Improving Lookup Latency in Distributed Hash Table systems Using Random Sampling

Presented by 曾胤燁

2005/11/11

IEEE/ACM TRANSACTIONS ON NETWORKING,

VOL. 13, NO. 5, OCTOBER 2005

Outline

- Introduction
  - DHT
  - Chord
- Lookup-Parasitic Random Sampling(LPRS)
- Simulation
- Conclusion

Distributed hash table (DHT)

- Provide routing infrastructures for scalable information storage and retrieval.
- Support scalable and distributed storage and retrieval of (key, data) pairs on the overlay network.
- Do this by associating each node in the network with a portion of the key space.

- In most DHT systems, a request will take θ (logN) overlay hops in average

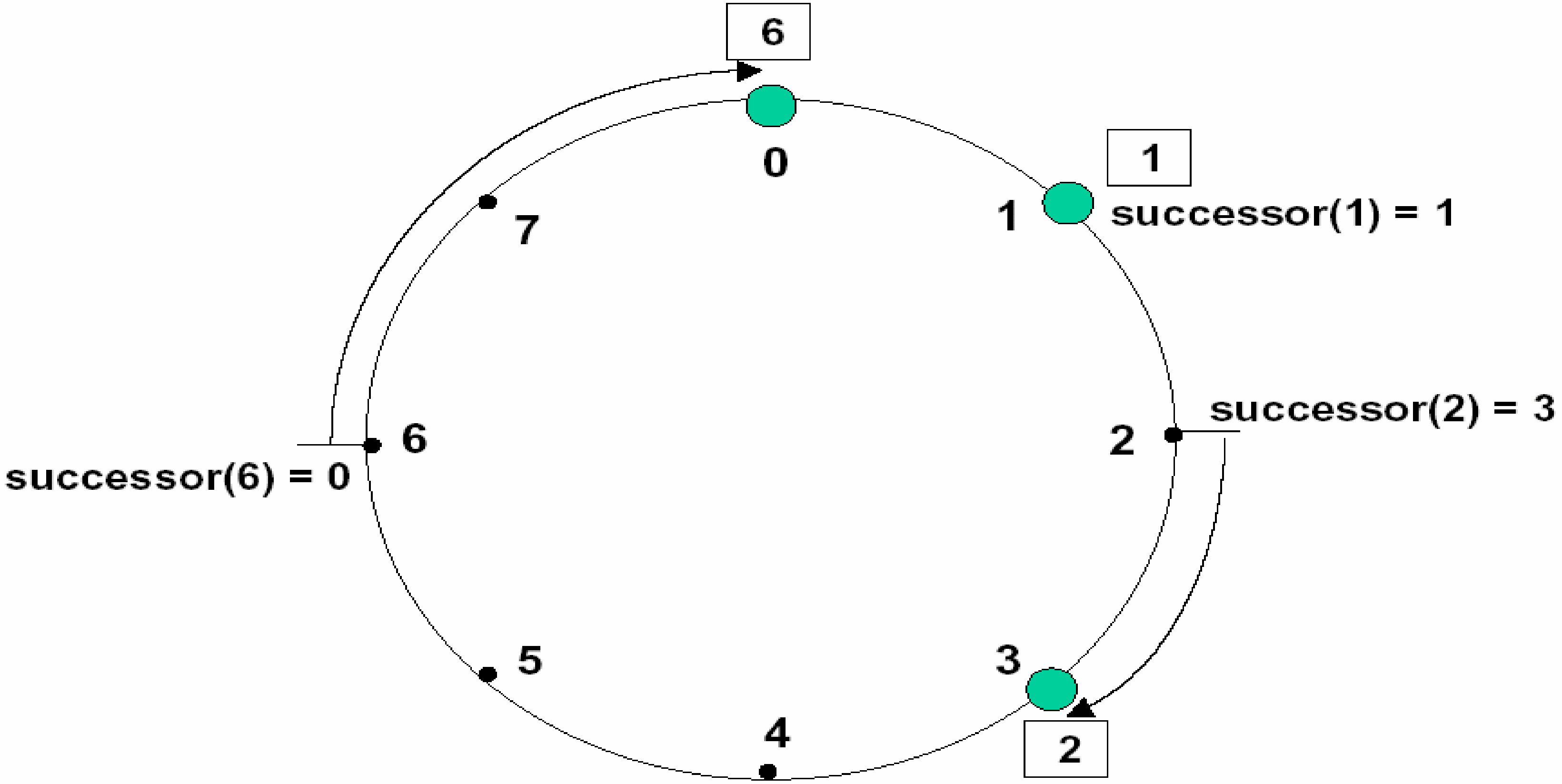
(N：the network size)

- A DHT network might have a large latency if the overlay network topology is not congruent with the underlying IP topology.

poor lookup performance

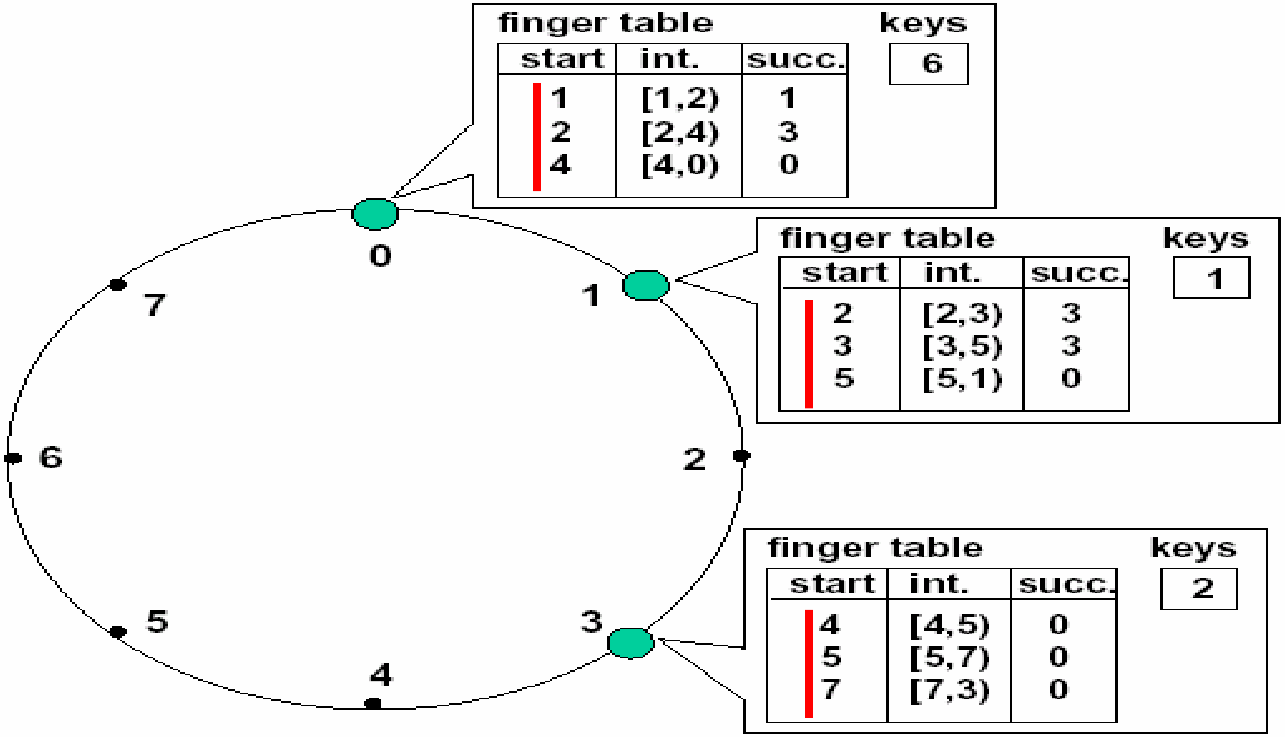
Chord

- Chord assigns each overlay node in the network an m-bit identifier (NodeID).
- Chord uses consistent hashing to assign keys to nodes.
- Each key is assigned to that node in the overlay whose node ID is equal to the key identifier, or follows it in the *key space* That node is called the *successor* of the key.


6
0
1
1
successor(1) = 1
7
successor(2) = 3
2
6
successor(6) = 0
5
3
4
2

- Each node, n, maintains a finger table with m entries.
- The i^{th} entry contains the identity of the first node, s, that succeeds n by at least 2^{i-1} on the identifier circle.

S= successor($n+2^{i-1}$)



finger table — keys: 6

start	int.	succ.
1	[1,2)	1
2	[2,4)	3
4	[4,0)	0

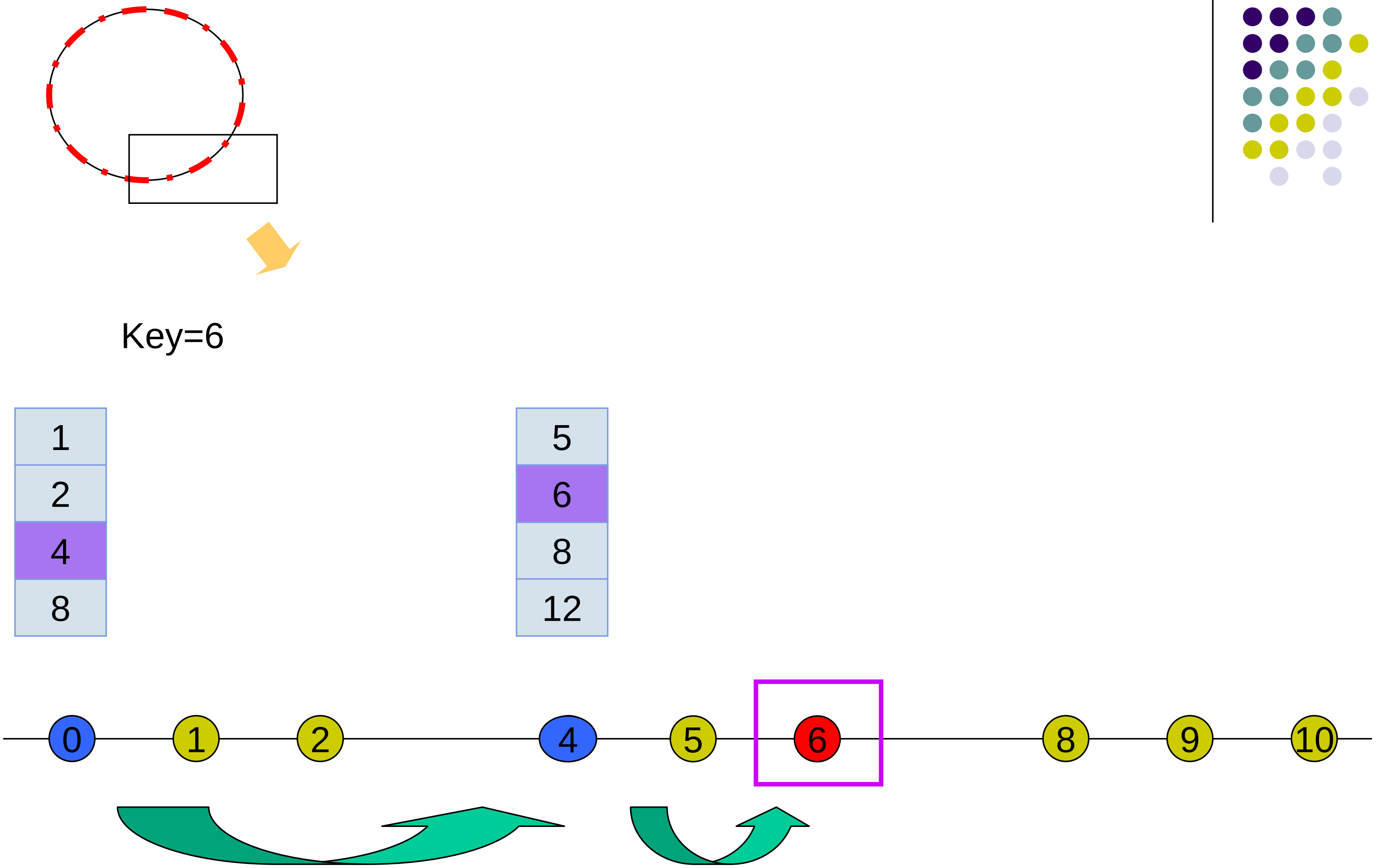
finger table — keys: 1

start	int.	succ.
2	[2,3)	3
3	[3,5)	3
5	[5,1)	0

finger table — keys: 2

start	int.	succ.
4	[4,5)	0
5	[5,7)	0
7	[7,3)	0

(b)



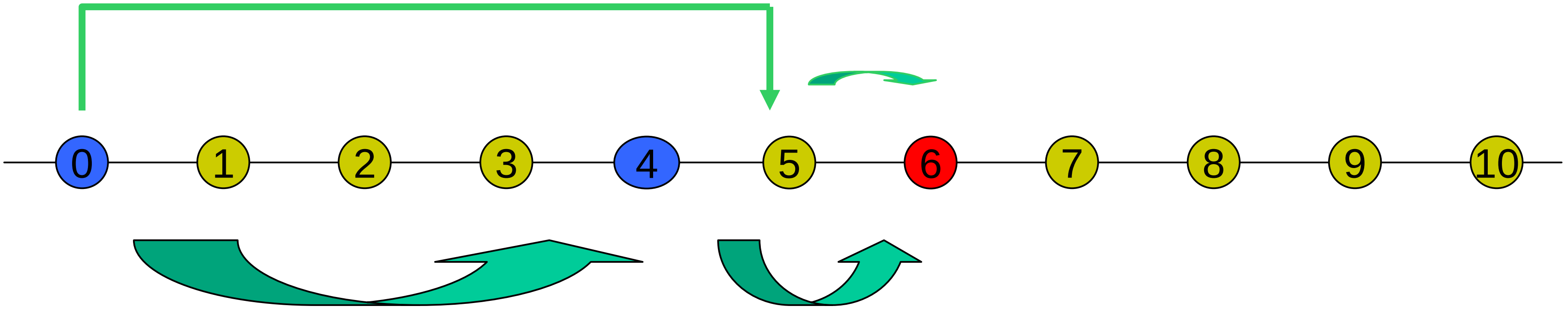
Key=6

1
2
4
8

5
6
8
12

- 若latency(0,4) > latency(0,5)

1
2
5
8



better?

- Each node must have a pointer to a good (i.e., low-latency)node in each geometrically decreasing range of the key-space.

- Simple way: to randomly sample a small number of nodes from this range.

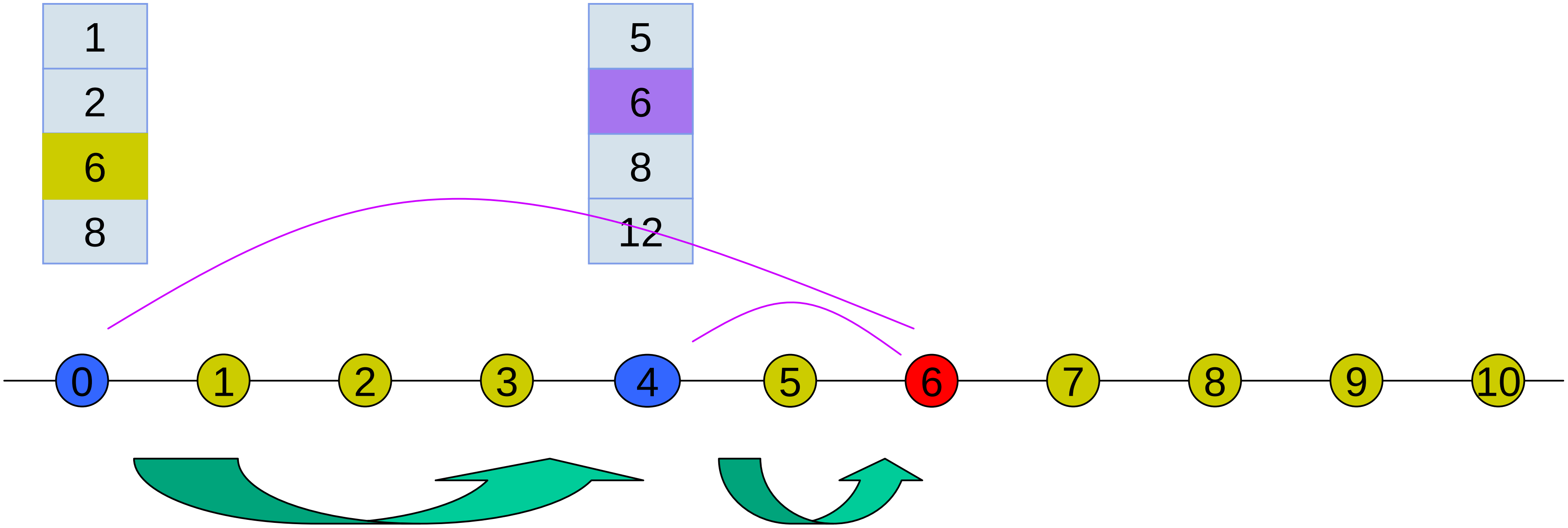
Lookup-Parasitic Random Sampling

- Range: For a node j, its ith range is the interval of the key space defined as $[j+2^{i-1}, j+2^{i}]$
- Sampling: When we say node x samples node y, we mean that x measures the latency (e.g., by using ping)to y.

LPRS

- When a request completes, each node on the request path samples the target, and updates its finger table accordingly.

If(latency(0,4) > latency(0,6))



1
2
6
8

5
6
8
12

Modification

- Each intermediate hop appends its IP address to a lookup message.
- A latency-sensitive replacement scheme to update its finger table entries based on the samples obtained.

Algo.

(1) A node n maintains one finger table entry for each of its m ranges.

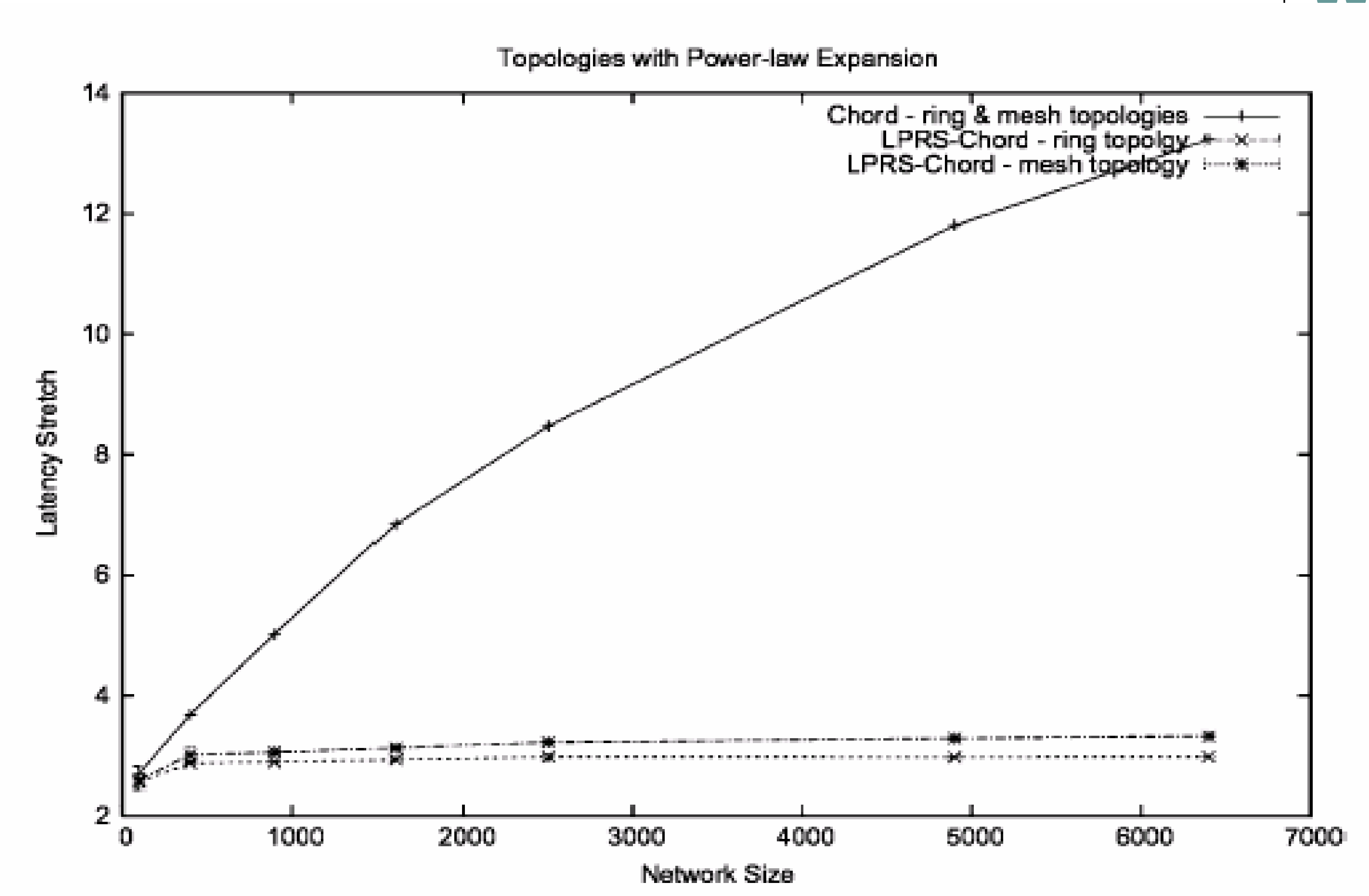
(2) For node p, n finds the range k it lies in.

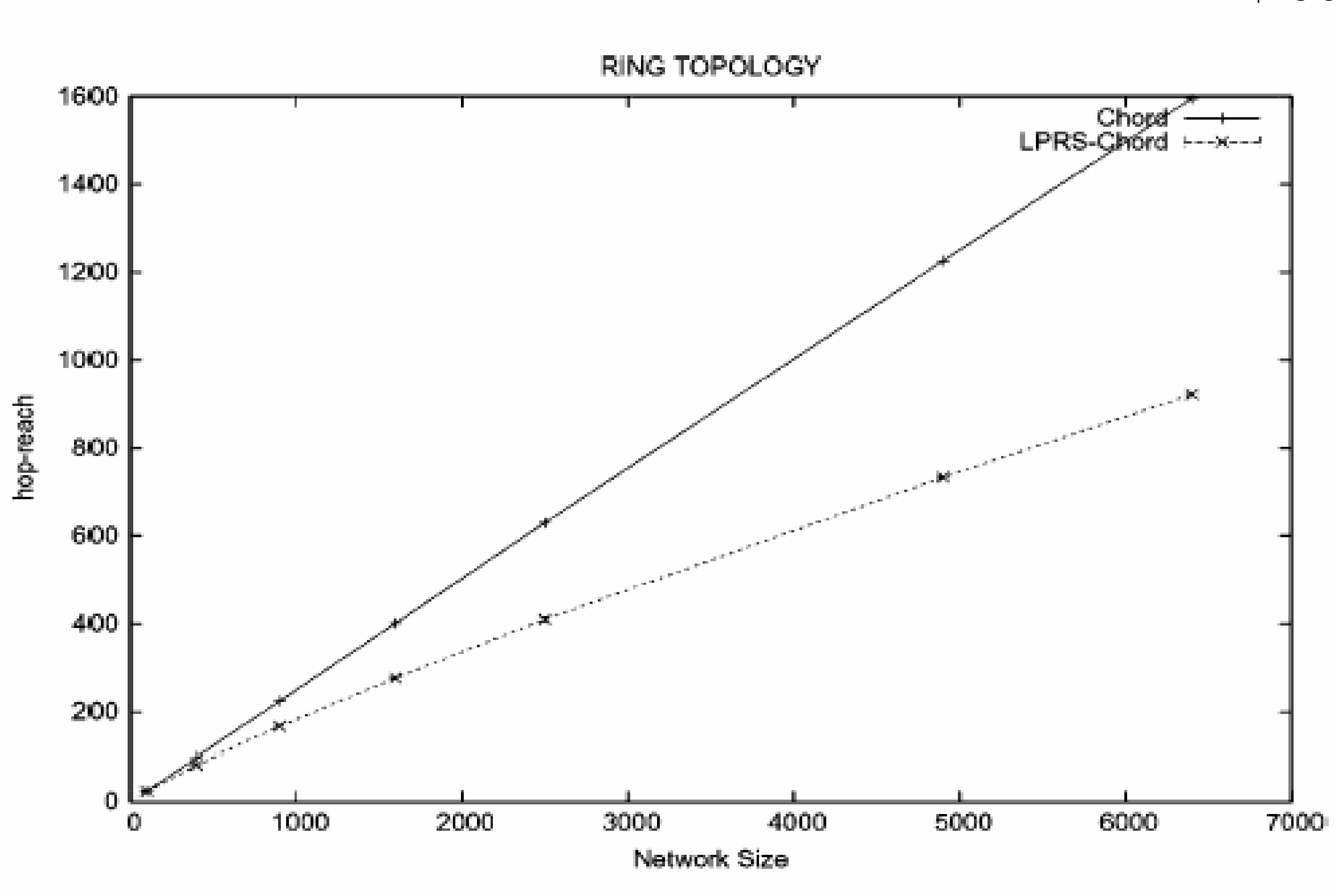
(3) If p is closer than the current kth successor for node n, make p the new kth successor of n .

Simulation

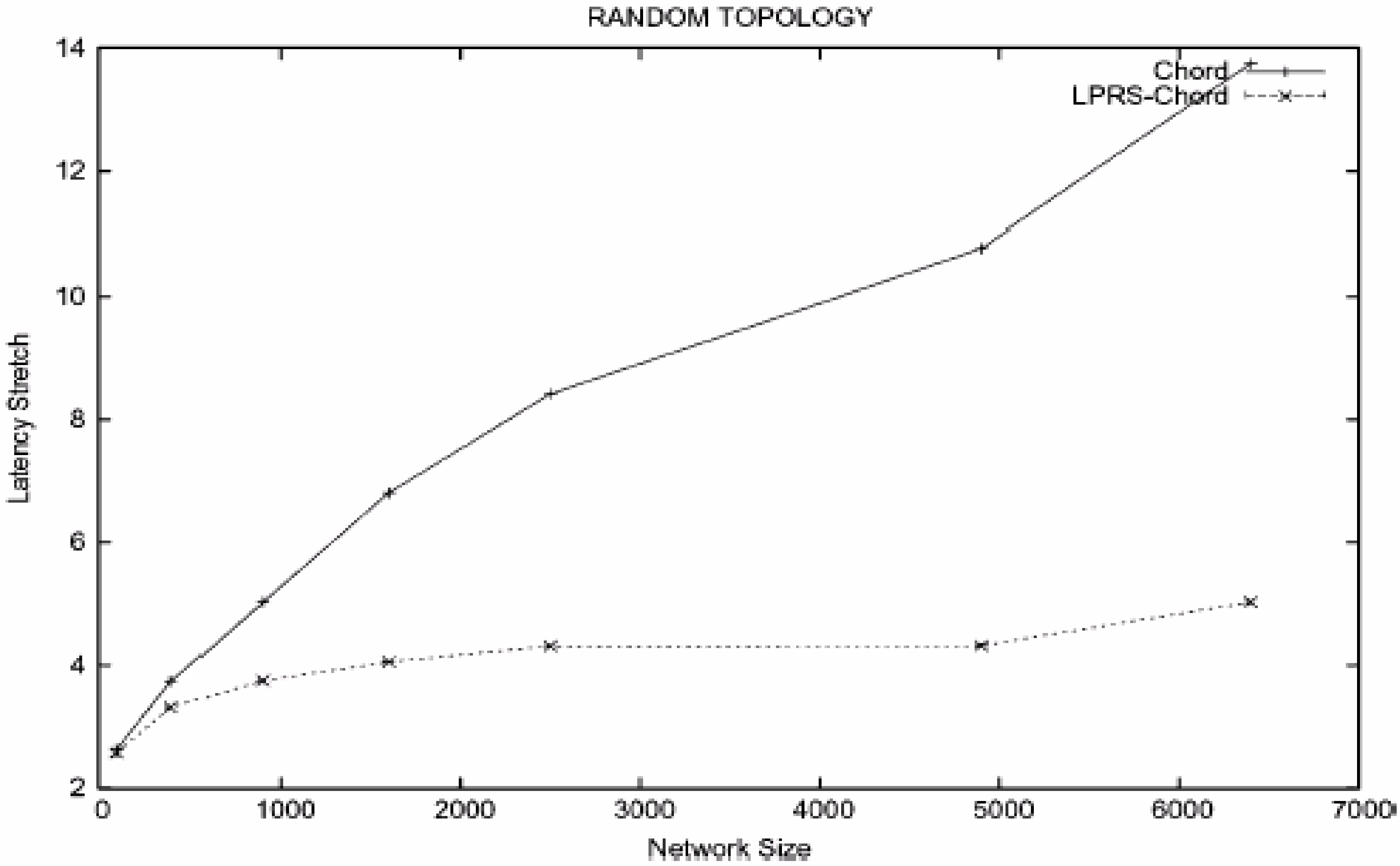
- First phase: N nodes join the network.

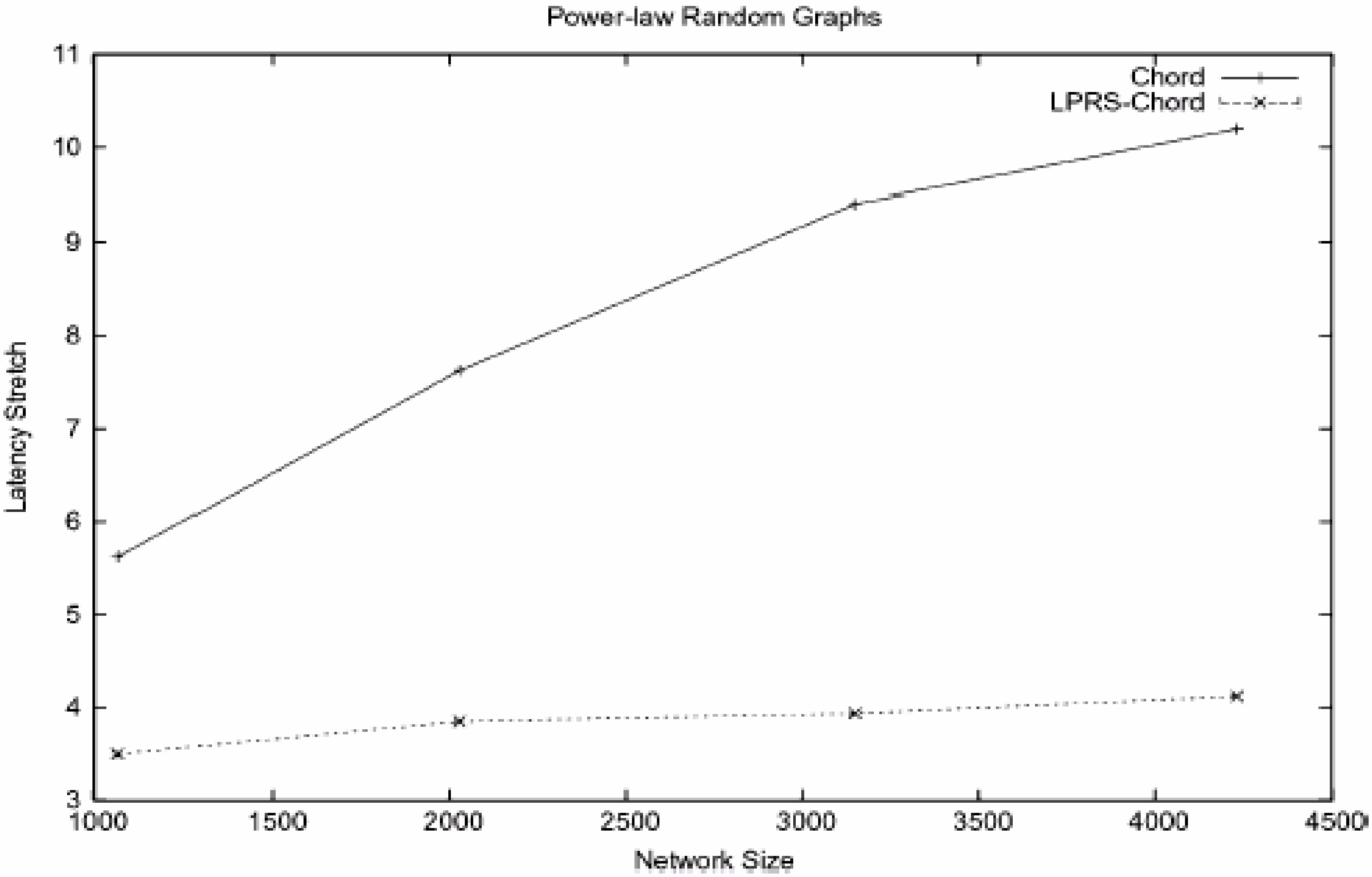
  (We use the Chord join algorithms to build the finger tables.)
- Second phase: Nodes insert four documents into the network.
- Third phase: Each node generates, on average 3logN request.

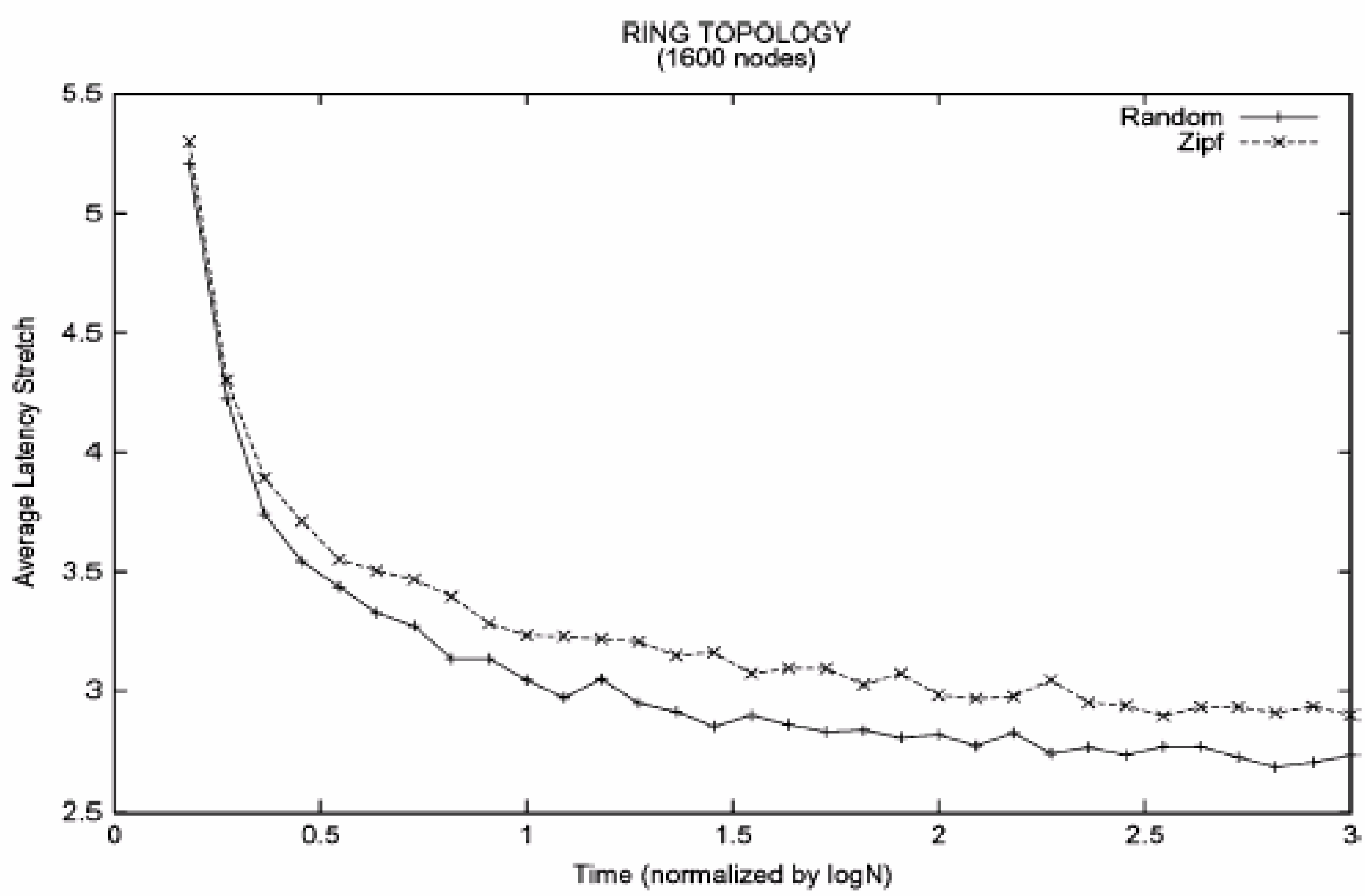

Topologies with Power-law Expansion
Latency Stretch
Network Size
14
12
10
8
6
4
2
0
1000
2000
3000
4000
5000
6000
7000
Chord - ring & mesh topologies
LPRS-Chord - ring topolgy
LPRS-Chord - mesh topology



Hop-reach: the average latency on the underlying network incurred by each hop on the overlay network.


RANDOM TOPOLOGY
Chord
LPRS-Chord
Latency Stretch
2
4
6
8
10
12
14
0
1000
2000
3000
4000
5000
6000
7000
Network Size


Power-law Random Graphs
Chord
LPRS-Chord
Latency Stretch
Network Size
11
10
9
8
7
6
5
4
3
1000
1500
2000
2500
3000
3500
4000
4500


RING TOPOLOGY
(1600 nodes)
Average Latency Stretch
Time (normalized by logN)
Random
Zipf
5.5
5
4.5
4
3.5
3
2.5
0
0.5
1
1.5
2
2.5
3

conclusion

- This paper describes LPRS, a fast random sampling technique

- Simulation based comparison with Chord establish the performance advantages of LPRS.

Thank you☺