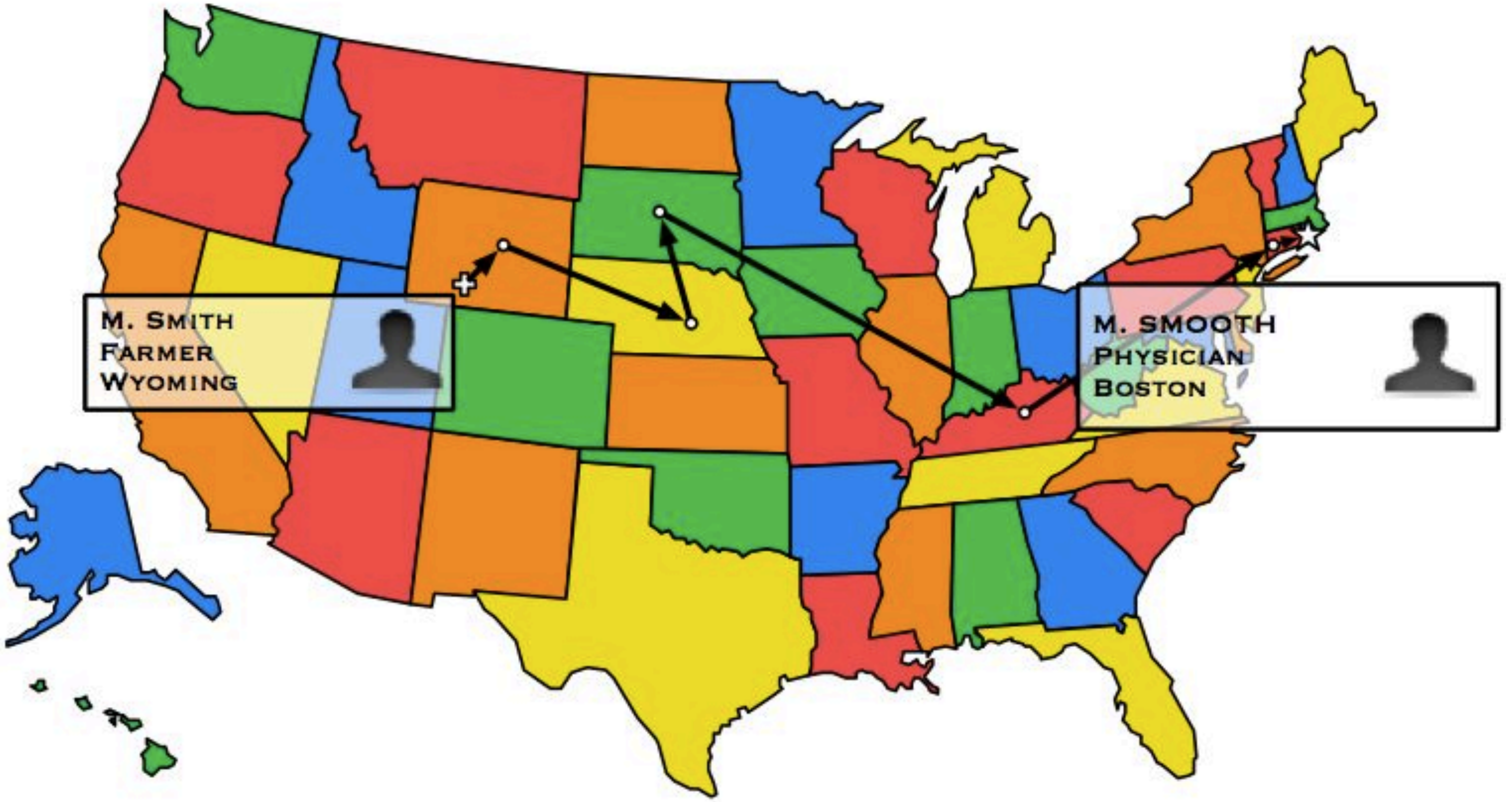
Search Algorithms in Small Worlds

Nicolas Schabanel
CNRS - LIAFA, Université Paris Diderot - IXXI, École normale supérieure de Lyon

Includes joint work with:
George Giakkoupis, University of Calgary, Canada
Emmanuelle Lebhar, École normale supérieure de Lyon, France (at that time)

History: Stanley Milgram experiment (1967)

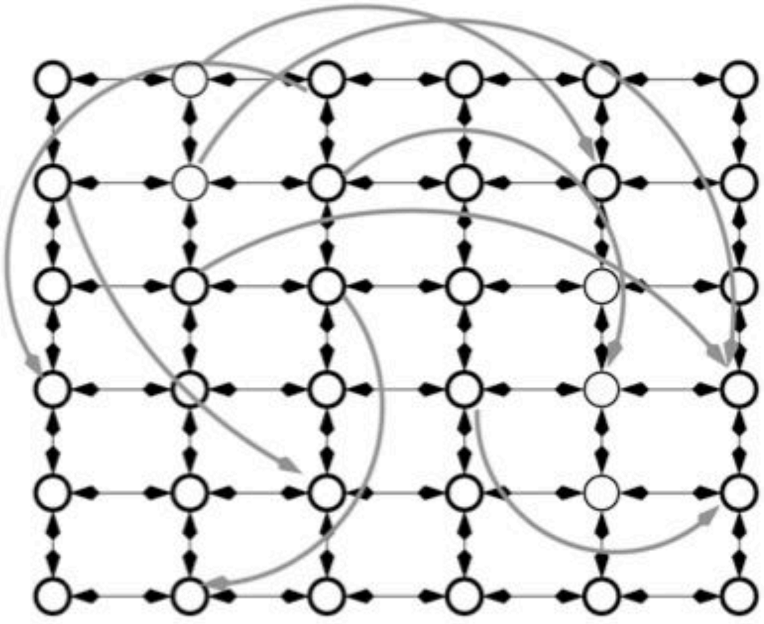
Stanley Milgram (1960s)



- **(1968)** Every individual can **find short paths** (length ≤ 6 on average) towards an arbitrary individual ***based on his own local view only!***

The network model: Jon Kleinberg (2000)

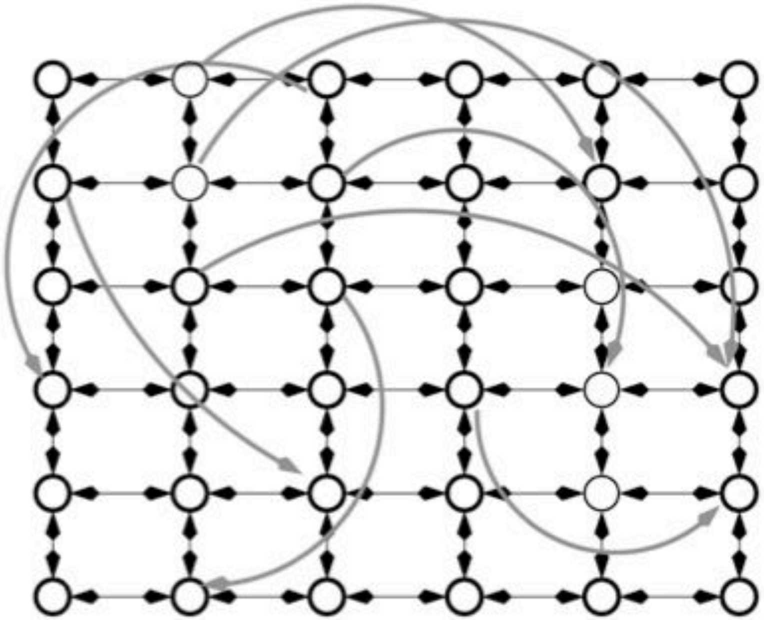
Jon Kleinberg's small world networks (2000)



$$\Pr\{u \to v\} \propto \frac{1}{d_{\mathbf{grid}}(u,v)^{\alpha}}$$

- **Decentralized search algorithm:**
  - Does **not know** the long range links *beforehand*
  - Can only ask for the long range contacts of **already visited nodes**
- **Example: Greedy algorithm** - Go to the grid-closest contact to the target

Jon Kleinberg's small world networks (2000)



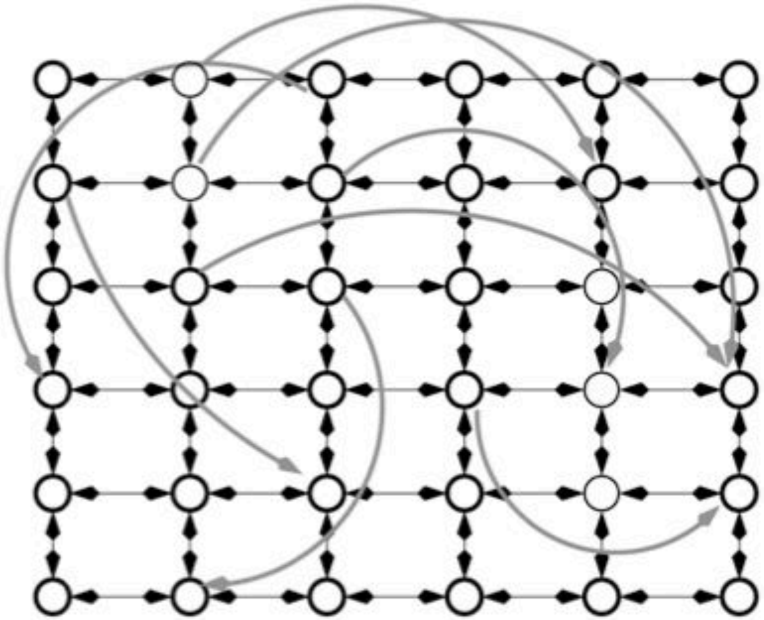
Parameter $\boldsymbol{\alpha}$

$$\Pr\{u \to v\} \propto \frac{1}{d_{\mathbf{grid}}(u,v)^{\alpha}}$$

Geographic distance
(w.r. the grid)

- **Decentralized search algorithm:**
  - Does **not know** the long range links *beforehand*
  - Can only ask for the long range contacts of **already visited nodes**
- **Example: Greedy algorithm** - Go to the grid-closest contact to the target

Jon Kleinberg's small world networks (2000)



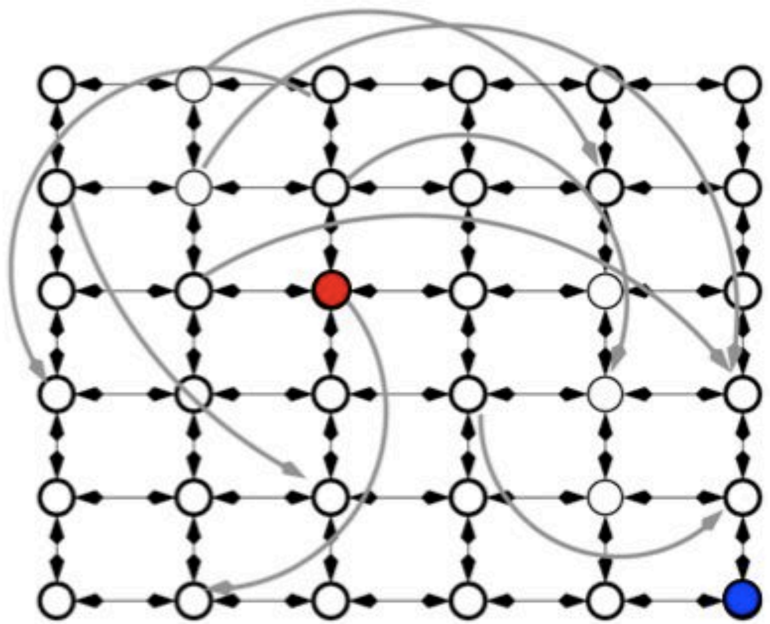
Parameter **α**

$$\Pr\{u \to v\} \propto \frac{1}{d_{\mathbf{grid}}(u,v)^{\alpha}}$$

Geographic distance
(w.r. the grid)

- **Decentralized search algorithm:**
  - Does **not know** the long range links *beforehand*
  - Can only ask for the long range contacts of **already visited nodes**
- **Example: Greedy algorithm** - Go to the grid-closest contact to the target

Jon Kleinberg's small world networks (2000)



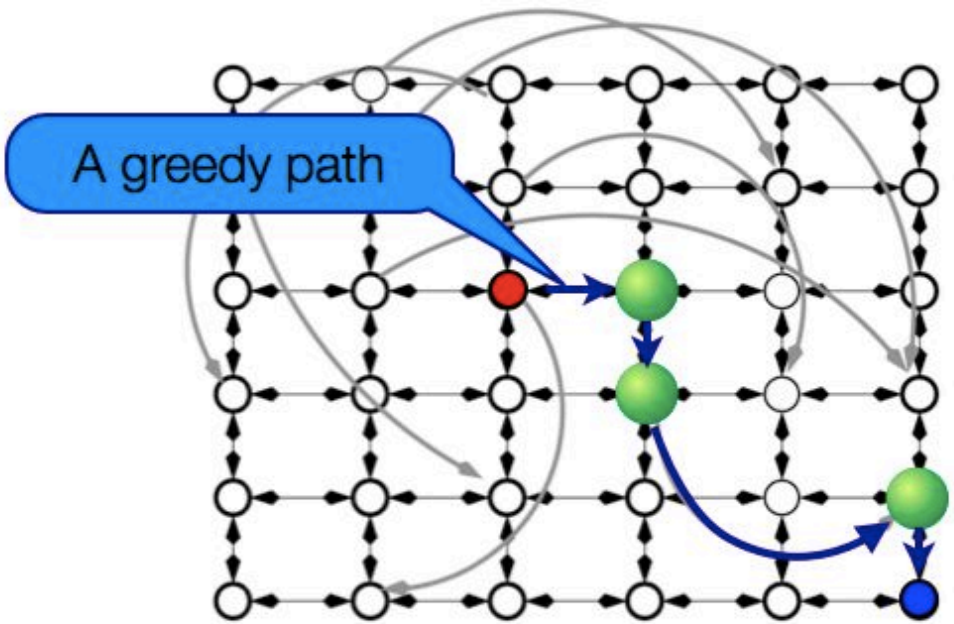
$$\Pr\{u \to v\} \propto \frac{1}{d_{\mathbf{grid}}(u,v)^{\alpha}}$$

Parameter **α**

Geographic distance (w.r.t. the grid)

- **Decentralized search algorithm:**
  - Does **not know** the long range links *beforehand*
  - Can only ask for the long range contacts of **already visited nodes**
- **Example: Greedy algorithm** - Go to the grid-closest contact to the target

Jon Kleinberg's small world networks (2000)



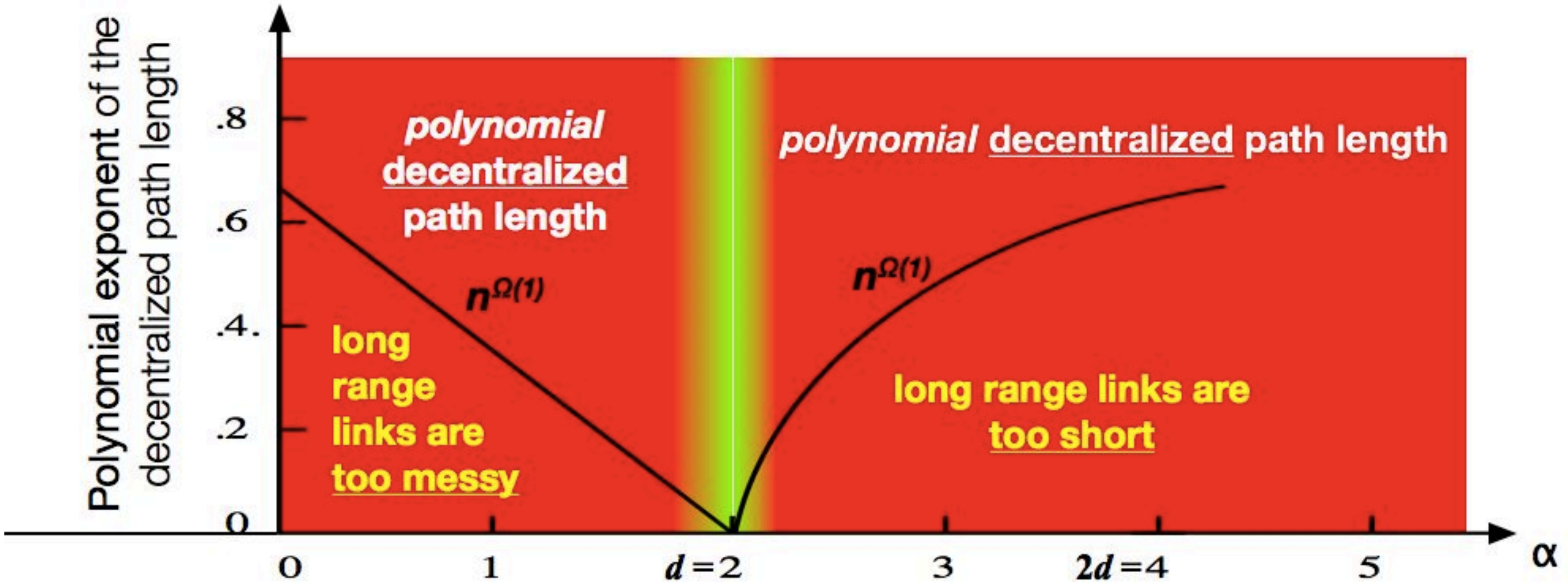
Parameter $\boldsymbol{\alpha}$

$$\Pr\{u \to v\} \propto \frac{1}{d_{\mathbf{grid}}(u,v)^{\alpha}}$$

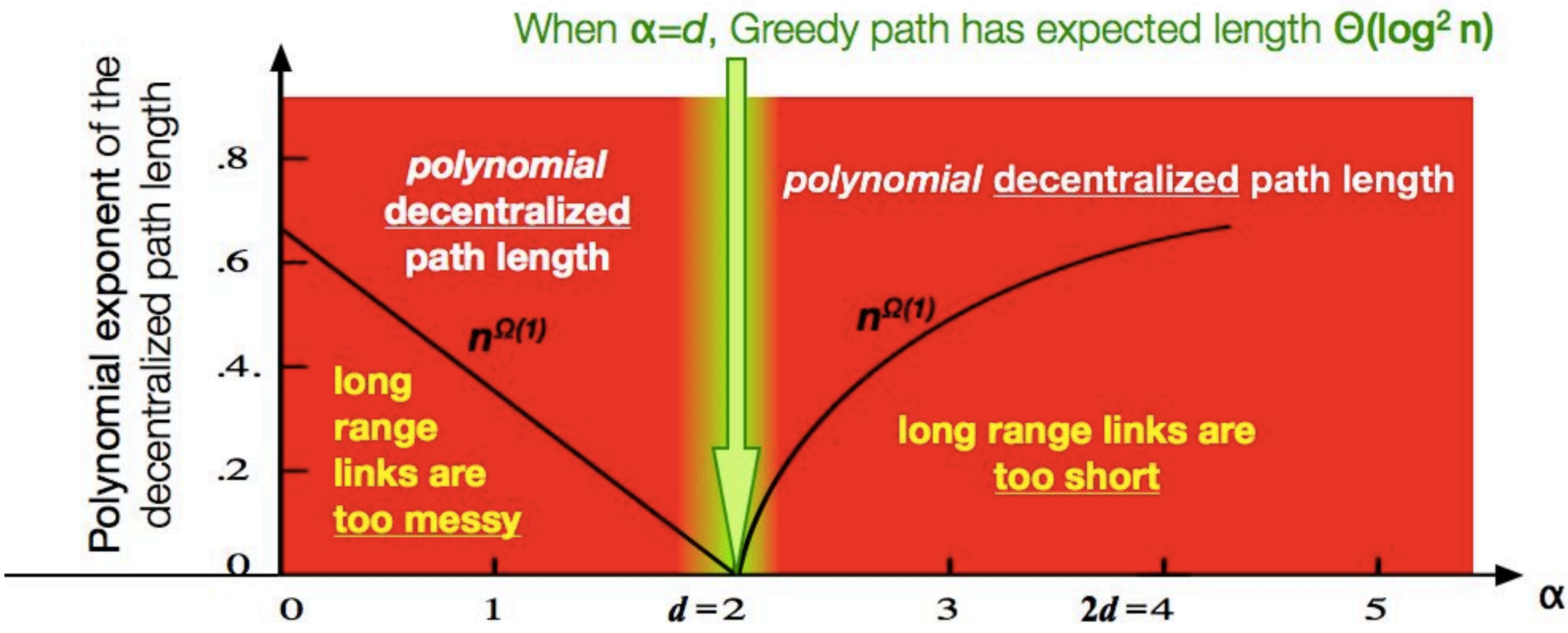
Geographic distance
(w.r.t. the grid)

- **Decentralized search algorithm:**
  - Does **not know** the long range links *beforehand*
  - Can only ask for the long range contacts of **already visited nodes**
- **Example: Greedy algorithm** - Go to the grid-closest contact to the target

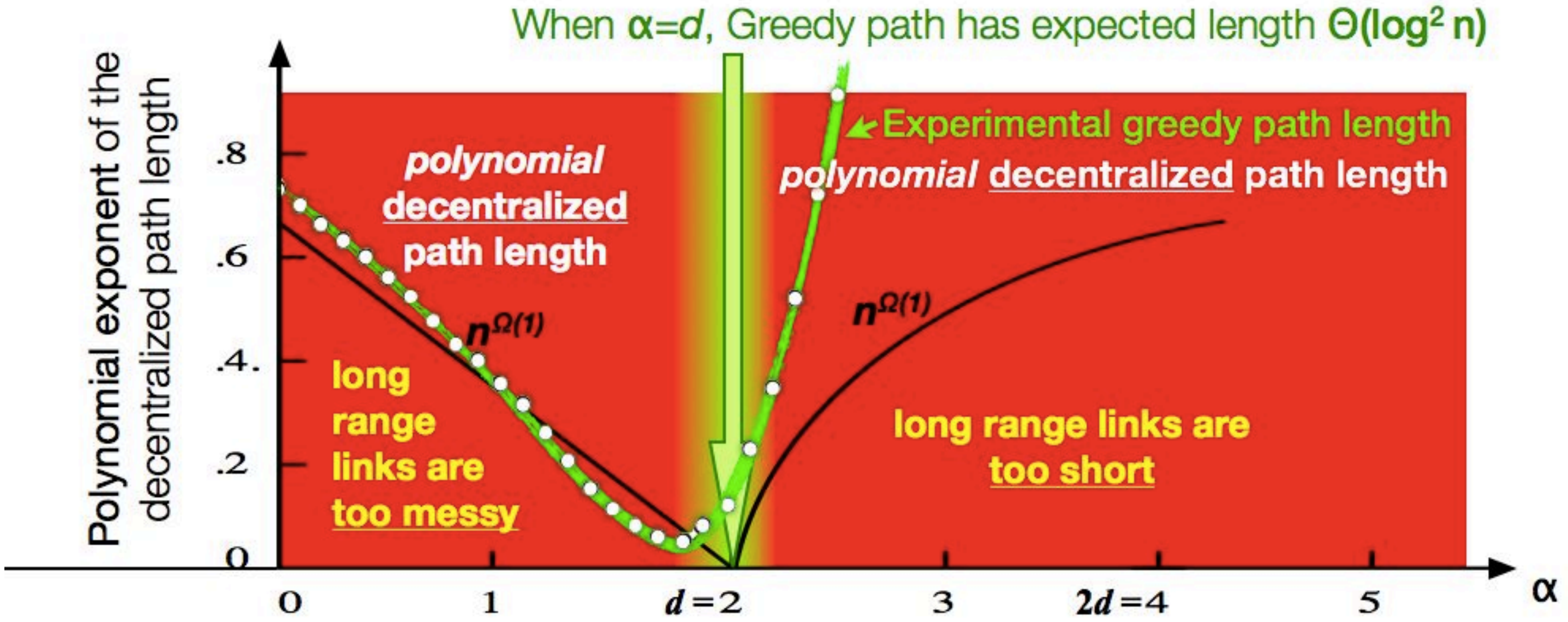
When is decentralized search possible/efficient? Lower bounds (Kleinberg, 2000)



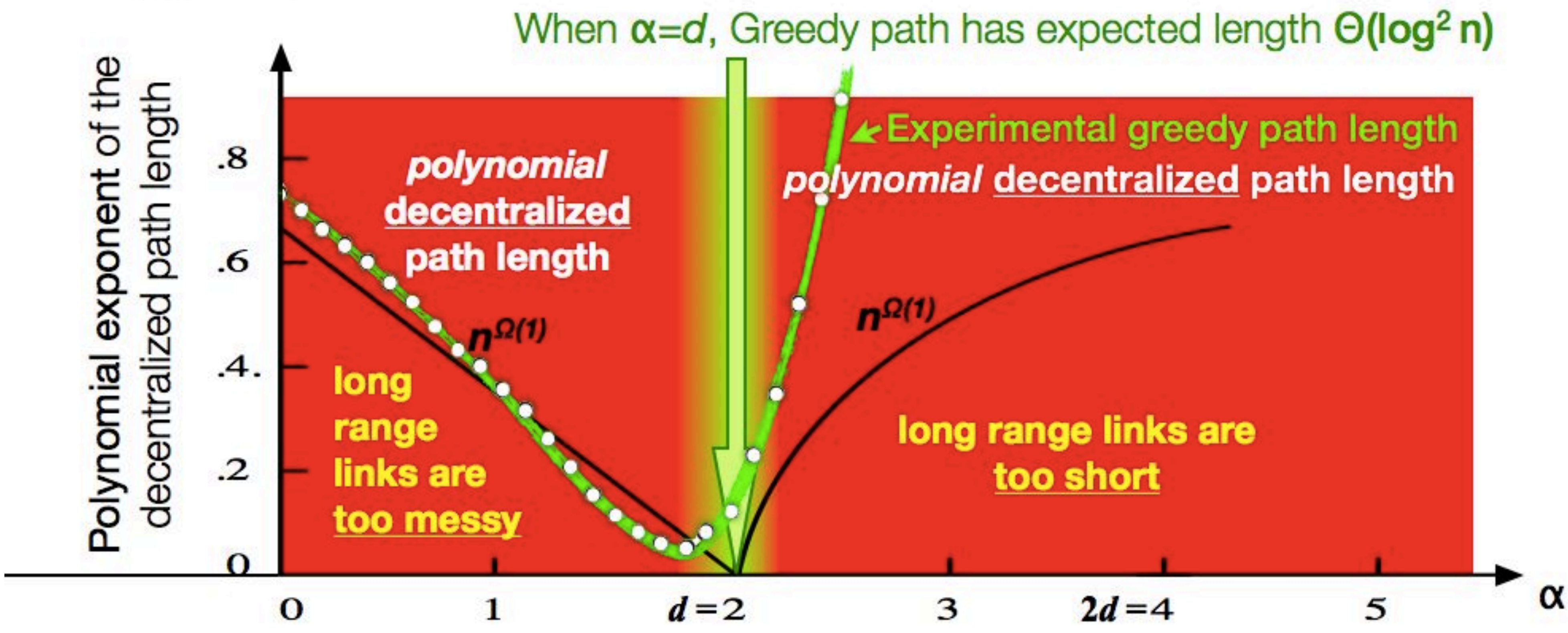
When is decentralized search possible/efficient? Lower bounds (Kleinberg, 2000)



When is decentralized search possible/efficient? Lower bounds (Kleinberg, 2000)



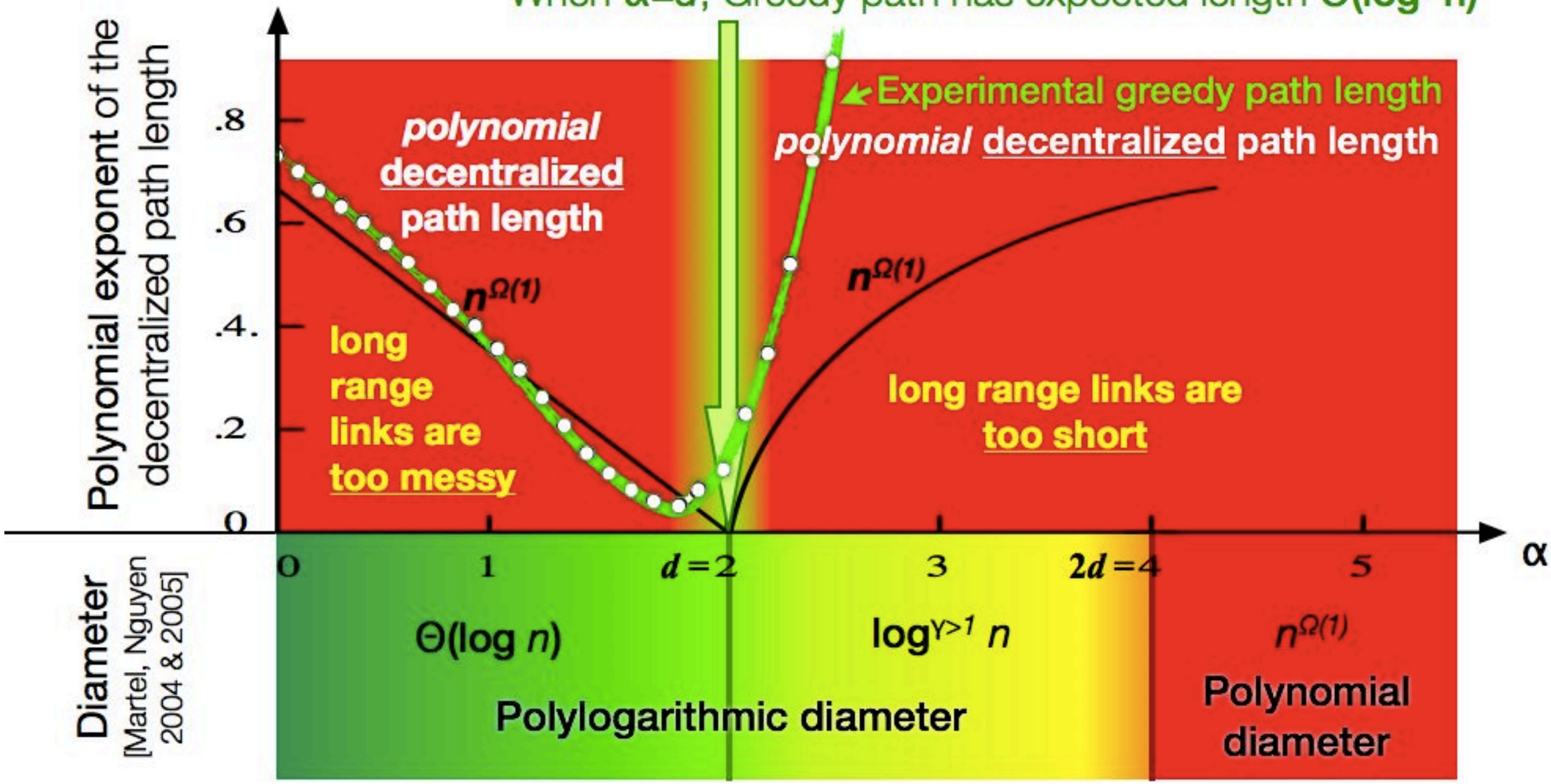
Kleinberg's result (2000): Long-range contacts distribution has to be ***correlated to the underlying geography (α=d) for decentralized search to work***



When is decentralized ... Lo...

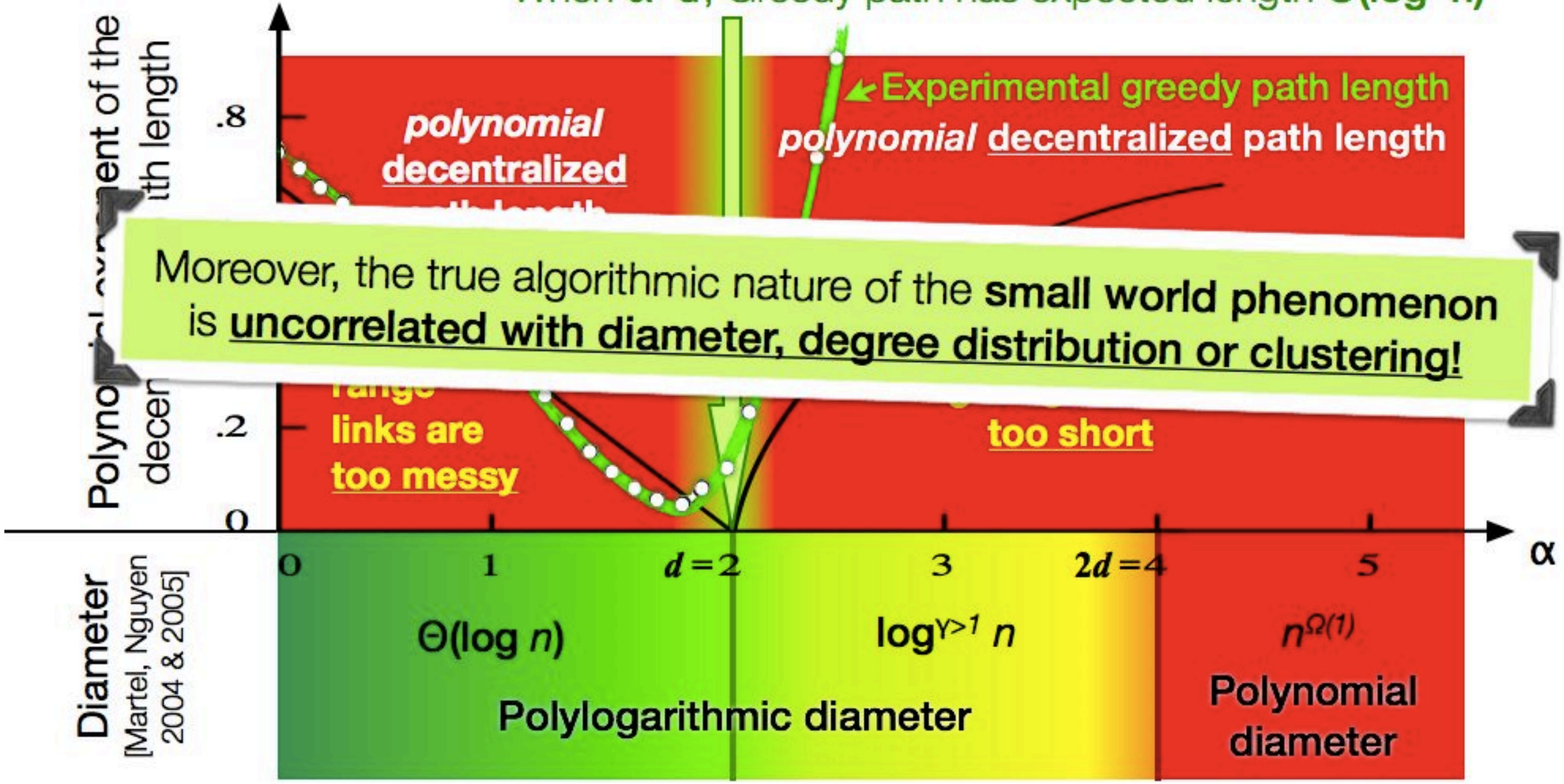
Kleinberg's result (2000): Long-range contacts distribution has to be ***correlated to the underlying geography (α=d) for decentralized search to work***

When **α**=*d*, Greedy path has expected length **Θ(log² n)**



Kleinberg's result (2000): Long-range contacts distribution has to be ***correlated to the underlying geography (α=d) for decentralized search to work***

When **α=*d***, Greedy path has expected length **Θ(log² n)**



Moreover, the true algorithmic nature of the small world phenomenon is **uncorrelated with diameter, degree distribution or clustering!**

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm?
- Can we **reach the true diameter** of the graph?
- What would be an **optimal** search algorithm?

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?
- How do **real people use** the social links?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm?
- Can we **reach the true diameter** of the graph?
- What would be an **optimal** search algorithm?

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?
- How do **real people use** the social links?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm? ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph?
- What would be an **optimal** search algorithm?

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?
- How do **real people use** the social links?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm? ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph? ➡ ***Yes:*** if grid-dimension ≥ 2
  ➡ ***No:*** if grid-dimension $= 1$
- What would be an **optimal** search algorithm? ➡ $\exists$ an exploration-based algorithm finds paths of **optimal** length **[GS'11]**

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?

- How do **real people use** the social links?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm? ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph? ➡ ***Yes:*** if grid-dimension ≥ 2
  ➡ ***No:*** if grid-dimension = 1
- What would be an **optimal** search algorithm? ➡ ∃ an exploration-based algorithm finds paths of **optimal** length **[GS'11]**

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?

- How do **real people use** the social links?

Kleinberg's model & results raises
two main types of questions

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm? ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph? ➡ ***Yes:*** if grid-dimension ≥ 2
  ➡ ***No:*** if grid-dimension = 1
- What would be an **optimal** search algorithm? ➡ ∃ an exploration-based algorithm finds paths of **optimal** length **[GS'11]**

Social network & behaviors modeling:

- How does this **"link" topology emerge** ? ➡ **Several propositions:**
  - **[CM'08]:** rewiring until matching a random maximum path length
  - **[CFL'08]:** friends follow random walk and we forget them

  ➡ ***Mostly still open...***
- How do **real people use** the social links?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm?
  ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph?
  ➡ ***Yes:*** if grid-dimension ≥ 2
  ➡ ***No:*** if grid-dimension = 1
- What would be an **optimal** search algorithm?
  ➡ ∃ an exploration-based algorithm finds paths of **optimal** length **[GS'11]**

Social network & behaviors modeling:

- How does this **"link" topology emerge** ?
  ➡ **Several propositions:**
  - **[CM'08]:** rewiring until matching a random maximum path length
  - **[CFL'08]:** friends follow random walk and we forget them

  ➡ ***Mostly still open...***
- How do **real people use** the social links?
  ➡ Is greedy algorithm a good model for human behavior in Milgram's experiment?
  ➡ Could other algorithms would make better sense?
  ➡ How to test that?

Kleinberg's model & results raises **two main types of questions**

Algorithms & peer-to-peer protocol design:

- Can we **beat** the **greedy** algorithm? ➡ ***Yes:*** We can find shorter paths
  ➡ ***No:*** Every algorithm has to visit at least as many nodes **[NW'04]**
- Can we **reach the true diameter** of the graph? ➡ ***Yes:*** if grid-dimension ≥ 2
  ➡ ***No:*** if grid-dimension = 1
- What would be an **optimal** search algorithm? ➡ ∃ an exploration-based algorithm finds paths of **optimal** length **[GS'11]**

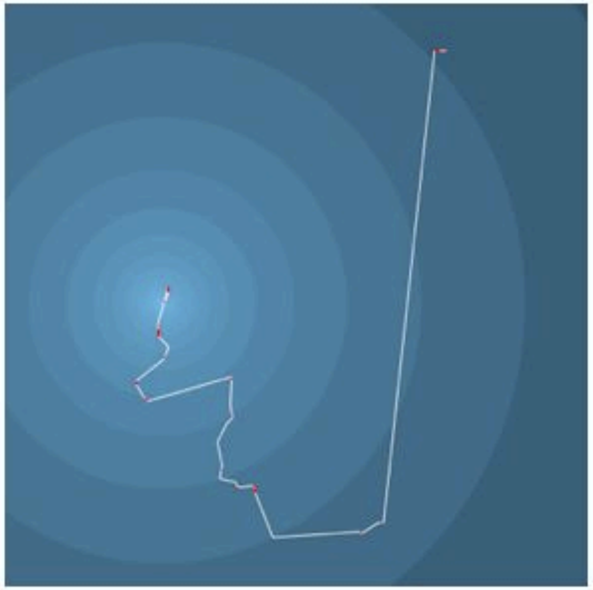
[GS'11] Kleinberg-based peer-to-peer protocols should thus rather use **bi-dimensional grid** than an 1-dimensional ring

- **[CFL'08]:** friends follow random walk and we forget them
  ➡ ***Mostly still open...***

- How do **real people use** the social links? ➡ Is greedy algorithm a good model for human behavior in Milgram's experiment?
  ➡ Could other algorithms would make better sense?
  ➡ How to test that?

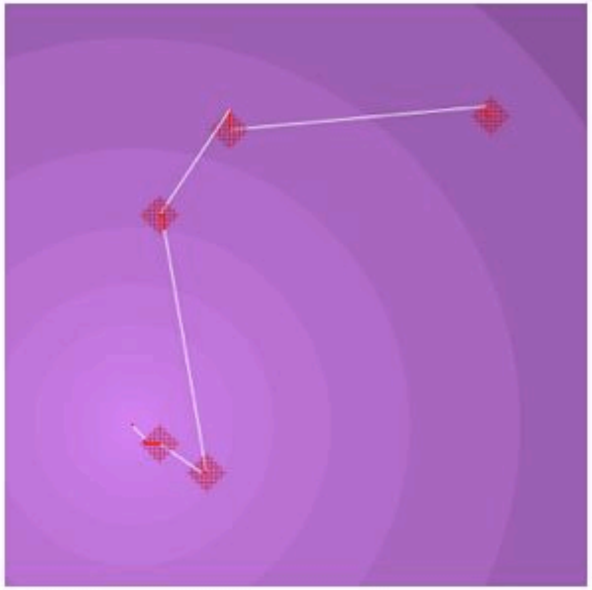
Three main types of algorithms proposed in literature to beat the greedy

Greedy algorithm



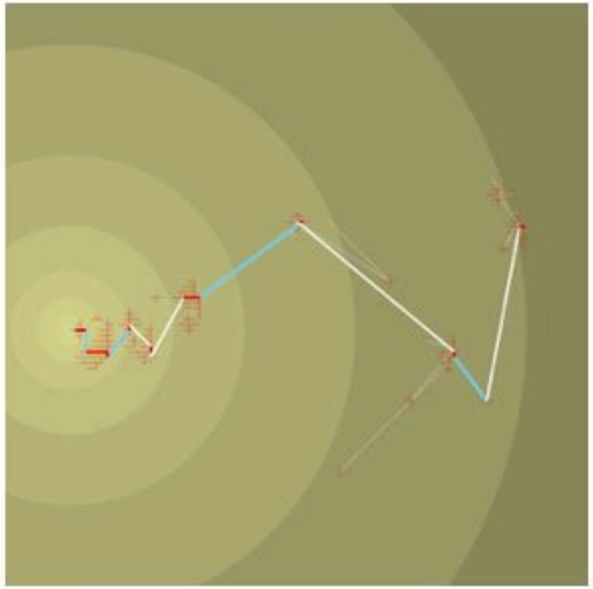
[K'00, BFKK'01]

Algorithms based on **local exploration**



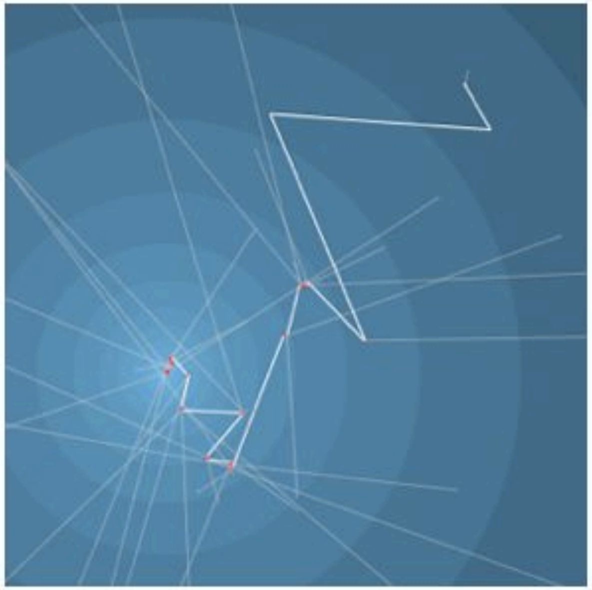
[CGS'02, NW'04, **FGP'04**]

Algorithms based on **non-local exploration**



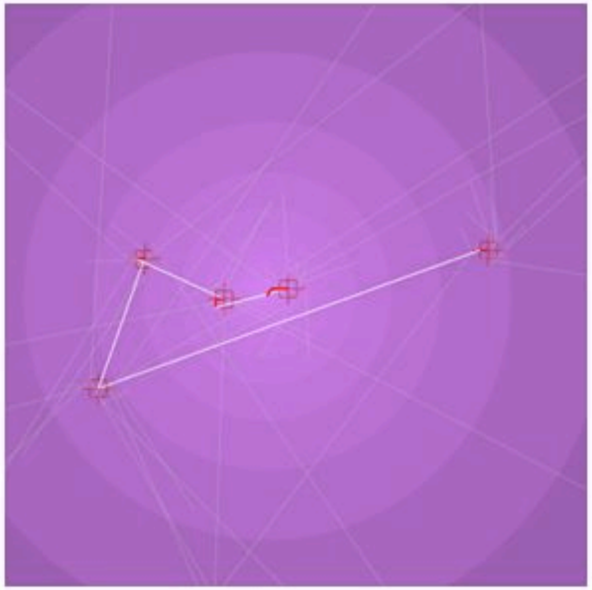
[LS'04, LS'05, GS'11]

The greedy algorithm



- **Algorithm [K'00]:**
  Always forward to the closest (local or long-range) neighbor w.r.t. the target
- Computes paths of expected length **$\Theta(\log^2 n)$**
- Visit **$\Theta(\log^2 n)$** nodes on expectation

Algorithms based on local exploration

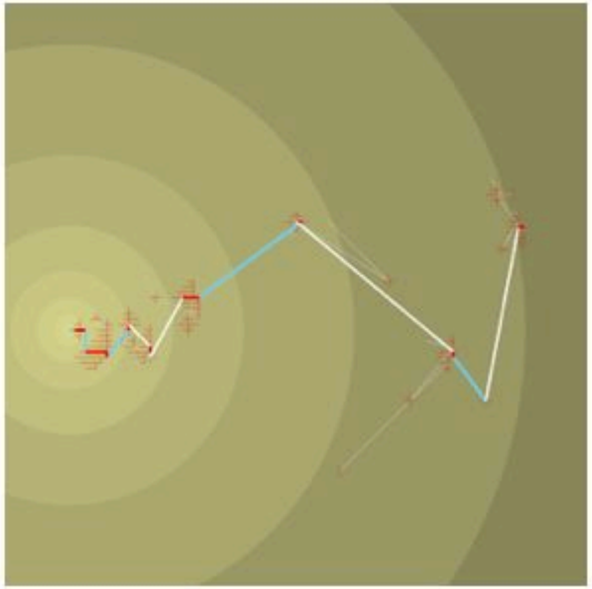


- **Algorithm [FGP'04]:**
  - Scan all the long-range contacts of the **log *n*** local neighbors close by and
  - Forward to the closest long-range contact w.r.t. the target

- Computes paths of expected length $\mathbf{\Theta(\log^{1+1/d} n)}$

- Visit $\mathbf{\Theta(\log^2 n)}$ nodes on expectation

Algorithms based on non-local exploration



- **Algorithm [LS'04, LS'05, GS'11]:**
  - Scan at most $\mathbf{log^{1+\varepsilon}}$ ***n*** long-range or local neighbors *nearby*** and
  - Forward to the closest long-range contact w.r.t. the target

- Computes paths of **optimal expected length**:
  - **Θ(log *n*)** if d ≥ 2
  - **Θ(log *n* loglog *n*)** if d = 1

- Visit **Θ(log$^{2+\varepsilon}$ *n*)** nodes on expectation

*** the meaning of "nearby" will be detailled later on*

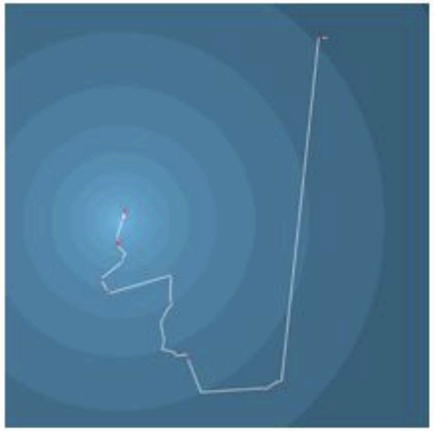
Comparing the three types of algorithms

Some differences:

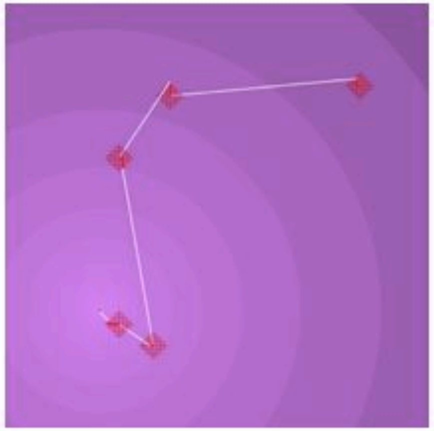
- The **length** and **structure** of the path vary
- The **rate** and **length** of used long-range links

Some similarities:

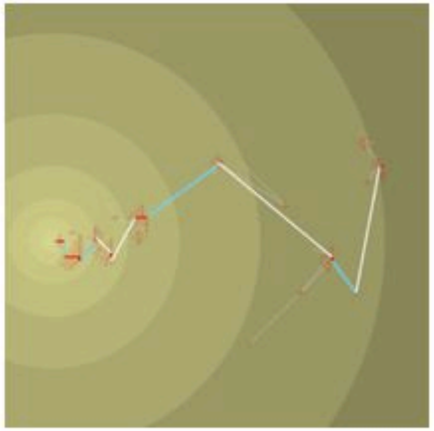
- Significant progresses towards the target are made with **some very long long-range links** which are **spaced from each other**



Greedy algorithm



Algorithms based on local exploration



Algorithms based on non-local exploration

Comparing the three types of algorithms

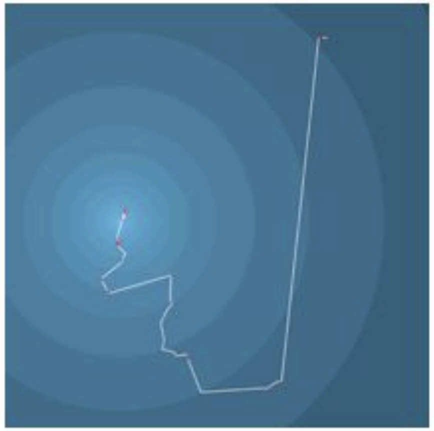
How do we analyze these three search algorithms?

Some differences:

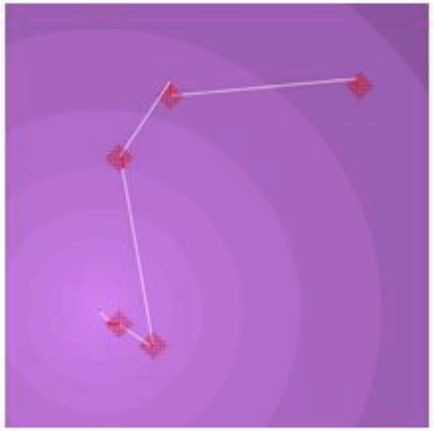
- The **length** and **structure** of the path vary
- The **rate** and **length** of used long-range links

Some similarities:

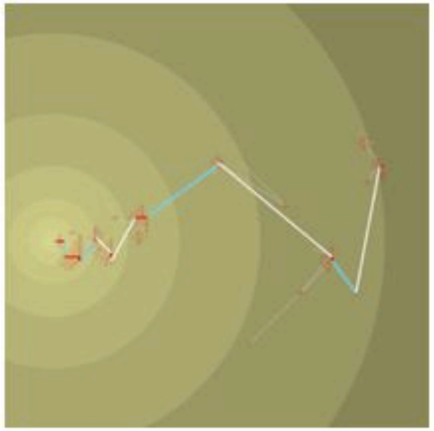
- Significant progresses towards the target are made with **some very long long-range links** which are **spaced from each other**



Greedy algorithm



Algorithms based on local exploration



Algorithms based on non-local exploration

Comparing the three types of algorithms

How do we analyze these three search algorithms?

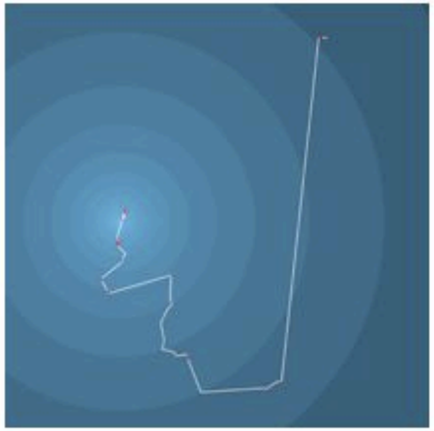
Some differences:

- The **length** and **structure** of the path vary
- The **rate** and **length** of used long-range links

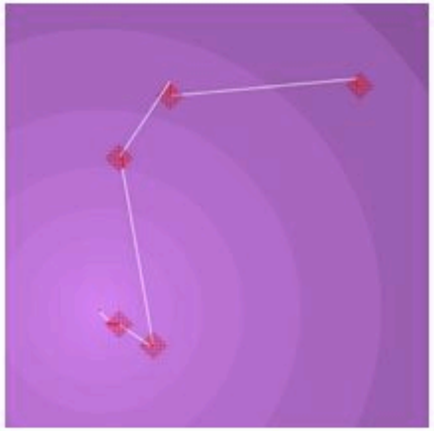
Indeed, TRUE for every efficient decentralized search algorithm [GS'11]

Some similarities:

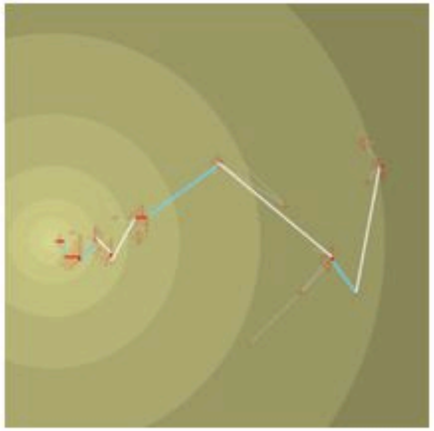
- Significant progresses towards the target are made with **some very long long-range links** which are **spaced from each other**



Greedy algorithm



Algorithms based on local exploration



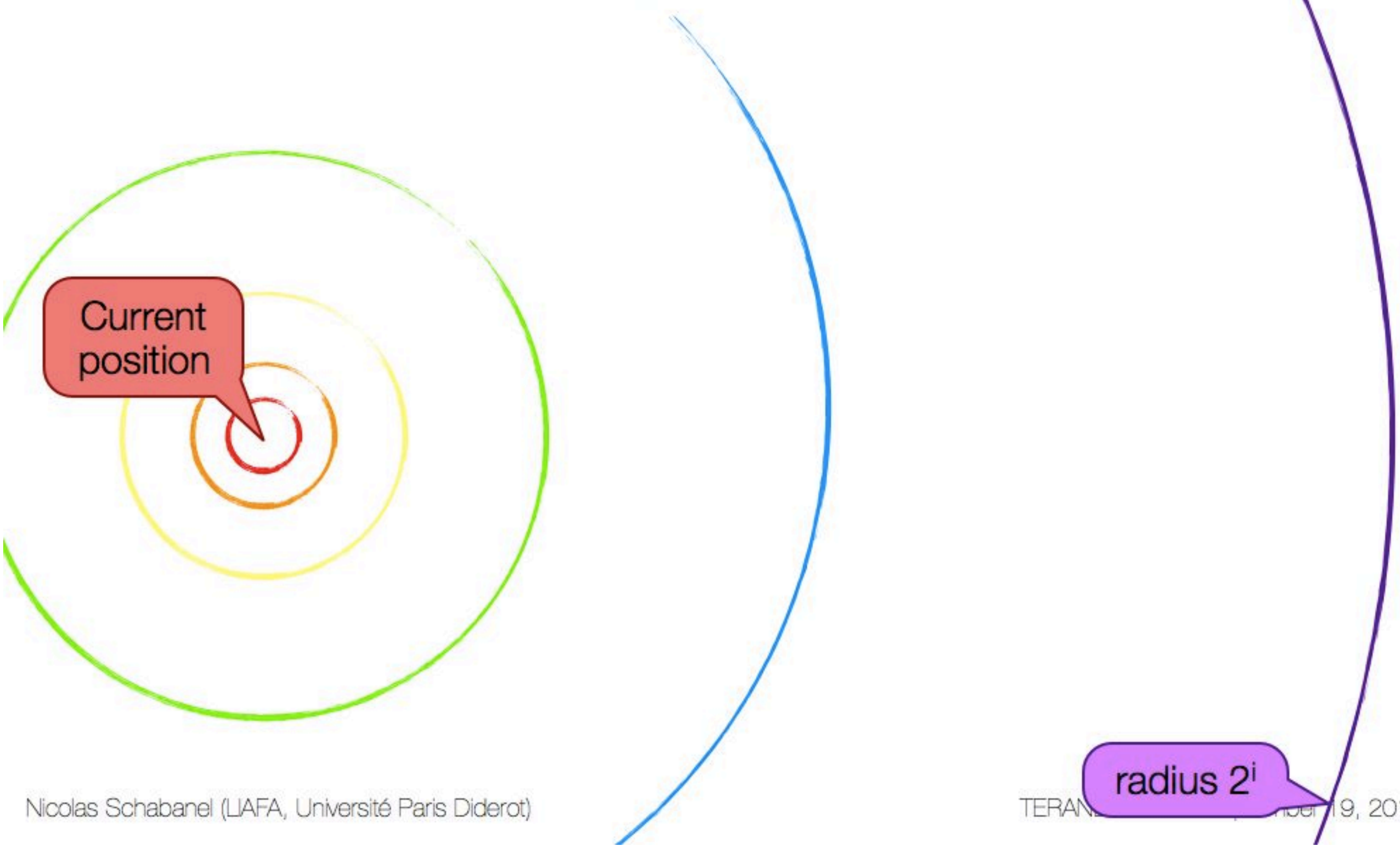
Algorithms based on non-local exploration

Using **simple geometric arguments**
to analyze of these algorithms

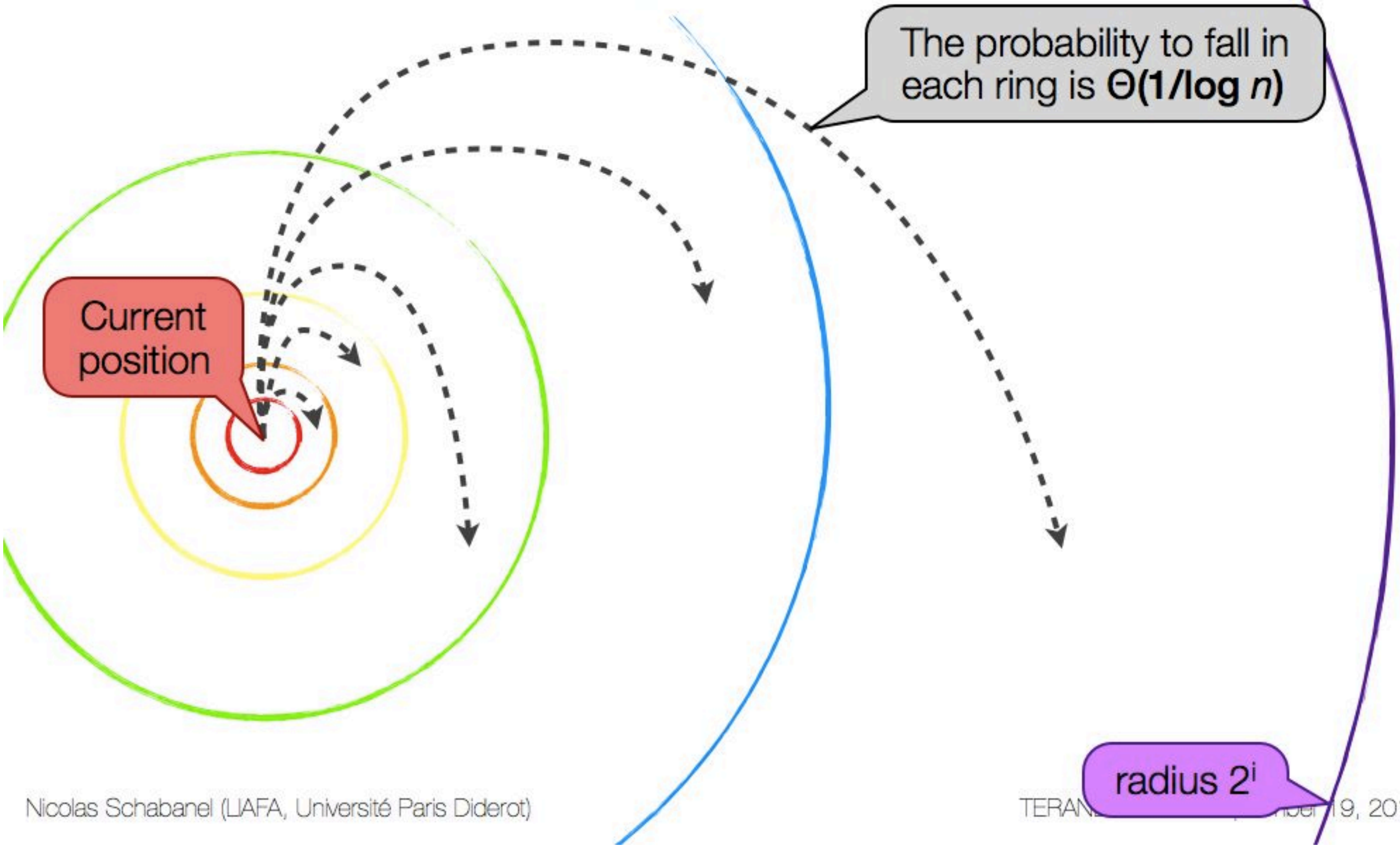
The **keys** to Kleinberg's Greedy algorithm's **$\Theta(\log^2 n)$ expected path length analysis**

Current position

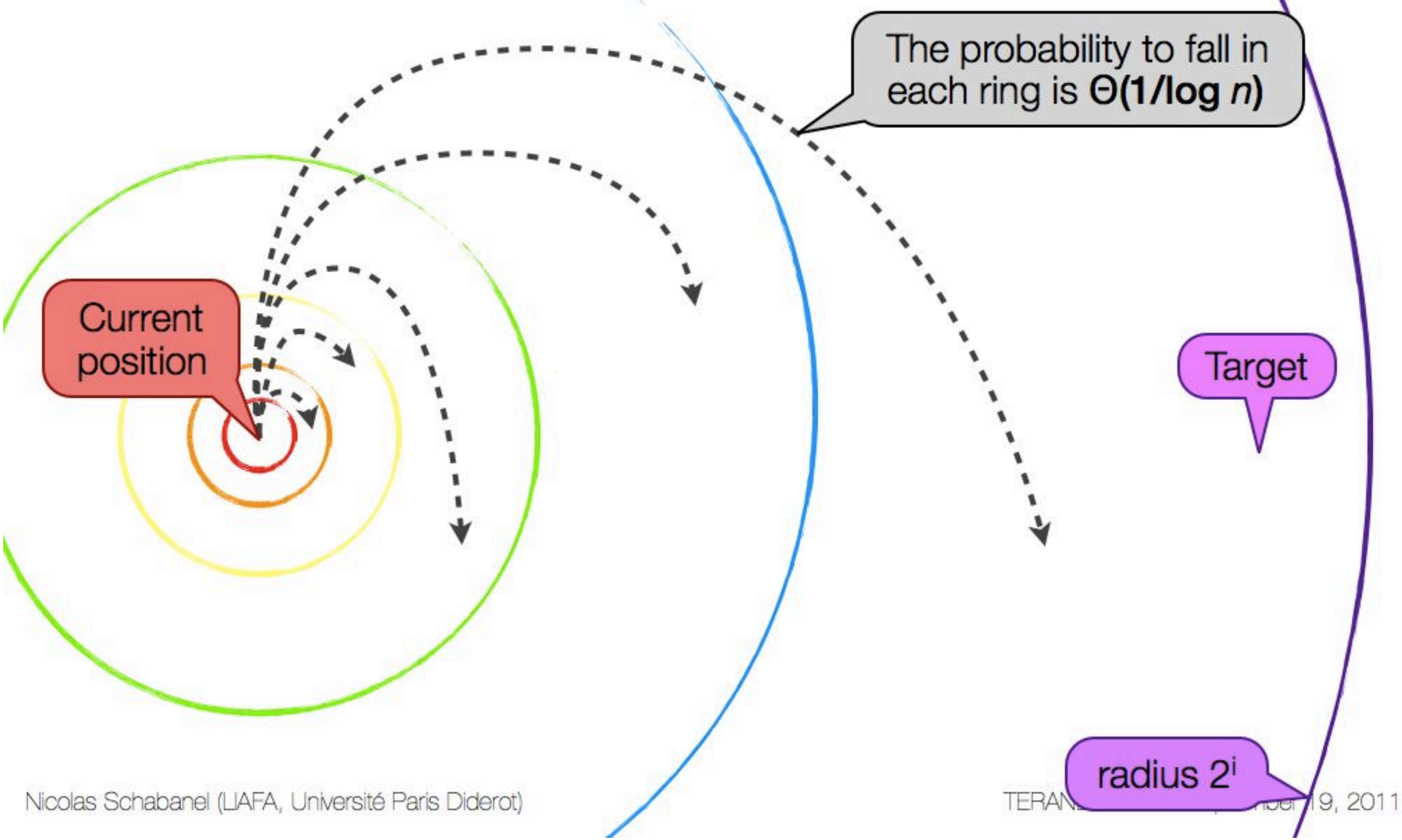
The **keys** to Kleinberg's Greedy algorithm's **$\Theta(\log^2 n)$** expected path length analysis



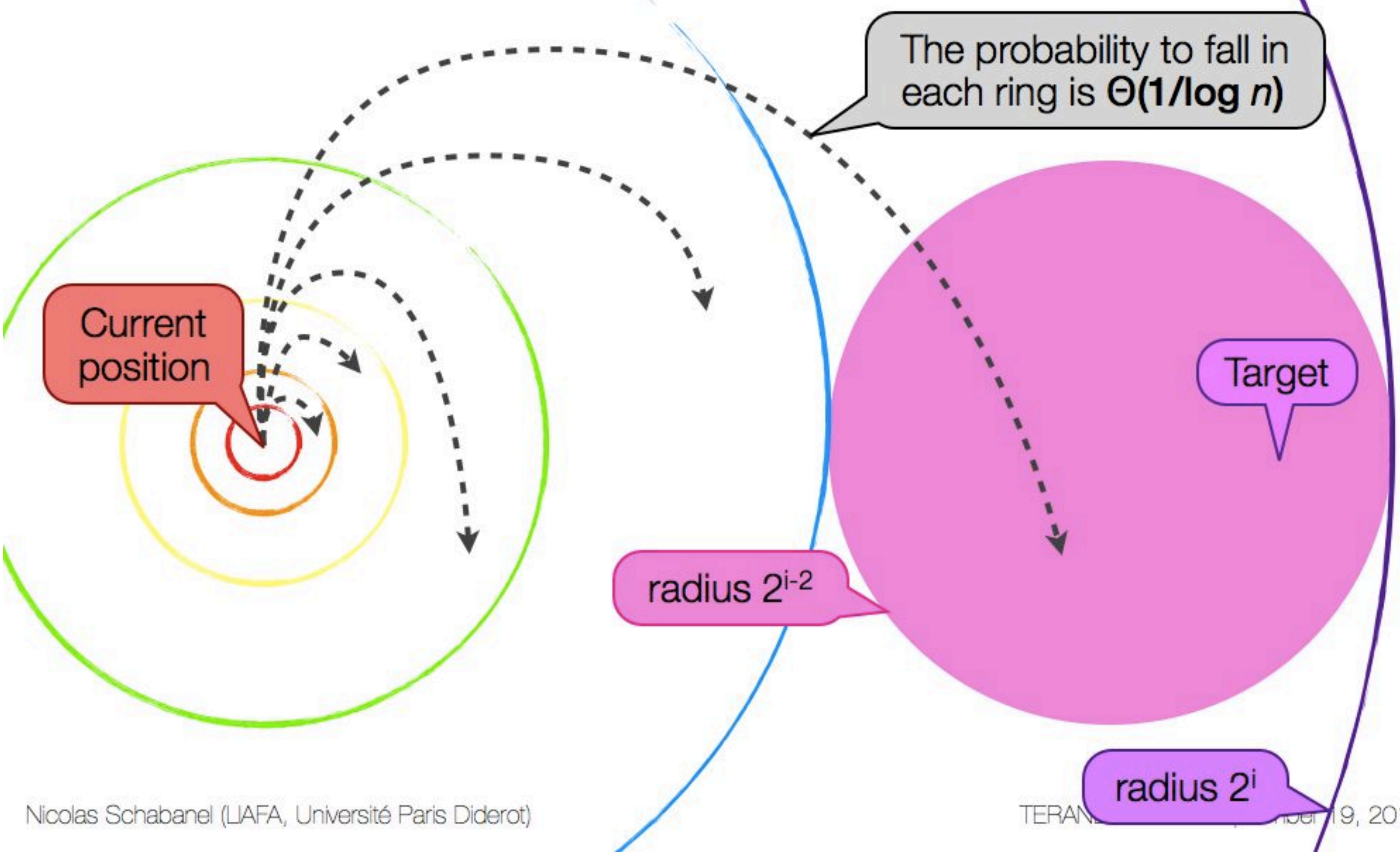
The **keys** to Kleinberg's Greedy algorithm's **$\Theta(\log^2 n)$** expected path length analysis



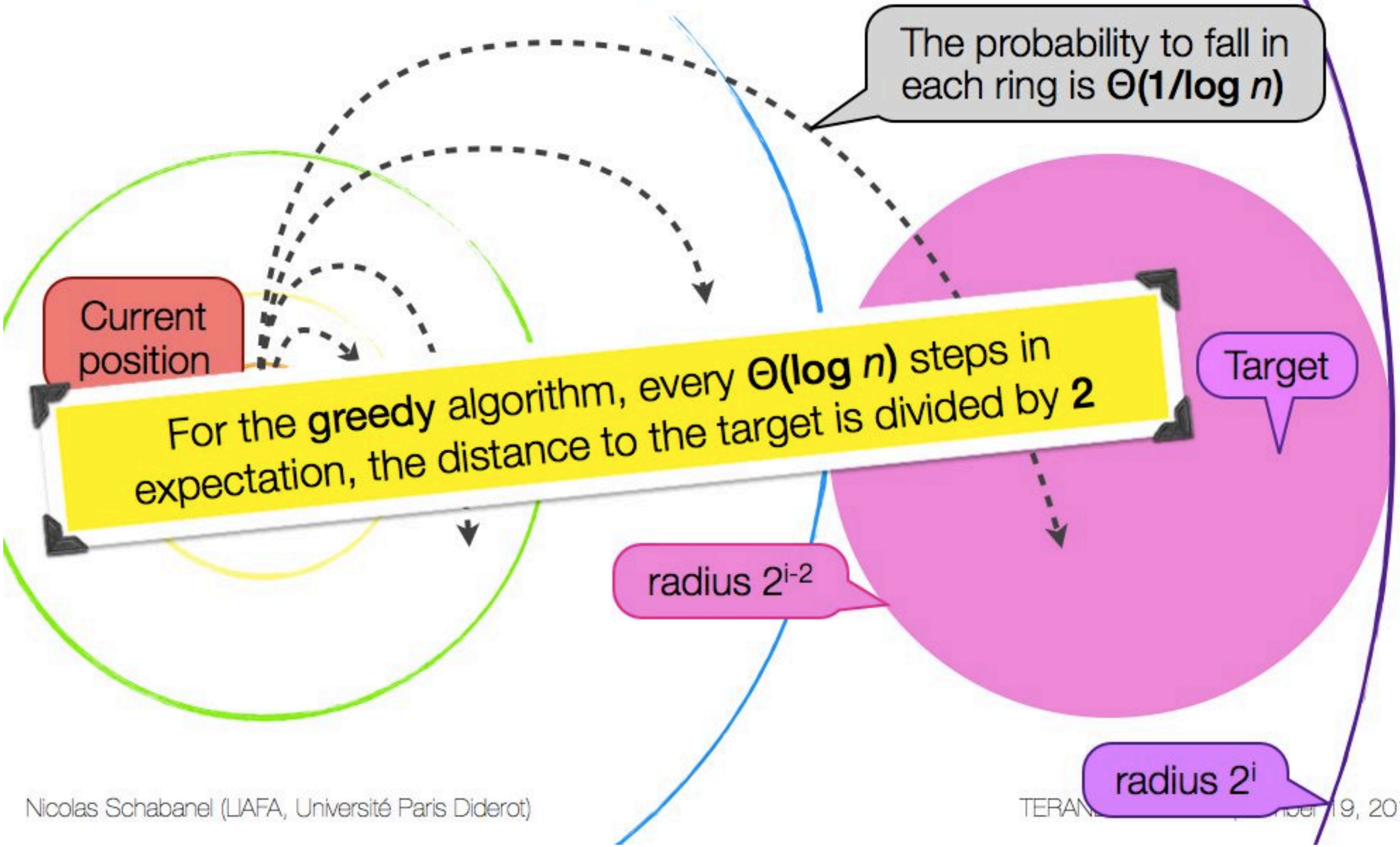
The **keys** to Kleinberg's Greedy algorithm's **$\Theta(\log^2 n)$** expected path length analysis



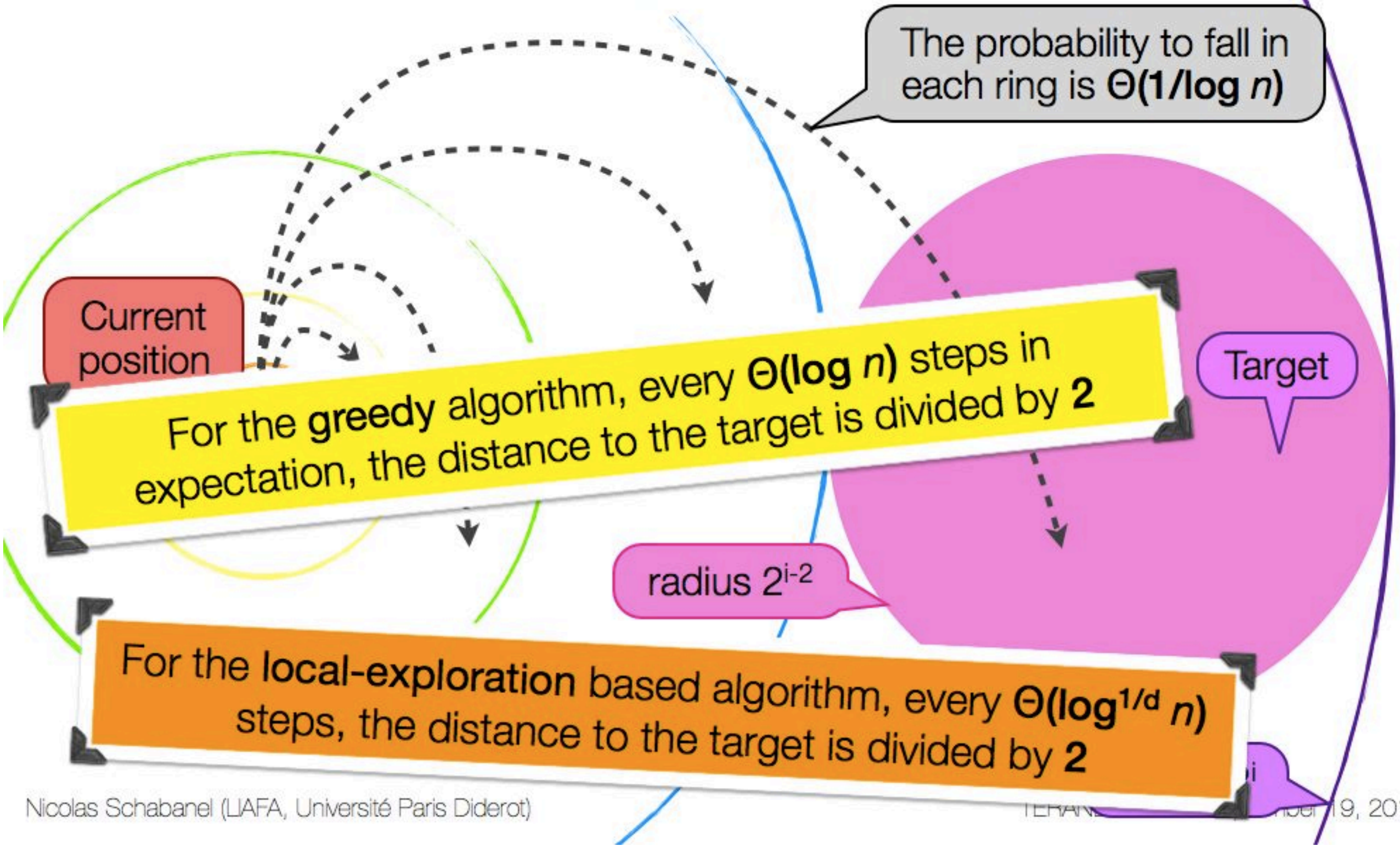
The **keys** to Kleinberg's Greedy algorithm's **$\Theta(\log^2 n)$** expected path length analysis



The **keys** to Kleinberg's Greedy algorithm's $\Theta(\log^2 n)$ expected path length analysis



The **keys** to Kleinberg's Greedy algorithm's $\Theta(\log^2 n)$ expected path length analysis



For the **greedy** algorithm, every **$\Theta(\log n)$** steps in expectation, the distance to the target is divided by **2**

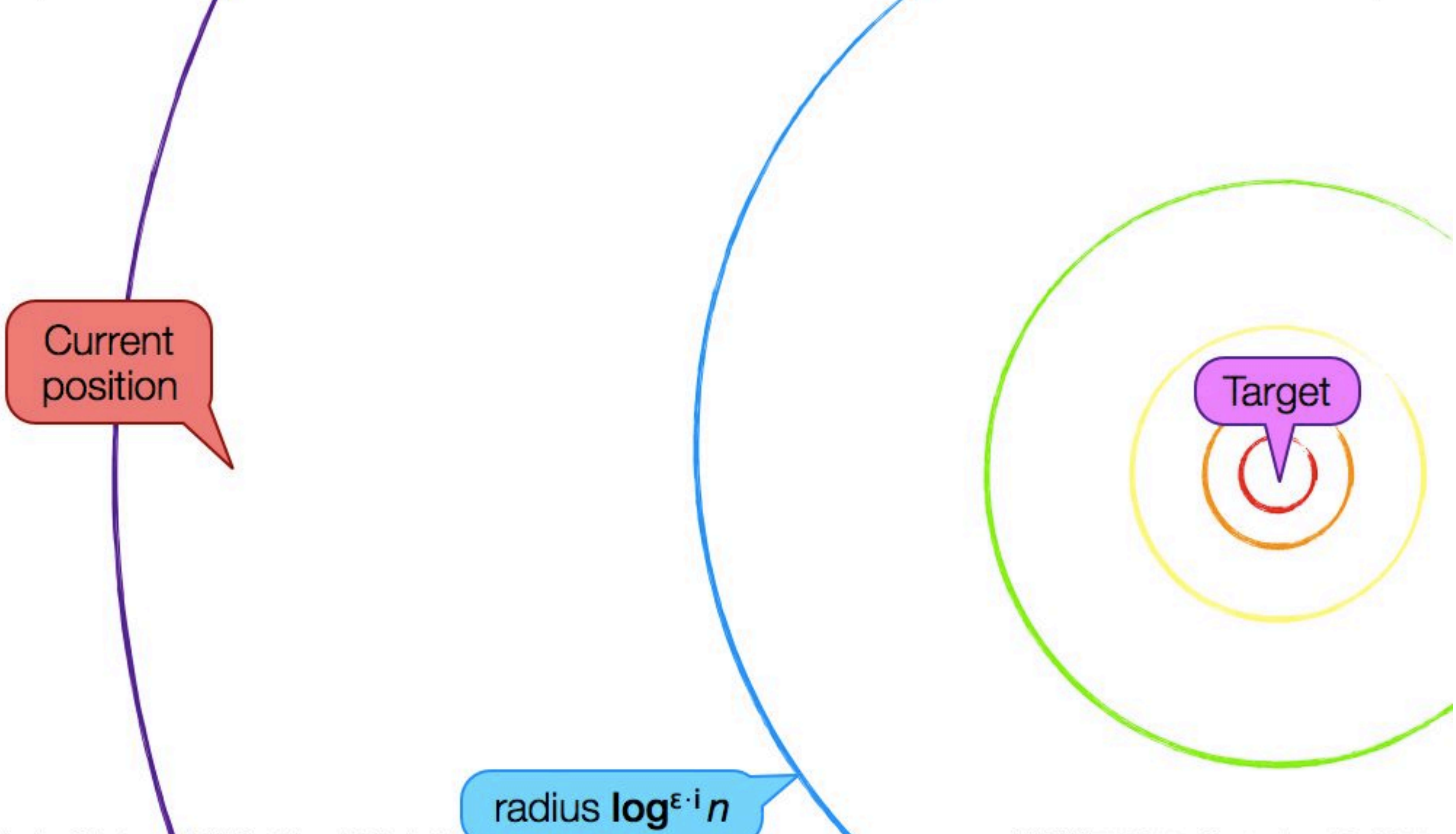
For the **local-exploration** based algorithm, every **$\Theta(\log^{1/d} n)$** steps, the distance to the target is divided by **2**

Analyzing the non-local exploration-based algorithm [LS'04, GS'11]

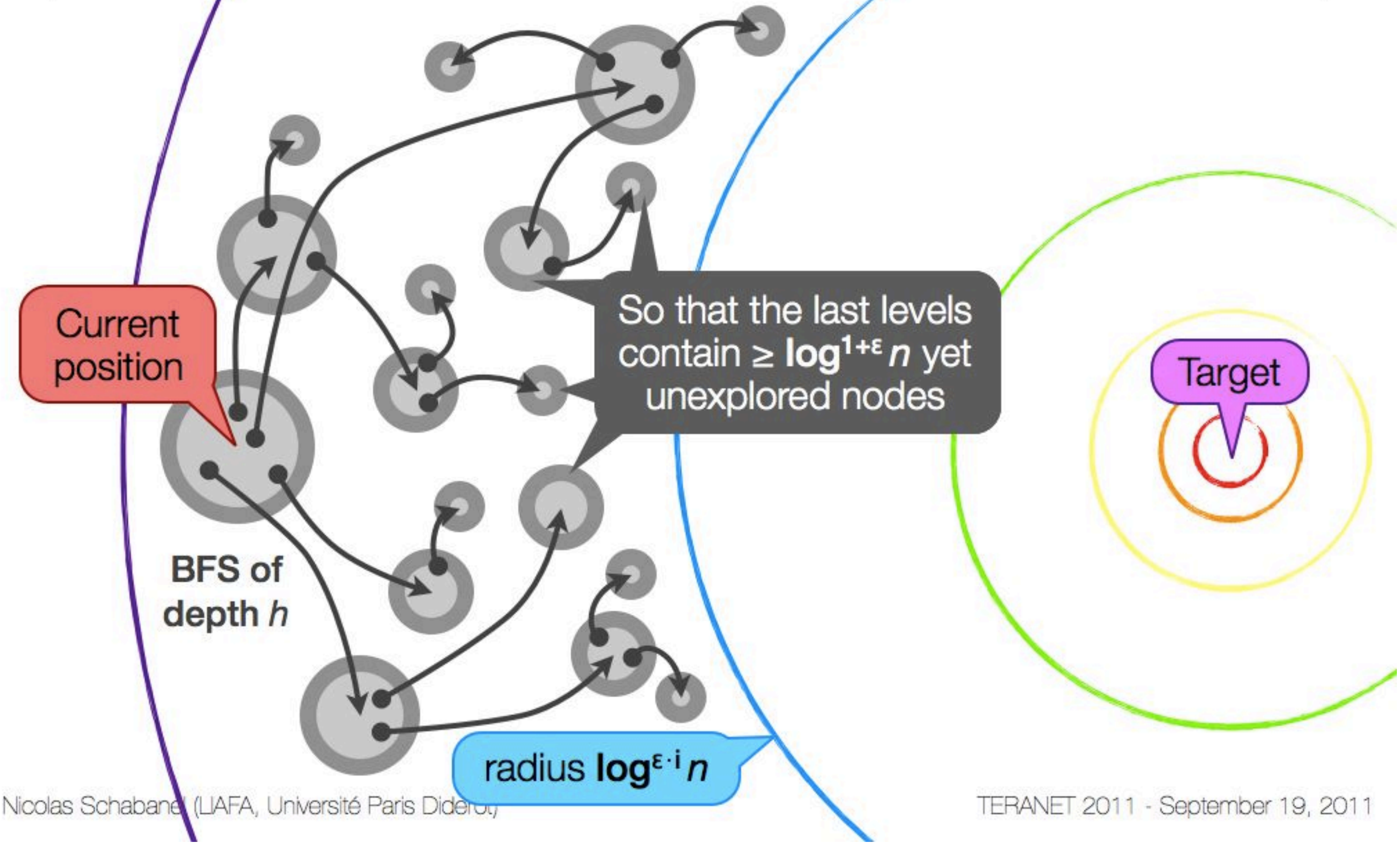
Current position

Target

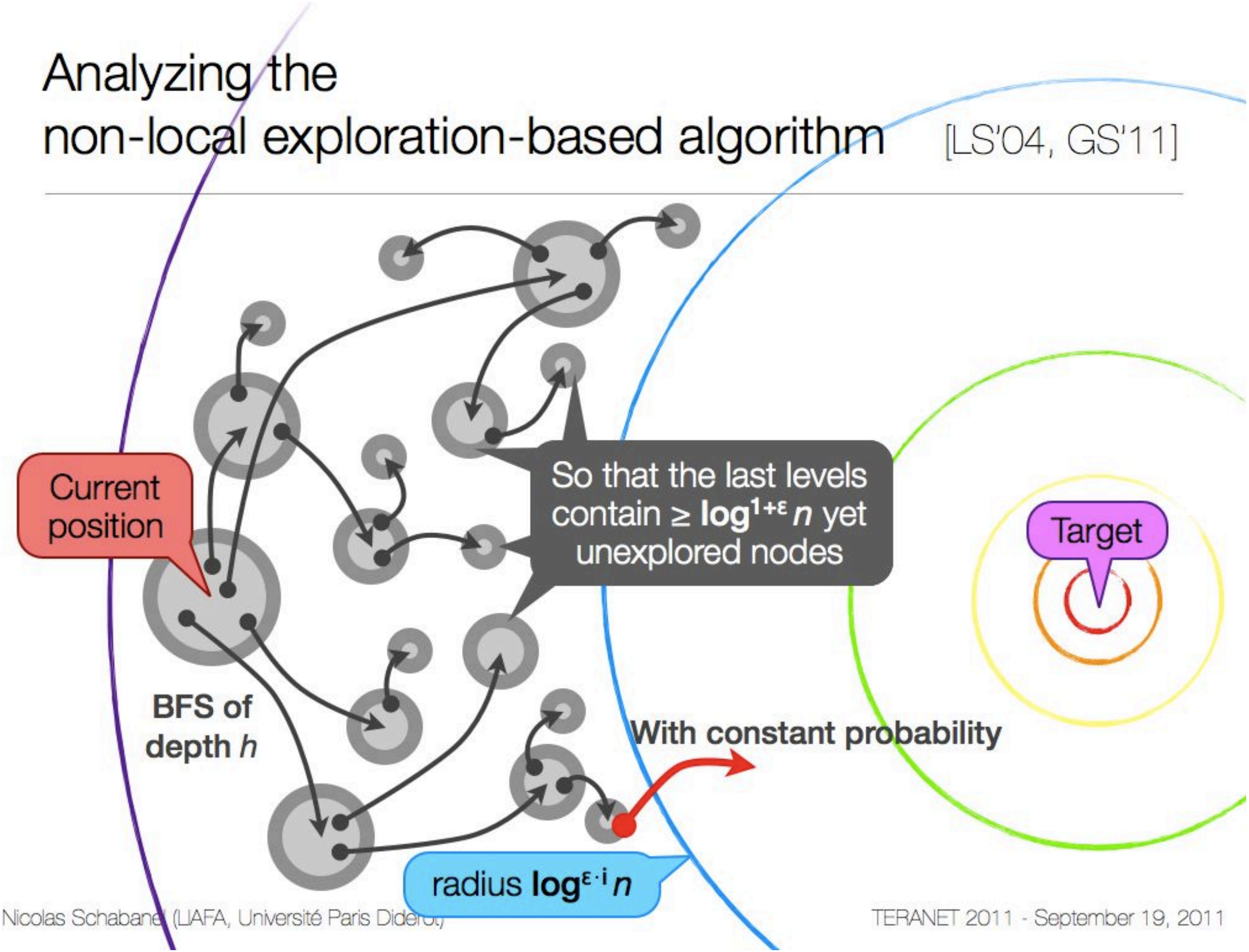
Analyzing the non-local exploration-based algorithm [LS'04, GS'11]



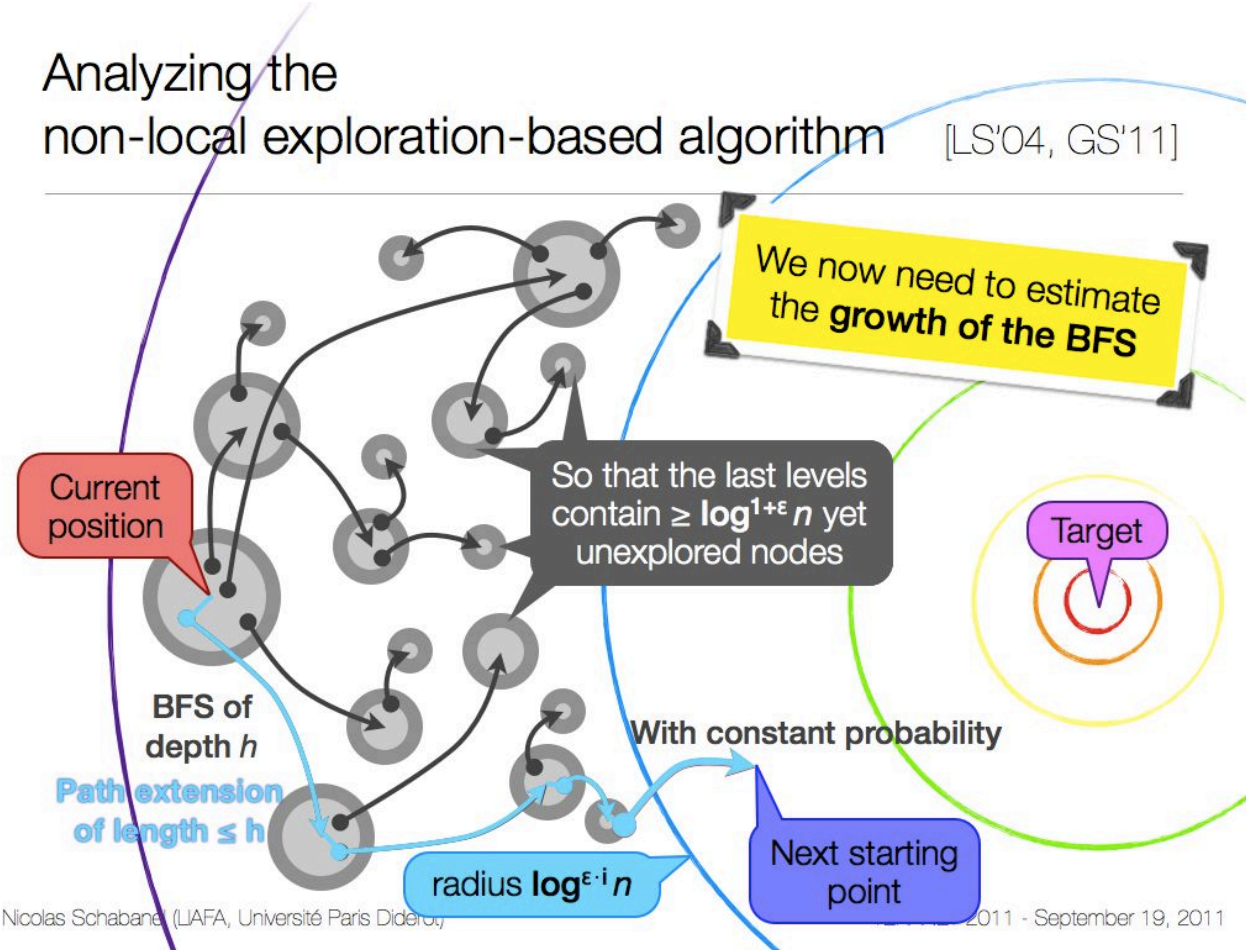
Analyzing the non-local exploration-based algorithm [LS'04, GS'11]



Analyzing the non-local exploration-based algorithm [LS'04, GS'11]



Analyzing the non-local exploration-based algorithm [LS'04, GS'11]

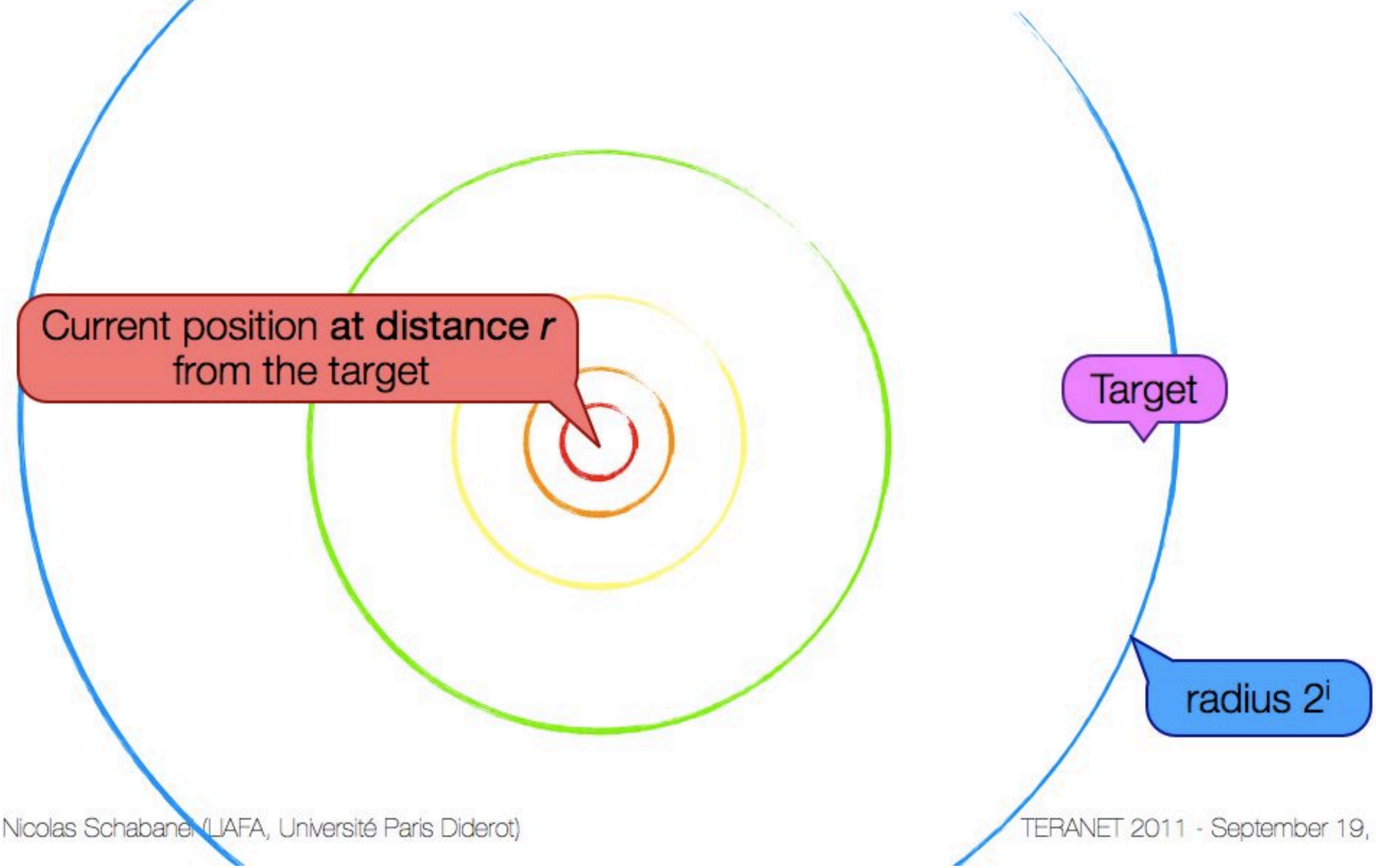


Estimation of the growth of the BFS:
Probability for a long-range link to be in it

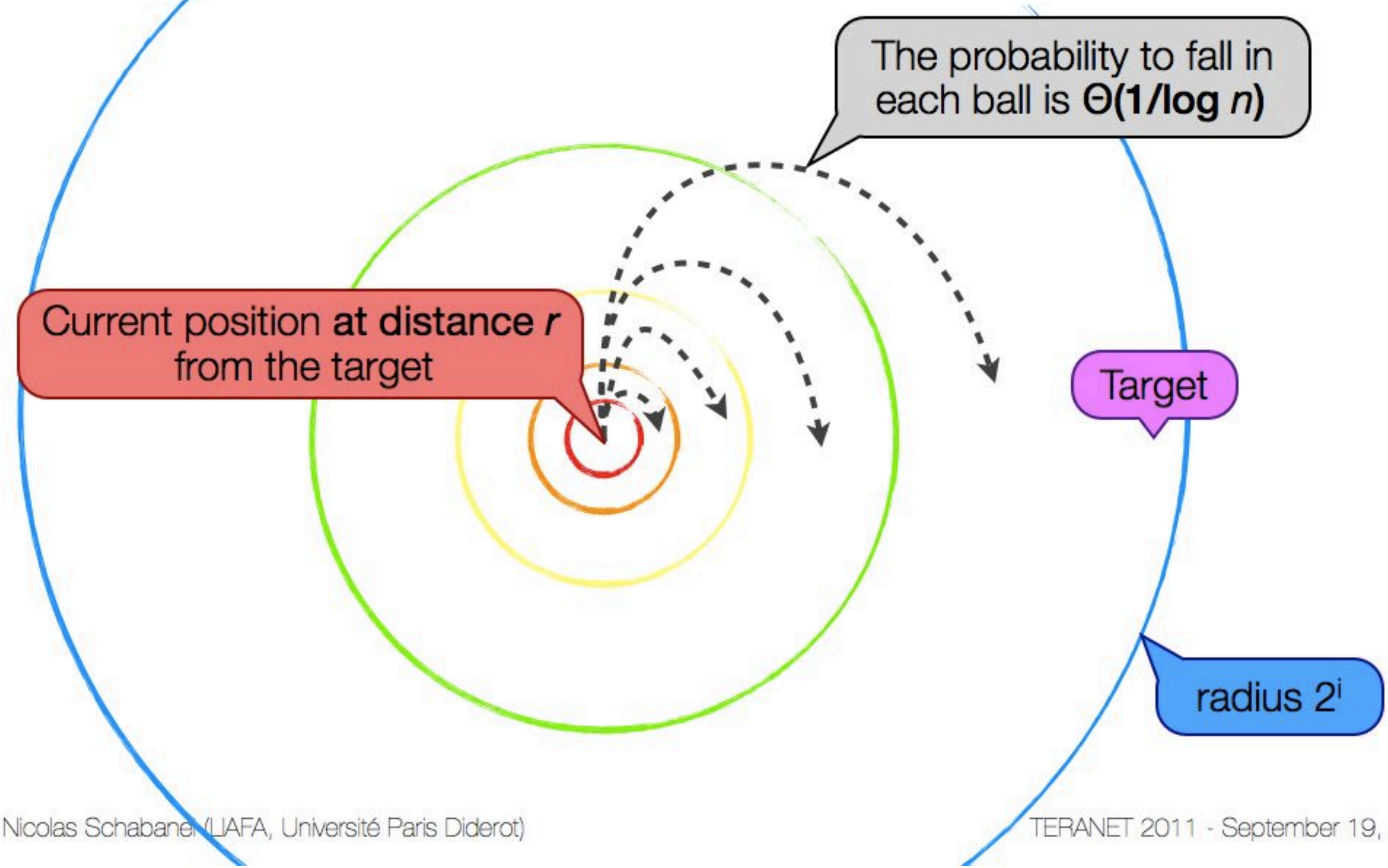
Current position **at distance *r*** from the target

Target

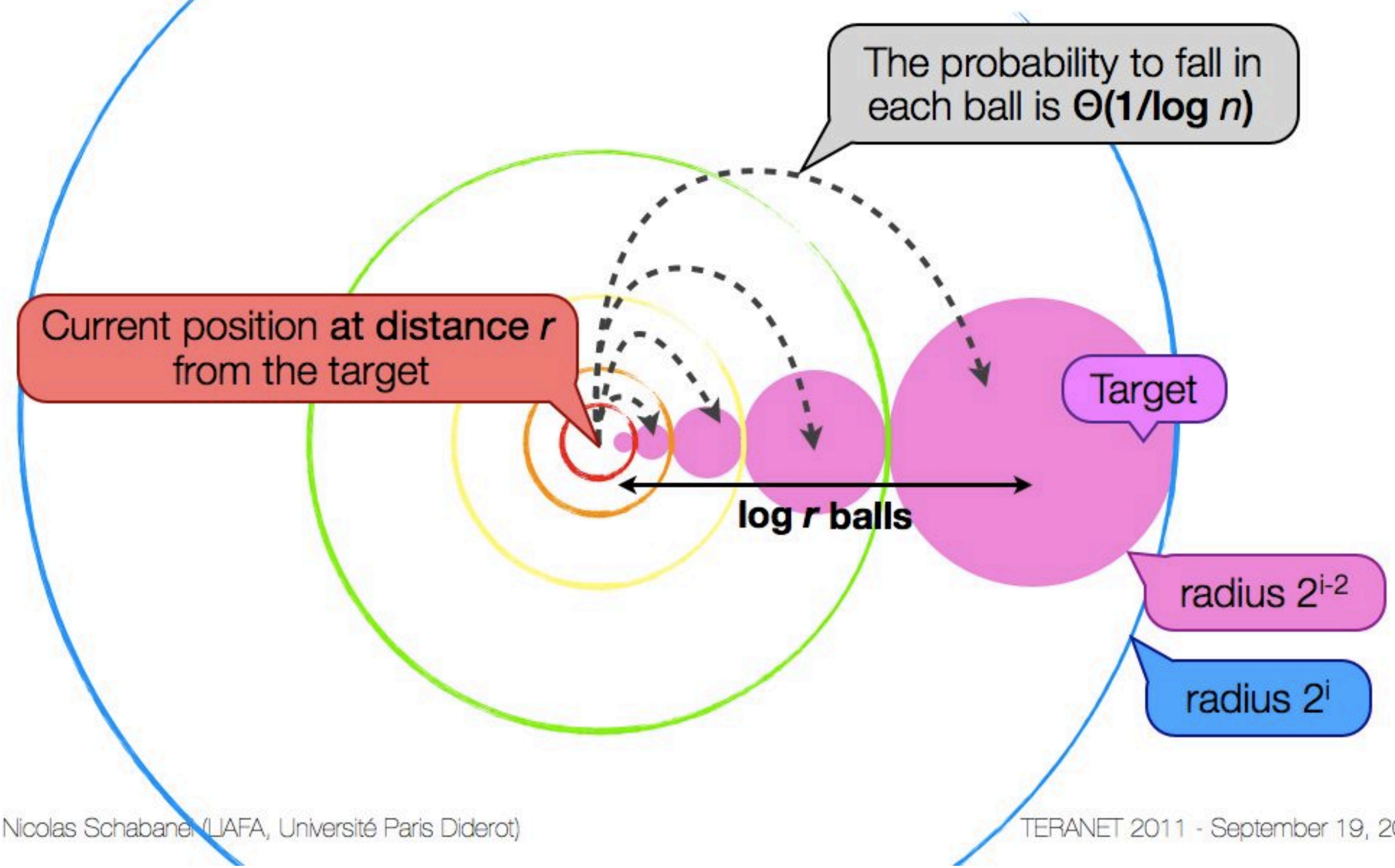
Estimation of the growth of the BFS:
Probability for a long-range link to be in it



Estimation of the growth of the BFS:
Probability for a long-range link to be in it

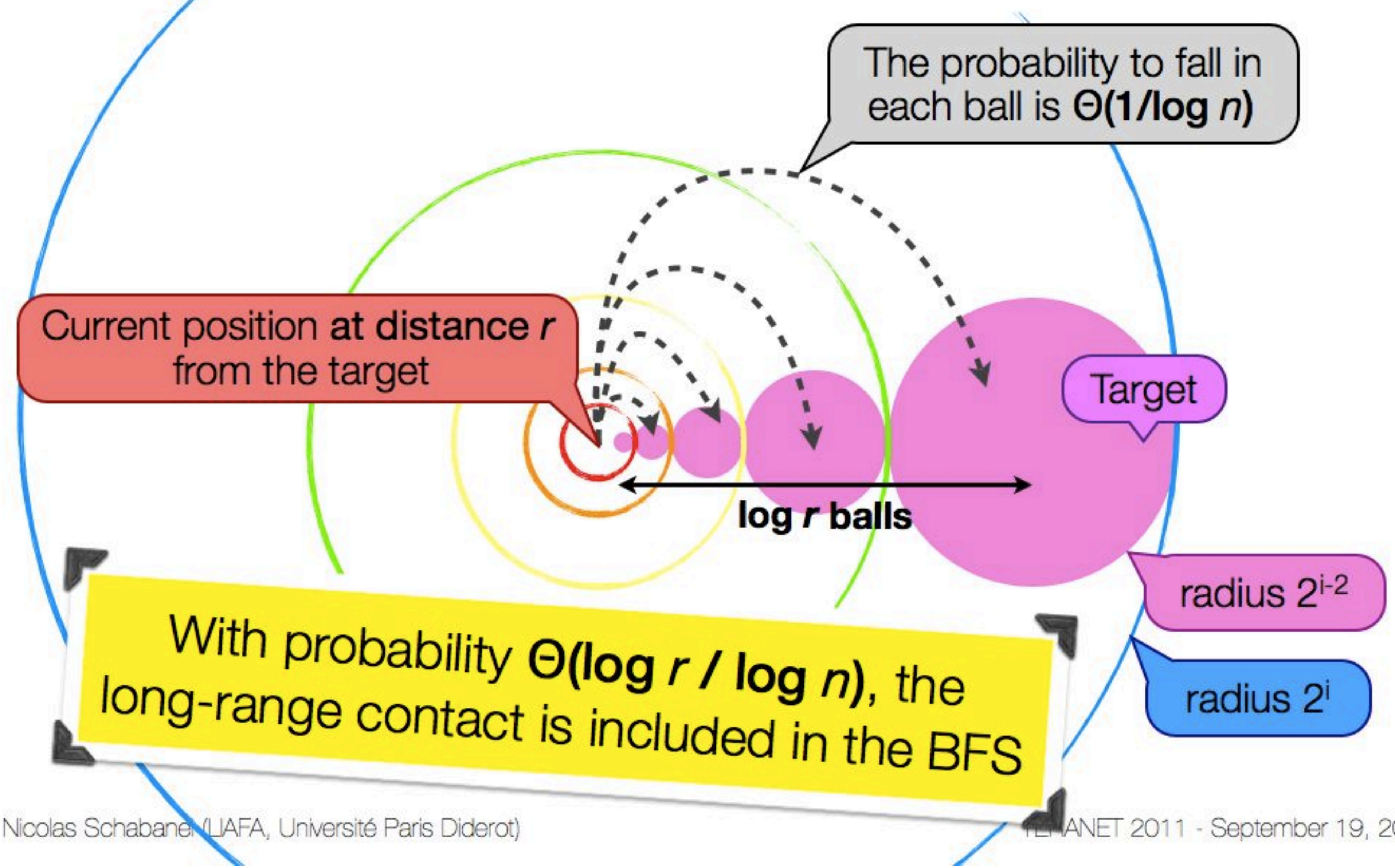


Estimation of the growth of the BFS:
Probability for a long-range link to be in it



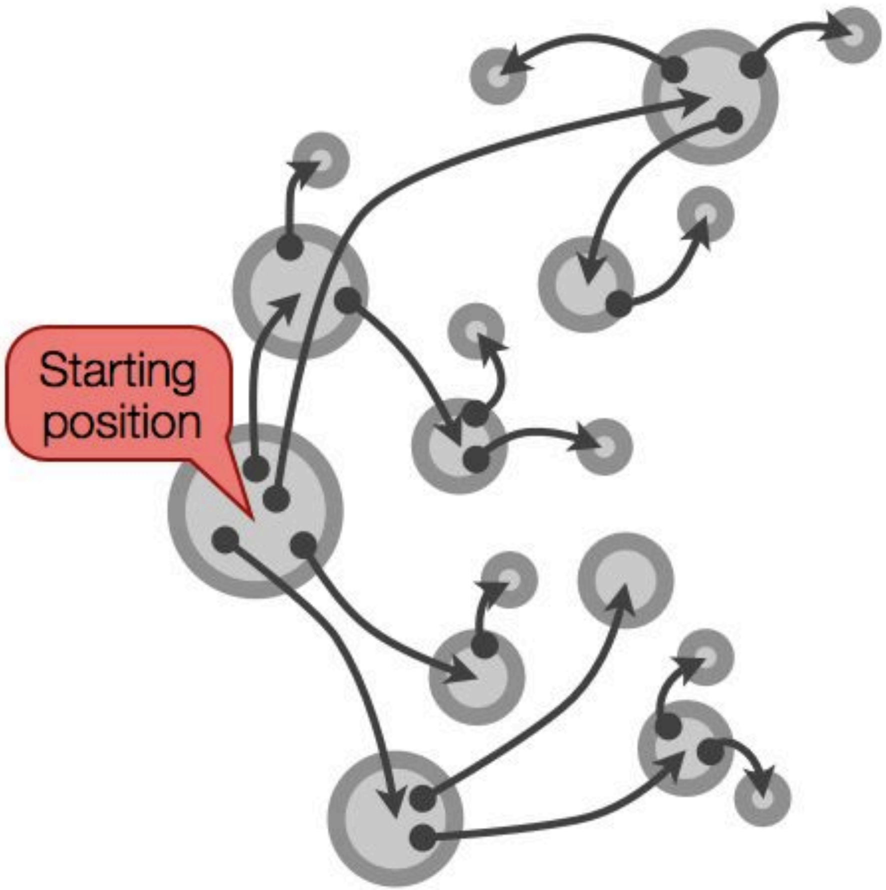
Estimation of the growth of the BFS:

Probability for a long-range link to be in it



With probability **Θ(log *r* / log *n*)**, the long-range contact is included in the BFS

Estimation of the growth of the BFS



- **Two components:**
  - Number of balls
  - Growth of the ball borders

- **d = 1:** no growth of ball borders

  $\Rightarrow$ Size of level *i* in BFS is $\left(1+\frac{\log r}{\log n}\right)^i$

  $\Rightarrow$

$$h_{d=1}(r) = \frac{\log(\log^{1+\varepsilon} n)}{\log(1+\frac{\log r}{\log n})} \approx \frac{\log n \log\log n}{\log r}$$

- **d ≥ 2:** Ball border increases at least by +1

  $\Rightarrow$ Size of level *i* in BFS is $\left(1+\sqrt{\frac{\log r}{\log n}}\right)^i$

  $\Rightarrow$

$$h_{d\geqslant 2}(r) = \frac{\log(\log^{1+\varepsilon} n)}{\log(1+\sqrt{\frac{\log r}{\log n}})} \approx \frac{\sqrt{\log n}\cdot\log\log n}{\sqrt{\log r}}$$

Resulting expected path length

$$r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log \log n}$$

Resulting expected path length

- $r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log \log n}$

Resulting expected path length

- $r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log\log n}$

- **For d = 1:** $h_{d=1}(r_i) \approx \dfrac{\log n \log\log n}{i\epsilon \log\log n} = \dfrac{1}{i\epsilon} \log n$

$$\mathbb{E}[\ell(\mathcal{P}_{d=1})] \lesssim \sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{i\epsilon} \log n \underbrace{+ \log n \log\log n}_{\text{Greedy final steps}} = O(\log n \log\log n)$$

Resulting expected path length

- $r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log\log n}$
- **For d = 1:**

$$h_{d=1}(r_i) \approx \frac{\log n \log\log n}{i\epsilon \log\log n} = \frac{1}{i\epsilon}\log n$$

$$\mathbb{E}[\ell(\mathcal{P}_{d=1})] \lesssim \sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{i\epsilon}\log n \underbrace{+ \log n \log\log n}_{\text{Greedy final steps}} = O(\log n \log\log n)$$

- **For d = 2:**

$$h_{d\geqslant 2}(r_i) \approx \frac{\sqrt{\log n}\log\log n}{\sqrt{i\epsilon \log\log n}} = \frac{\sqrt{\log n \log\log n}}{\sqrt{i\epsilon}}$$

$$\mathbb{E}[\ell(\mathcal{P}_{d\geqslant 2})] \lesssim \sqrt{\log n \log\log n} \underbrace{\sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{\sqrt{i\epsilon}}}_{\frac{1}{\epsilon}\sqrt{\frac{\log n}{\log\log n}}} \underbrace{+ \sqrt{\log n} \cdot \log 2^{\sqrt{\log n}}}_{\text{Greedy local-search-based final steps}} = O(\log n)$$

Resulting expected path length

- $r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log\log n}$

- **For d = 1:**

$$h_{d=1}(r_i) \approx \frac{\log n \log\log n}{i\epsilon \log\log n} = \frac{1}{i\epsilon} \log n$$

$$\mathbb{E}[\ell(\mathcal{P}_{d=1})] \lesssim \sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{i\epsilon} \log n \underbrace{+ \log n \log\log n}_{\text{Greedy final steps}} = O(\log n \log\log n)$$

- **For d = 2:**

$$h_{d\geqslant 2}(r_i) \approx \frac{\sqrt{\log n} \log\log n}{\sqrt{i\epsilon \log\log n}} = \frac{\sqrt{\log n \log\log n}}{\sqrt{i\epsilon}}$$

$$\mathbb{E}[\ell(\mathcal{P}_{d\geqslant 2})] \lesssim \sqrt{\log n \log\log n} \underbrace{\sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{\sqrt{i\epsilon}}}_{\frac{1}{\epsilon}\sqrt{\frac{\log n}{\log\log n}}} \underbrace{+ \sqrt{\log n} \cdot \log 2^{\sqrt{\log n}}}_{\text{Greedy local-search-based final steps}} = \boxed{O(\log n)}$$

= O(diameter) thus optimal

Resulting expected path length

- $r_i = \log^{i\epsilon} n, \quad \text{for } i = 1 \ldots \frac{\log n}{\epsilon \log\log n}$
- **For d = 1:**

$$h_{d=1}(r_i) \approx \frac{\log n \log\log n}{i\epsilon \log\log n} = \frac{1}{i\epsilon}\log n$$

>> diameter! Is there a matching lower bound?

$$\mathbb{E}[\ell(\mathcal{P}_{d=1})] \lesssim \sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{i\epsilon}\log n + \underbrace{\log n \log\log n}_{\text{Greedy final steps}} = O(\log n \log\log n)$$

- **For d = 2:**

$$h_{d\geqslant 2}(r_i) \approx \frac{\sqrt{\log n}\log\log n}{\sqrt{i\epsilon \log\log n}} = \frac{\sqrt{\log n \log\log n}}{\sqrt{i\epsilon}}$$

$$\mathbb{E}[\ell(\mathcal{P}_{d\geqslant 2})] \lesssim \sqrt{\log n \log\log n} \underbrace{\sum_{i \leqslant \frac{\log n}{\epsilon \log\log n}} \frac{1}{\sqrt{i\epsilon}}}_{\frac{1}{\epsilon}\sqrt{\frac{\log n}{\log\log n}}} \underbrace{+ \sqrt{\log n} \cdot \log 2^{\sqrt{\log n}}}_{\text{Greedy local-search-based final steps}} = O(\log n)$$

= O(diameter) thus optimal

Analysis of the non-local exploration based search algorithm

Our decentralized search algorithm:

- **As long as $r > R$** [where $R = \log^2 n$ (for $d = 1$) or $R = 2^{\sqrt{\log n}}$ (for $d \geq 2$)]
  - Do a **BFS** upto *depth* $h(r)$
    but stop before if $\log^{1+\varepsilon} n$ nodes are encountered on a BFS-level
  - Go to the **contact** ***grid-closest*** to the target among the current nodes
- **As soon as $r \leq R$**, then use Greedy local search

This decentralized algorithm **visits $O(\log^{2+O(\varepsilon)} n)$ nodes** and computes **optimal expected length paths $O(\log n)$ for $d \geq 2$**

What about $d = 1$?

Proving the **observed similarities** between all efficient decentralized search algorithms:
Our lower bound for $d = 1$ [GS'11]

Looking for a matching lower bound for d = 1

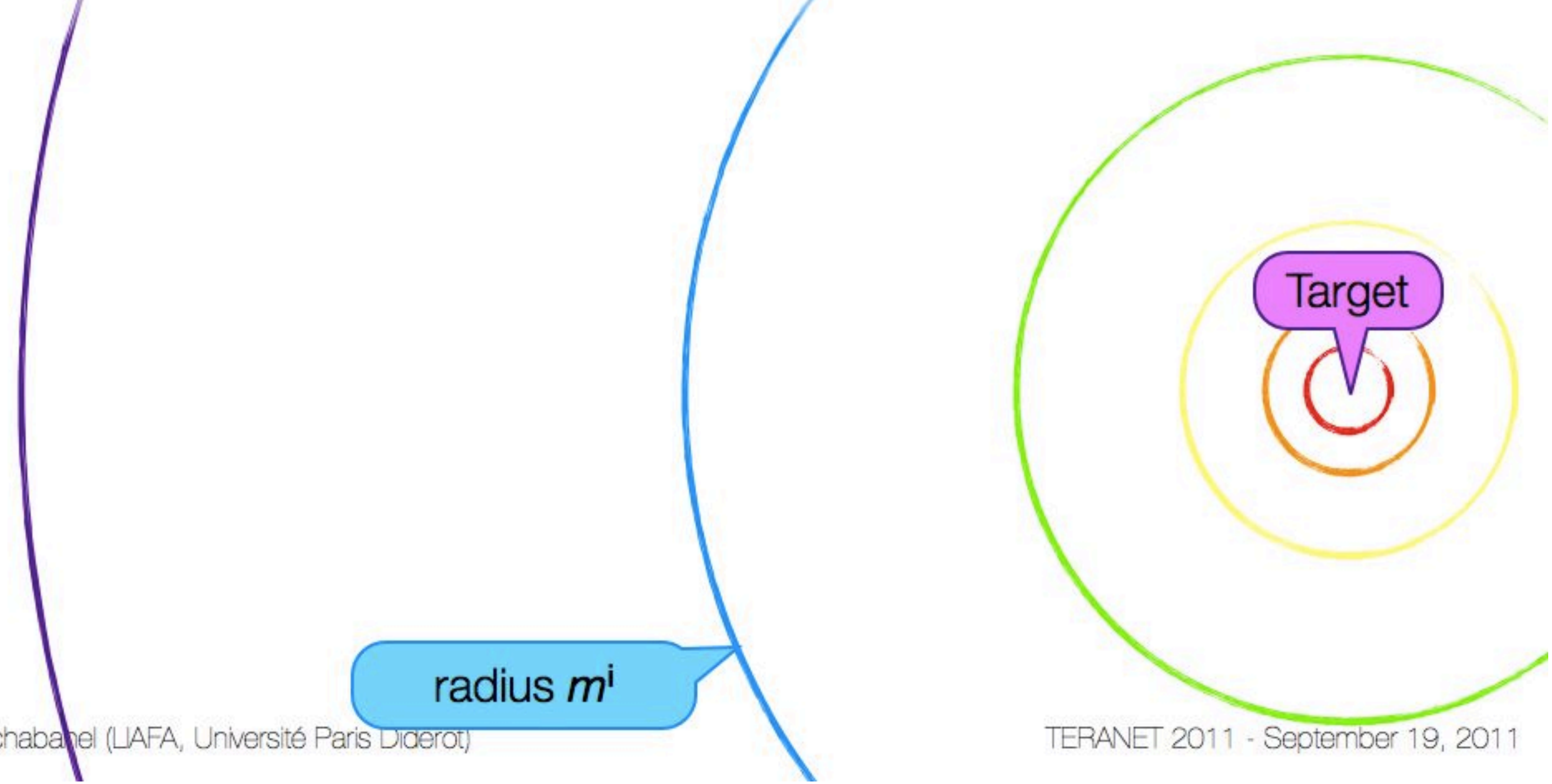
- There is a O(√n)-time decentralized algorithm [Martel, Nguyen, 2004]

➡**What happens if we bound the time to be ≤ m = O(log$^{\gamma}$ n) for some γ > 0 ?**

Looking for a matching lower bound for d = 1

- There is a O(√n)-time decentralized algorithm [Martel, Nguyen, 2004]

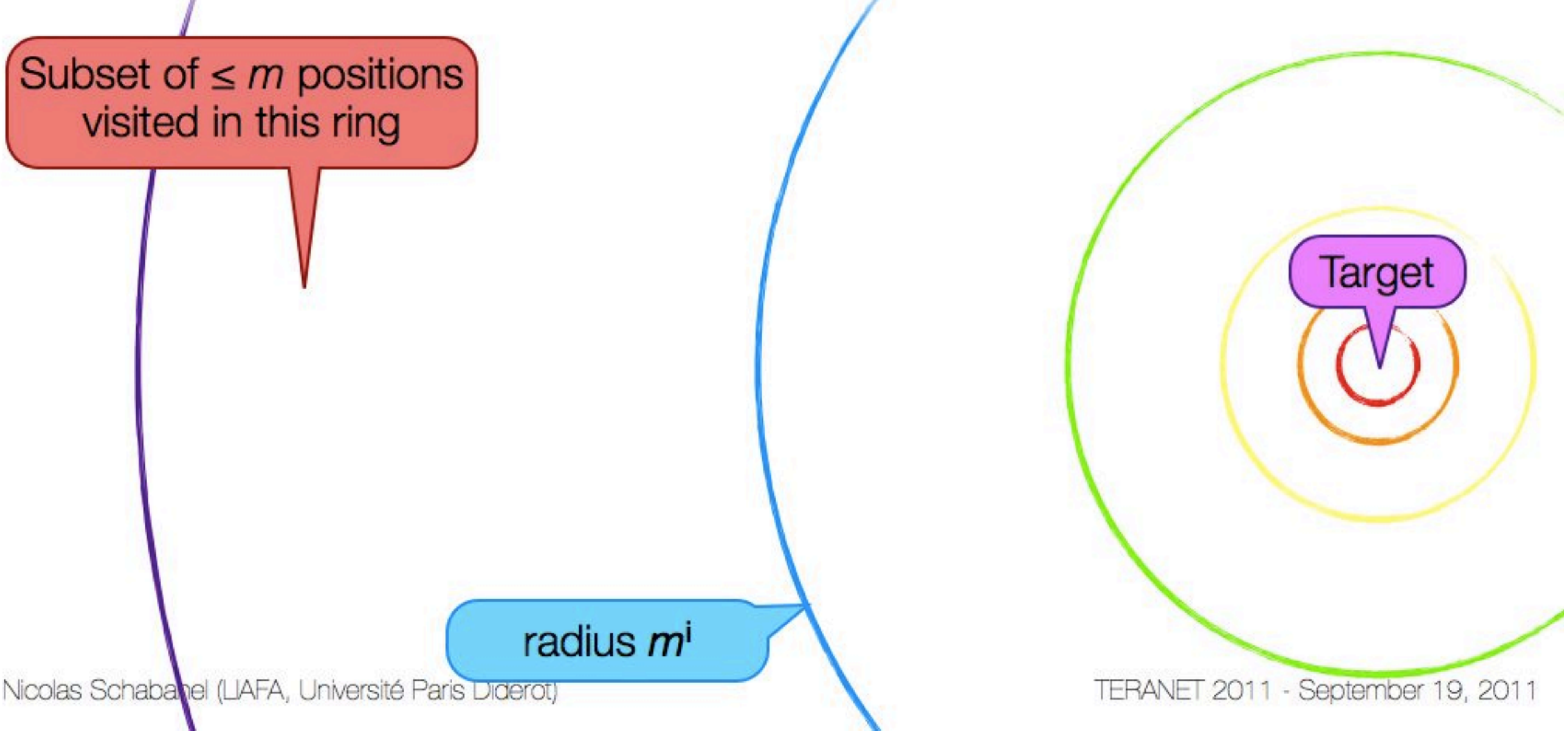
➡**What happens if we bound the time to be ≤ m = O(log$^\gamma$ n) for some γ > 0 ?**



Looking for a matching lower bound for d = 1

- There is a O(√n)-time decentralized algorithm [Martel, Nguyen, 2004]

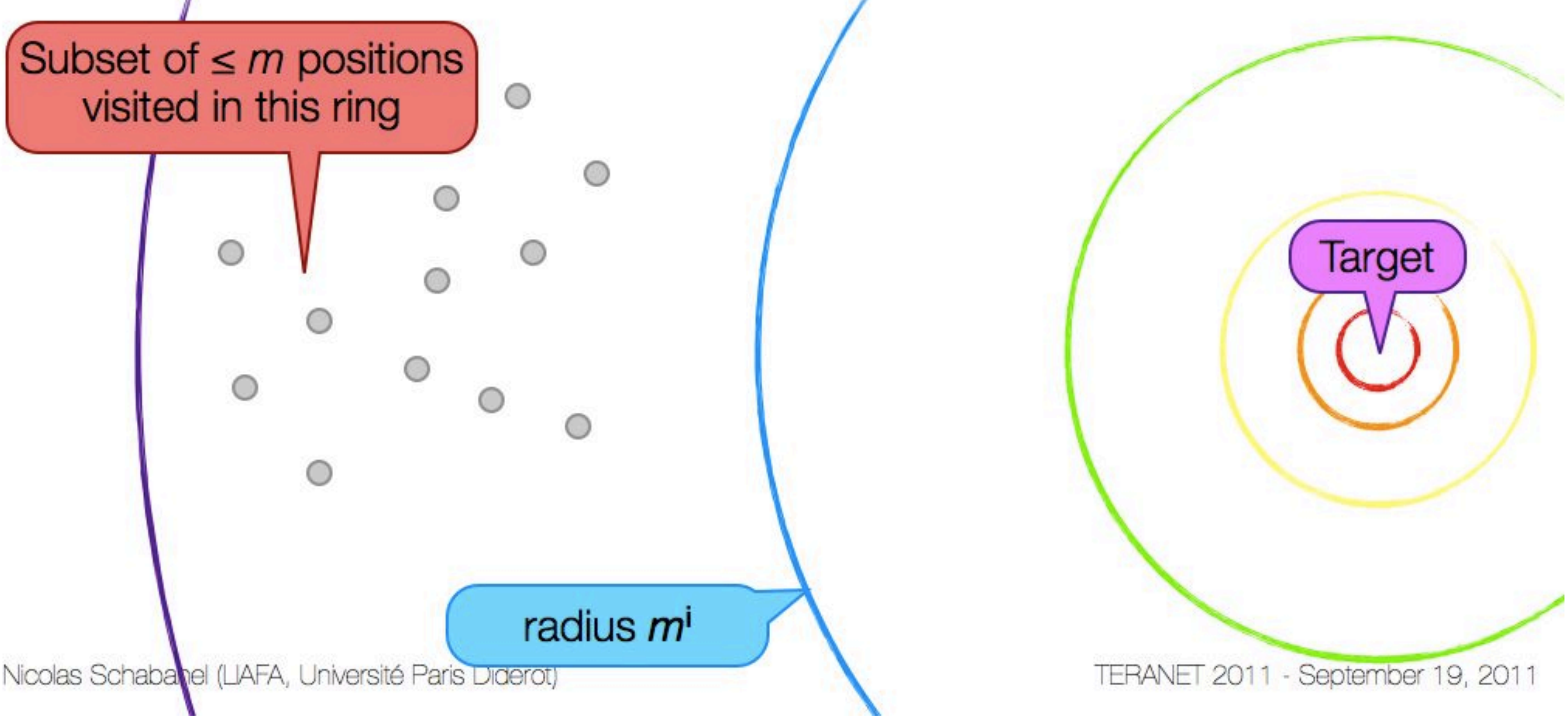
➡**What happens if we bound the time to be ≤ m = O(log$^{\gamma}$ n) for some γ > 0 ?**



Looking for a matching lower bound for d = 1

- There is a $O(\sqrt{n})$-time decentralized algorithm [Martel, Nguyen, 2004]

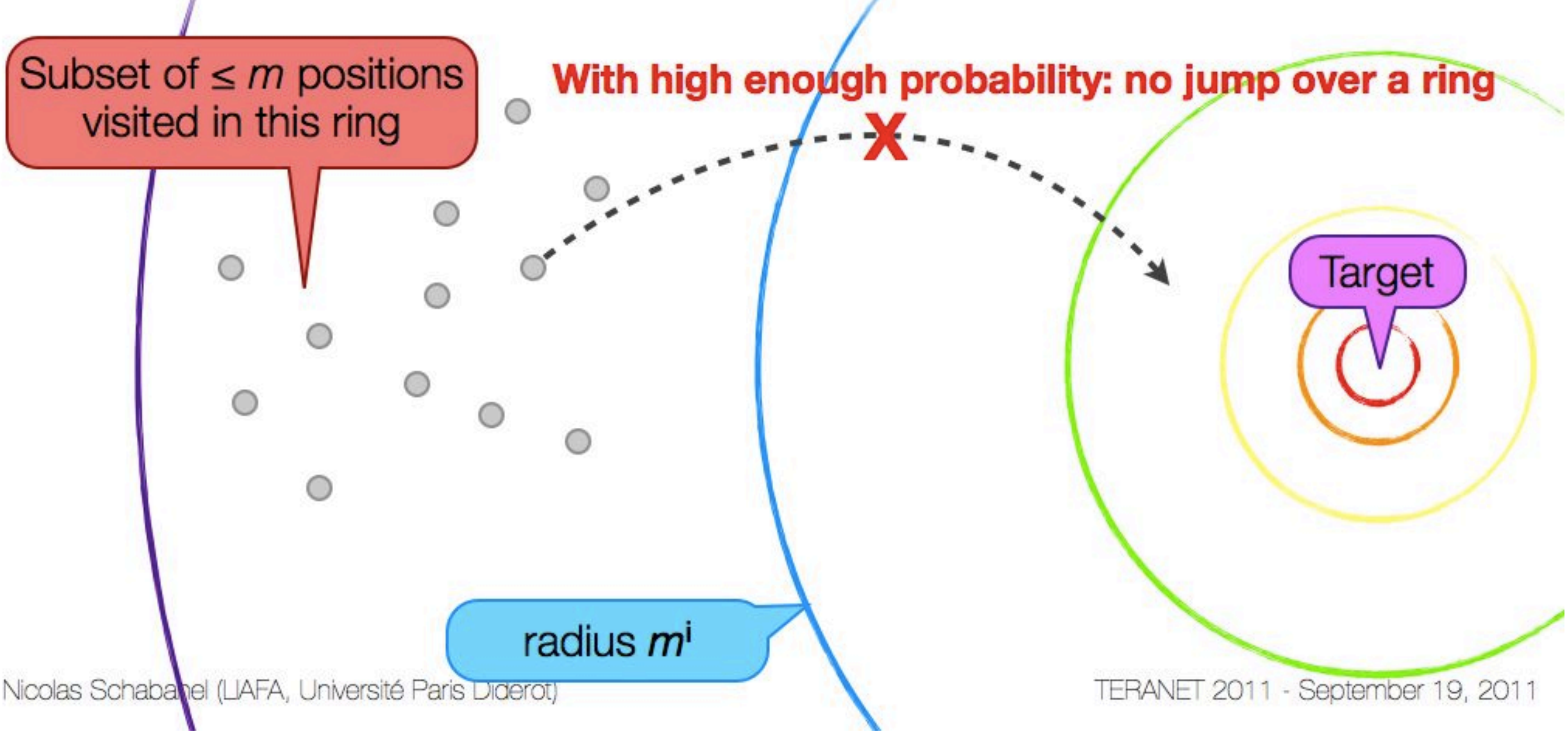
➡**What happens if we bound the time to be $\leq m = O(\log^{\gamma} n)$ for some $\gamma > 0$?**



Looking for a matching lower bound for $d = 1$

- There is a $O(\sqrt{n})$-time decentralized algorithm [Martel, Nguyen, 2004]

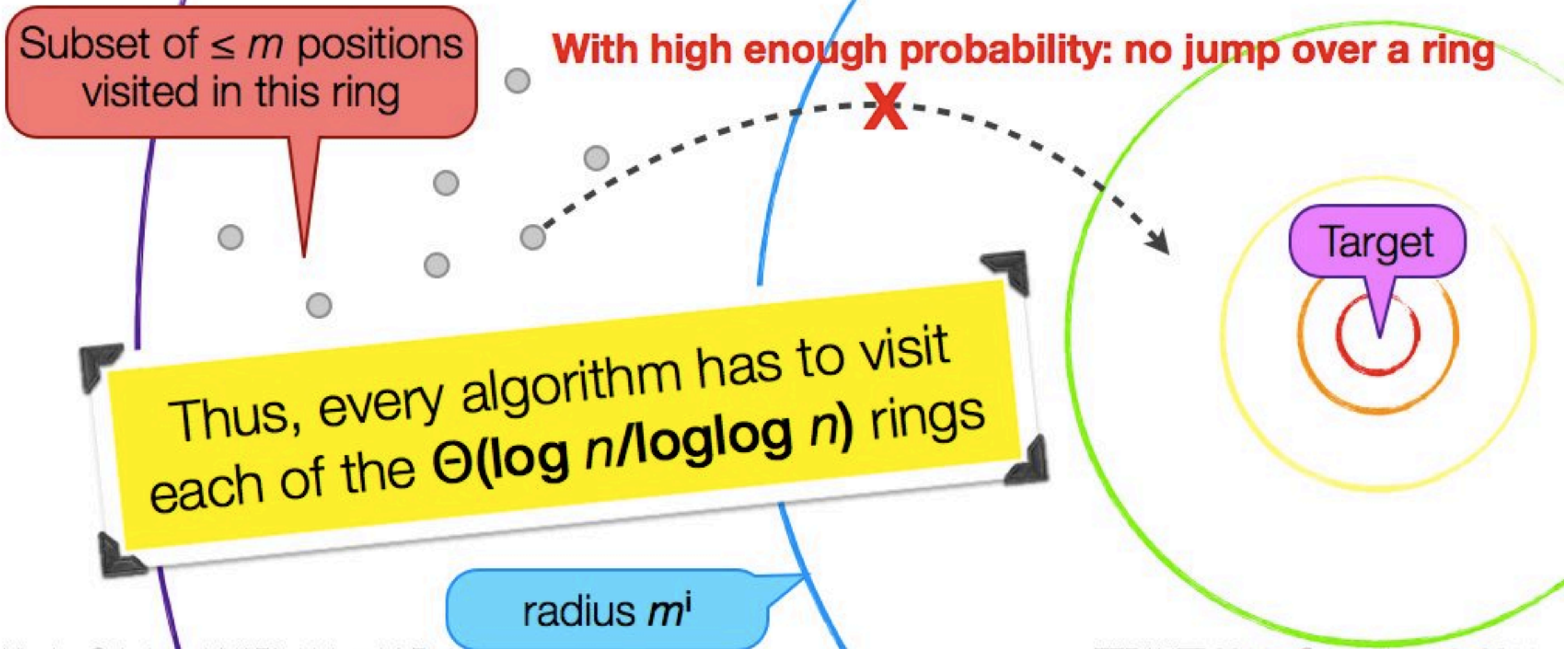
➡**What happens if we bound the time to be $\leq m = O(\log^{\gamma} n)$ for some $\gamma > 0$?**



Looking for a matching lower bound for d = 1

- There is a O(√n)-time decentralized algorithm [Martel, Nguyen, 2004]

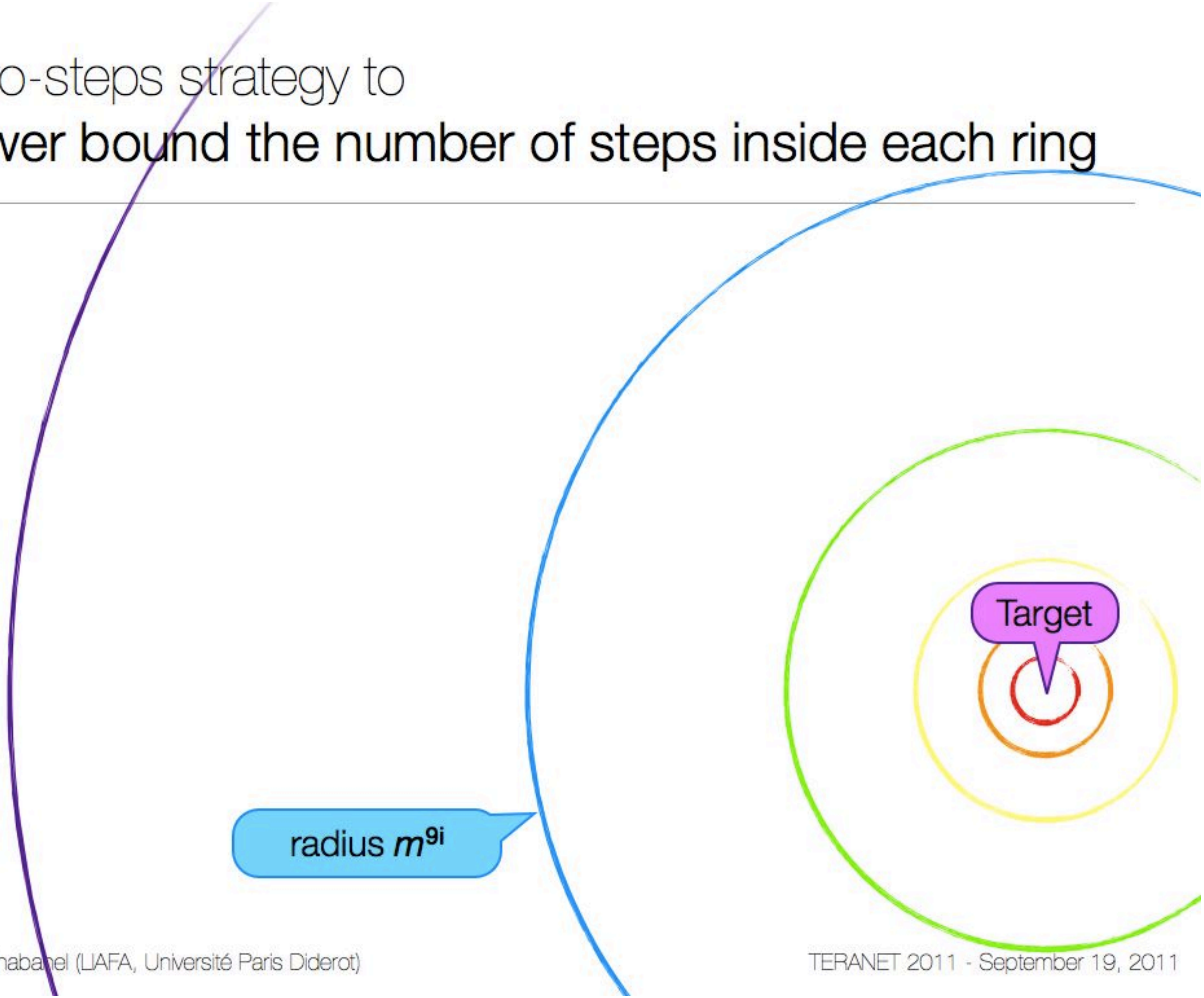
➡**What happens if we bound the time to be ≤ m = O(log$^{\gamma}$ n) for some γ > 0 ?**



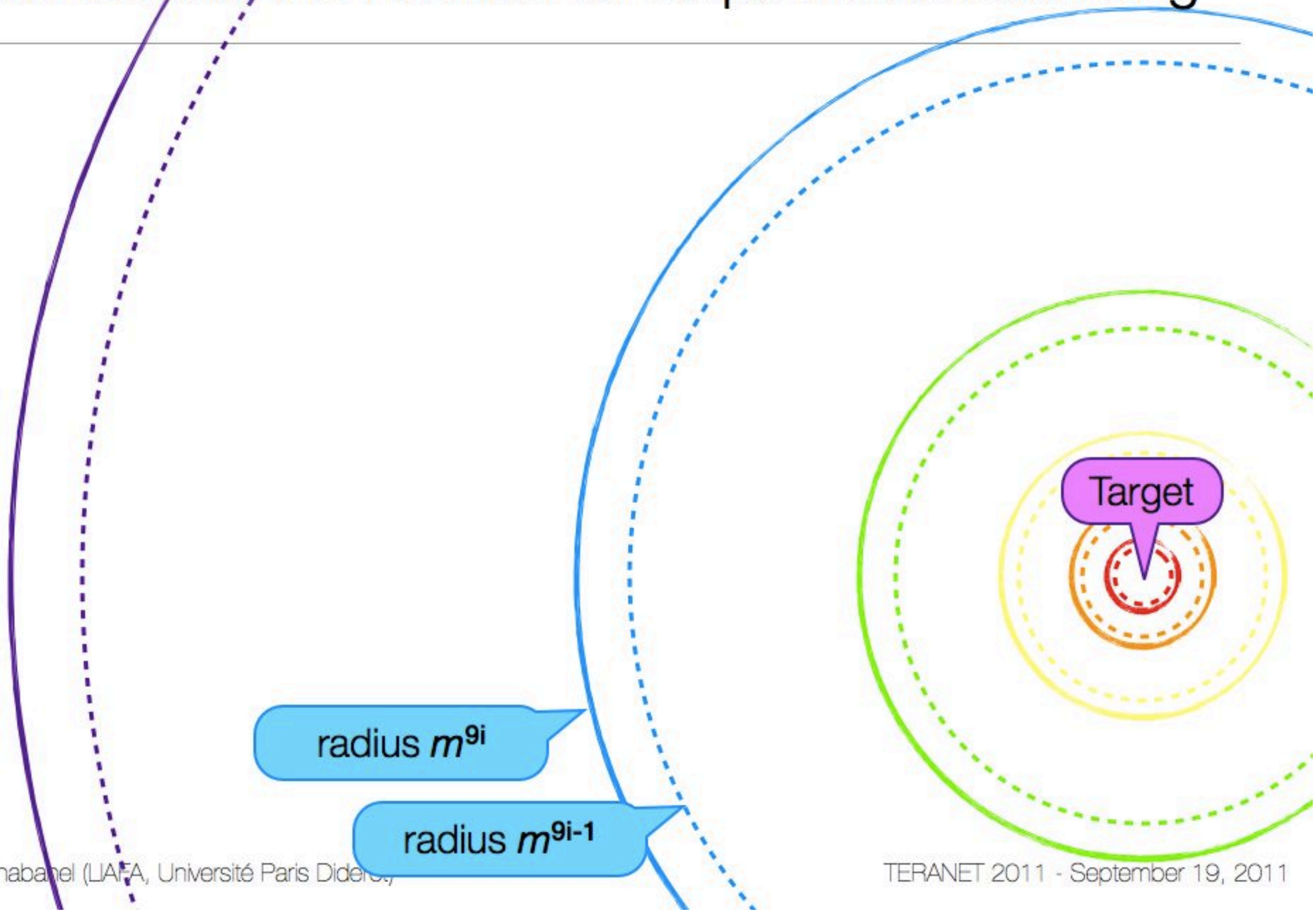
Two-steps strategy to lower bound the number of steps inside each ring

Two-steps strategy to

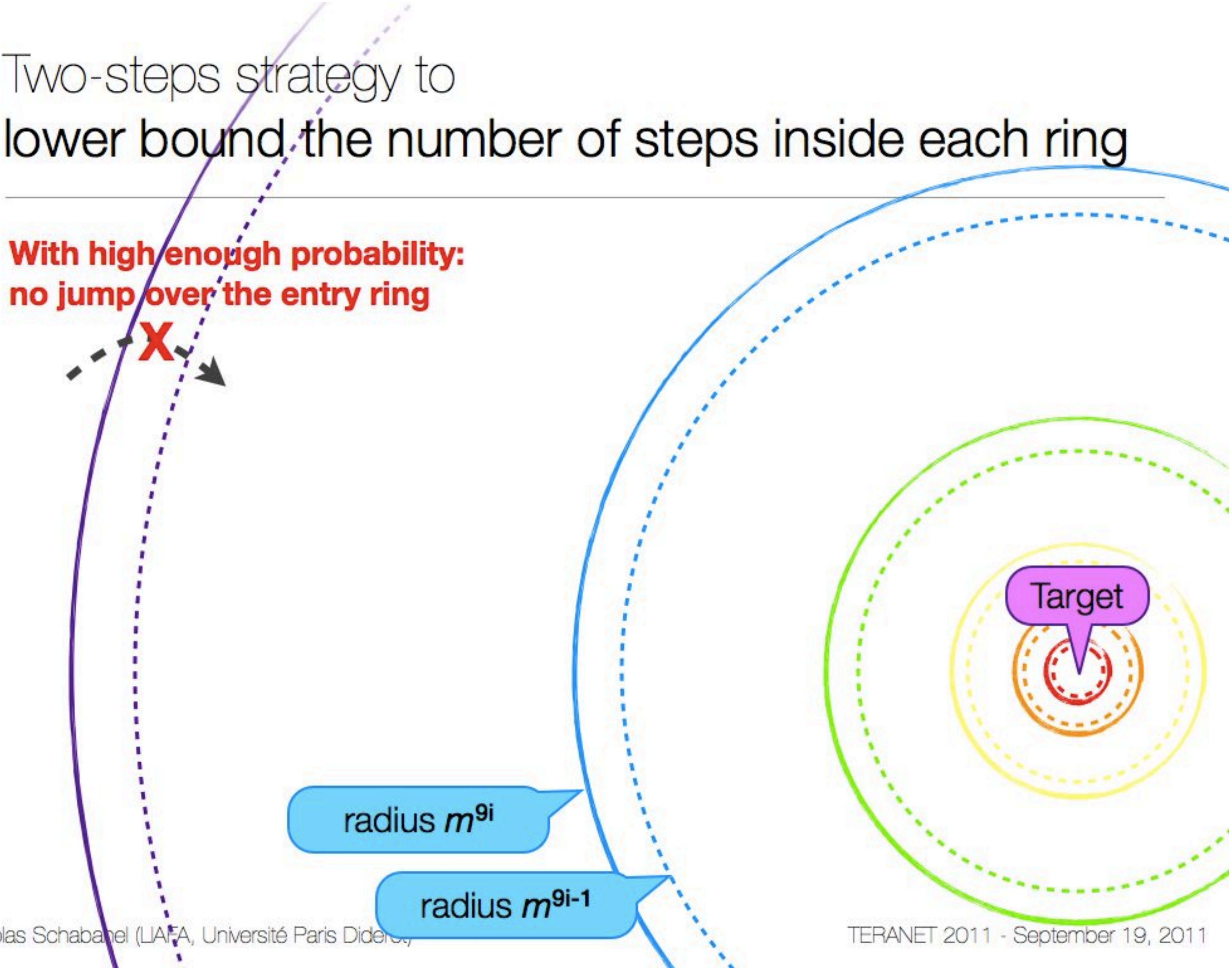
lower bound the number of steps inside each ring



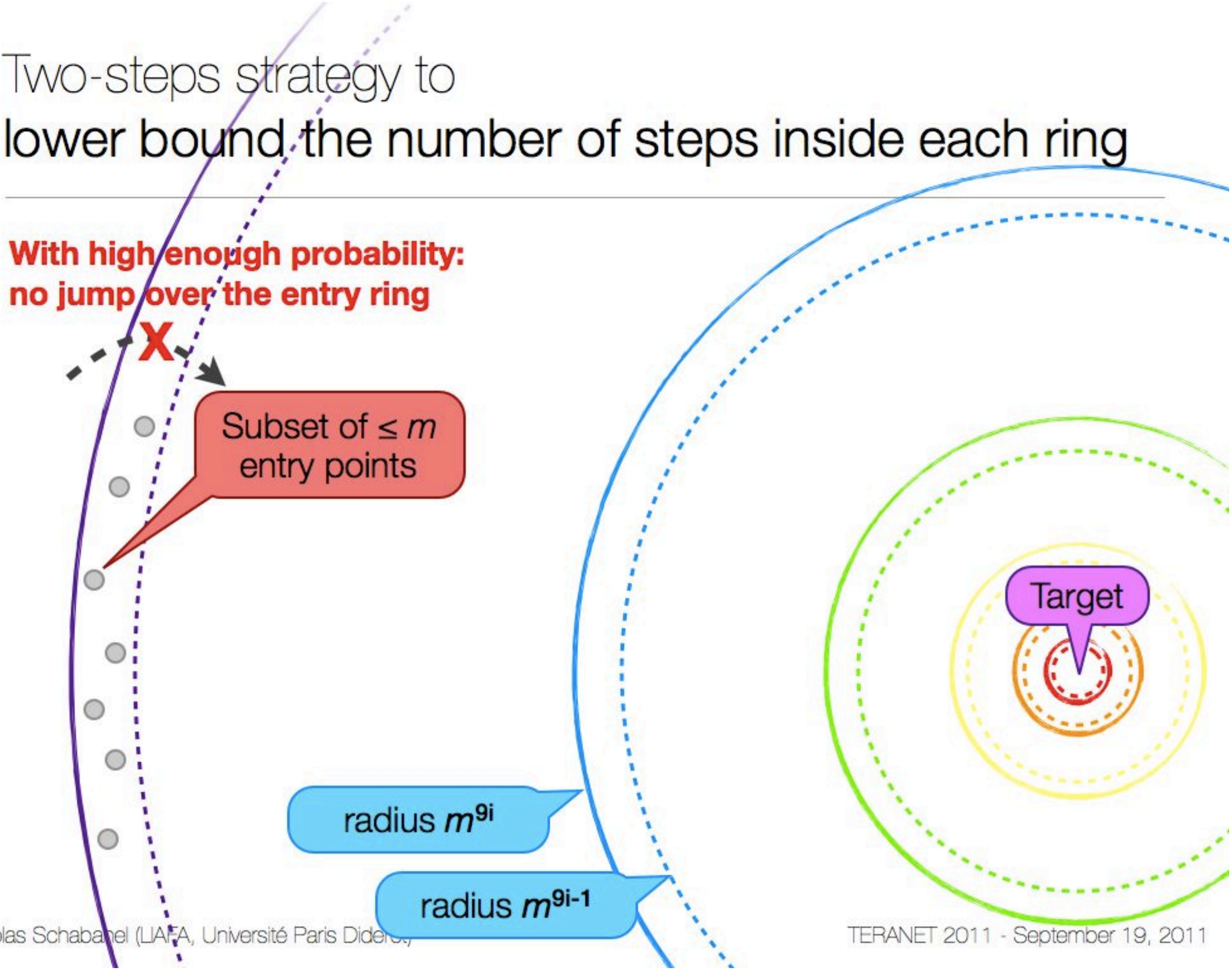
Two-steps strategy to
lower bound the number of steps inside each ring



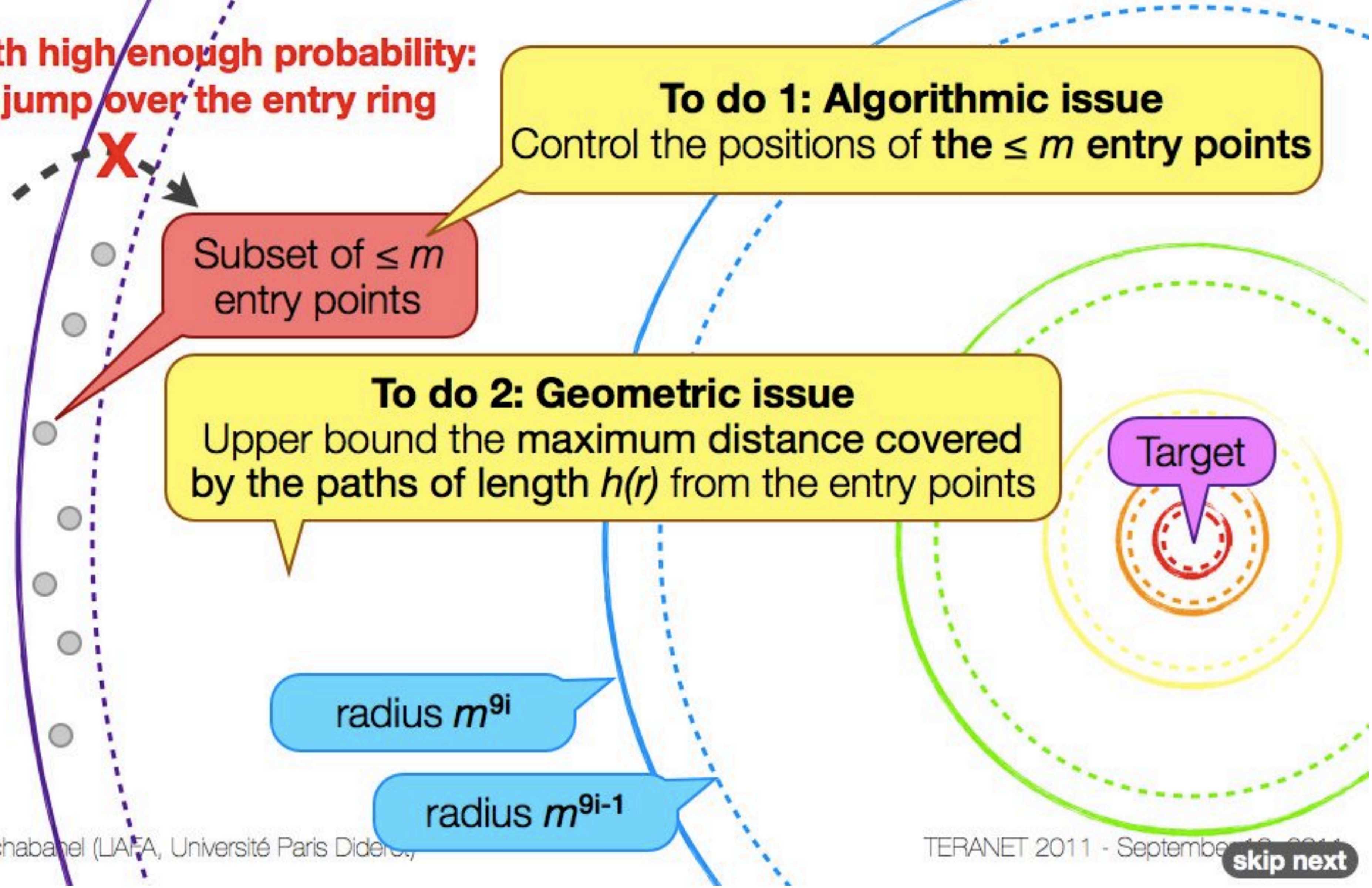
Two-steps strategy to
lower bound the number of steps inside each ring



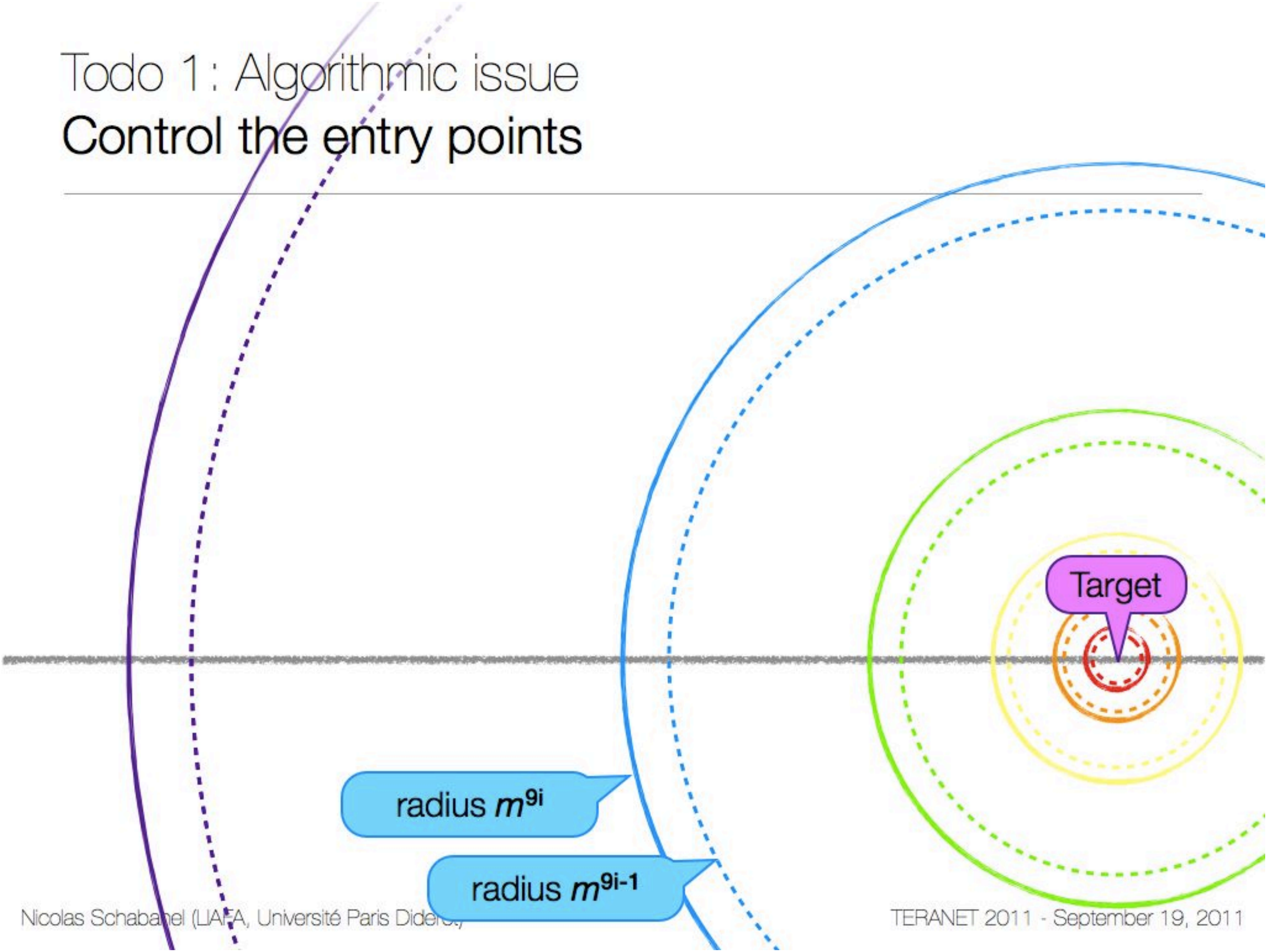
Two-steps strategy to
lower bound the number of steps inside each ring



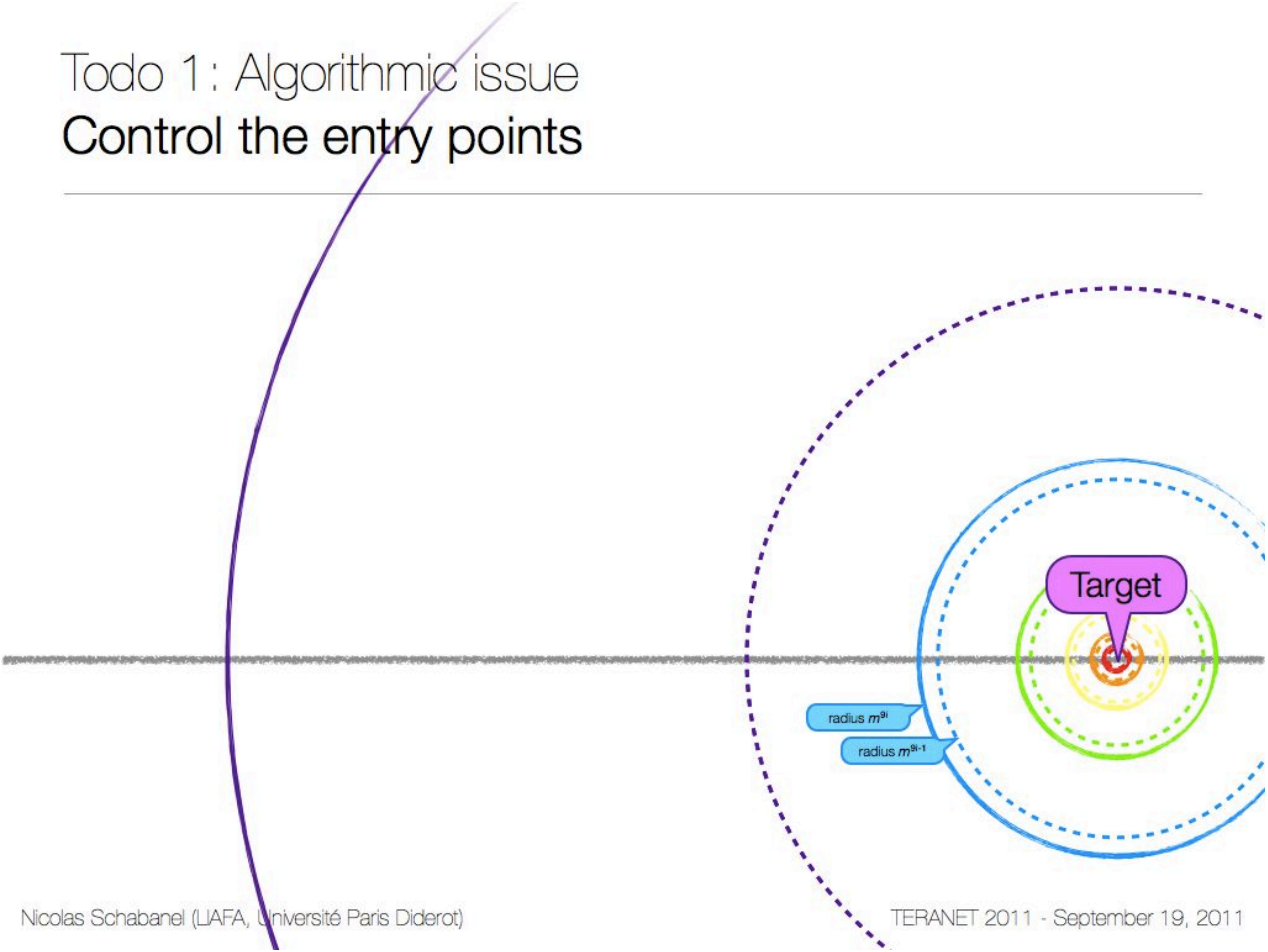
Two-steps strategy to
lower bound the number of steps inside each ring



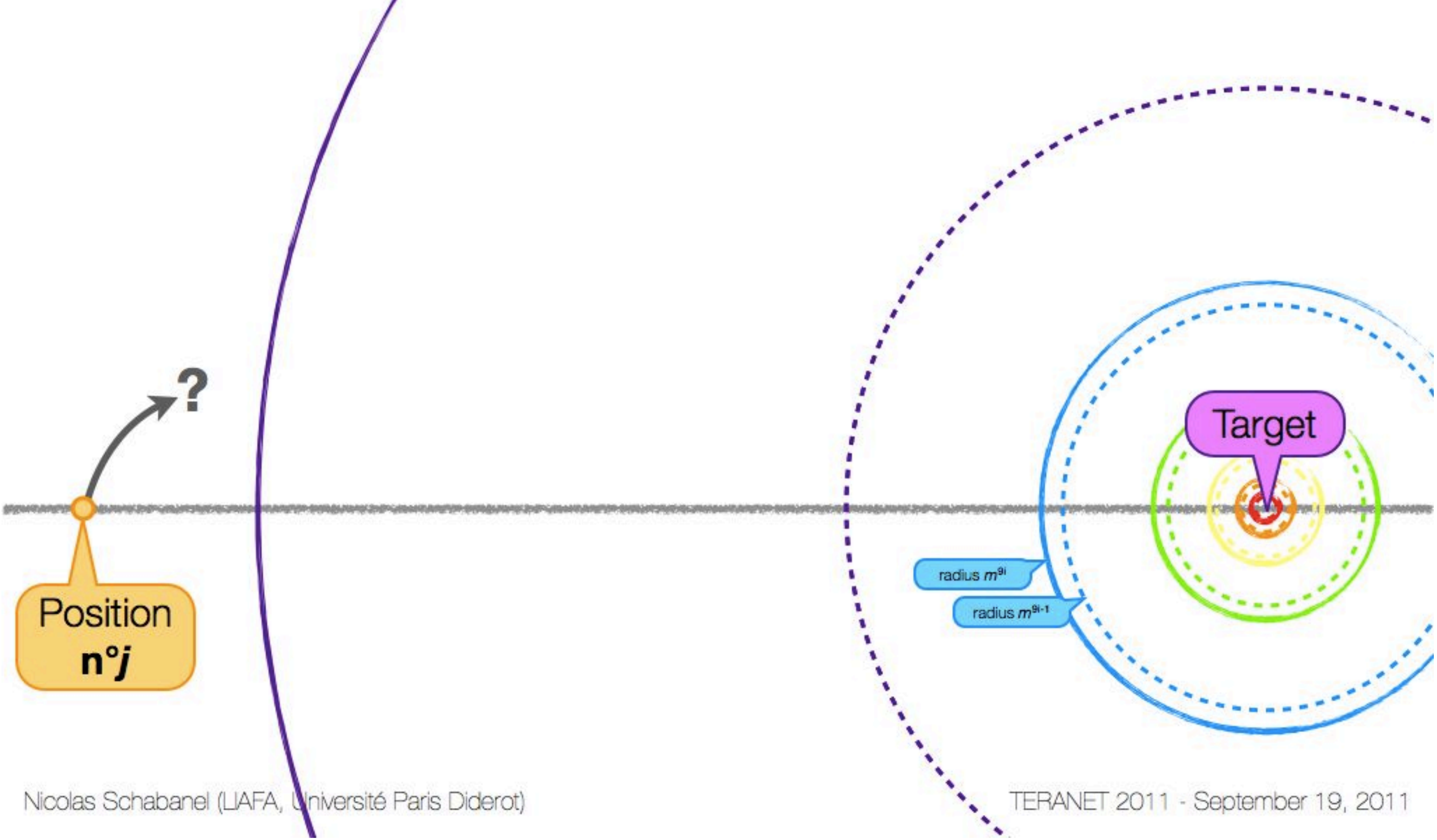
Todo 1: Algorithmic issue
Control the entry points



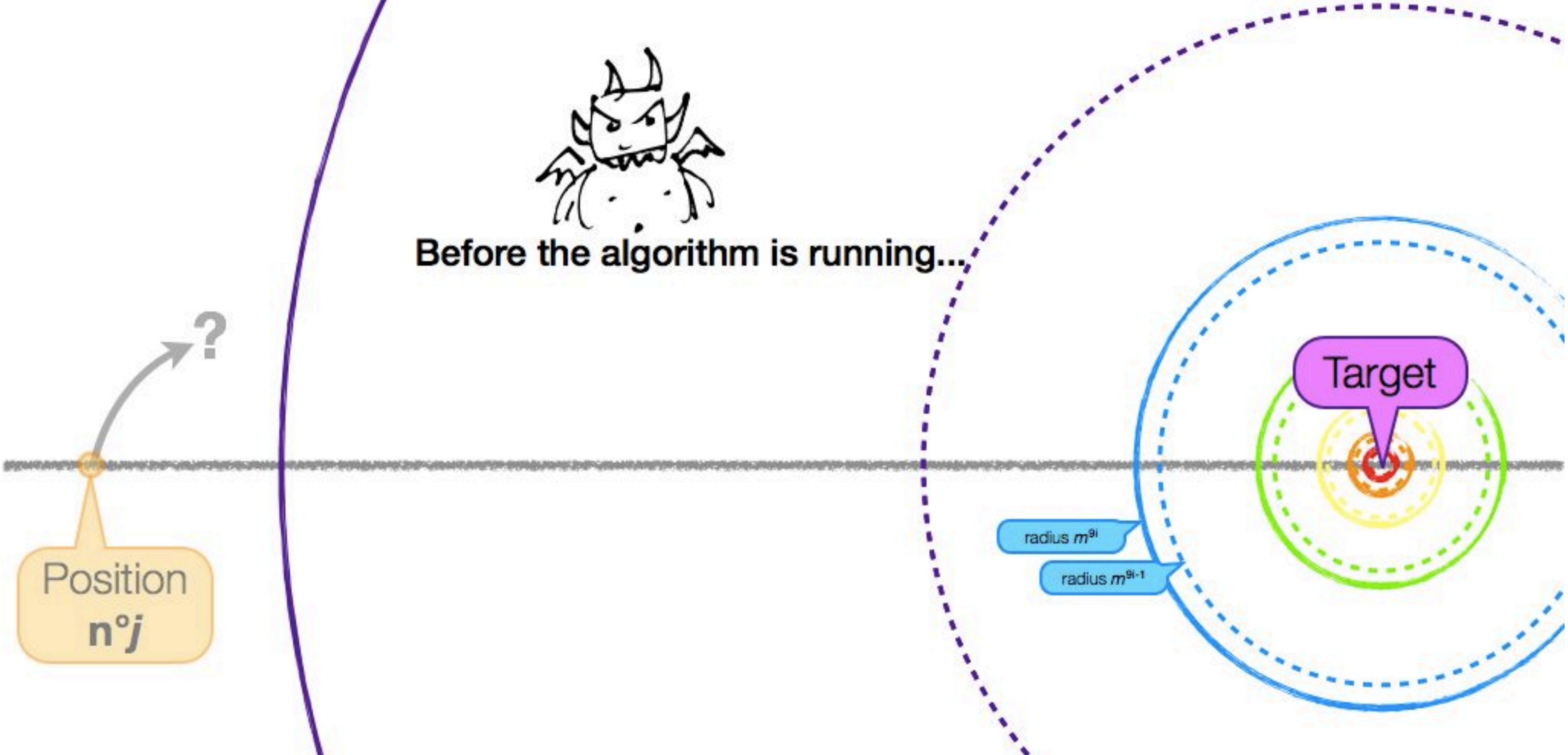
Todo 1: Algorithmic issue
Control the entry points



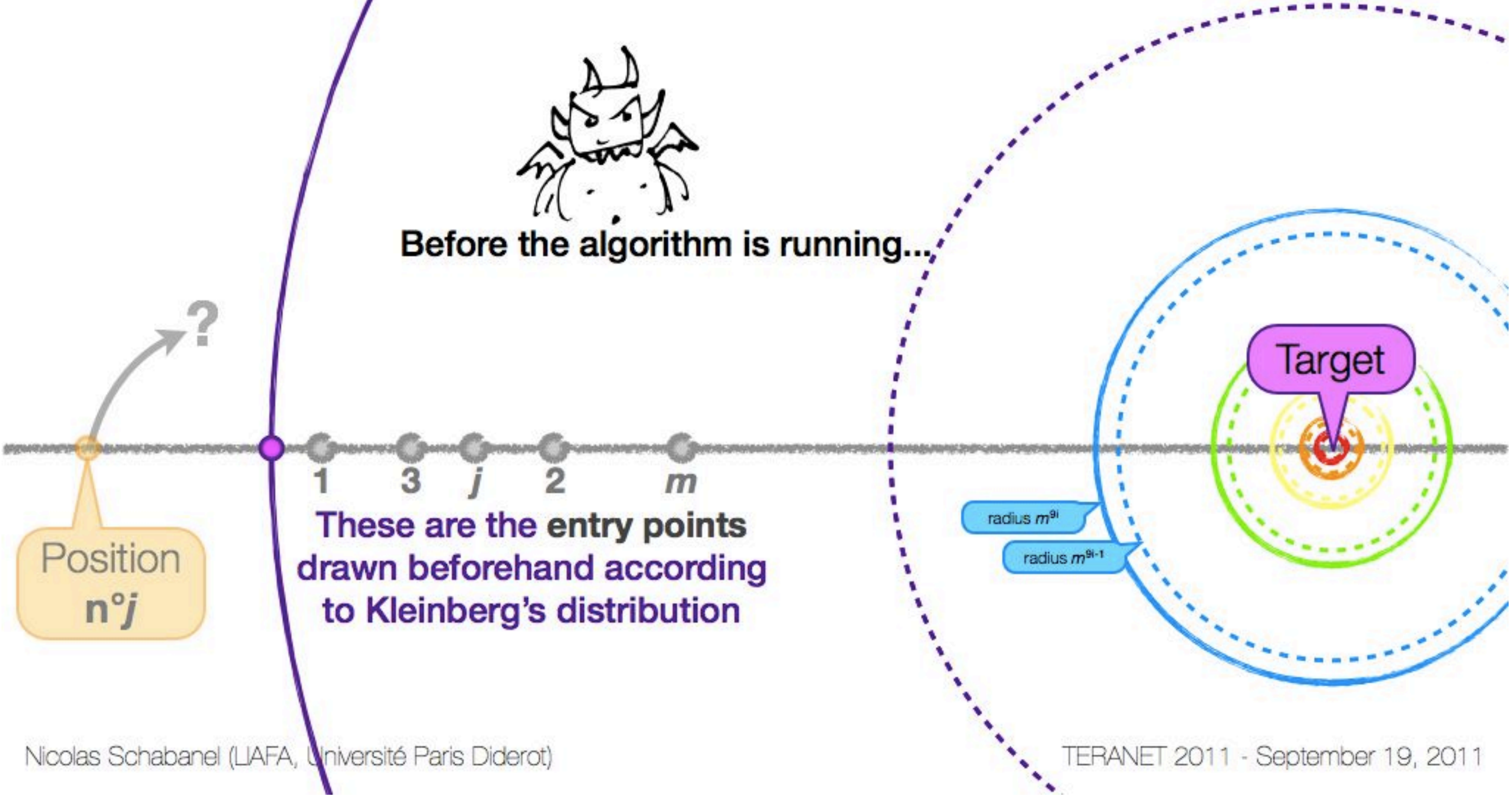
Todo 1 : Algorithmic issue
Control the entry points



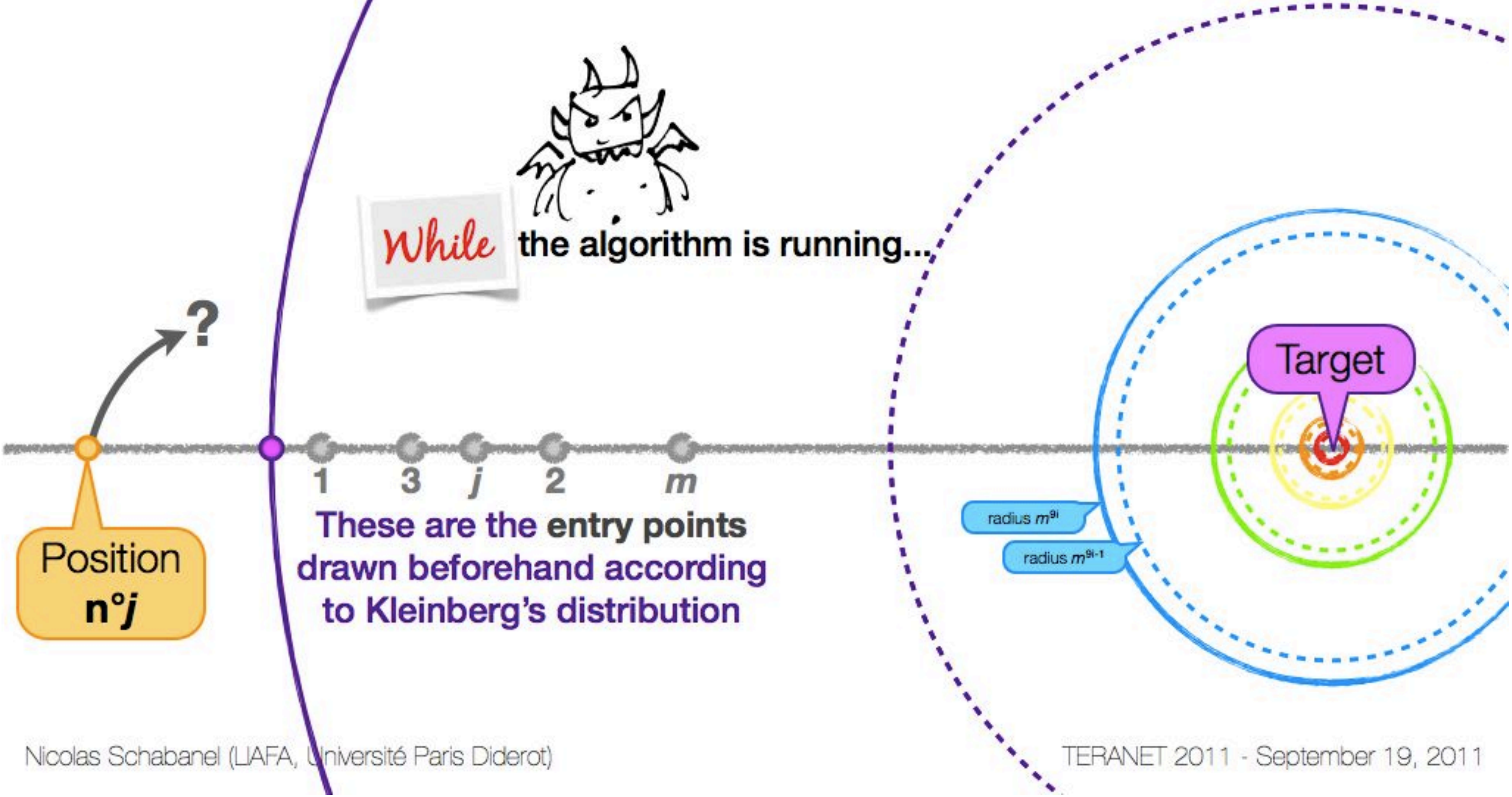
Todo 1: Algorithmic issue
Control the entry points



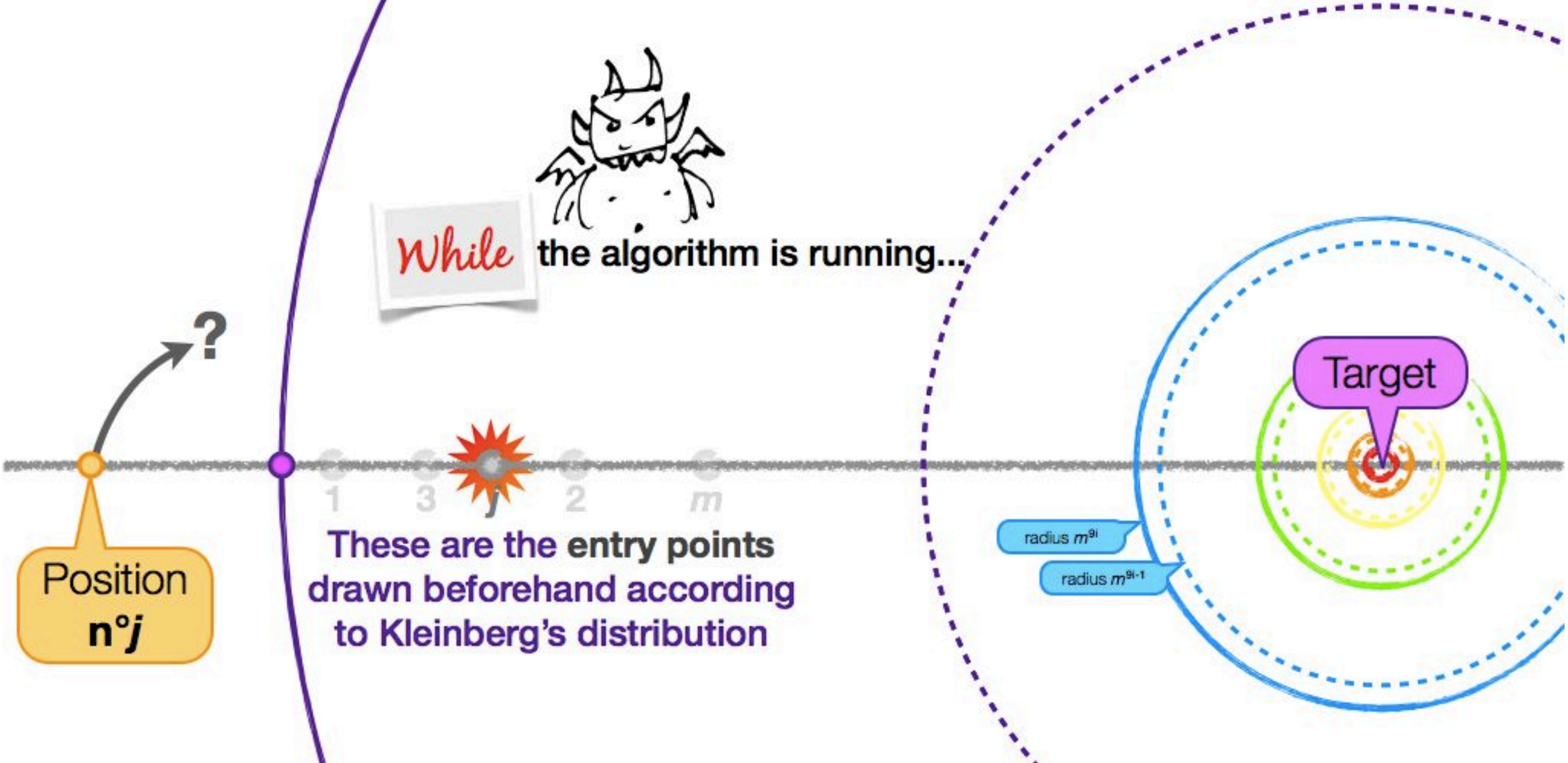
Todo 1: Algorithmic issue
Control the entry points



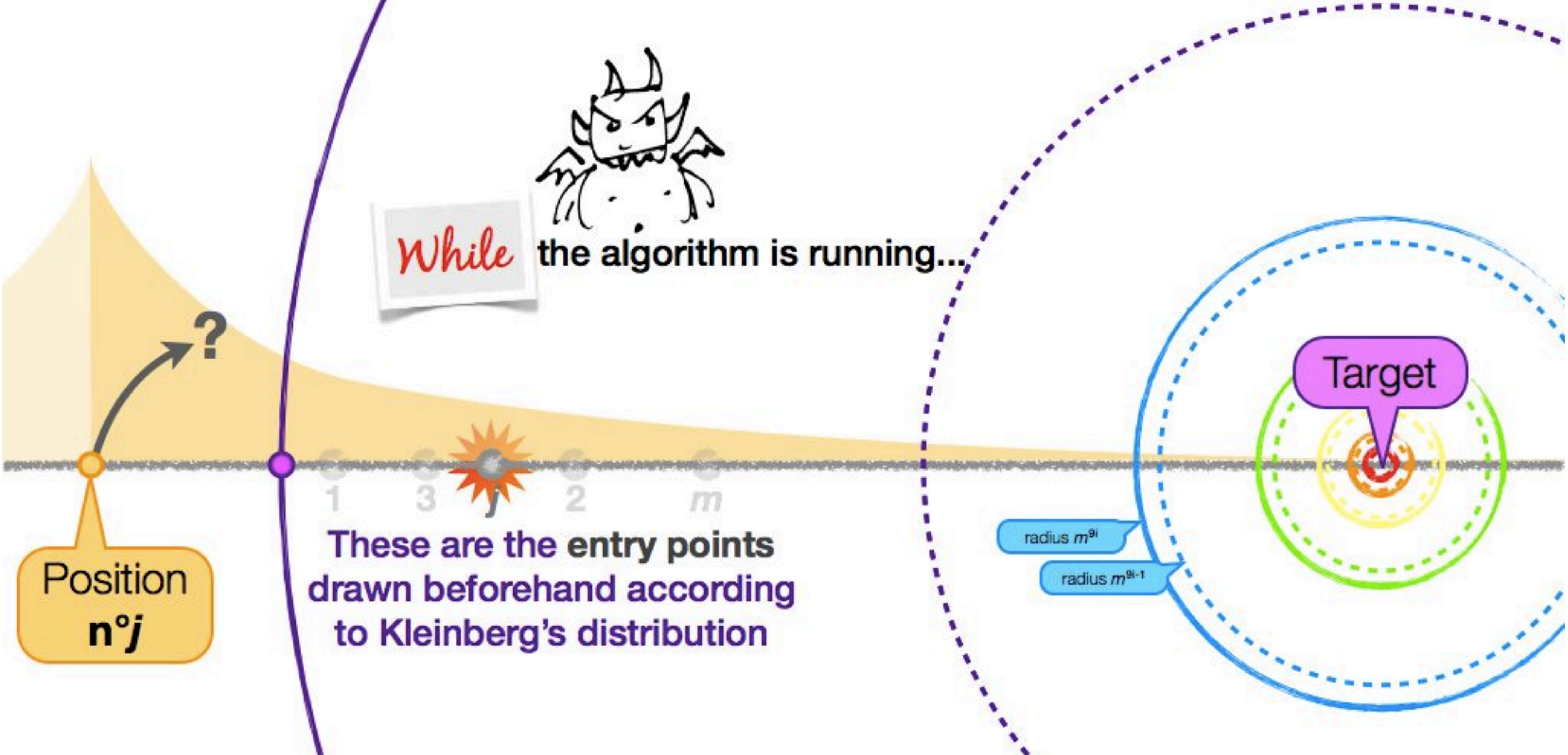
Todo 1: Algorithmic issue
Control the entry points



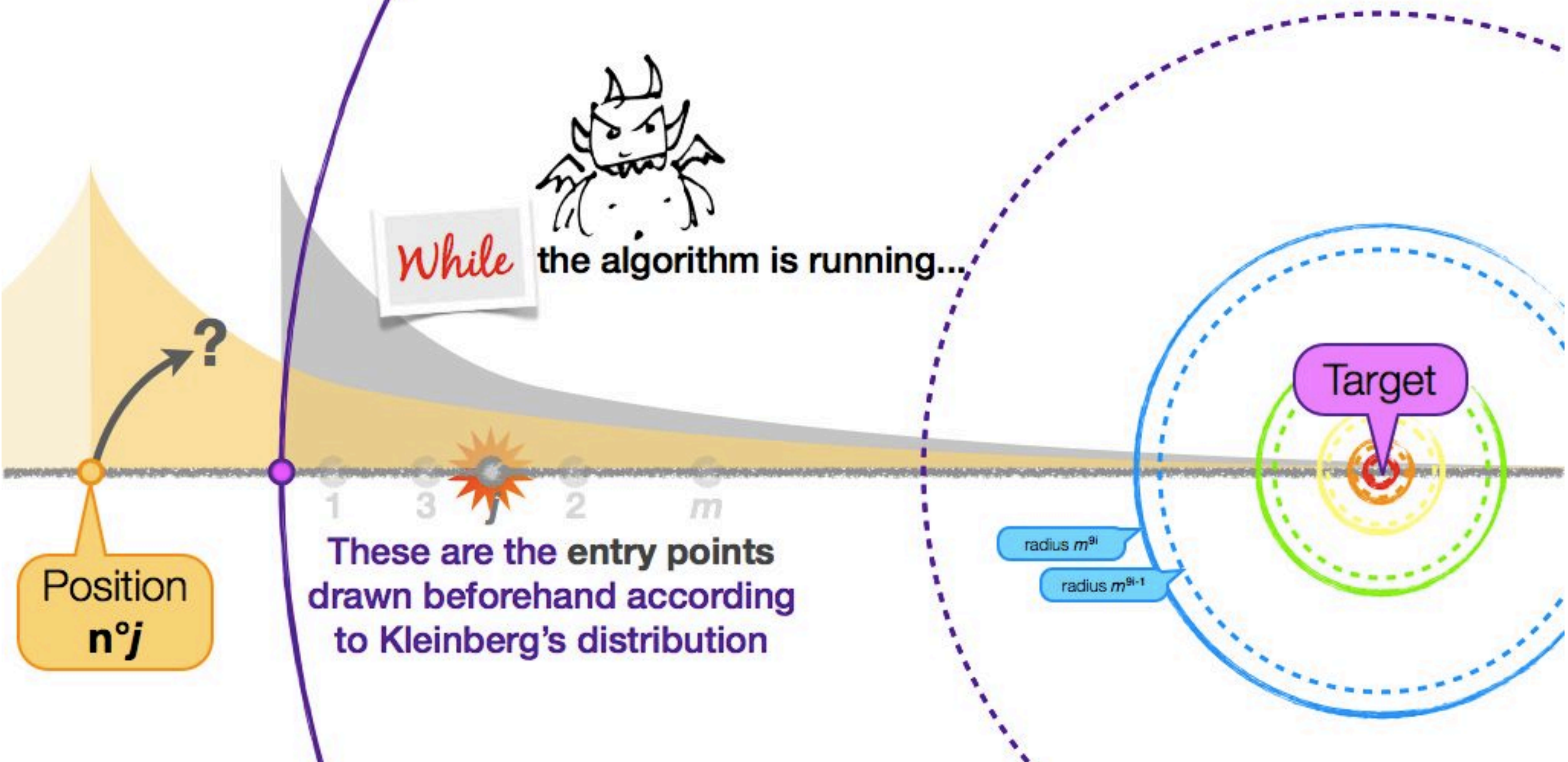
Todo 1: Algorithmic issue
Control the entry points



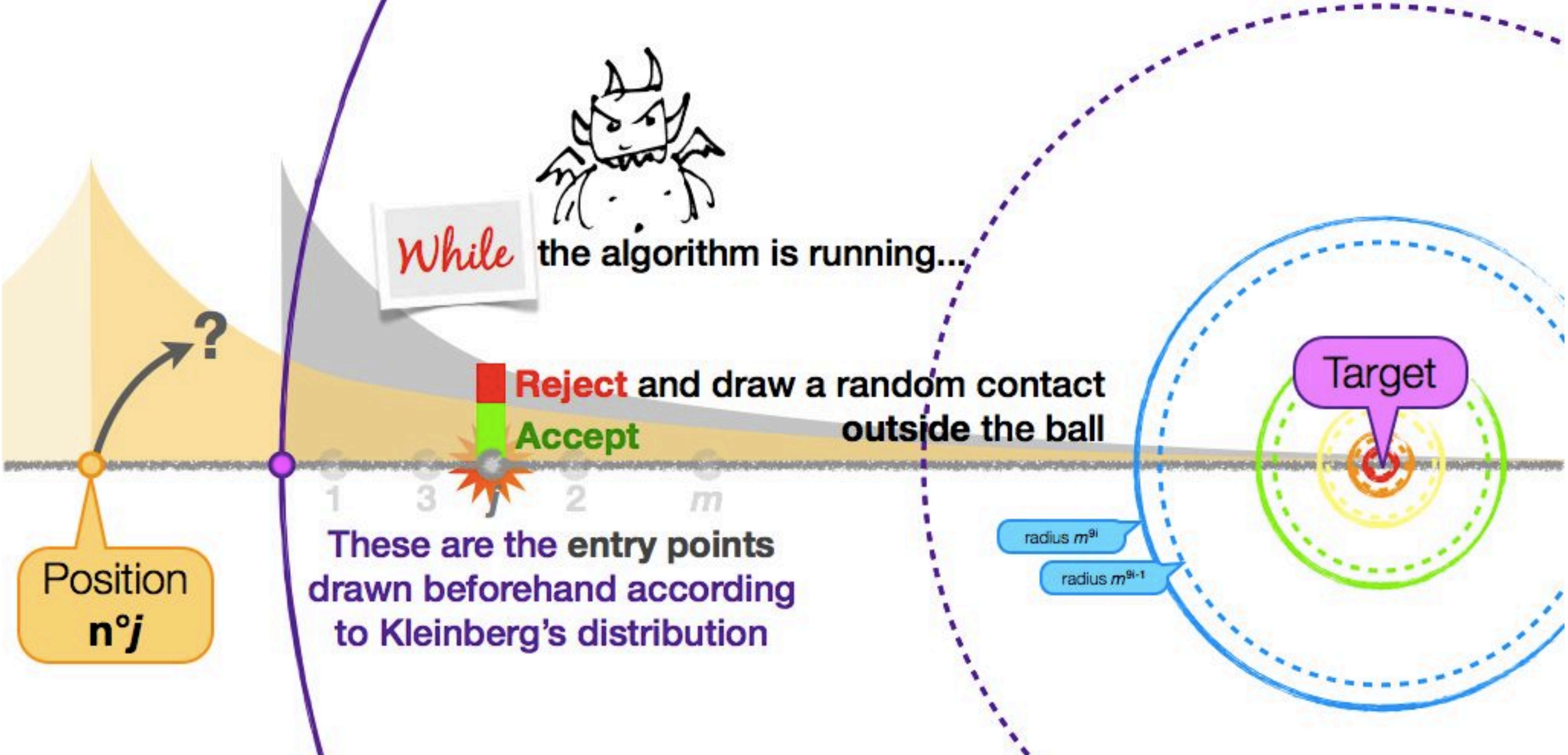
Todo 1: Algorithmic issue
Control the entry points



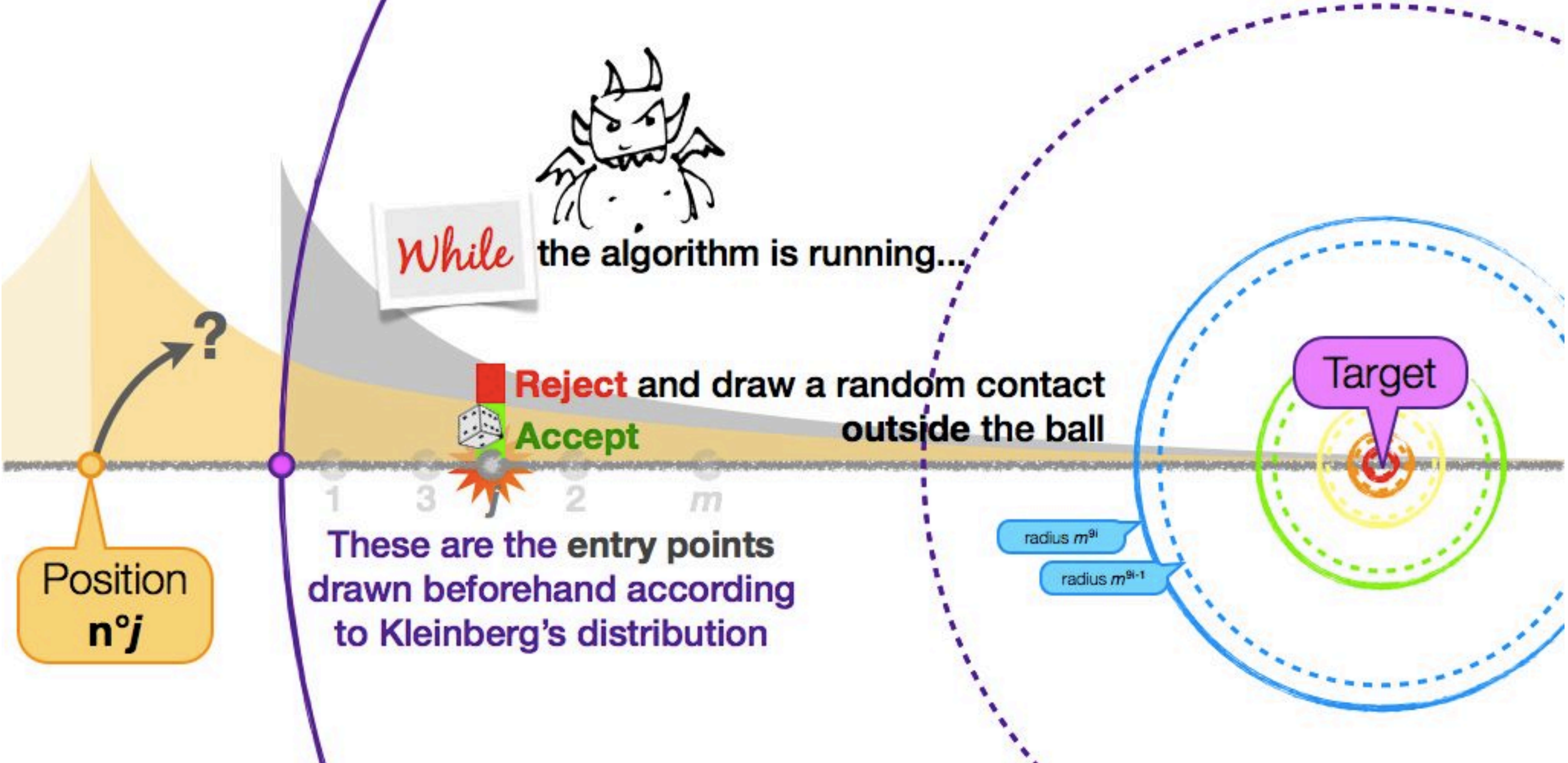
Todo 1: Algorithmic issue
Control the entry points



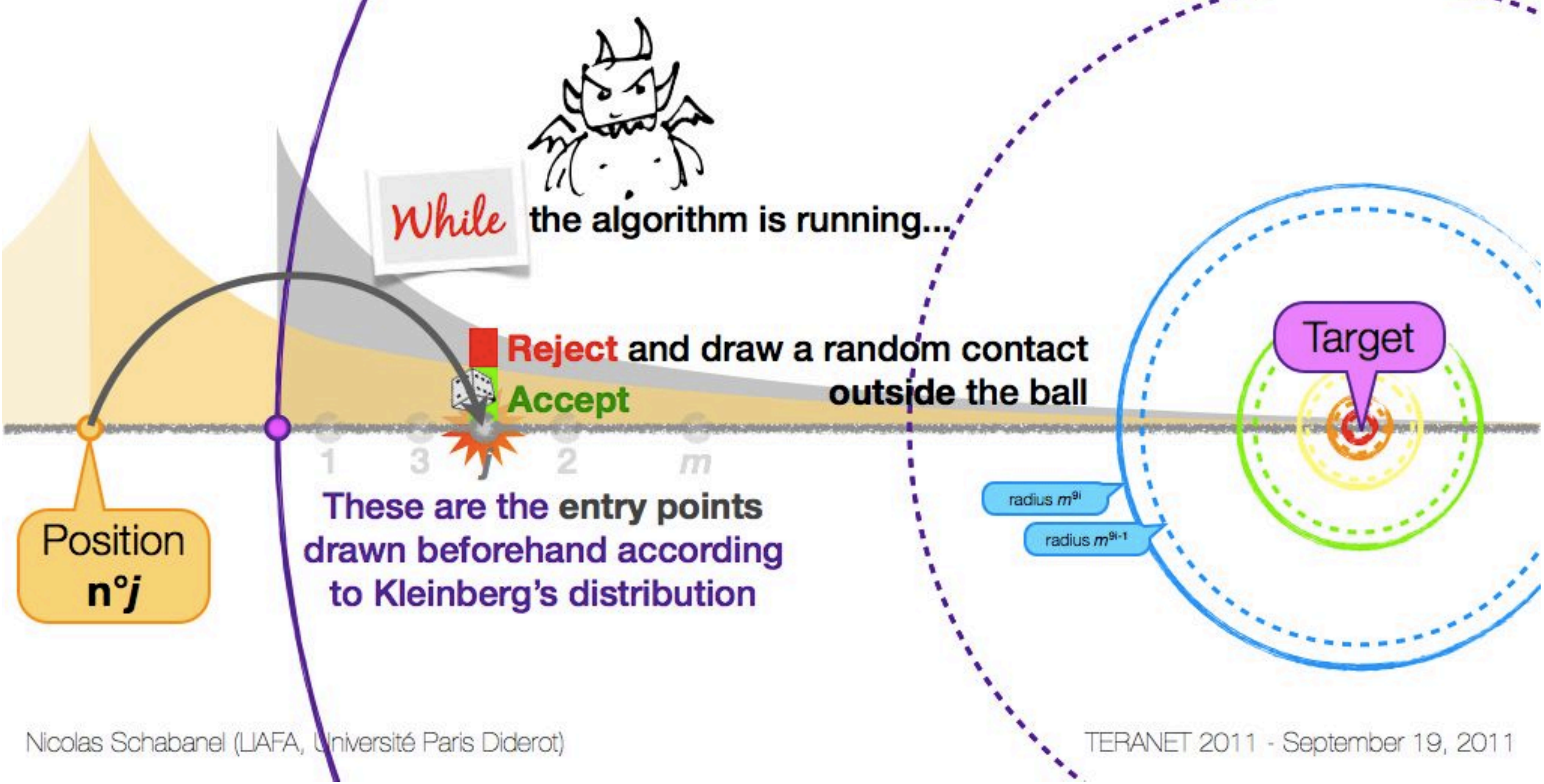
Todo 1: Algorithmic issue
Control the entry points



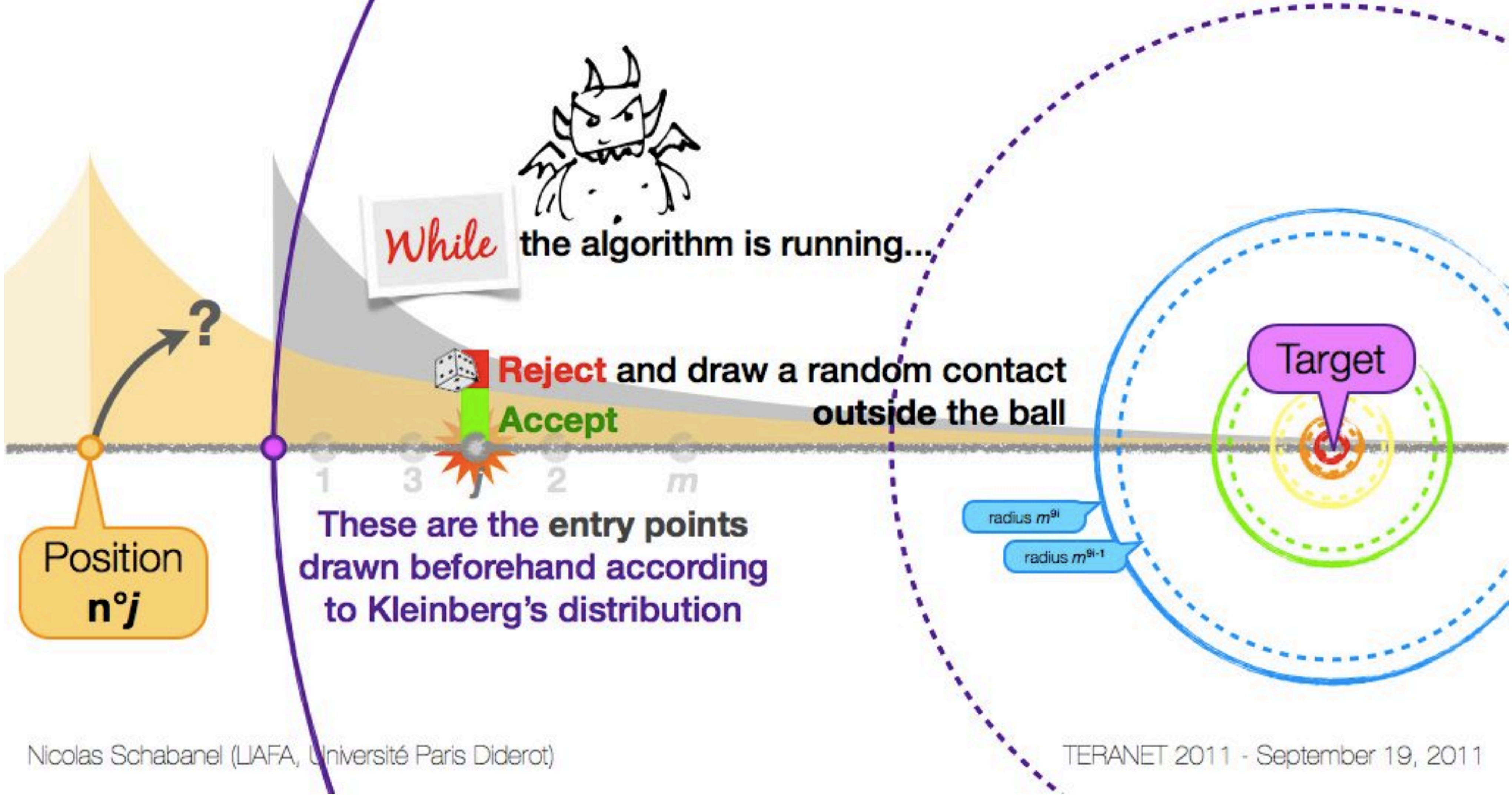
Todo 1: Algorithmic issue
Control the entry points



Todo 1: Algorithmic issue
Control the entry points

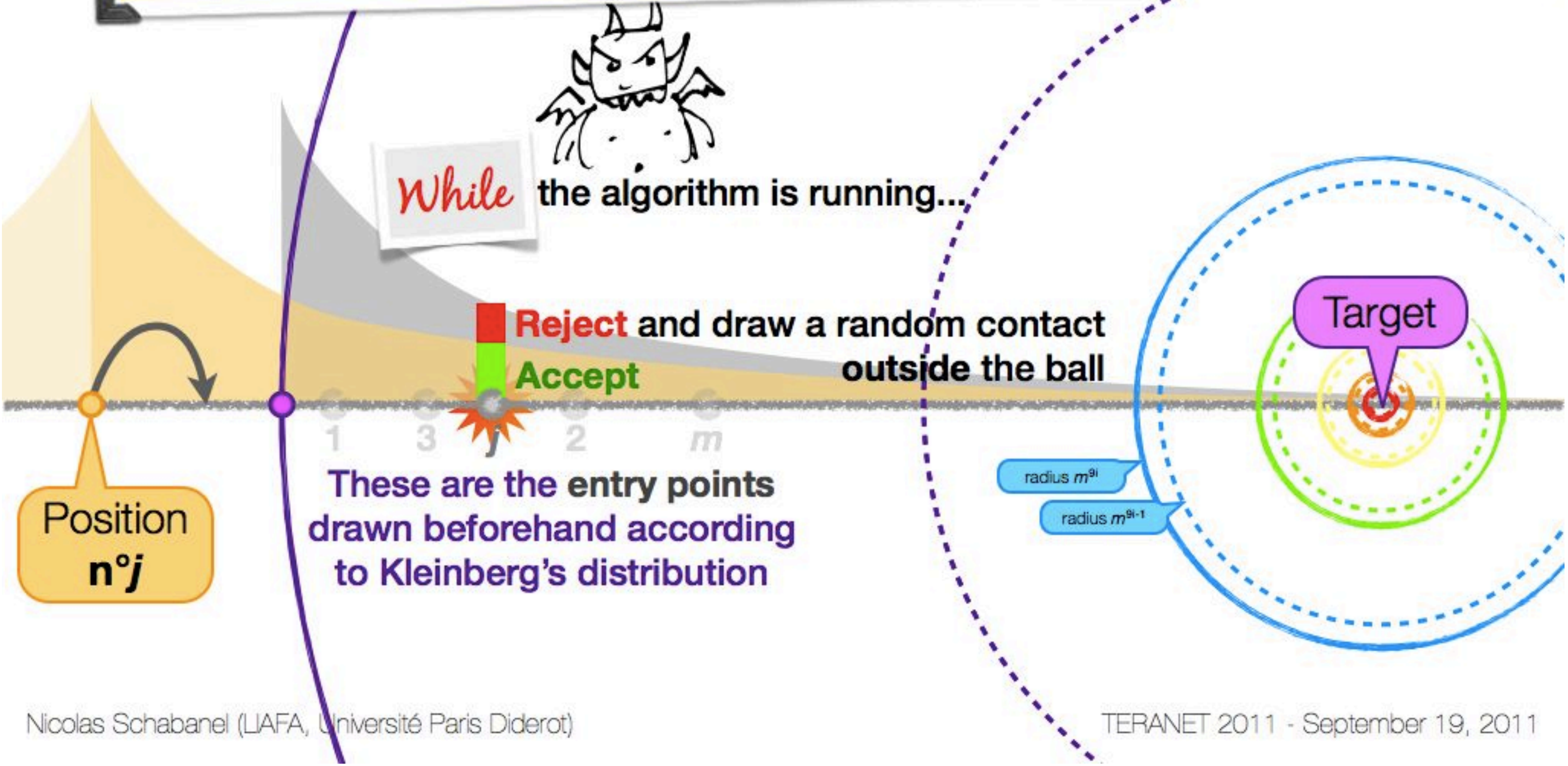


Todo 1: Algorithmic issue
Control the entry points

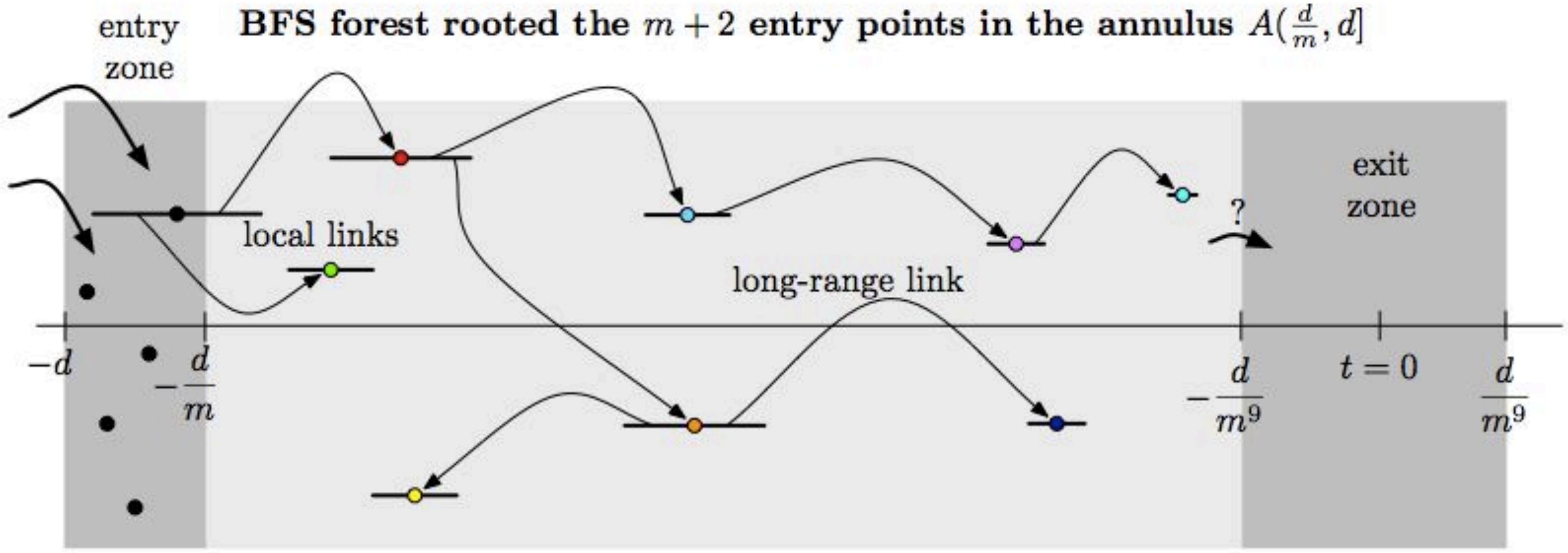


Todo 1: Algorithmic issue
Control the entry points

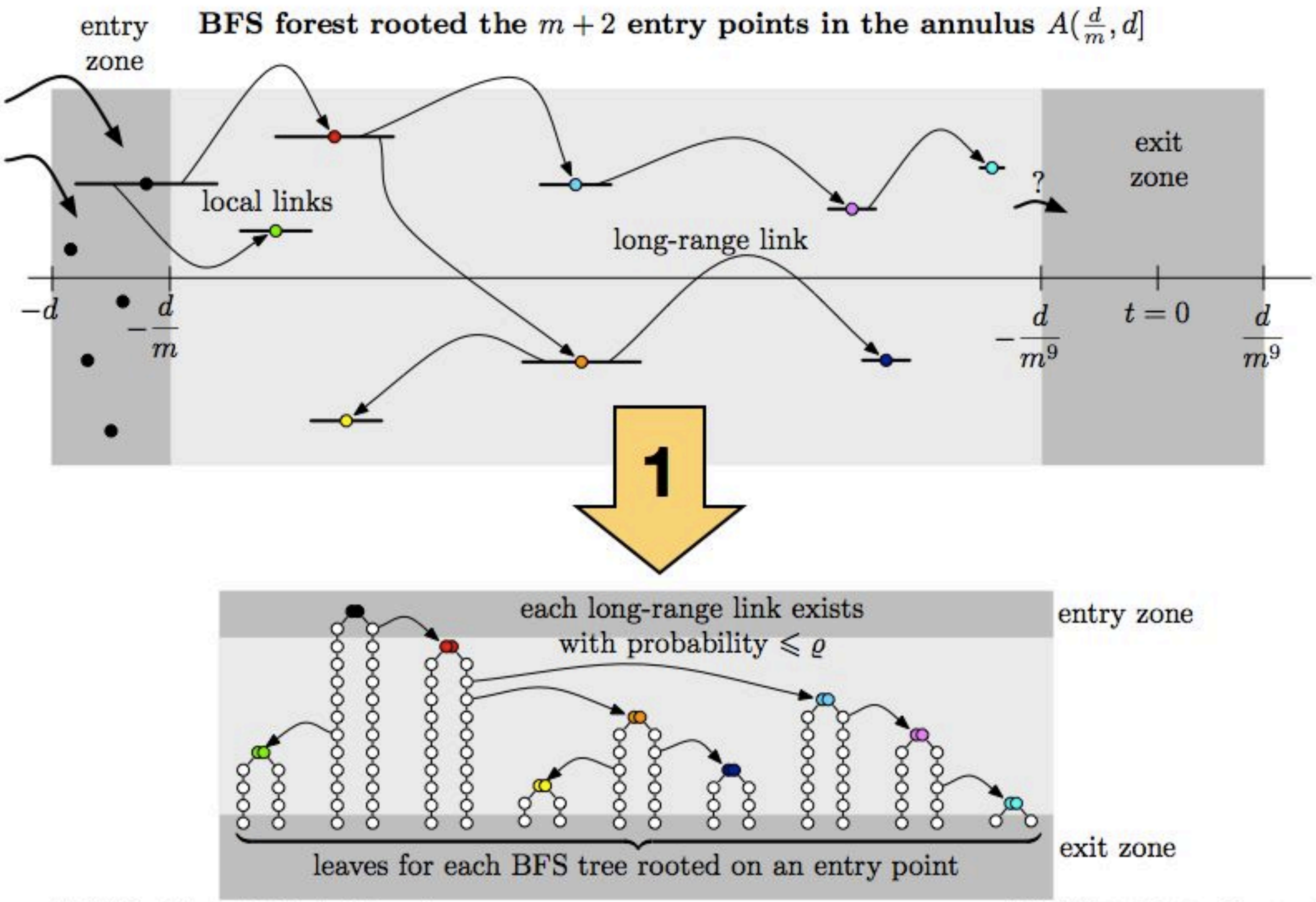
For the algorithm, the distribution is **indistinguishable from the original**
But the entry points are now **independent** of the algorithm



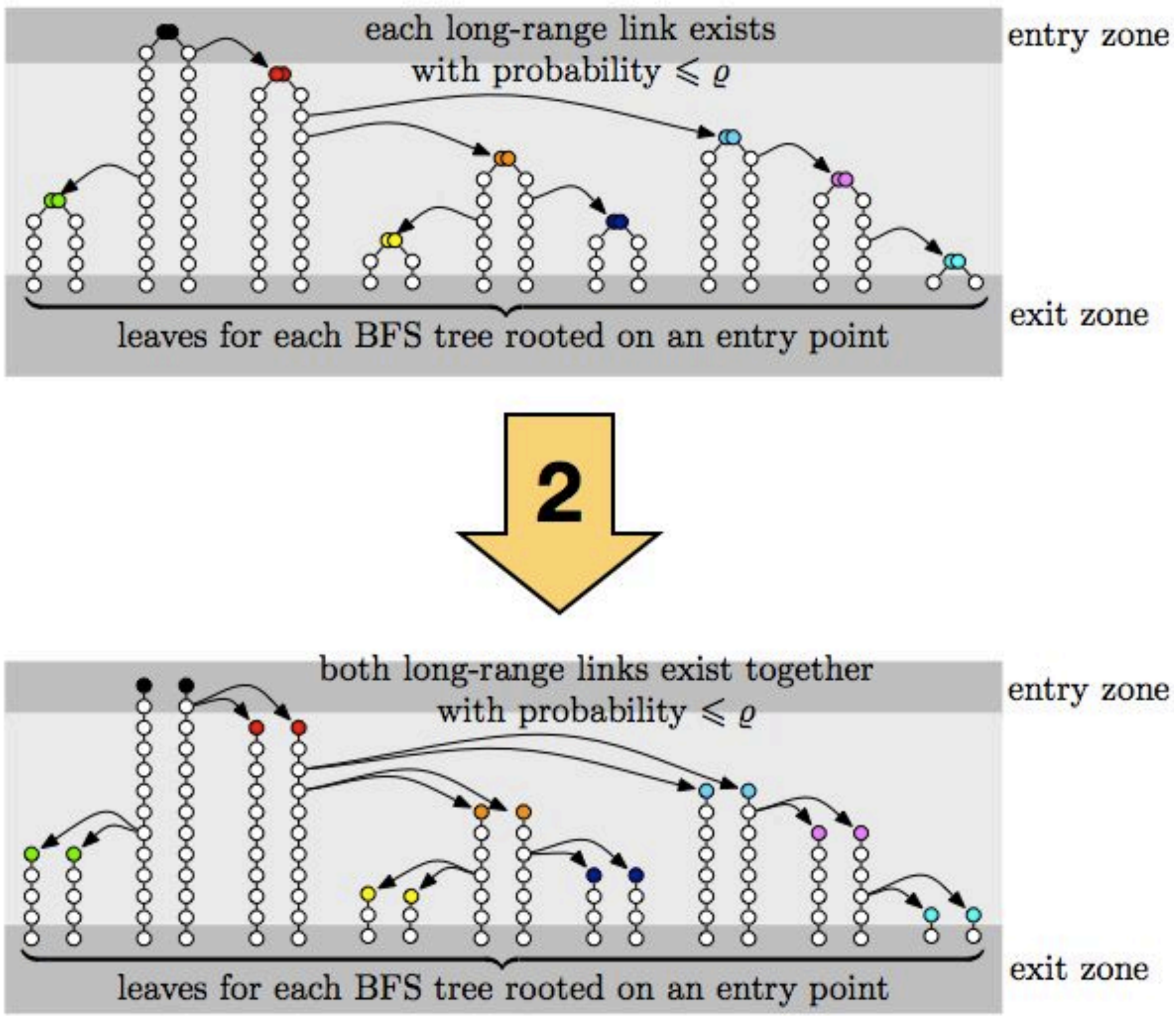
Todo 2: Geometric issue
Upper bounding the distance covered by the paths



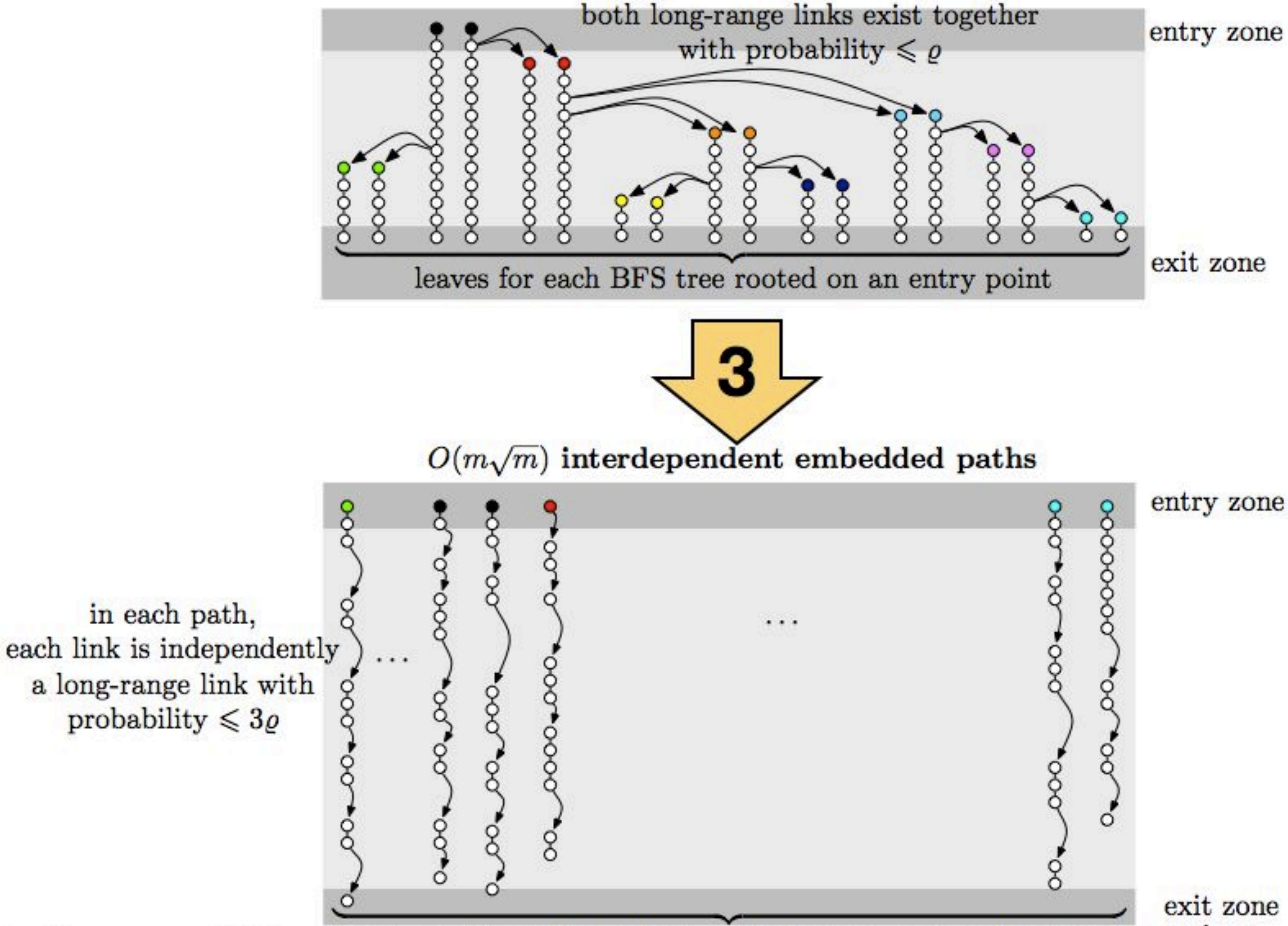
Todo 2: Geometric issue
Upper bounding the distance covered by the paths



Todo 2: Geometric issue
Upper bounding the distance covered by the paths



Todo 2: Geometric issue
Upper bounding the distance covered by the paths

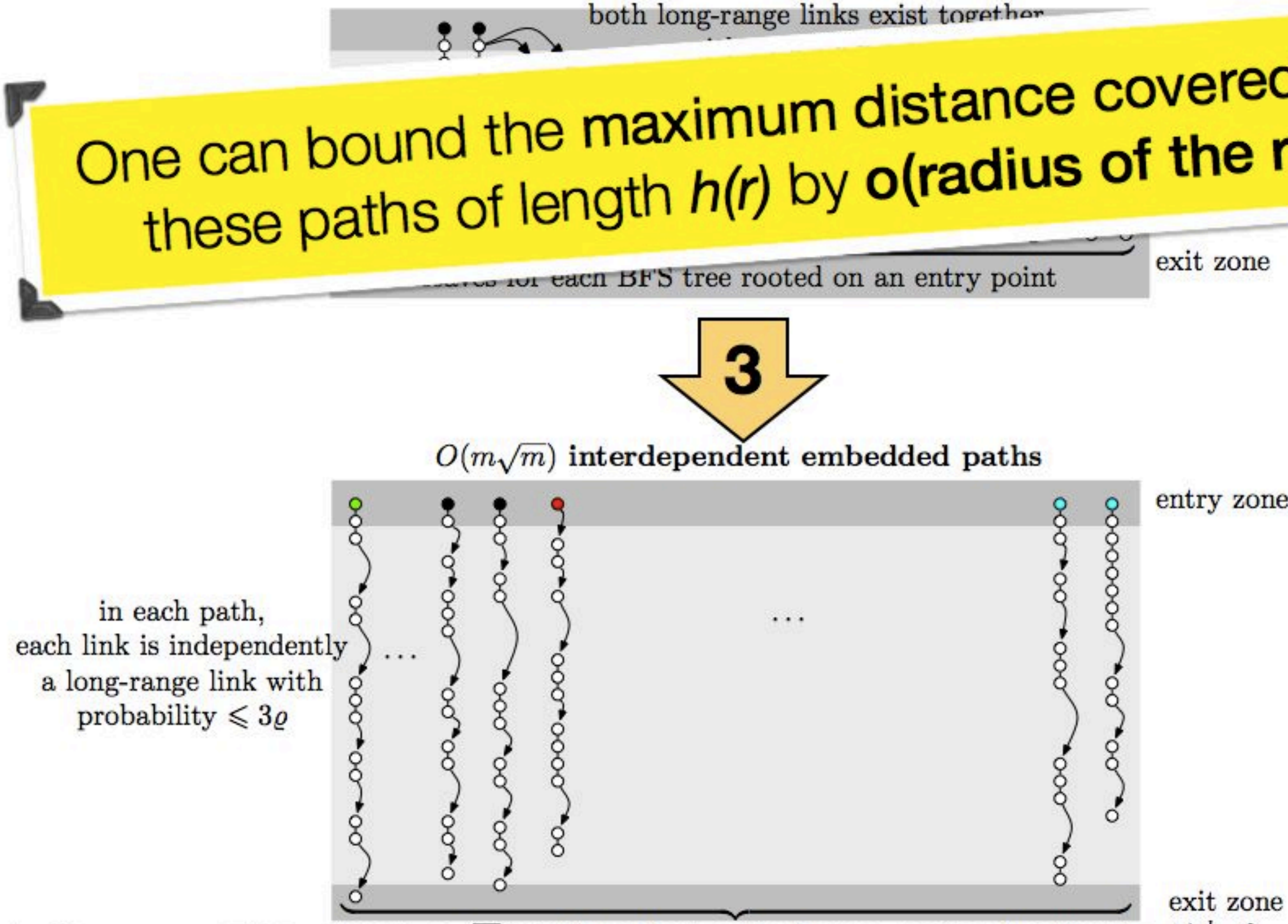


w.h.p., at most $m\sqrt{m}$ interdependent paths from m random locations in $A(\frac{d}{m}, d]$

Todo 2: Geometric issue

Upper bounding the distance covered by the paths

One can bound the **maximum distance covered** by all these paths of length *h(r)* by **o(radius of the ring)**



Lower bound for $d = 1$ for every efficient decentralized search algorithm

- **Theorem.**
  For $d = 1$, for all efficient decentralized search algorithm, the expected length of the computed path is **$\Omega(\log n \log\log n)$ for almost all source-target pairs.**

- **Corollary.**
  The [GS'11] decentralized algorithm is **asymptotically optimal** among search algorithms visiting at most a polylogarithmic number of nodes.

- **Corollary.**
  Every efficient decentralized search algorithm **stays for a little while within each ring before jumping** to the smaller ring**.**

Let's now go back to the other subject:

How to connect these works with **sociology** ?

Towards an analysis of relevant parameters

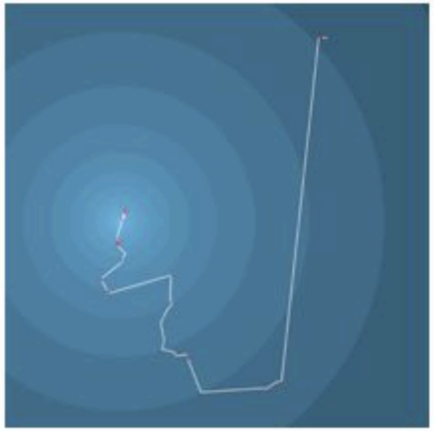
Differentiating the three types of algorithms

Some differences:

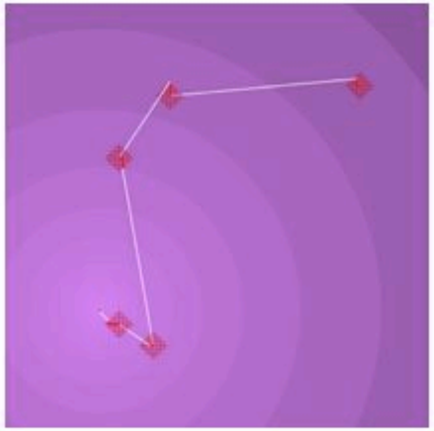
- The **length** of the path varies
- The **rate** and **length** of used long-range links

Some similarities:

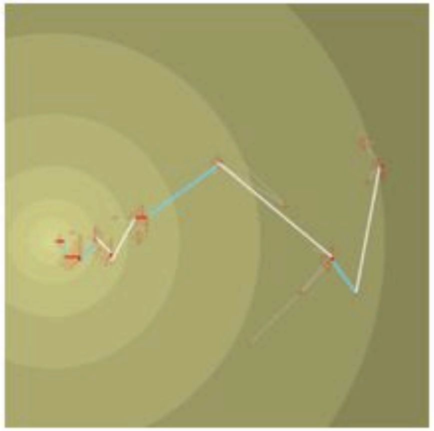
- Significant progresses towards the target are made with **some very long long-range links** which are **spaced from each other**



Greedy algorithm



Algorithms based on local exploration



Algorithms based on non-local exploration

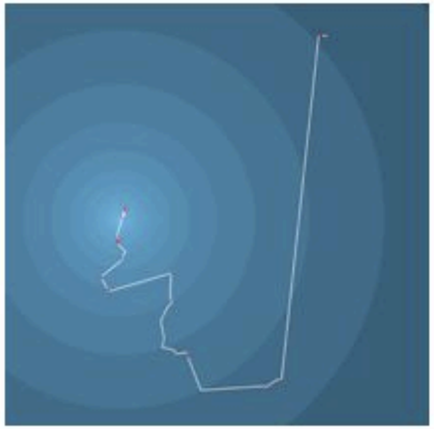
Differentiating the three types of algorithms

Some differences:

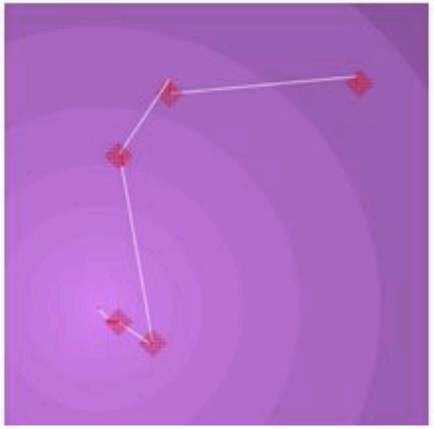
- The **length** of the path varies
- The **rate** and **length** of used long-range links

Relevant differentiating parameters should:

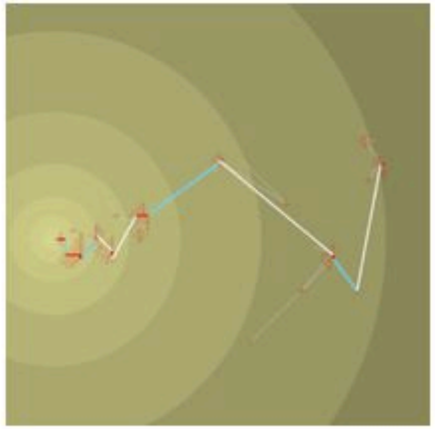
- take **different values** for each algorithms
- be **easily measured** in **real** experiments



Greedy algorithm



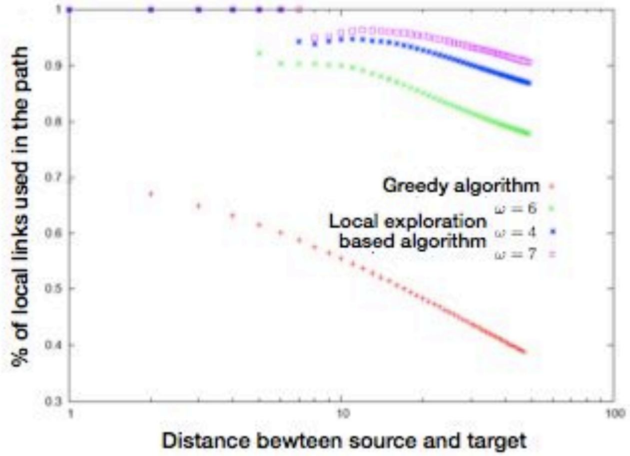
Algorithms based on local exploration



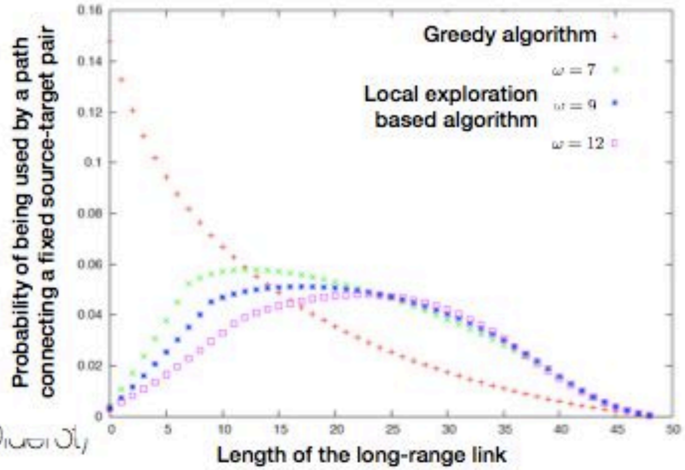
Algorithms based on non-local exploration

Examples of other relevant parameters [LS'05]

- **% of local links** used in the path as a **function of the distance to the target**

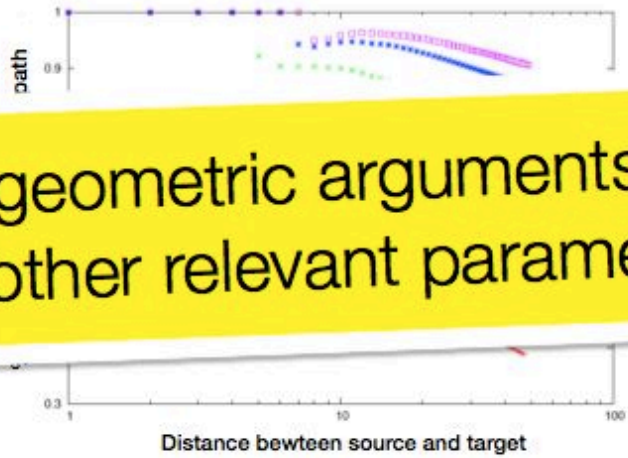


- **Probability to use a long-range link as a function of its length** (the load of a link in peer-to-peer networks)



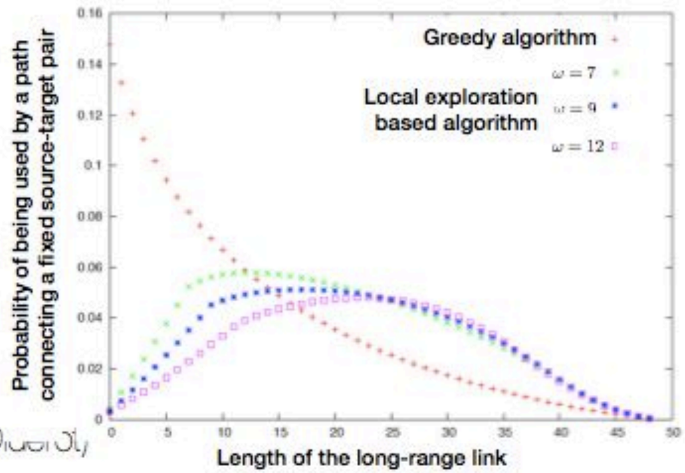
Examples of other relevant parameters [LS'05]

- **% of local links** used in the path as a **function of the distance to the target**



Can we use our geometric arguments to analyze these and other relevant parameters?

- **Probability to use a long-range link as a function of its length** (the load of a link in peer-to-peer networks)



Conclusions & Open questions

✓ Kleinberg's model is now pretty well (fully?) understood:

- when $\alpha \neq d$: decentralized (K. 2000) & diameter (M. & N. 2004)
- when $\alpha = d$: diameter (M. & N. 2004) & decentralized + natural matching algorithms (our result)

✓ A surprising gap between decentralized and centralized search when d=1
(the geometric extra space offered when $d \geq 2$ allows to go around this problem)

✓ Practical consequence: better use 2-dimensional grids than rings for P2P

❀ Human behavior modelization: Which algorithm are used by human? Use our technics to analysis the statistics of the different "path type".

❀ A further gap for $d = 1$ and $d \geq 2$?

- [N.&W. 2004] Every algorithm has to visit $\Omega(\log^2 n)$ nodes
- [L.&S. 2004]'s algorithm visits $O(\log^2 n)$ nodes and gets a path of length $O(\log n \cdot (\log \log n)^2)$
- Our algorithm visits $O(\log^{2+O(\varepsilon)} n)$ nodes and gets a path of length $O(\log n \cdot \log \log n)$
- Conjecture: $O(\log n \cdot (\log \log n)^{2 \text{ or } 1})$ is tight for d=1 or ≥ 2 when visiting $O(\log^2 n)$ nodes.

❀ Does our result extend to other general smallwordization processes?
(such as [Fraigniaud, Giakkoupis, 2010])

& Conclusions Open questions

Thank you
Any question?

✓ **Kleinberg's model is now pretty well (fully?) understood:**

- when $\alpha \neq d$: decentralized (K. 2000) & diameter (M. & N. 2004)
- when $\alpha = d$: diameter (M. & N. 2004) & decentralized + natural matching algorithms (our result)

✓ **A surprising gap** between decentralized and centralized search when d=1
(the geometric extra space offered when $d \geq 2$ allows to go around this problem)

✓ **Practical consequence:** better use 2-dimensional grids than rings for P2P

❀ **Human behavior modelization:** Which algorithm are used by human? Use our technics to analysis the statistics of the different "path type".

❀ **A further gap for d = 1 *and* $d \geq 2$?**

- [N.&W. 2004] Every algorithm has to visit $\Omega(\log^2 n)$ nodes
- [L.&S. 2004]'s algorithm visits $O(\log^2 n)$ nodes and gets a path of length $O(\log n \cdot (\log \log n)^2)$
- Our algorithm visits $O(\log^{2+O(\varepsilon)} n)$ nodes and gets a path of length $O(\log n \cdot \log \log n)$
- <u>Conjecture:</u> $O(\log n \cdot (\log \log n)^{2 \text{ or } 1})$ is tight for d=1 or ≥ 2 when visiting $O(\log^2 n)$ nodes.

❀ **Does our result extend to other general smallwordization processes?**
(such as [Fraigniaud, Giakkoupis, 2010])