COST 295 - 1st WORKSHOP ON DYNAMIC NETWORKS

Università di Salerno – May 7-8 2007

Degree–Optimal Routing for P2P Systems

Giovanni Chiola, Gennaro Cordasco, Luisa Gargano, Mikael Hammar, Alberto Negro, and Vittorio Scarano

Outline

- Peer-to-Peer
  - Distribute Hash Table and Overlay Network
  - Uniform Networks vs. Randomized Networks
  - Greedy Routing vs. Neighbor of Neighbor Routing
- Our Proposals
  - H-Networks
  - Overlay network with class
- Conclusions

Distributed Hash Table (DHT)

- Distributed version of a hash table data structure
- Stores (key, value) pairs
  - The key is like a filename
  - The value can be file contents
- Goal: Efficiently insert/lookup/delete (key, value) pairs
- Each peer stores a subset of (key, value) pairs in the system
- Core operation: Find node responsible for a key
  - Map key to node
  - Efficiently route insert/lookup/delete requests to this node

Overlay Network: Performance

- Routing table size (degree)
  - Storage cost
  - Measure the cost of self-stabilization for adapting to node joins/leaves
- Diameter and Average path length
  - Time cost
  - Fault tolerance
- Congestion
- Clustering

Chord

- Chord uses a one-dimensional circular key space (ring) of $N=2^m$ identifiers
- The node responsible for the key is the node whose identifier most closely follows the key
- Chord maintains two sets of neighbors:
  - A successor list of r nodes that immediately follows it in the key space
  - A finger list of $m = log\ N$ nodes spaced exponentially around the key space
- Routing consists in forwarding to the node closest, but not past, the key
- Performance:
  - Diameter: $log\ N$ ($O(log\ n)$ whp) where n denote the number of nodes present in the network
  - Routing table size: $log\ N$ ($O(log\ n)$ whp)
  - Average path length: ½ $log\ N$

Routing correctness

Routing efficiency

Classification

Pure P2P Networks

- Uniform Systems
  - Chord, CAN, Pastry, Tapestry…
- Non Uniform Systems
  - "Greedy" Routing
    - Randomized Networks and Neighbor of Neighbor Routing
  - Non Greedy Routing
    - Viceroy, De Bruijn graphs

Uniform Networks

- We consider a ring of N identifiers labeled from *0* to N-*1*
- A routing algorithm is *uniform* if for each identifier v, v is connected to u iff $v+w$ is connected to $u+w$ (i.e. : all the connection are symmetric).

Advantages

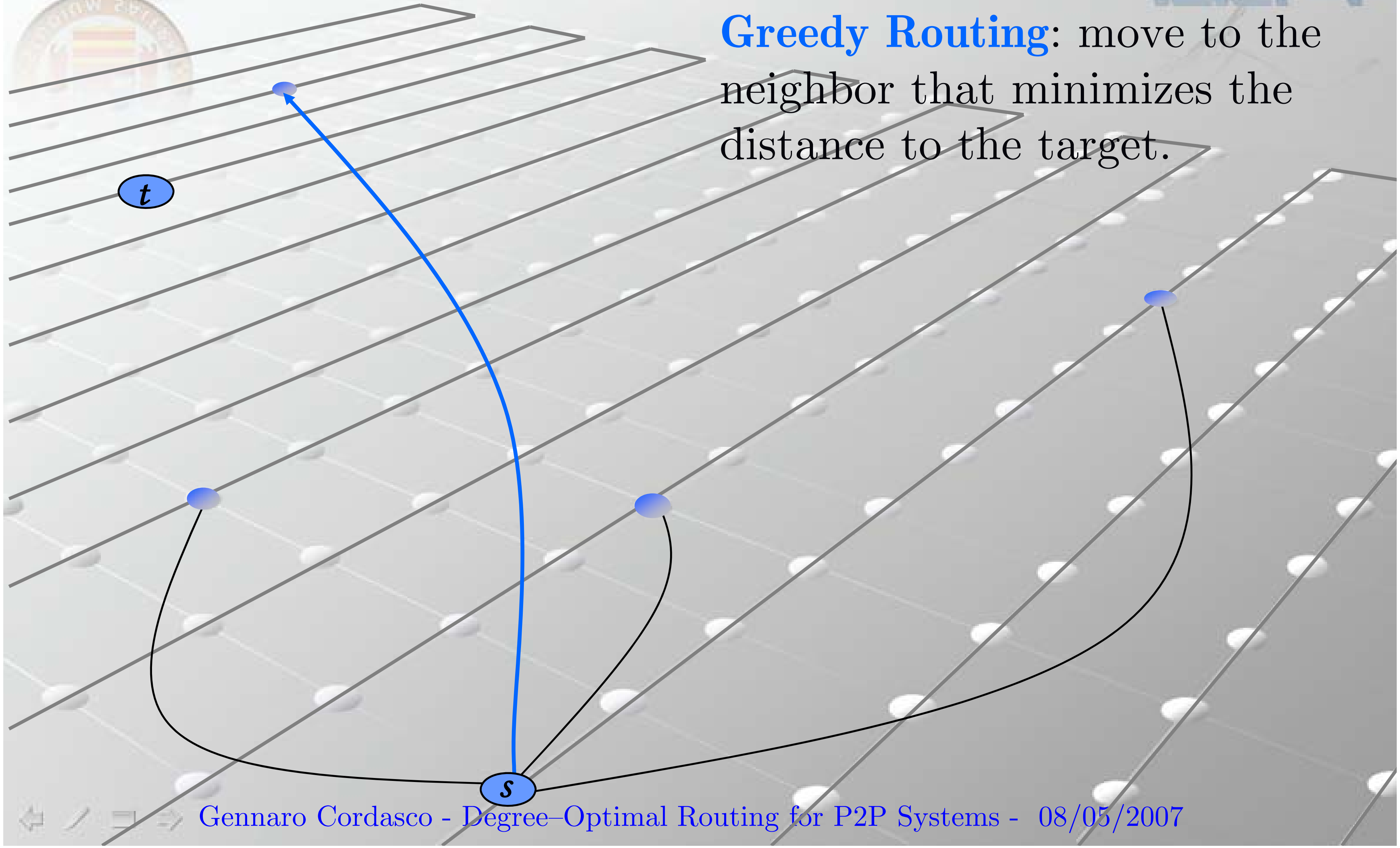
+ Easy to implement
+ Greedy algorithm
  + **Simple** – to understand and implement
  + **Local** – routing occurs inside the portion of ring that is delimited by source and destination
  + In some cases – (Hypercube, Chord) – the best we can do
+ No node congestion
+ Quick Bootstrap
+ High Clustering

Drawback

– Less powerful (De Bruijn Graph and Neighbor of Neighbor Greedy routing are more powerful)

Greedy Routing

Greedy Routing: move to the neighbor that minimizes the distance to the target.



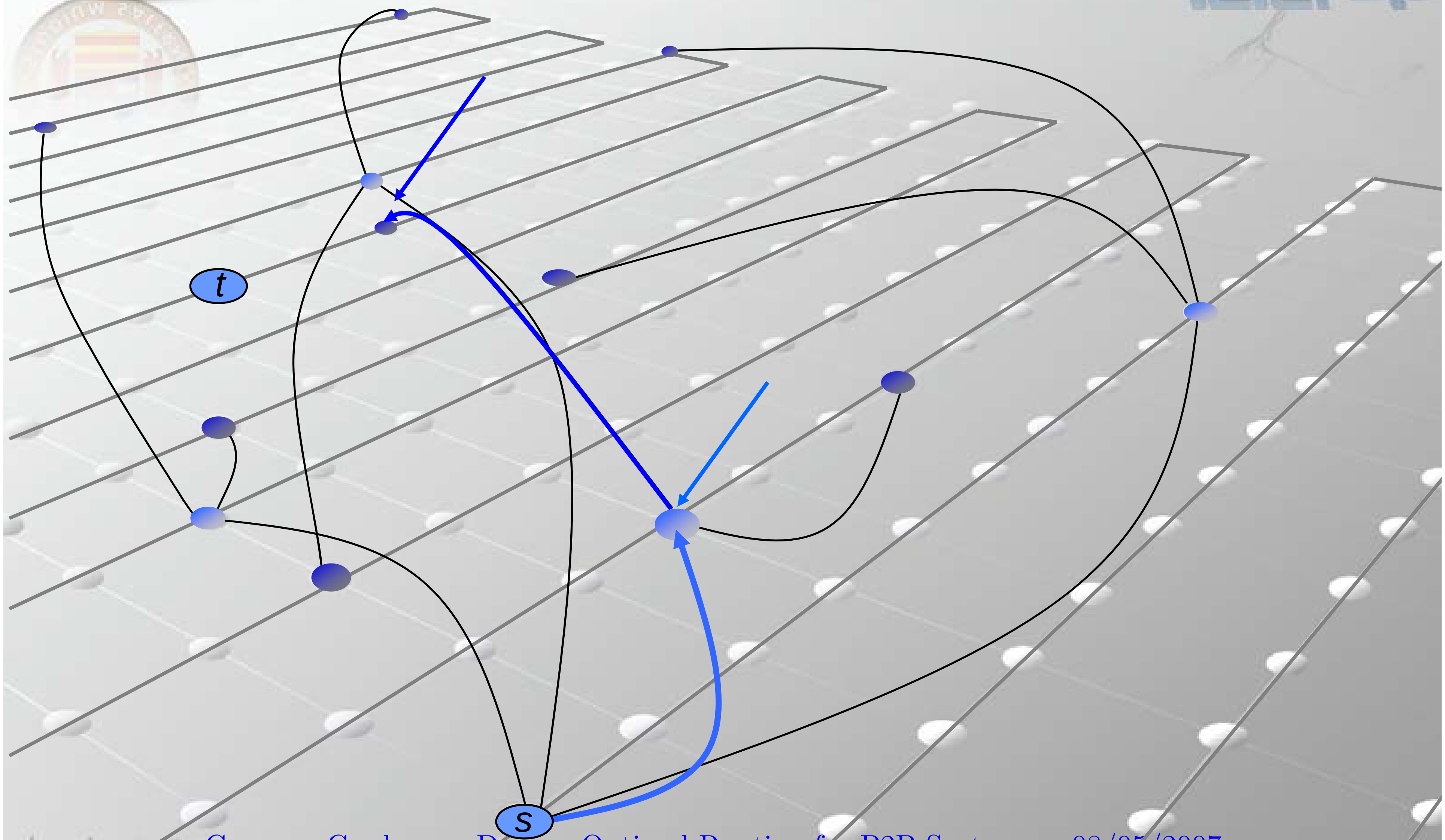
Randomized Networks

Networks whose nodes are connected to each other randomly

+ Small Average Path Length (APL)
– The global structure of the network cannot be a global information
– Routing is not local
– Optimal trade-off can be reached only through a estimation of the number of participants in the system
– The optimal trade-off will only occur for a range of n
– Small Clustering
– No Quick Bootstrap
– Congestion should be analyzed

- Examples:
  - Viceroy Network
    - Degree: *O(1)*
    - APL: *O(log N)*
  - Mariposa Network
    - Degree: $O(\delta)$
    - APL: $O(\log_\delta N)$

NoN Greedy routing

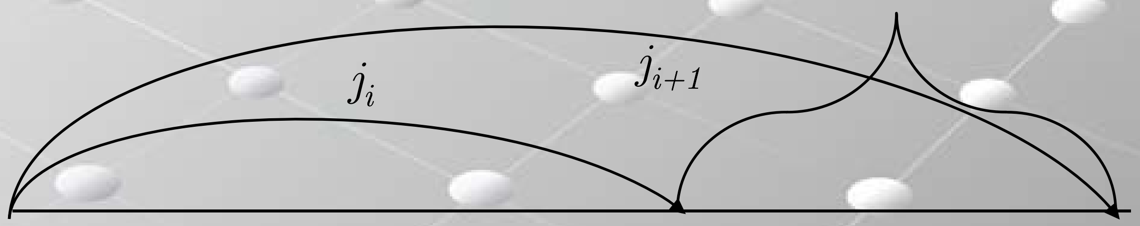


Neighbor of Neighbor Routing

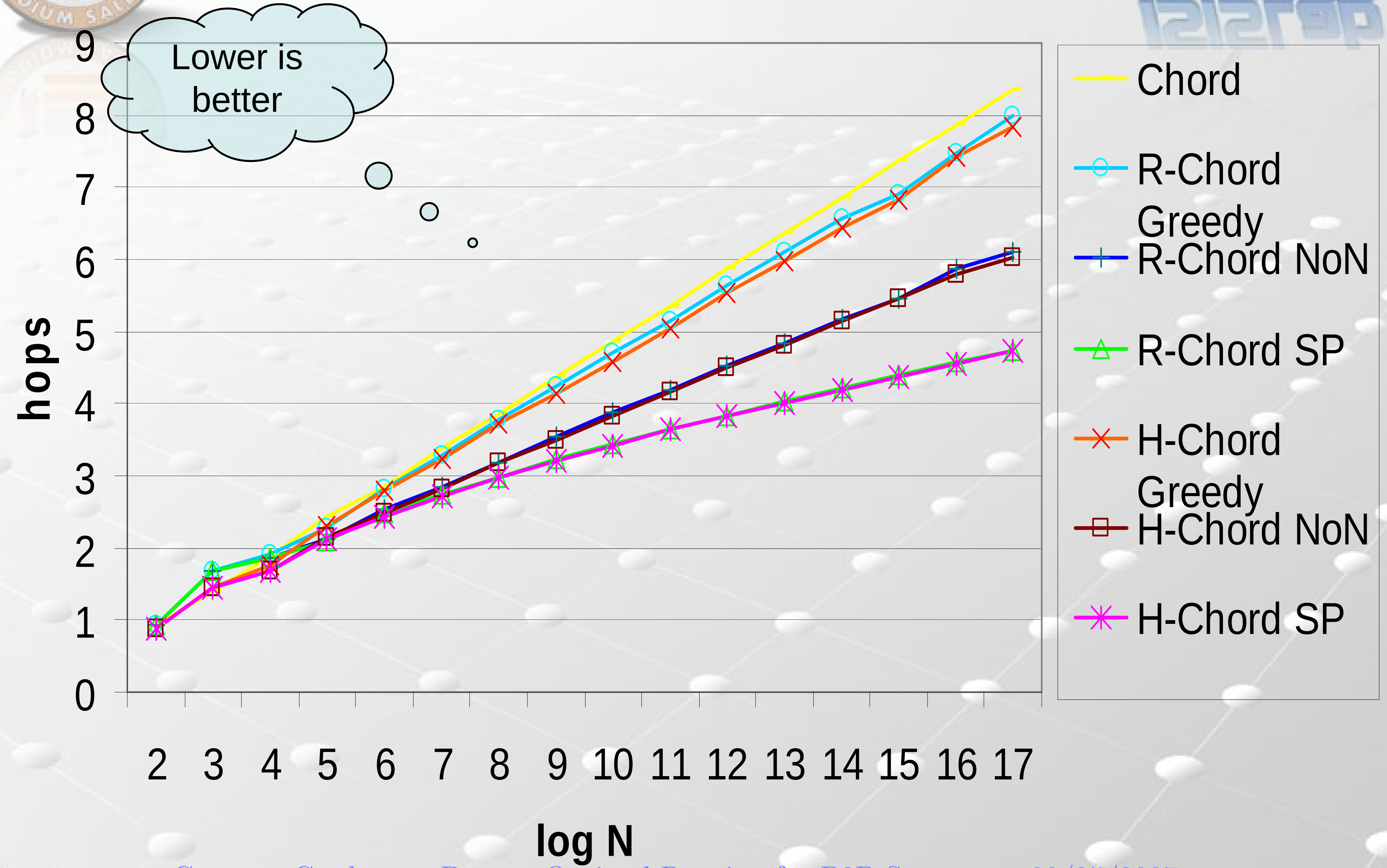
- Benefits
  - Efficient
  - Local Routing
  - Estimation of n not required
- Drawbacks
  - Neighbor of Neighbor lists should be maintained
  - No Quick bootstrap
  - Small Clustering

H-Networks

- Consider a deterministic network U (based on a ring) having n nodes and degree δ
- Let $j_1, j_2, \ldots, j_\delta$ all the jumps of U (ordered by their size). (i.e.: For each node v and for each $i=1,\ldots,\delta$, v is connected with $v+j_i \bmod n$)
- Let $H()$ denote a good hash function, that maps an id on the interval $[0,1)$
- The network H-U is obtained from U as follows:
  - For each $i=1,\ldots,\delta$, node v is connected to the nodes $v+j_i+(j_{i+1}-j_i)H(v) \bmod n$ (where $j_{\delta+1} = n$)



Chord



Lower is better
hops
0
1
2
3
4
5
6
7
8
9
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
log N
Chord
R-Chord Greedy
R-Chord NoN
R-Chord SP
H-Chord Greedy
H-Chord NoN
H-Chord SP

Why to reduce randomization?

- The use of randomization increases the difficulties in the implementation and testing of applications.
- Quick bootstrap
- The smaller is the randomization the higher is the clustering coefficient of the considered network
  - Clustering represents a fundamental feature that a network model, designed to describe complex network, must hold.
  - The resilience of a network grows with the clustering coefficient.
  - An high clustering implies an improved ability to handle heavy traffic workload.

How Much randomization is required to have an efficient Network?

Overlay Network with class

- Split nodes into a c classes
  - Routing tables are uniform within the same class
  - Different classes follow different patterns
- Each peer chooses its class pseudo-randomly (based on Hash digest of its identifier)
- Quick bootstrap is obtained when finding a neighbor belonging to the same class

Overlay Network with class

- Let U be uniform overlay network (based on a ring) having n nodes and degree δ.
- Let $j_1, j_2, \ldots, j_\delta$ all the jumps of U (ordered by their size).
- Let c be any given positive integer in the interval $[1, 2^m]$ and $H()$ be a cryptographic hash function that maps an id on the interval $[0, 1)$.
- Consider c real numbers $\lambda_0, \lambda_1, \ldots, \lambda_{c-1}$ in the interval $[0, 1)$ such that $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{c-1} < 1$
- The network H_c-U is obtained from U as follows:
  - For each $i = 0, \ldots, \delta - 1$, node v is connected by an edge to the nodes

$$v + \lfloor j_i + \lambda_{c_v}(j_{i+1} - j_i) \rfloor,$$

where $c_v = \lfloor cH(v) \rfloor$.

H_c-Chord

- *Theorem*

  Let $c > 1$, the average path length is $O(\log_c n + \log n / \log\log n)$ hops for the NoN Greedy algorithm on H_c-Chord

- *Corollary*

  When $c = \log n$, the average path length is $O(\log n / \log\log n)$ hops and the number of hops for the completion of the join operation are $O(\log n \; \log\log n)$ (w.h.p.)

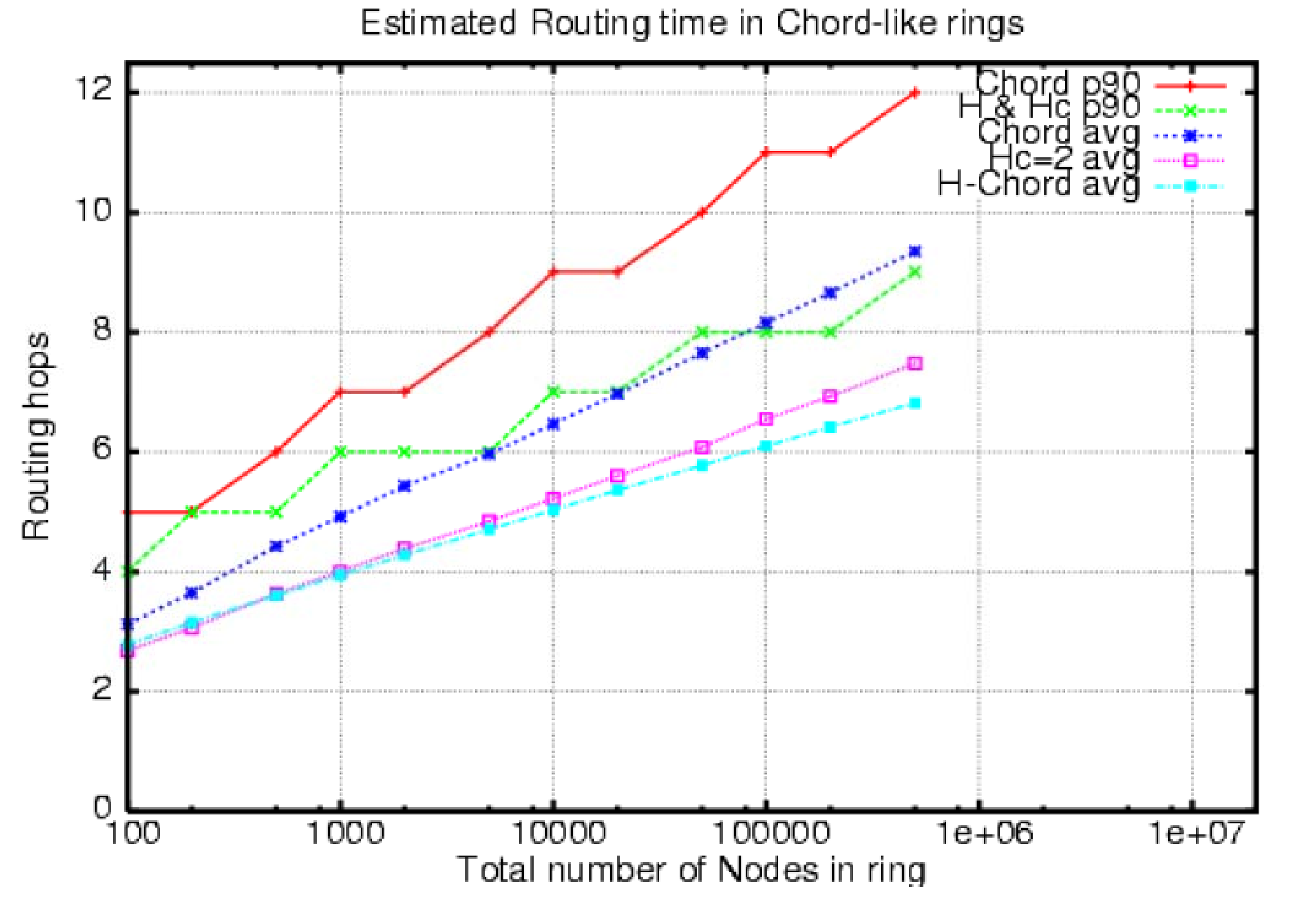
H_c-Chord

- *Theorem*

  Both the diameter and the average (shortest) path length of an H_c-Network with degree $O(\log n)$ and $c > 1$ classes are $\Omega(\log_c n)$

H_c-Chord is asymptotically optimal

Simulation Results



Overlay Network with class

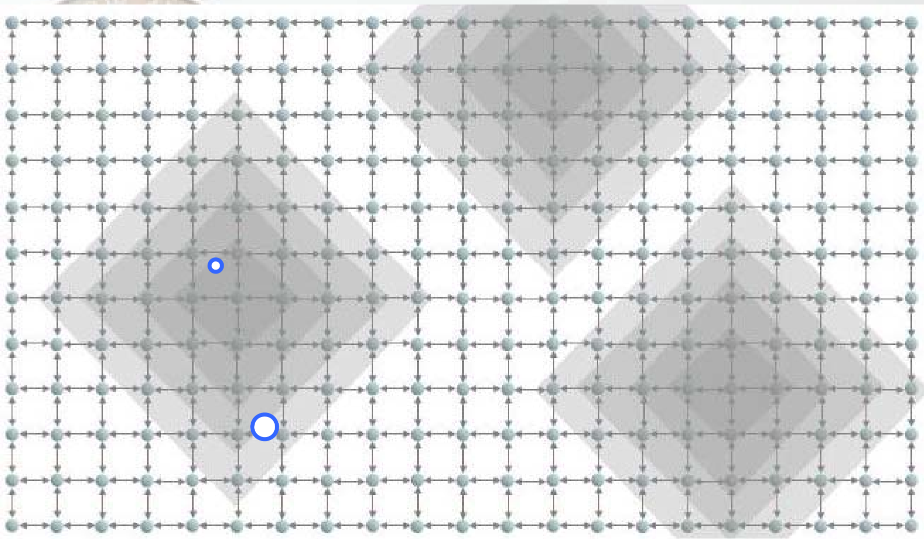
- H_c-Networks:
  - NoN routing on few Classes gives almost the same performance as NoN routing on truly randomized systems
  - Few Classes guarantee a quick bootstrap with very high probability
  - 2 Classes are sufficient for DHTs with up to hundred thousand peers
  - 4 Classes are sufficient for DHTs with billions of peers

Overlay Network with class

- Extension to other networks:
  - Symphony -> H_c-Symphony
  - Hypercube -> H_c-Hypercube
  - Small-World -> …

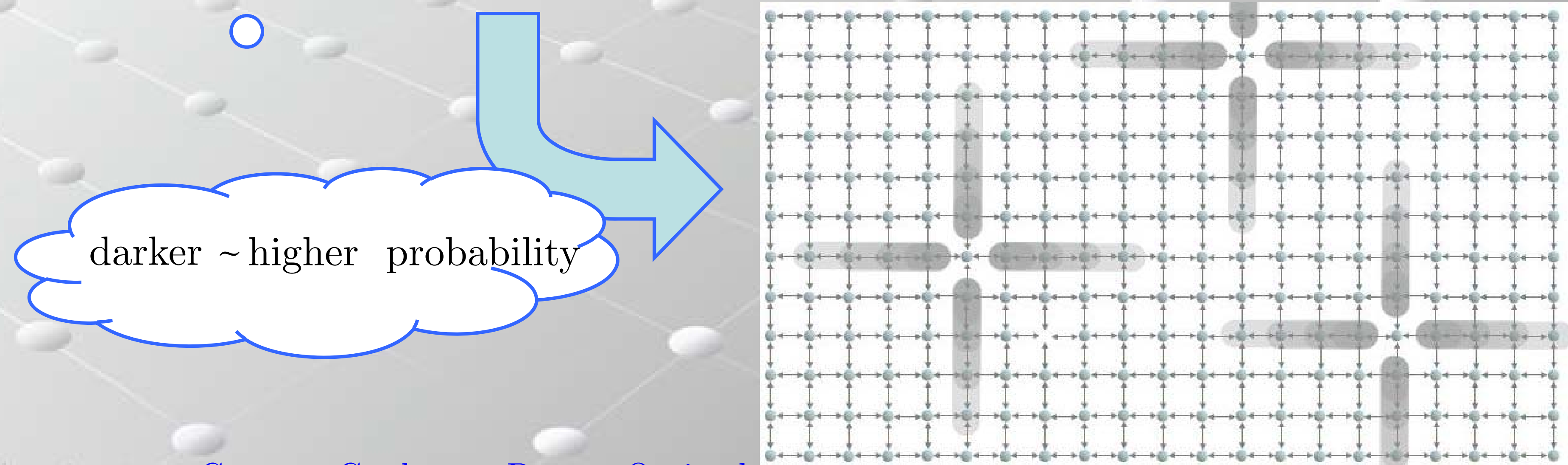
How Much Independent Should Individual Contacts be to Form a Small-World?

Our Proposals

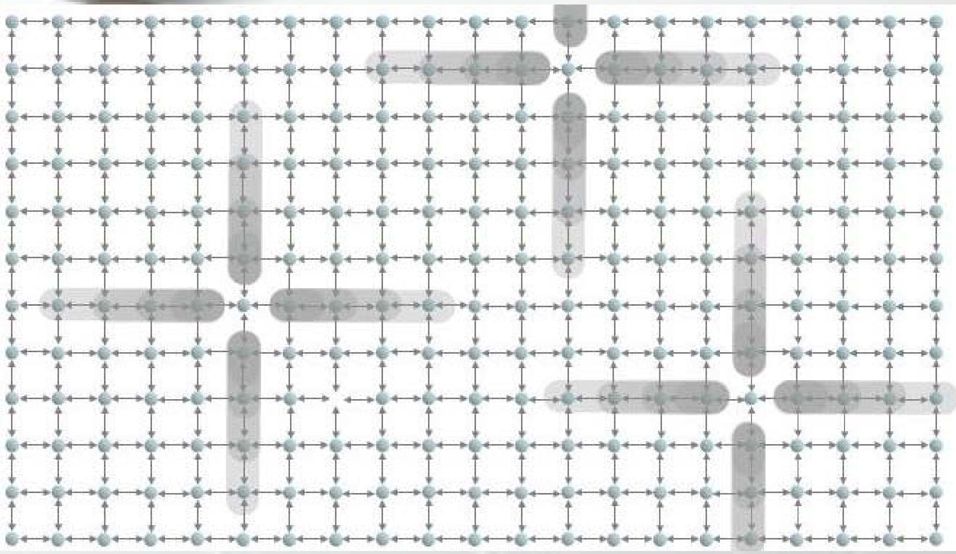


Restricted Small World:

Long-range connections are allowed only with nodes that differ in exactly one coordinate



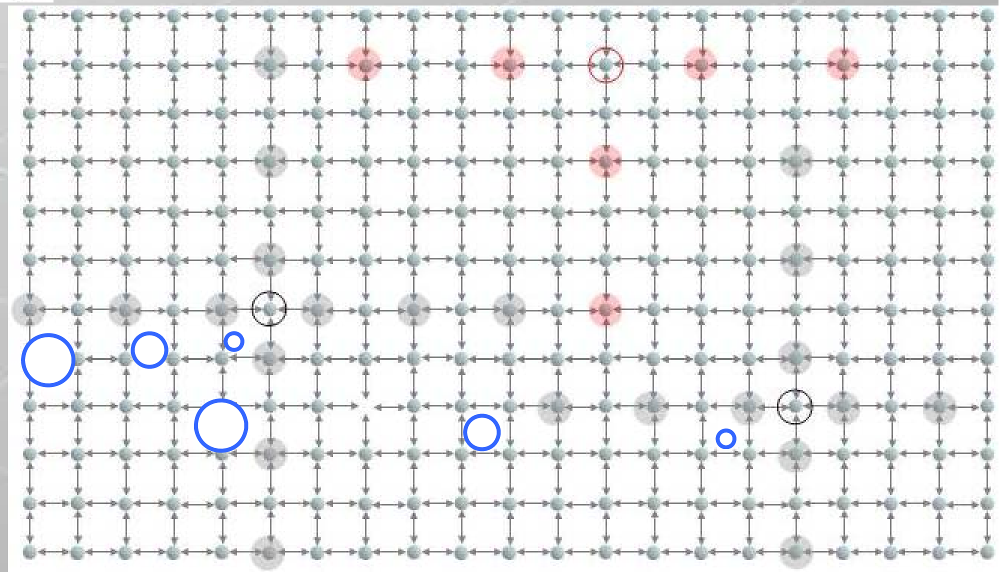
Our Proposals



Small World with communities:

Each node randomly chooses one of the communities to belong to and selects its long-range contacts only among a subset of nodes depending on the chosen community.

Same community means same long-range distances



Conclusions

- We showed that it is not necessary to use a completely eclectic network in order to obtain an efficient network.
- Our networks presents "almost" all the advantages of uniform networks (high clustering, no congestion, quick bootstrap).
- The proposed families are parameterized with a positive integer c which measures the amount of randomization injected in the network
  - $c = 1$ *Uniform network*
  - $c = \log n$ *"Almost uniform" network*
  - $c >> n$ *Randomized network*

Thanks for your attention

Questions?