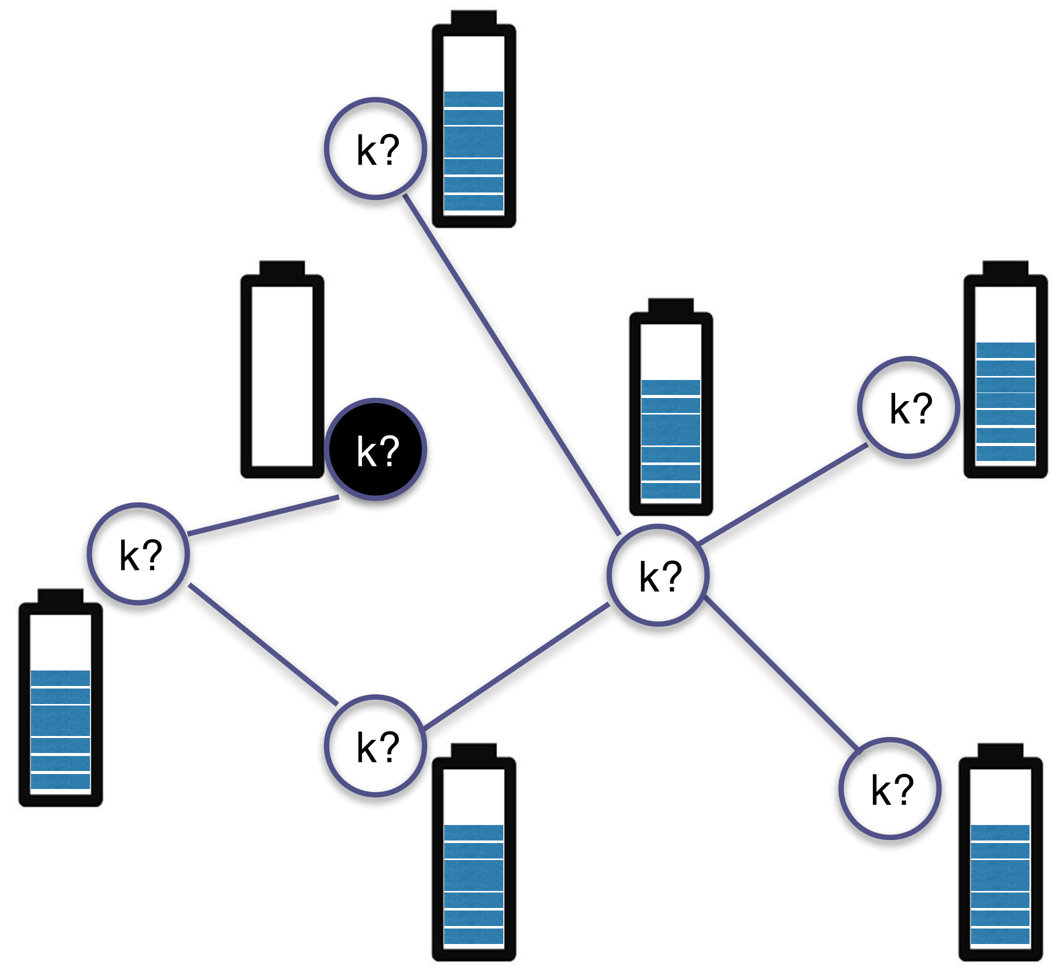
- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

(to let blacks remove "enough" potential ρ)

r(k) rounds:

(to "average" the current potentials Φ)



MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

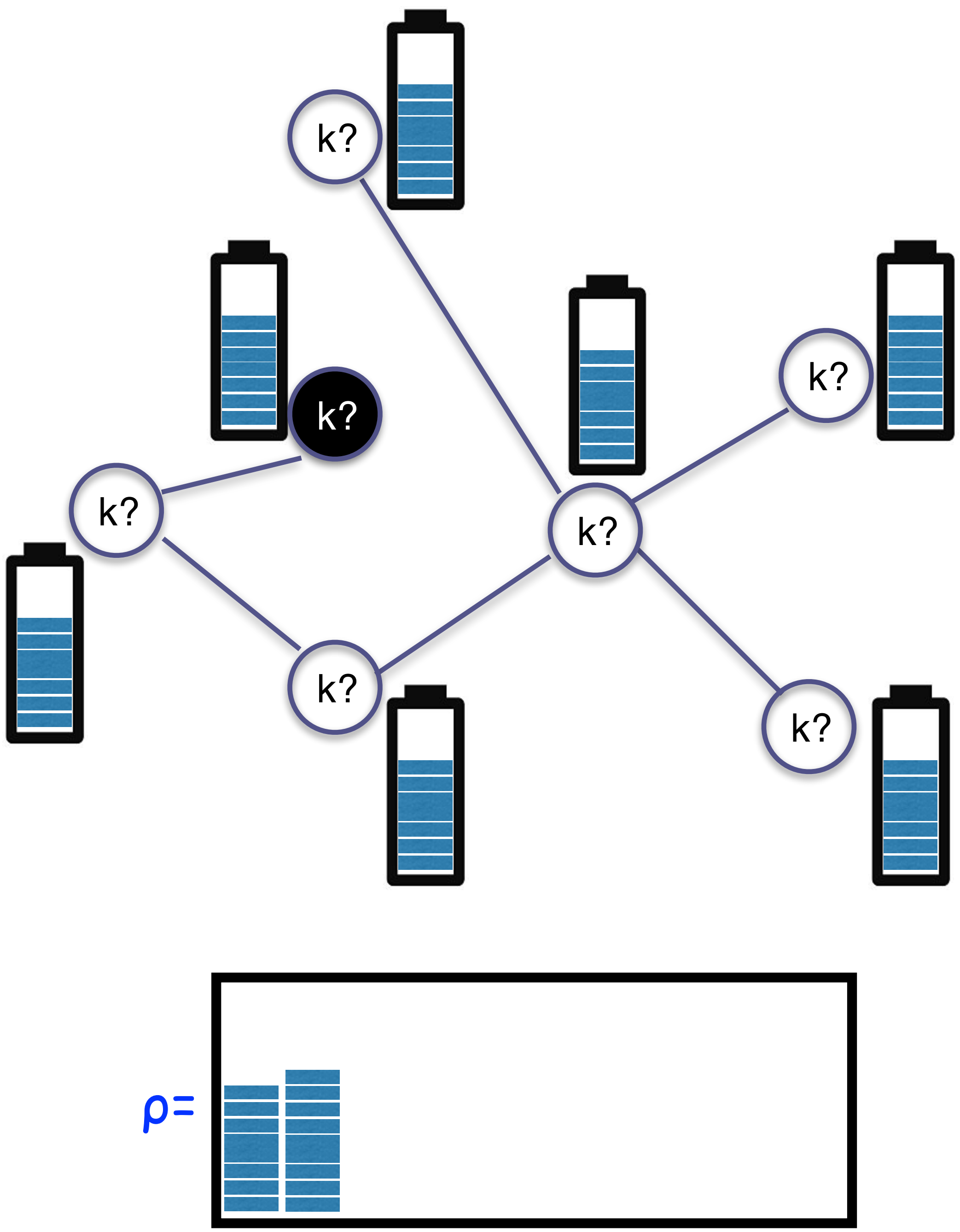
(to let blacks remove "enough" potential ρ)

r(k) rounds:

(to "average" the current potentials Φ)

mass distribution:

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



MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

(to let blacks remove "enough" potential ρ)

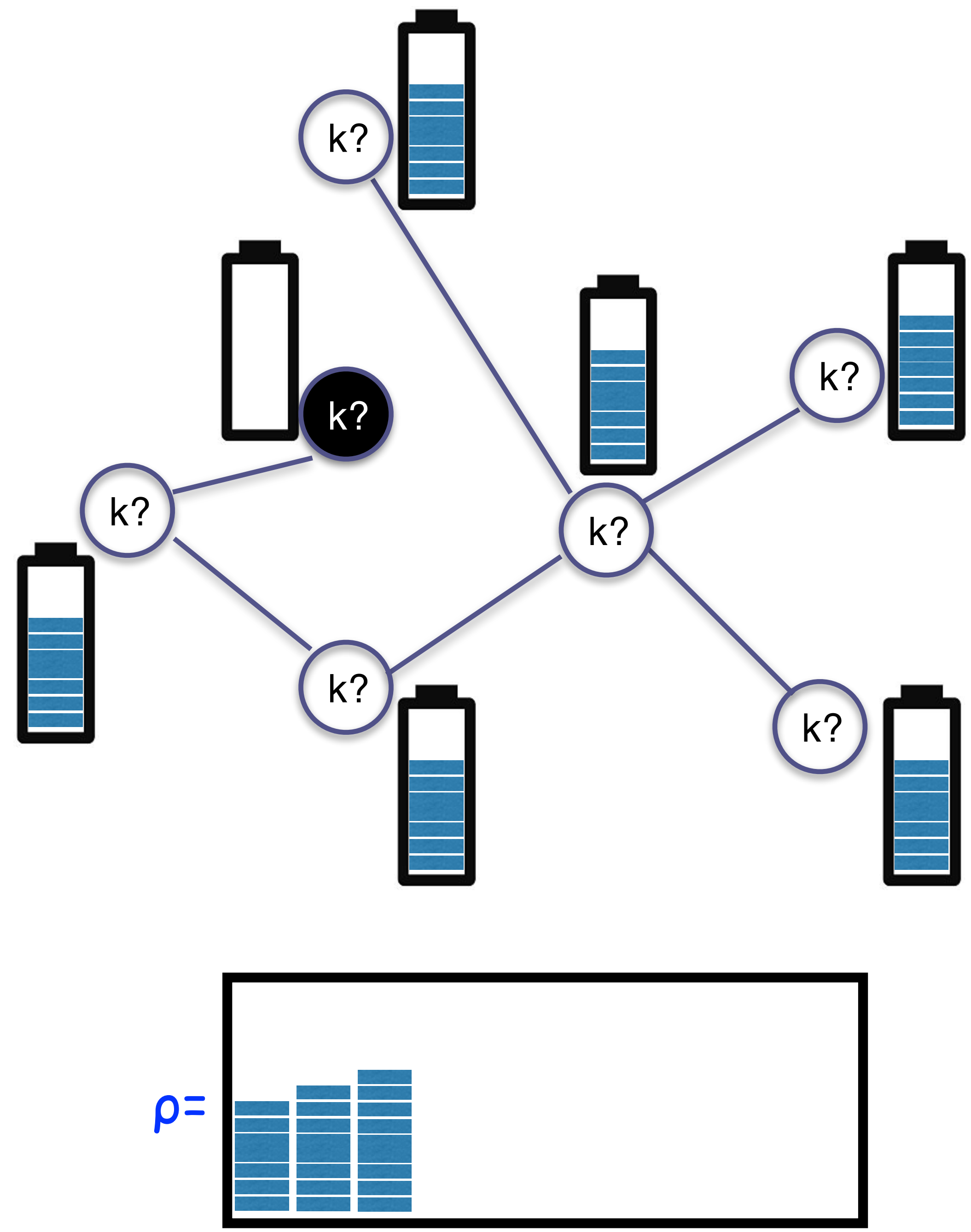
$r(k)$ rounds:

(to "average" the current potentials Φ)

mass distribution:

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

- blacks "remove" their potential: $\rho=\rho+\Phi$, $\Phi=0$



MMC Epoch Example

epochs:

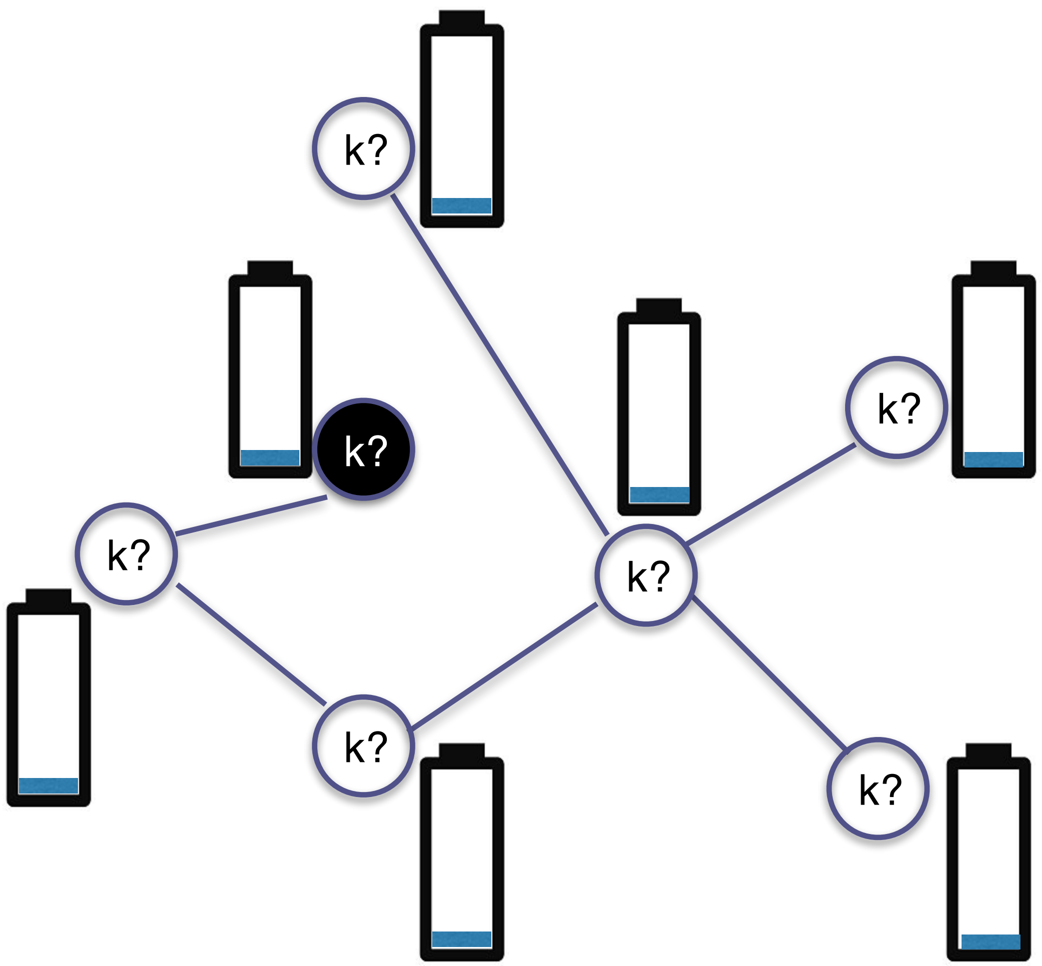
- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

$p(k)$ phases:

(to let blacks remove "enough" potential ρ)

$r(k)$ rounds:

(to "average" the current potentials Φ)



$\rho=$

MMC Epoch Example

epochs:

- one for each estimate $k=l+1, 2(l+1), 4(l+1), \ldots$
- initially, "potential" value: $\Phi_{white}=l$, $\Phi_{black}=0$

p(k) phases:

(to let blacks remove "enough" potential ρ)

r(k) rounds:

(to "average" the current potentials Φ)

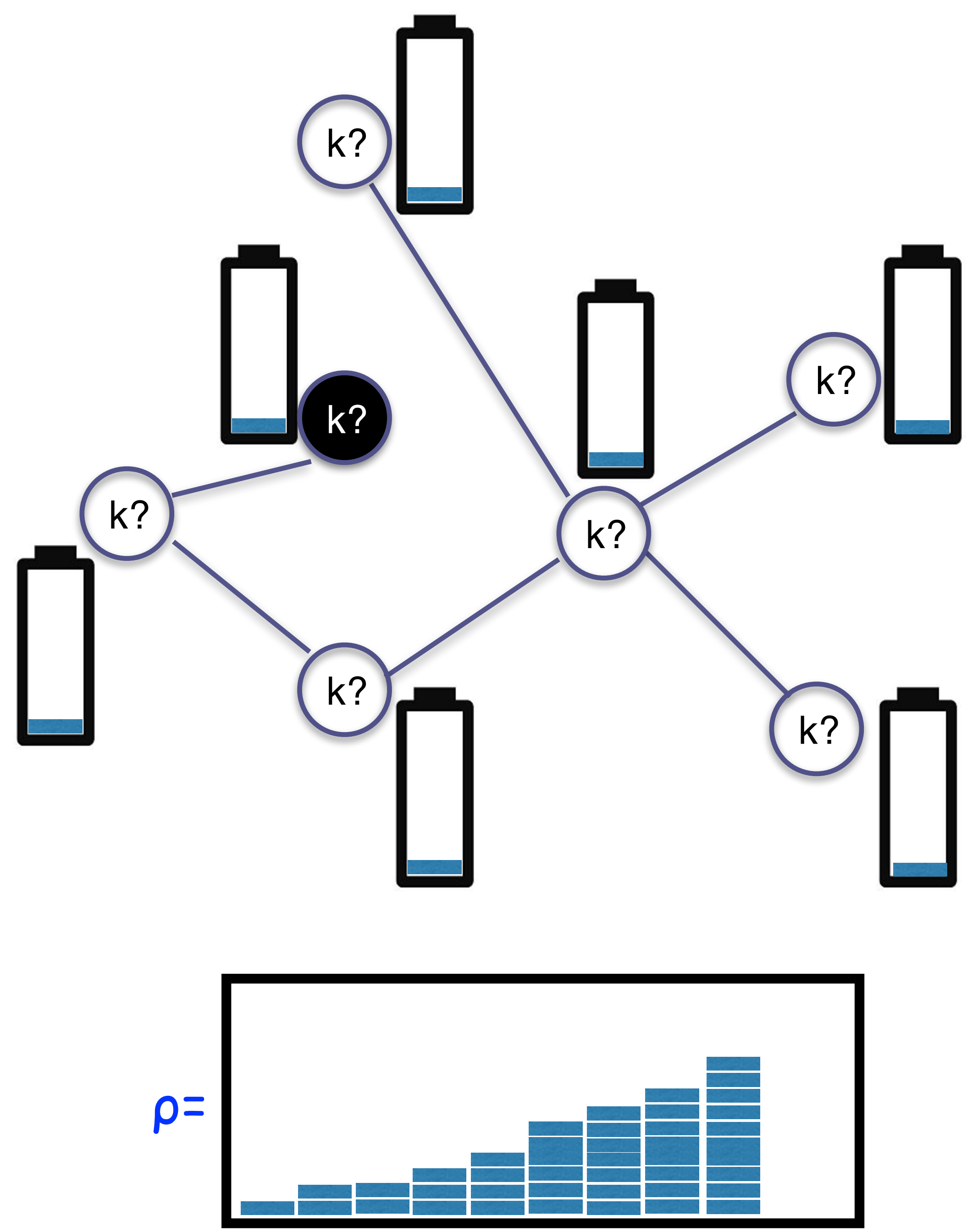
mass distribution:

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

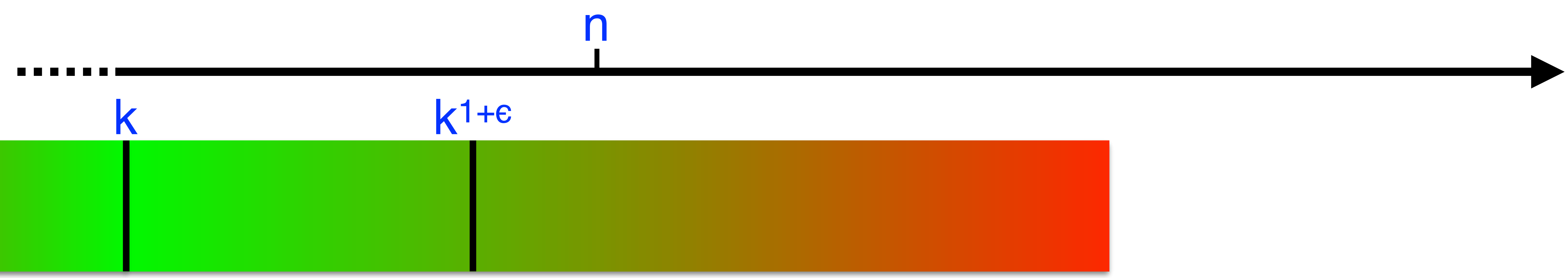
- blacks "remove" their potential: $\rho=\rho+\Phi$, $\Phi=0$

- blacks decide according to ρ
- blacks notify if $k >= n$
- try next k if needed

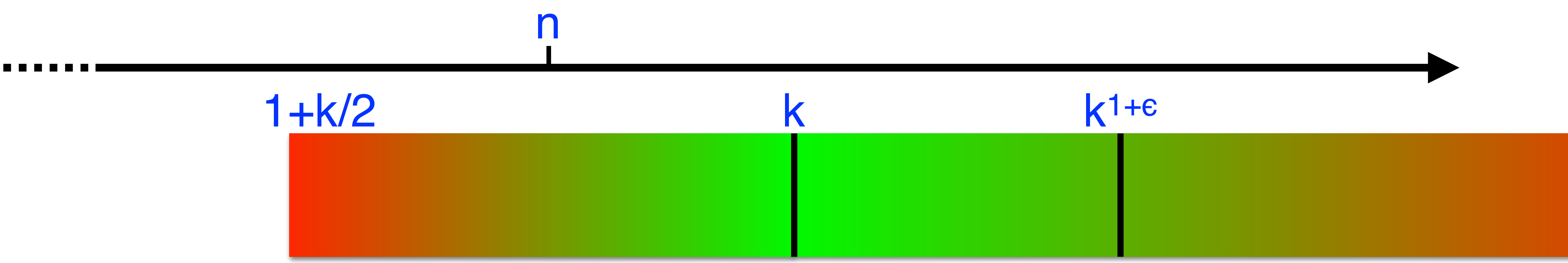
After p(k) phases…



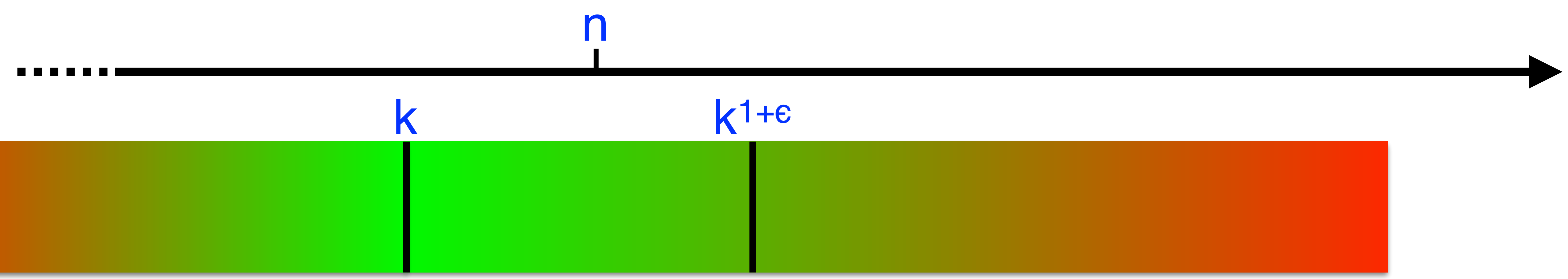
MMC Alarms



MMC Alarms

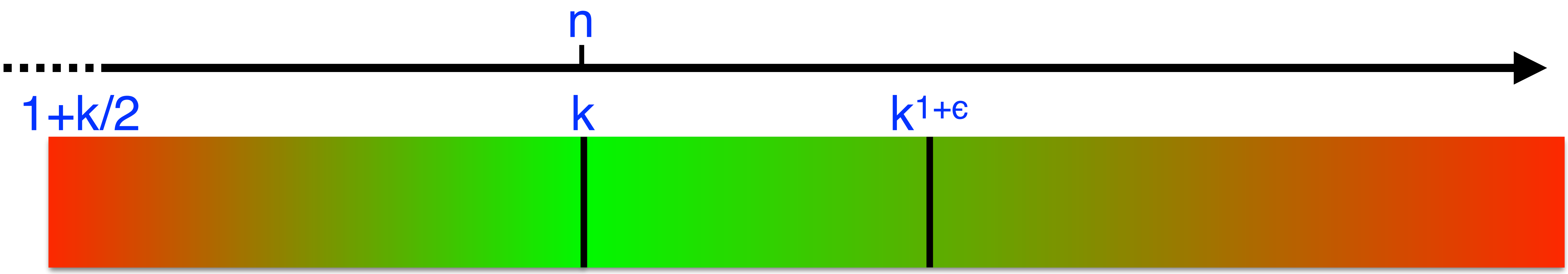


MMC Alarms

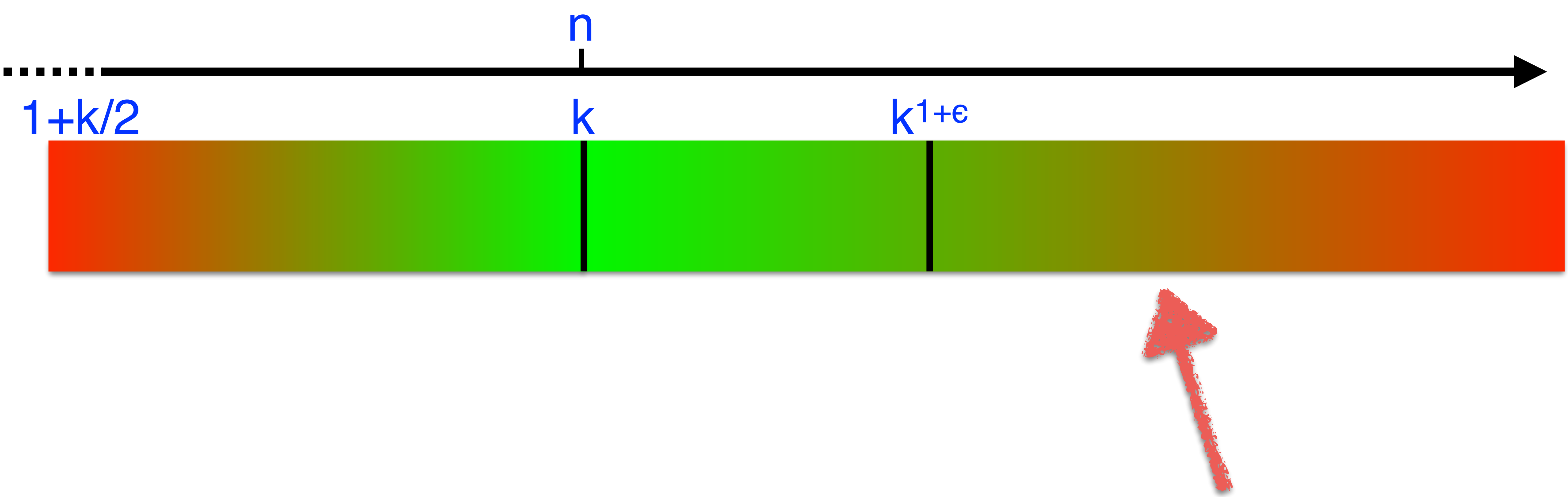


MMC Alarms

MMC Alarms



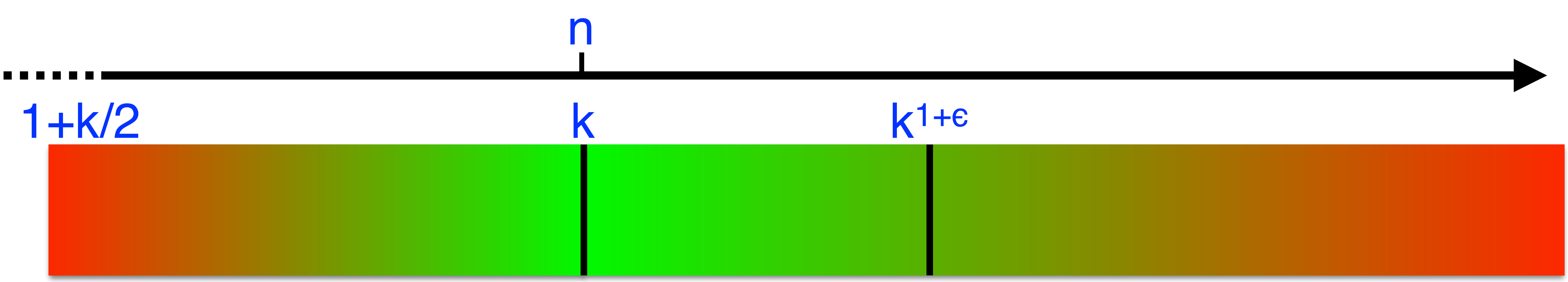
MMC Alarms



If n is "far" from k then not "many" nodes have "low" potential after phase 1,

so, blacks receive alarm from nodes with "high" potential "soon" after phase 1.

MMC Alarms

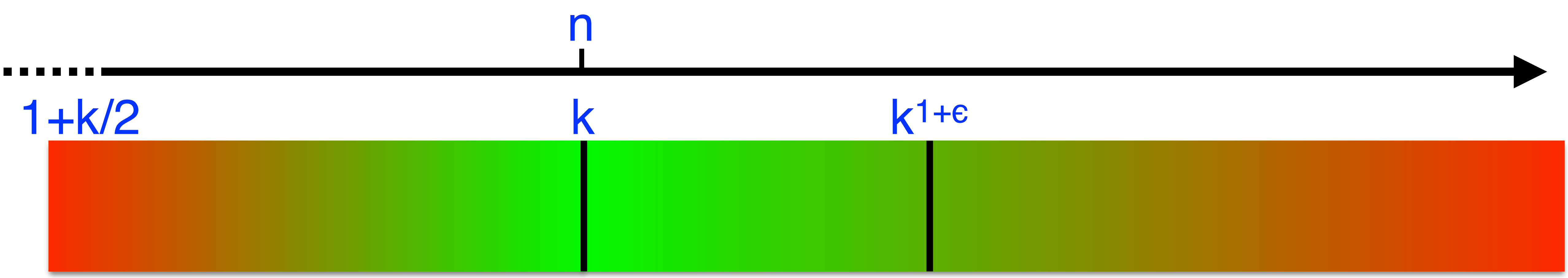


If n is “close” to k
from above
then blacks remove
“too much” potential.

If n is “far” from k then not
“many” nodes have “low”
potential after phase 1,

so, blacks receive alarm from
nodes with “high” potential
“soon” after phase 1.

MMC Alarms



If n below k then blacks remove "too little" potential.

If n is "close" to k from above then blacks remove "too much" potential.

If n is "far" from k then not "many" nodes have "low" potential after phase 1,

so, blacks receive alarm from nodes with "high" potential "soon" after phase 1.

Randomized Counting

As we showed, if ℓ is unknown or zero, $\exists$ executions that do not stop

$\Rightarrow$ we need black nodes, and we need to know how many,

we aim (stochastically) for $\ell=1$.

Randomized Counting

LLMC Key Ingredients:

- consider consecutive powers of 2 as values of K
- for each K
- each node chooses to be black with probability inverse of K
- run MMC for $\ell = 1$ and $k \leqslant K$
- if $K \geqslant n$ and there is one black node $\Rightarrow$ done

Randomized Counting

LLMC Key Ingredients:

- consider consecutive powers of 2 as values of K
- for each K
- each node chooses to be black with probability inverse of K
- run MMC for $\ell=1$ and $k \leqslant K$
- if $K \geqslant n$ and there is one black node $\Rightarrow$ done

Challenges:

- How to detect that $K \geqslant n$?
- If $\ell = 0$ no count, but what if $\ell > 1$?

Randomized Counting

LLMC Key Ingredients:

- consider consecutive powers of 2 as values of K
- for each K
- each node chooses to be black with probability inverse of K
- run MMC for $\ell = 1$ and $k \leqslant K$
- if $K \geqslant n$ and there is one black node $\Rightarrow$ done

Two additional techniques:

- Run parallel threads:
  - if # threads with $\ell = 0$ is large enough, $K \geqslant n$ is likely
- Take max count over threads:
  - count with $\ell > 1$ is smaller than with $\ell = 1$

LLMC

procedure
$K \leftarrow \lceil\lceil 12/(\epsilon) \rceil\rceil$ // $\lceil\lceil x \rceil\rceil$: the smallest power of 2 bigger than x
$Count \leftarrow \emptyset$ // set of potentially "good" estimates computed in threads
$EmptyThreads \leftarrow 0$ // number of threads with no black node detected
while $Count = \emptyset$ or $EmptyThreads \leq f(K)/2$ **do**
$Count \leftarrow \emptyset$, $EmptyThreads \leftarrow 0$
$K \leftarrow 2K$
Initiate $f(K) = 64\frac{\log(K/\epsilon)}{\log(e/(e-2))}$ parallel threads
// parallel computation and messages sharing same resources/medium
for each thread **do**
for each node **do**
Select to be a black node with probability $1/g(K)$, where $g(K) = K/2$
end for
$k \leftarrow MMC(K, 1)$ // refer to Figure 2
if $k > 0$ **then**
$Count \leftarrow Count \cup \{k\}$
end if
if no black node detected **then**
Increase $EmptyThreads$ by 1
end if
end for
end while
return $\max(Count)$ // Output the maximum number in $Count$ as the size n.
end procedure

LLMC

if K too small ⇒

prob of black too high ⇒

black nodes too big ⇒

empty threads too small

procedure

$K \leftarrow \lceil\lceil 12/(\epsilon) \rceil\rceil$ // $\lceil\lceil x \rceil\rceil$: the smallest power of 2 bigger than x

$Count \leftarrow \emptyset$ // set of potentially "good" estimates computed in threads

$EmptyThreads \leftarrow 0$ // number of threads with no black node detected

while $Count = \emptyset$ or $EmptyThreads \leq f(K)/2$ **do**

$Count \leftarrow \emptyset$, $EmptyThreads \leftarrow 0$

$K \leftarrow 2K$

Initiate $f(K) = 64\frac{\log(K/\epsilon)}{\log(e/(e-2))}$ parallel threads

// parallel computation and messages sharing same resources/medium

for each thread **do**

for each node **do**

Select to be a black node with probability $1/g(K)$, where $g(K) = K/2$

end for

$k \leftarrow MMC(K, 1)$ // refer to Figure 2

if $k > 0$ **then**

$Count \leftarrow Count \cup \{k\}$

end if

if no black node detected **then**

Increase $EmptyThreads$ by 1

end if

end for

end while

return $\max(Count)$ // Output the maximum number in $Count$ as the size n.

end procedure

when K gets close to n ⇒

even if ℓ =1, while K < n ⇒

it is k ⩽ K < n ⇒

no count

LLMC

if K too small ⇒

prob of black too high ⇒

black nodes too big ⇒

empty threads too small

```
procedure
    K ← ⌈⌈12/(ε)⌉⌉            // ⌈⌈x⌉⌉: the smallest power of 2 bigger than x
    Count ← ∅                 // set of potentially "good" estimates computed in threads
    EmptyThreads ← 0          // number of threads with no black node detected
    while Count = ∅ or EmptyThreads ≤ f(K)/2 do
        Count ← ∅, EmptyThreads ← 0
        K ← 2K
        Initiate f(K) = 64 log(K/ε)/log(e/(e−2)) parallel threads
          // parallel computation and messages sharing same resources/medium
        for each thread do
            for each node do
                Select to be a black node with probability 1/g(K), where g(K) = K/2
            end for
            k ← MMC(K, 1)                                    // refer to Figure 2
            if k > 0 then
                Count ← Count ∪ {k}
            end if
            if no black node detected then
                Increase EmptyThreads by 1
            end if
        end for
    end while
    return max(Count)  // Output the maximum number in Count as the size n.
end procedure
```

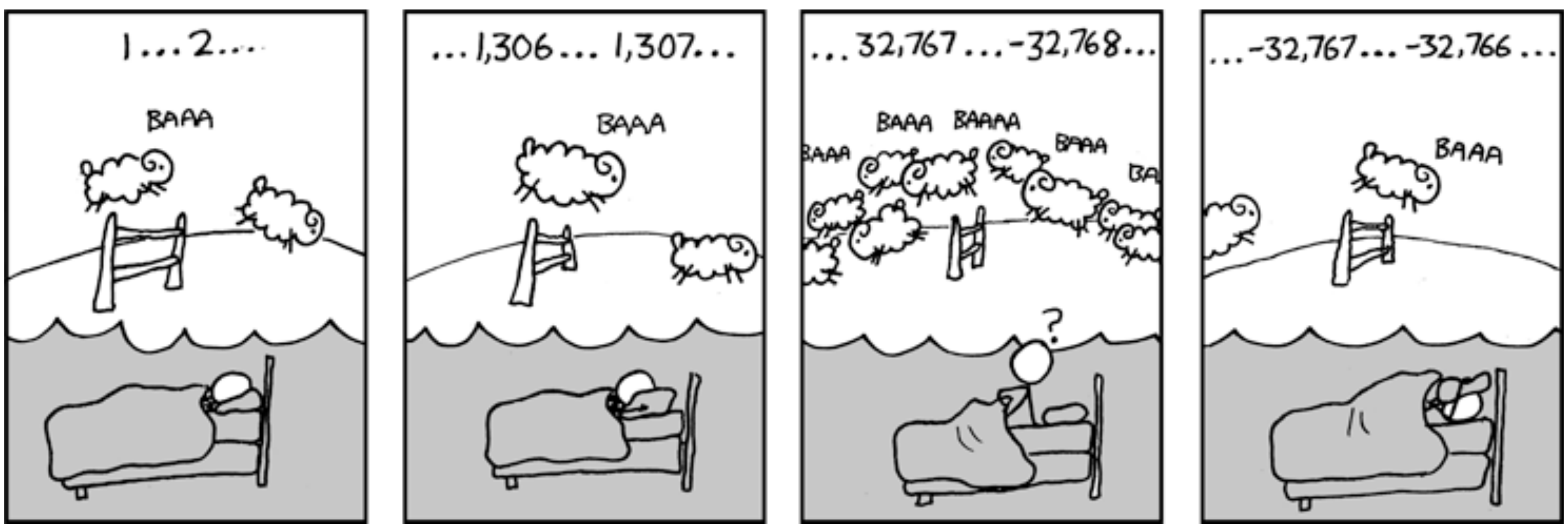
ICALP 2018 Open Questions

- Many distinguished nodes.
- Improve upper and/or lower bounds.
- Other computations in ADNs (beyond sum, avg, etc.).
- Asynchronous protocol.

ICALP 2018 2019 Open Questions

- Many distinguished nodes. ✔
- Improve upper and/or lower bounds. ✔
- Other computations in ADNs (beyond sum, avg, etc.).
- Asynchronous protocol.

Thank you!


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