oNe
eiburg

IIF
INSTITUT FÜR
INFORMATIK
FREIBURG

Peer-to-Peer Networks

05 Pastry

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

UNI
FREIBURG

- Peter Druschel
  - Rice University, Houston, Texas
  - now head of Max-Planck-Institute for Computer Science, Saarbrücken/ Kaiserslautern
- Antony Rowstron
  - Microsoft Research, Cambridge, GB
- Developed in Cambridge (Microsoft Research)
- Pastry
  - Scalable, decentralized object location and routing for large scale peer-to-peer-network
- PAST
  - A large-scale, persistent peer-to-peer storage utility
- Two names one P2P network
  - PAST is an application for Pastry enabling the full P2P data storage functionality
  - We concentrate on Pastry

+ Plaxton, Rajaraman, Richa |

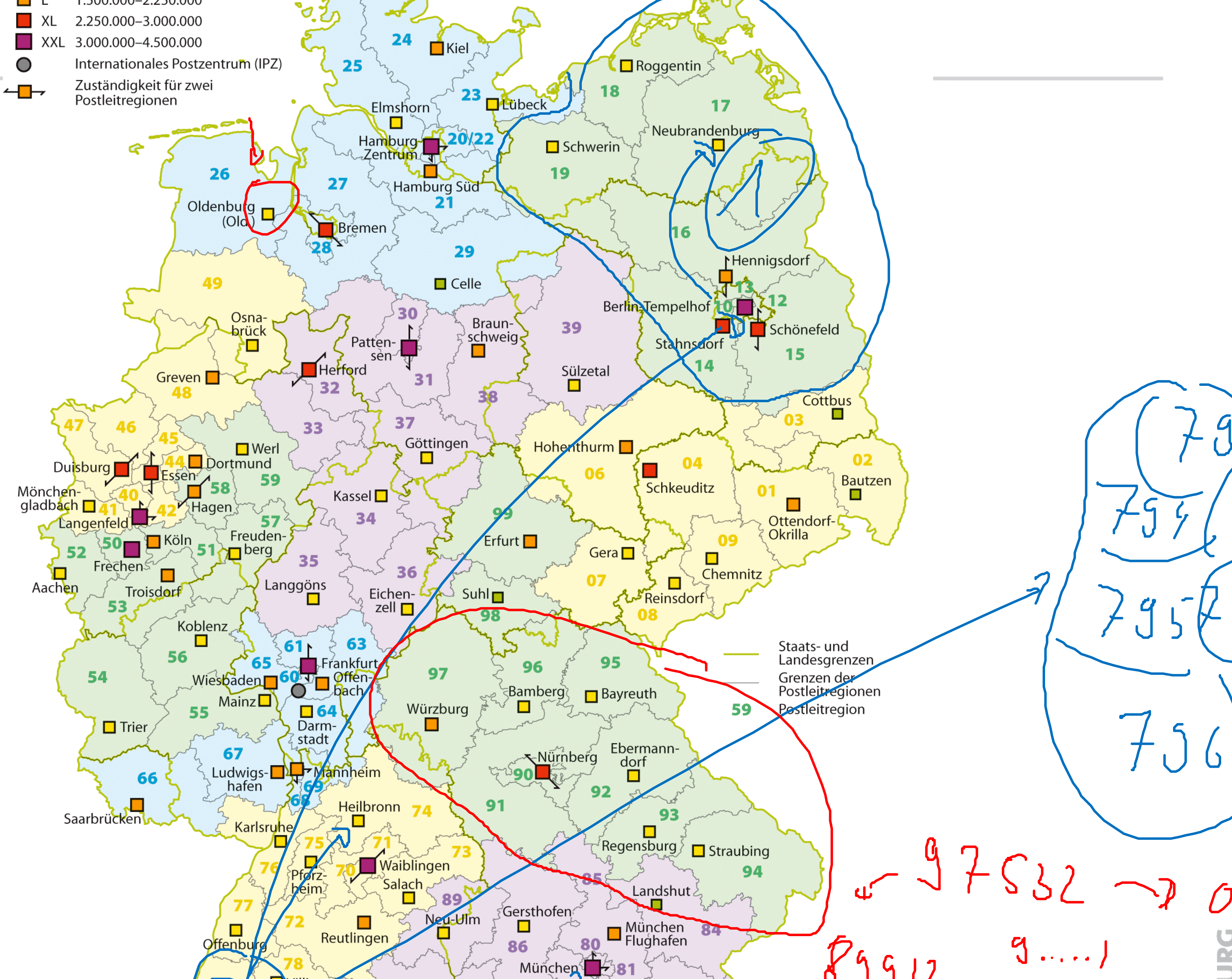
↳ „Plaxton-Routing"

Pastry, Tapestry

ZIP-codes

PLZ

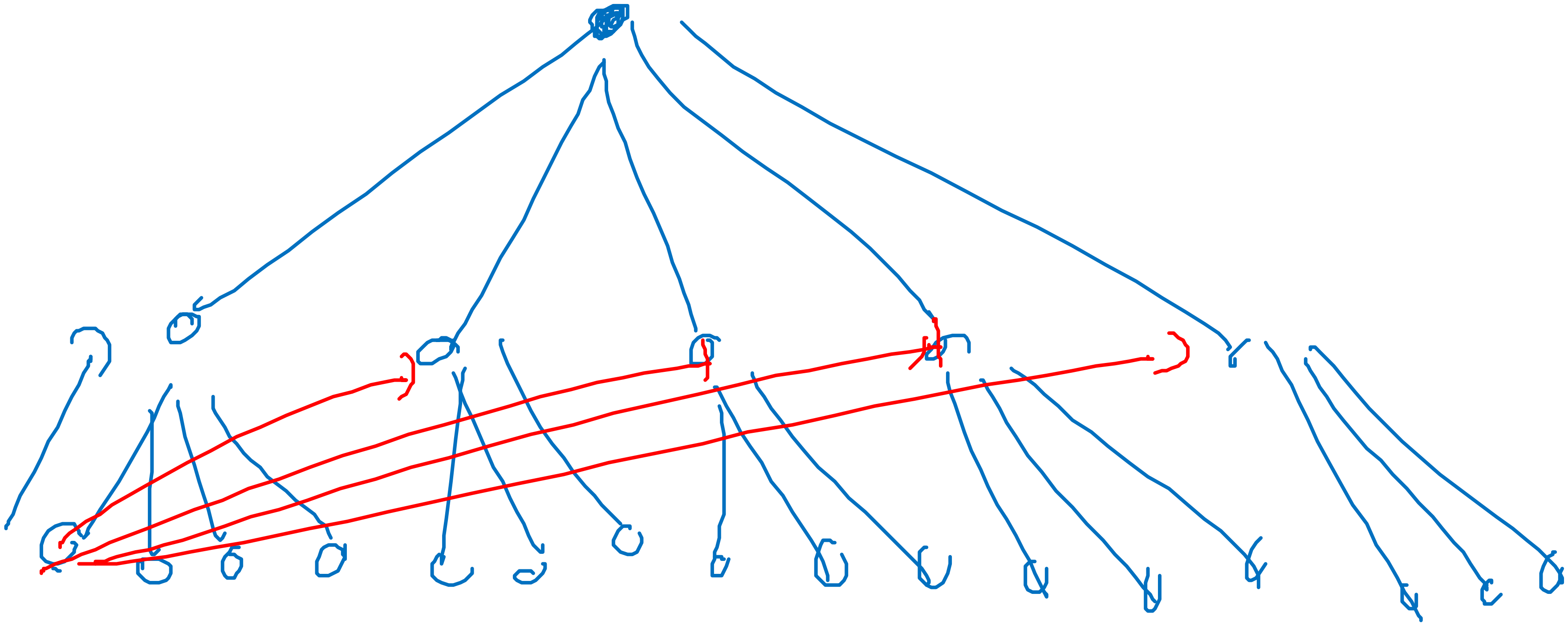
iburg

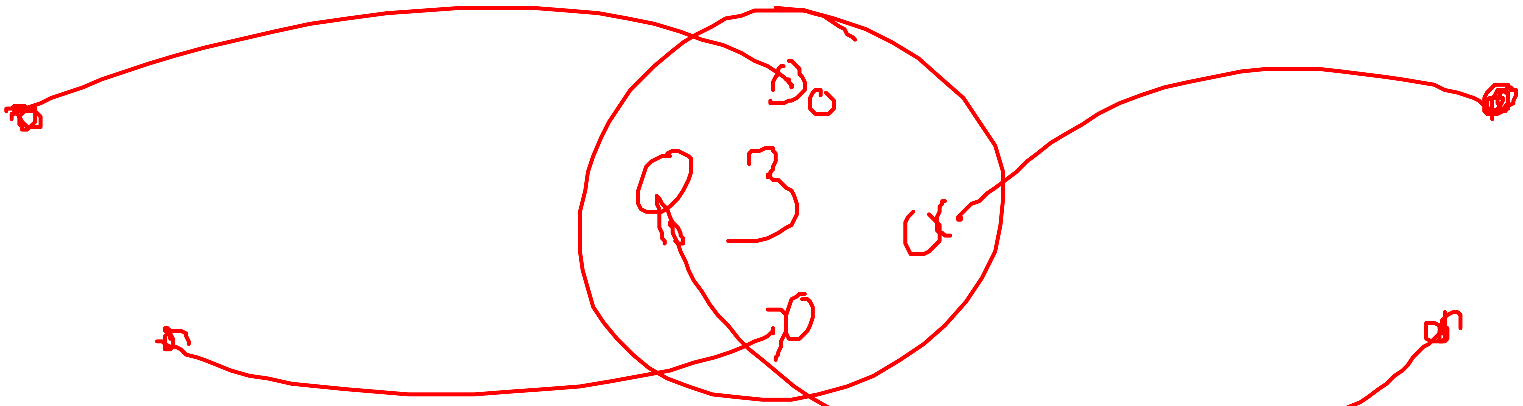


L 1.500.000–2.250.000
XL 2.250.000–3.000.000
XXL 3.000.000–4.500.000
Internationales Postzentrum (IPZ)
Zuständigkeit für zwei Postleitregionen
24
Kiel
25
23
Lübeck
Elmshorn
Hamburg Zentrum
20/22
Hamburg Süd
21
26
27
Oldenburg (Old)
Bremen
28
29
Celle
Roggentin
18
17
Neubrandenburg
Schwerin
19
16
Hennigsdorf
13
Berlin-Tempelhof
10
12
Schönefeld
Stahnsdorf
14
15
49
Osnabrück
30
Pattensen
31
Braunschweig
39
Herford
32
Greven
48
Sülzetal
38
33
37
Göttingen
47
46
45
Werl
Dortmund
44
59
Duisburg
Essen
58
Hagen
Mönchengladbach
40
41
42
Langenfeld
57
Kassel
34
Freudenberg
Köln
50
51
52
Frechen
Aachen
Troisdorf
53
35
Langgöns
36
Eichenzell
Cottbus
03
Hohenthurm
06
04
Schkeuditz
02
Bautzen
01
Ottendorf-Okrilla
99
Erfurt
Gera
09
Chemnitz
07
Reinsdorf
08
Suhl
98
Koblenz
56
54
61
63
Frankfurt
65
Offenbach
Wiesbaden
60
Mainz
64
Darmstadt
55
Trier
67
Ludwigshafen
Mannheim
69
68
66
Saarbrücken
97
96
95
Bamberg
Bayreuth
Würzburg
Nürnberg
90
Ebermanndorf
92
91
93
Regensburg
Straubing
94
Heilbronn
74
Karlsruhe
75
71
73
Waiblingen
76
Pforzheim
70
Salach
89
77
72
Reutlingen
Neu-Ulm
Gersthofen
86
85
Landshut
München Flughafen
84
80
München
81
Offenburg
78
Staats- und Landesgrenzen
Grenzen der Postleitregionen
59 Postleitregion
79
795
7957
756
97532
9....1

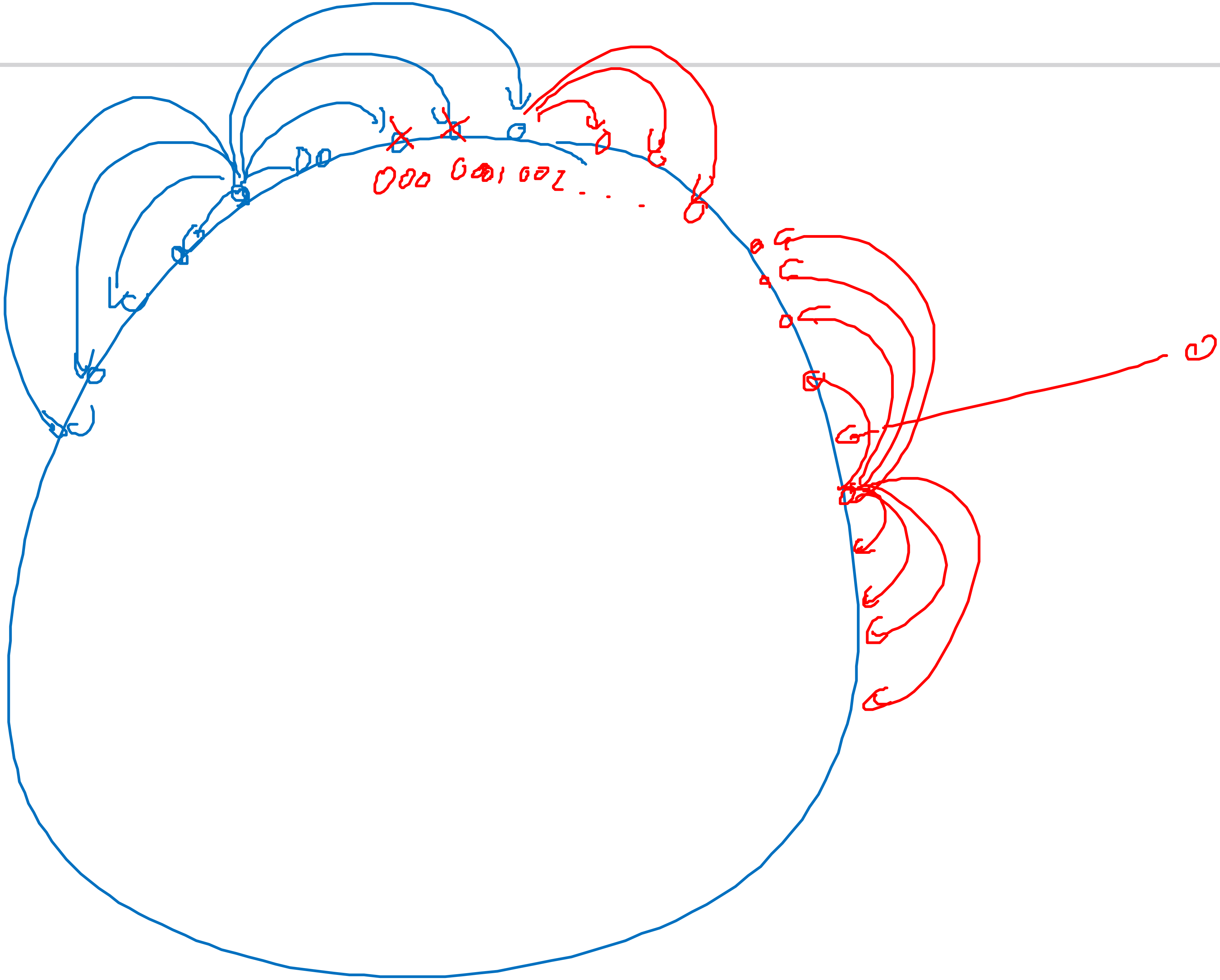
iburg

WRONG.





URG



- Each peer has a 128-bit ID: nodeID

$\frac{1}{2^{128}} \cdot n$ $\frac{1}{2^{128}}$

  - unique and uniformly distributed
  - e.g. use cryptographic function applied to IP-address
- Routing
  - Keys are matched to $\{0,1\}^{128}$
  - According to a metric messages are distributed to the neighbor next to the target
- Routing table has $O(2^b(\log n)/b) + \ell$ entries

79123

AB035

  - n: number of peers
  - ℓ: configuration parameter
  - b: word length
    - typical: b= 4 (base 16), $\ell = 16$
    - message delivery is guaranteed as long as less than $\ell/2$ neighbored peers fail
- Inserting a peer and finding a key needs $O((\log n)/b)$ messages

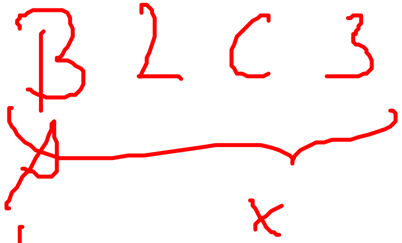
~~65A0BA14~~

70000000

- NodeId presented in base 2^b
  - e.g. NodeID: 65A0BA13
- For each prefix p and letter $x \in \{0,..,2^b-1\}$ add an peer of form px* to the routing table of NodeID, e.g.
  - b=4, 2^b=16
  - 15 entries for 0*,1*, .. F*
  - 15 entries for 60*, 61*,... 6F*
  - ...
  - if no peer of the form exists, then the entry remains empty
- Choose next neighbor according to a distance metric
  - metric results from the RTT (round trip time)
- In addition choose ℓ neighbors
  - $\ell/2$ with next higher ID
  - $\ell/2$ with next lower ID

0x	1x	2x	3x	4x	5x	6	7x	8x	9x	ax	bx	cx	dx	ex
60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex
650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex
65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex

Union Bound



$$\underbrace{(2^b)\cdot(2^b)\cdots\cdots(2^b)}_{x}$$

$$\frac{n}{(2^b)^x} \le \frac{1}{n^c}$$

$$2^{b\cdot x} \ge n^{c+1}$$

$$b\cdot x \ge (\log_2 n)\cdot(c+1)$$

$$x \ge \frac{1}{b}\cdot(\log_2 n)\cdot(c+1)$$

Base 4 = 2 = 2

Example b=2

Routing Table

- For each prefix p and letter x ∈ {0,..,2^b-1} add an peer of form px* to the routing table of NodeID

In addition choose ℓ neighbors

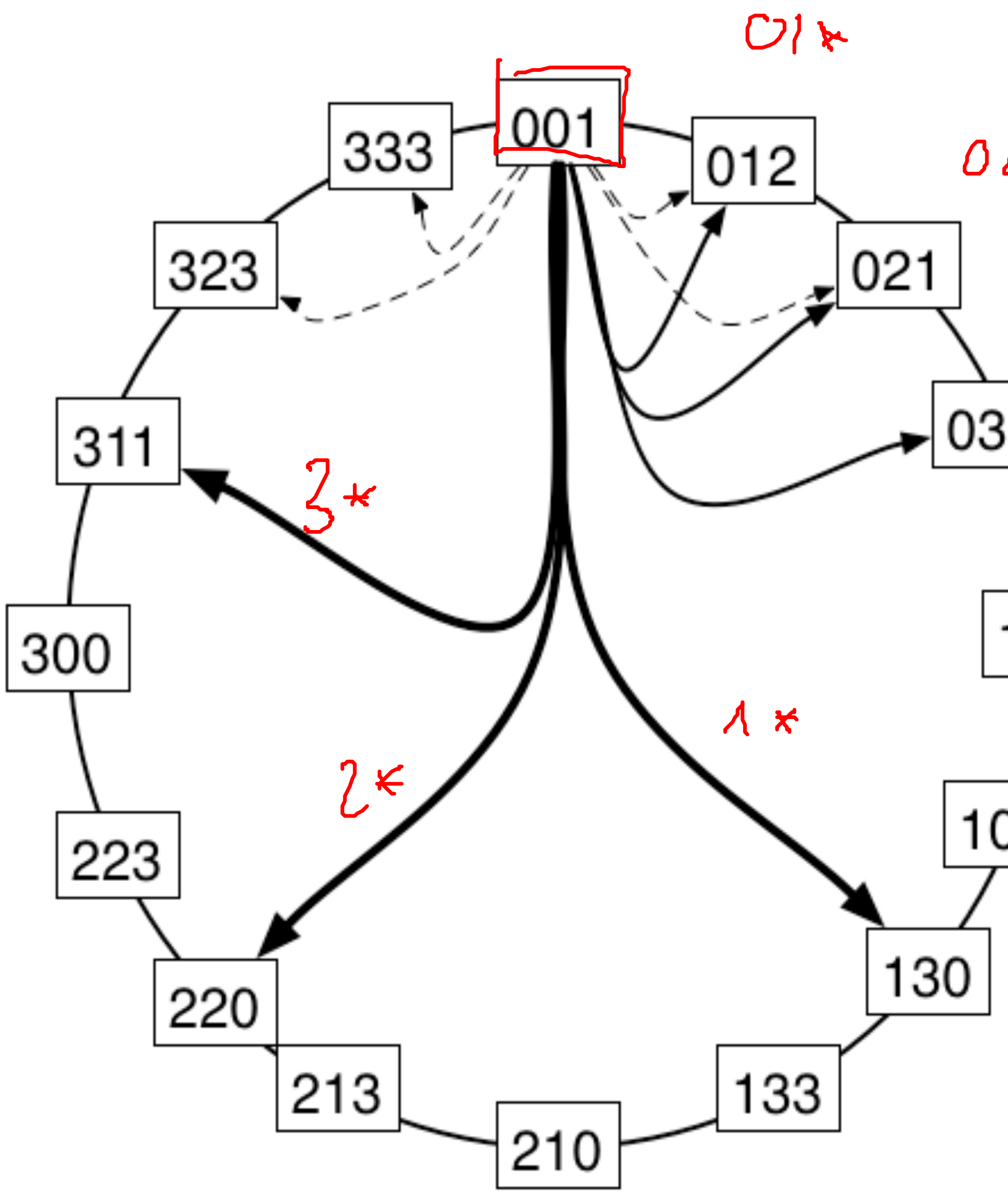
- ℓ/2 with next higher ID
- ℓ/2 with next lower ID

Observation

- The leaf-set alone can be used to find a target

Theorem

- With high probability there are at most O(2^b (log n)/b) entries in



Theorem

- With high probability there are at most $O(2^b (\log n)/b)$ entries in each routing table

Proof

- The probability that a peer gets the same m-digit prefix is $$2^{-bm}$$
- The probability that a m-digit prefix is unused is $$(1-2^{-bm})^n \leq e^{-n/2^{bm}}$$
- For m=c (log n)/b we get $$e^{-n/2^{bm}} \leq e^{-n/2^{c \log n}} \leq e^{-n/n^c} \leq e^{-n^{c-1}}$$
- With (extremely) high probability there is no peer with the same prefix of length (1+ε)(log n)/b
- Hence we have (1+ε)(log n)/b rows with 2^b-1 entries each

0x	1x	2x	3x	4x	5x		7x	8x	9x	ax	bx	cx	dx	ex	fx
60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex	6fx
650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex	65fx
65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex	65afx

Chord - recap.

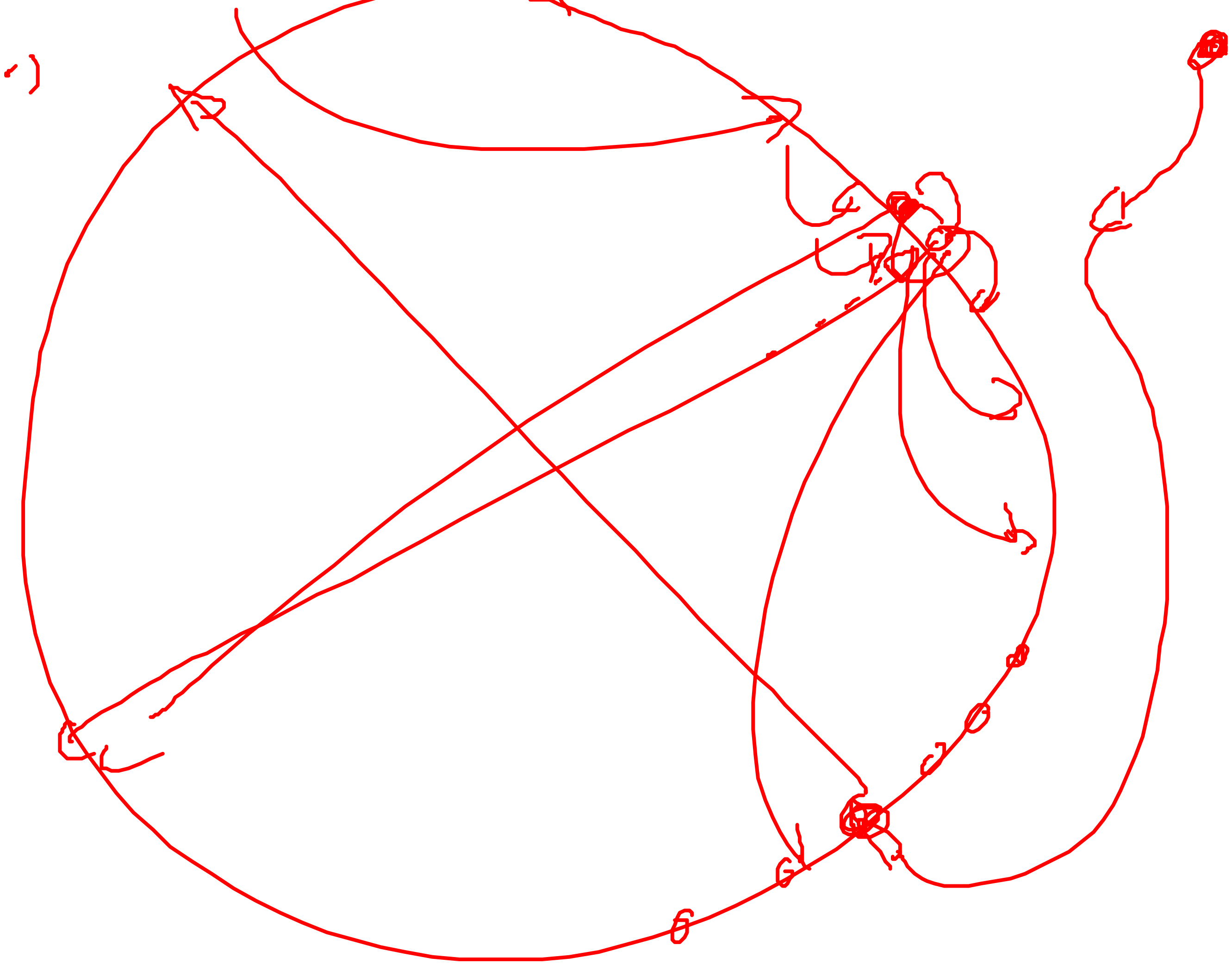
lookup

ing neighbor(s)

fingers

incoming

finger



You

17

lew node x sends message to the node
z with the longest common prefix p

z receives

- routing table of z
- leaf set of z

z updates leaf-set

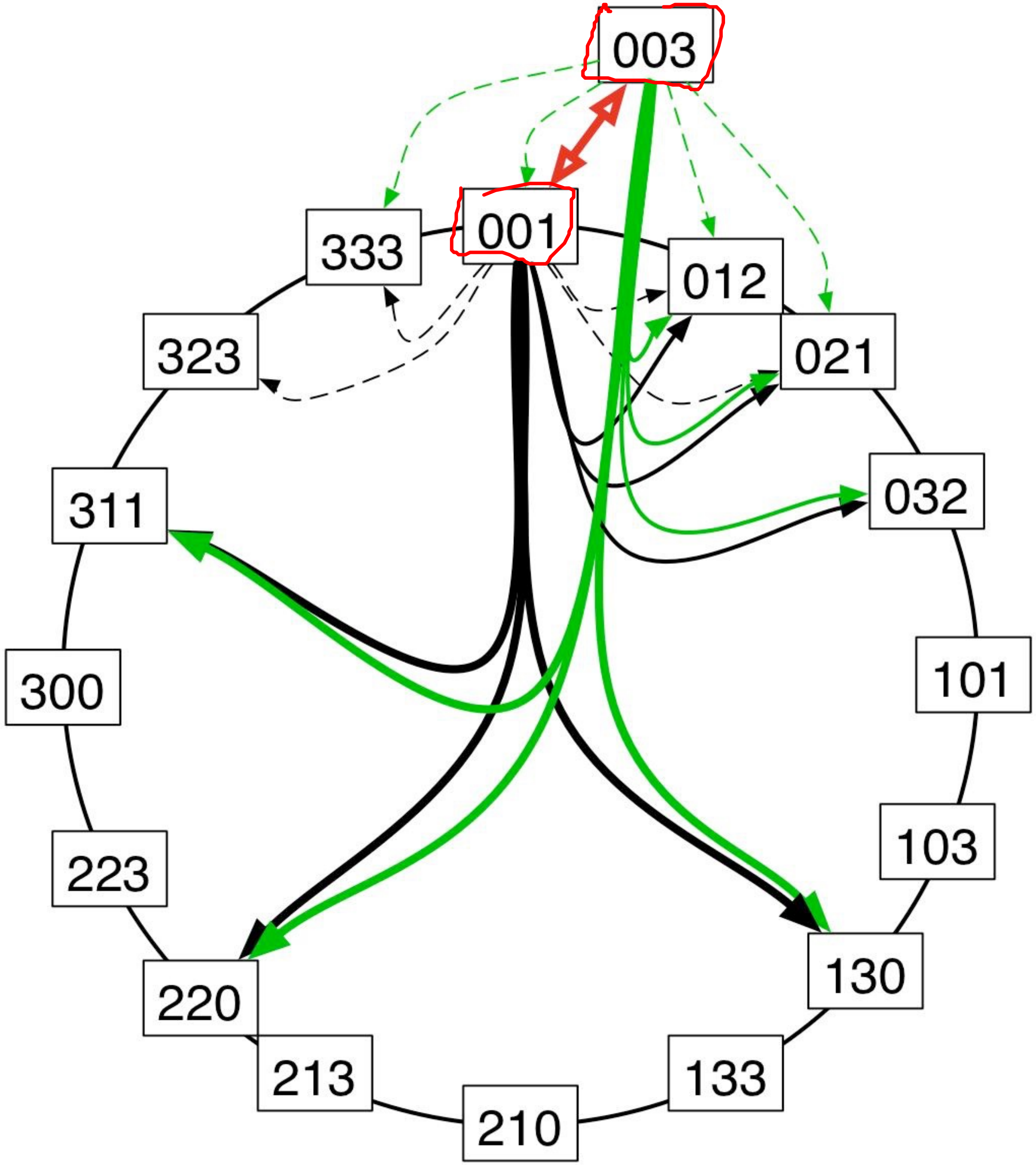
x informs ~~informiert~~ ℓ-leaf set

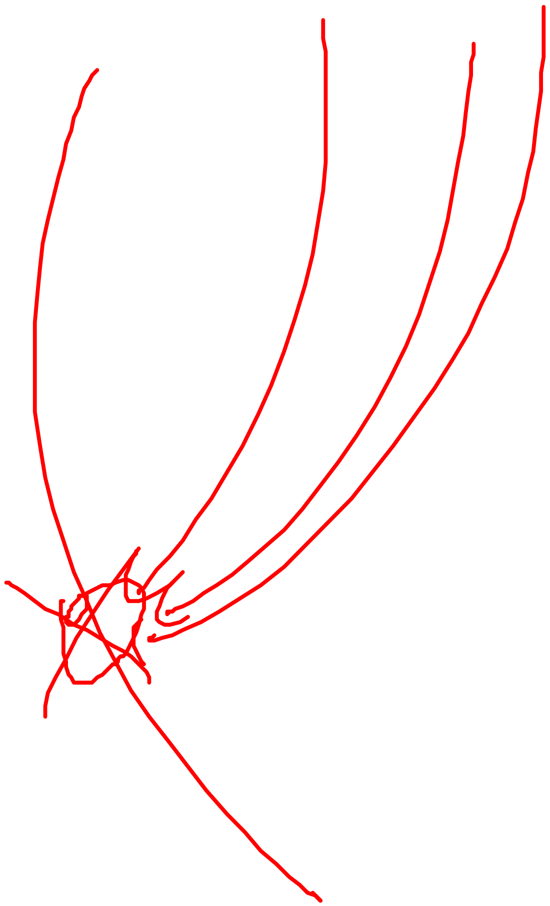
x informs peers in routing table

- with same prefix p (if $\ell/2 < 2^b$)

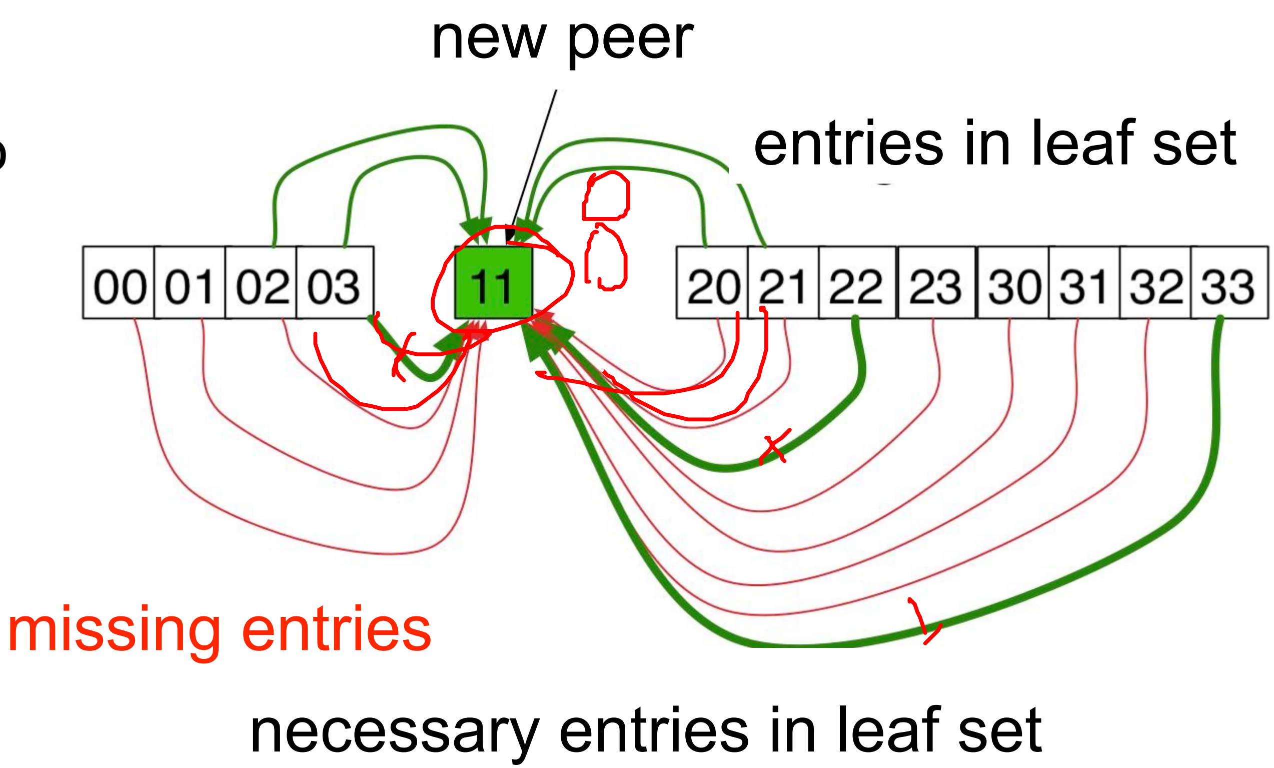
Number of messages for adding a peer

- ℓ messages to the leaf-set
- expected $(2^b - \ell/2)$ messages to node
  with common prefix
- one message to z with answer

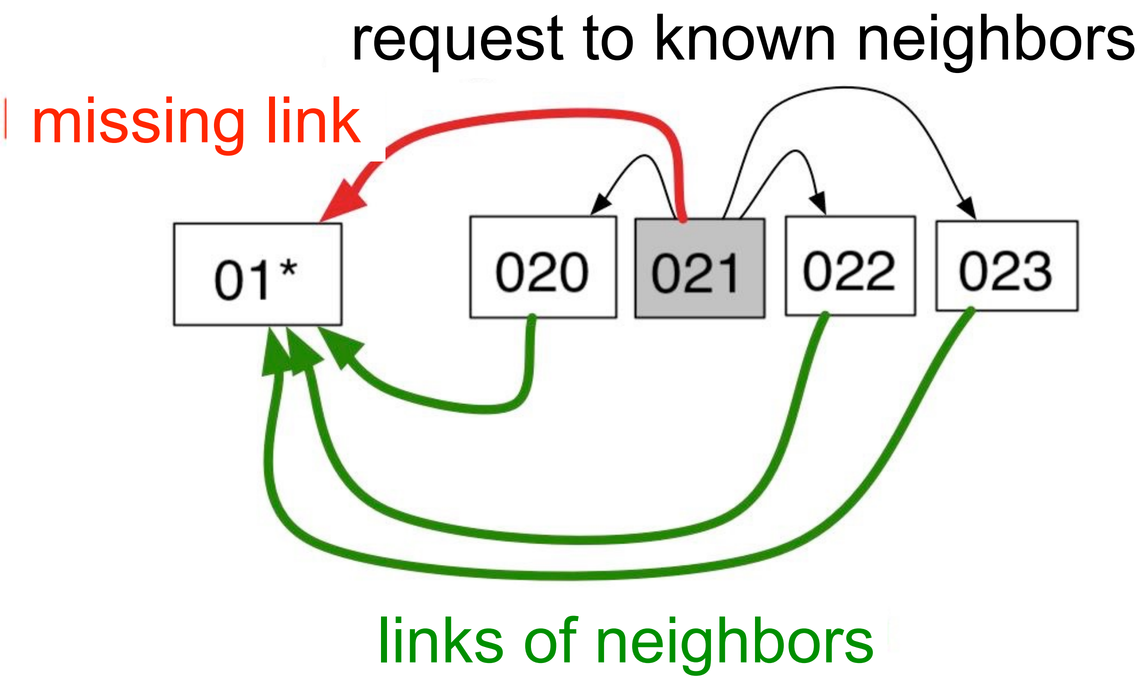




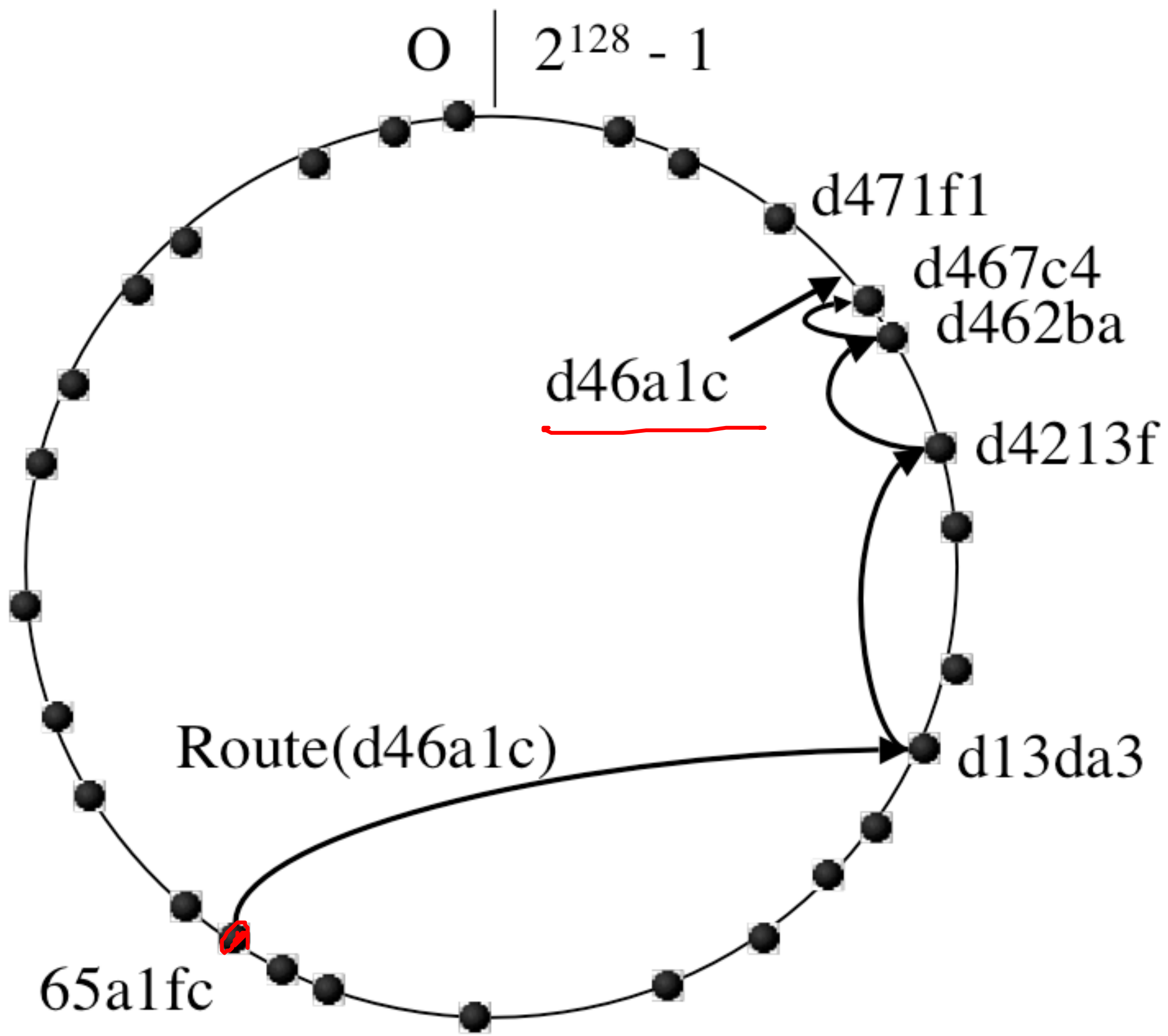
- Inheriting the next neighbor routing table does not allows work perfectly
- Example
  - If no peer with 1* exists then all other peers have to point to the new node
  - Inserting 11
  - 03 knows from its routing table
    - 22,33
    - 00,01,02
  - 02 knows from the leaf-set
    - 01,02,20,21
- 11 cannot add all necessary links to the routing tables



- Assume the entry R_i^j is missing at peer D
  - j-th row and i-th column of the routing table
- This is noticed if message of a peer with such a prefix is received
- This may also happen if a peer leaves the network
- Contact peers in the same row
  - if they know a peer this address is copied
- If this fails then perform routing to the missing link



- Compute the target ID using the hash function
- If the address is within the ℓ-leaf set
  - the message is sent directly
  - or it discovers that the target is missing
- Else use the address in the routing table to forward the mesage
- If this fails take best fit from all addresses



ℓ-leafset

~~routing table~~

nodes in the vicinity of D (according to RTT)

key

nodeID of current peer

: j-th row and i-th column of the routing table

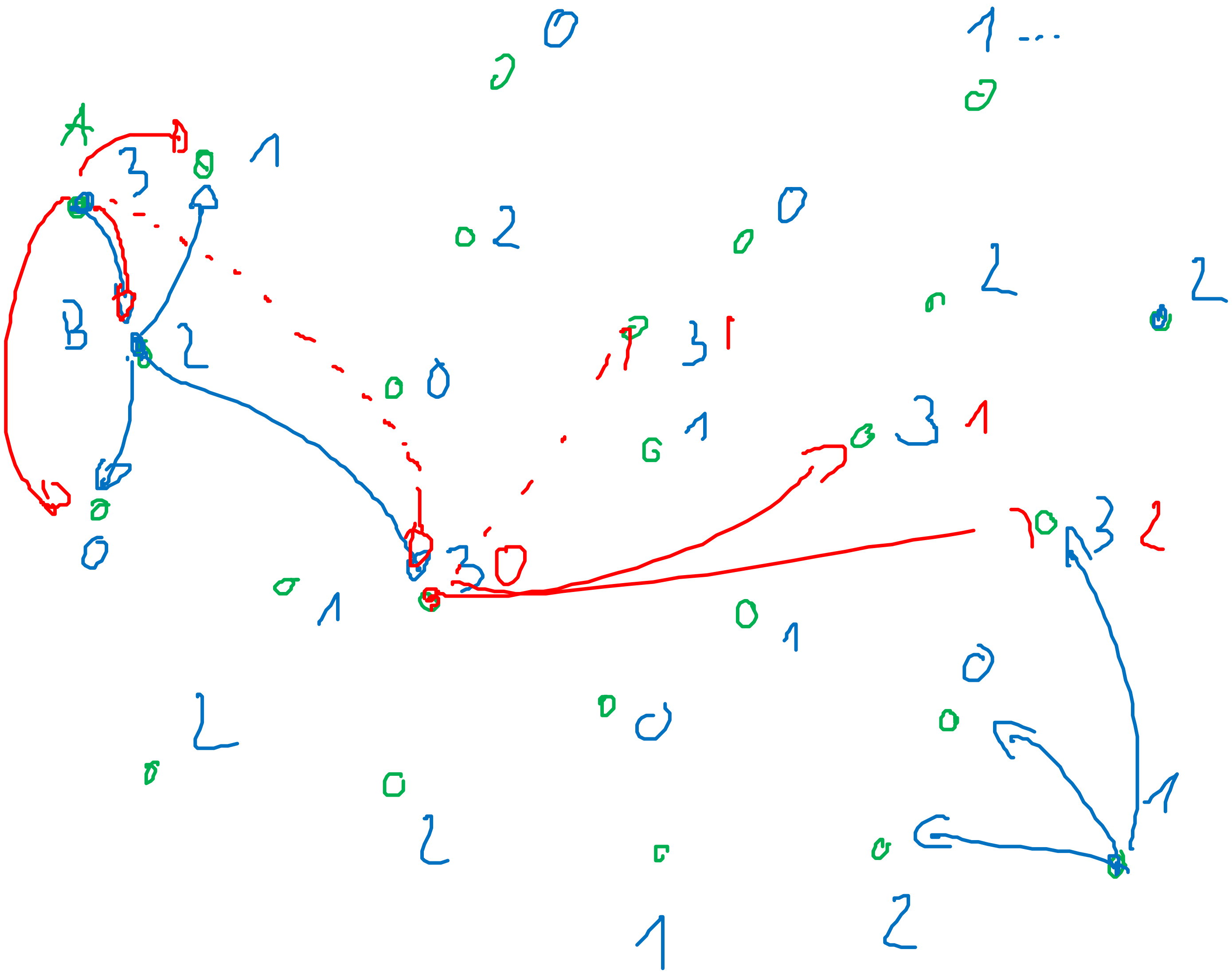
numbering of the leaf set

: i-th digit of key D

l(A): length of the largest common prefix of A and D (shared header length)

```
if (L_{-⌊|L|/2⌋} ≤ D ≤ L_{⌊|L|/2⌋}) {
    // D is within range of our leaf set
    forward to L_i, s.th. |D − L_i| is minimal;
} else {
    // use the routing table
    Let l = shl(D, A);
    if (R_l^{D_l} ≠ null) {
        forward to R_l^{D_l};
    }
    else {
        // rare case
        forward to T ∈ L ∪ R ∪ M, s.th.
            shl(T, D) ≥ l,
            |T − D| < |A − D|
    }
}
```

„magic So



- If the Routing-Table is correct

  $+ O(1)$

  - routing needs $\cancel{O}((\log n)/b)$ messages
- As long as the leaf-set is correct
  - routing needs $O(n/l)$ messages
  - unrealistic worst case since even damaged routing tables allow dramatic speedup
- Routing does not use the real distances
  - M is used only if errors in the routing table occur
  - using locality improvements are possible
- Thus, Pastry uses heuristics for improving the lookup time
  - these are applied to the last, most expensive, hops

- Leaf-set peers are not near, e.g.
  - New Zealand, California, India, ...

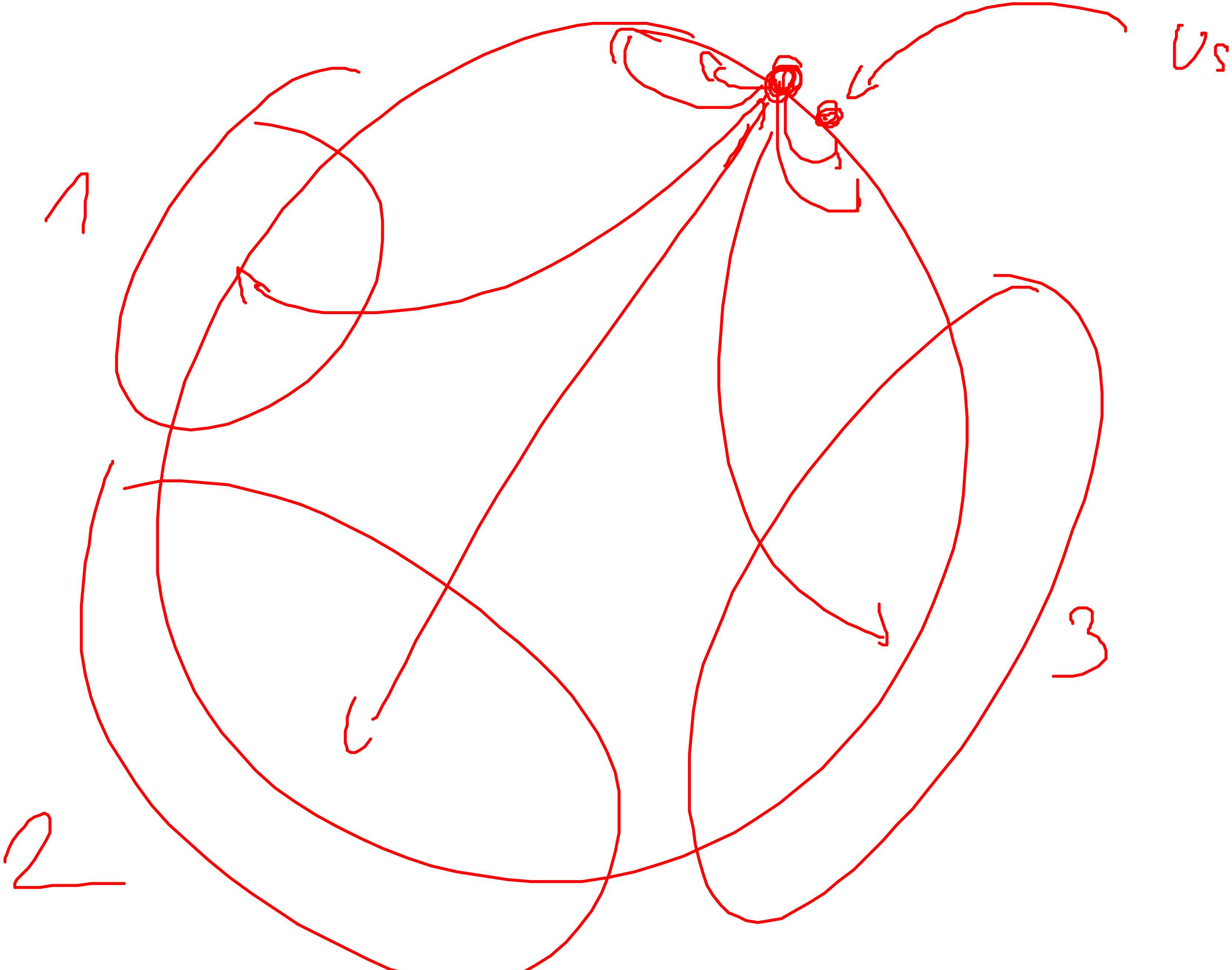
(x, y, z)

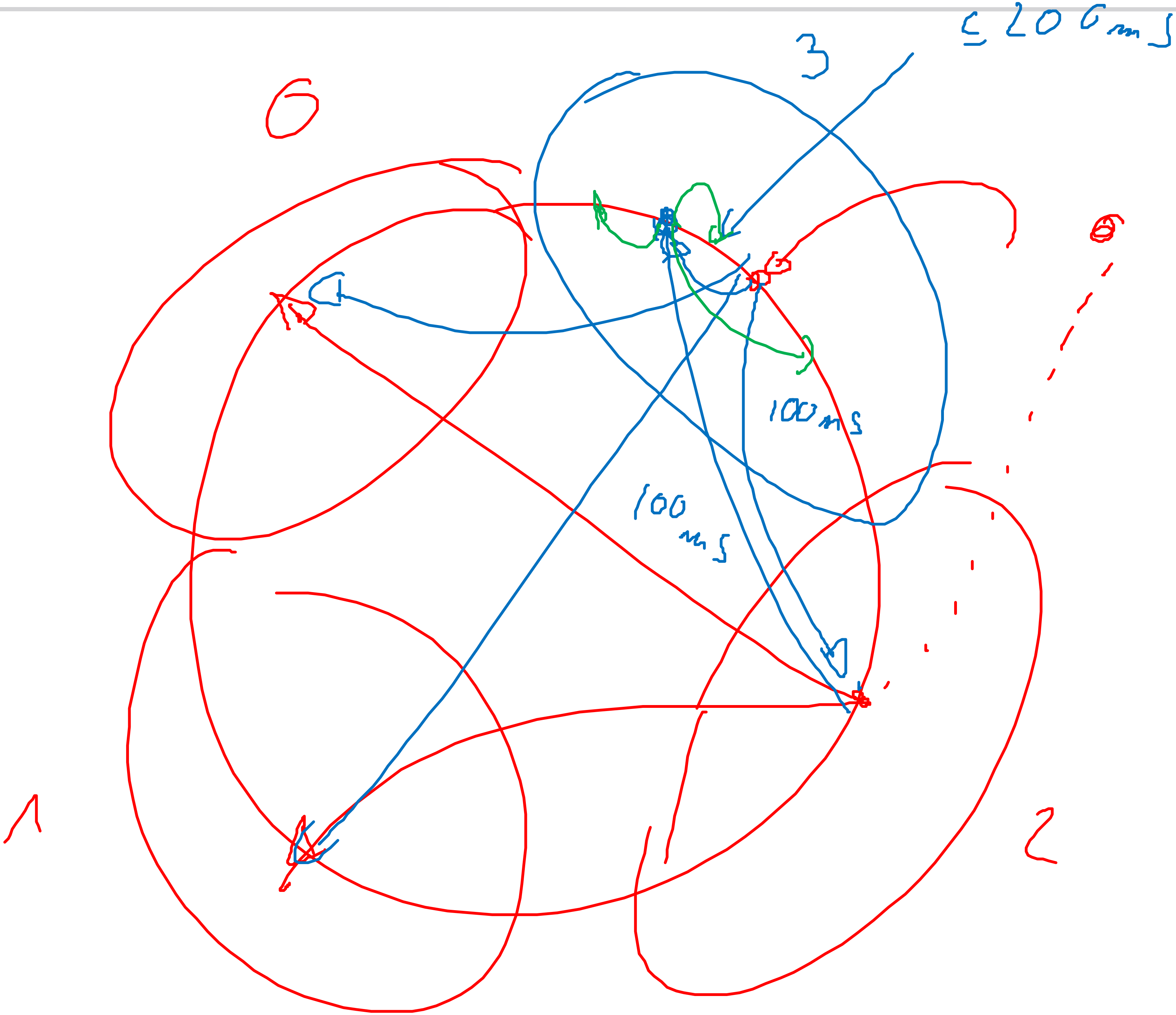
- TCP protocol measures latency
  - latencies (RTT) can define a metric
  - this forms the foundation for finding the nearest peers
- All methods of Pastry are based on heuristics
  - i.e. no rigorous (mathematical) proof of efficiency
- Assumption: metric is Euclidean

metric $d(A,B)$:

1. $d(A,A) = 0$
2. $d(A,B) = d(B,A)$
3. Triangle property

$d(A,B) \le d(A,C) + d(C,B)$


Vs
1
2
3



0
1
2
3
≤ 200 ms
100 ms
100 ms

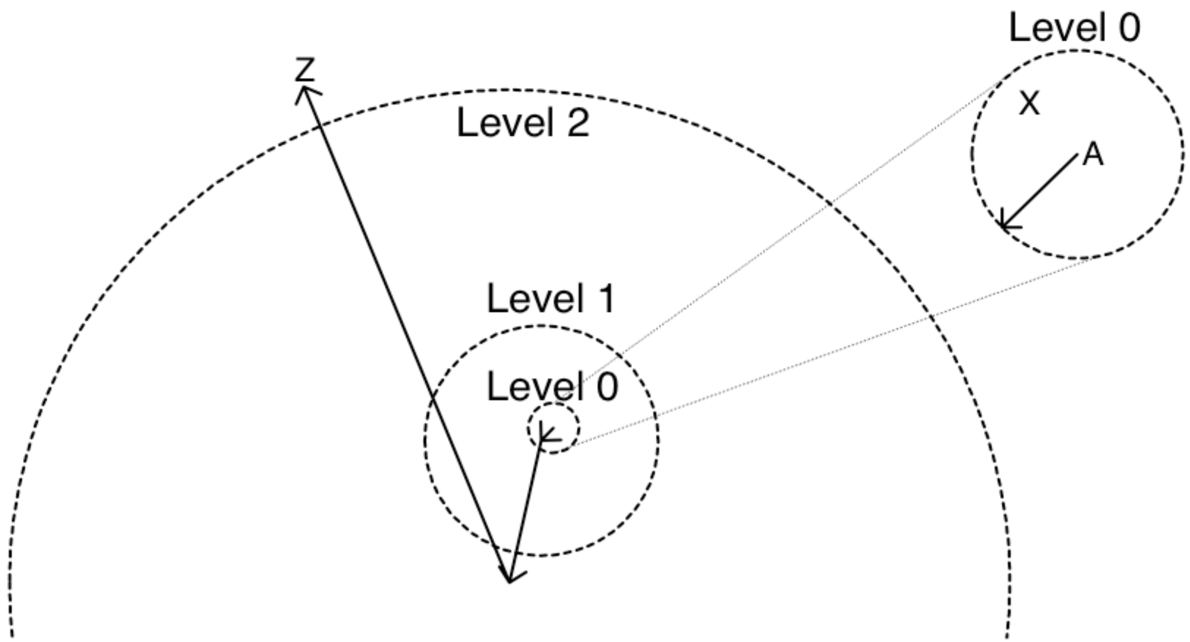
Assumption
- When a peer is inserted the peers contacts a near peer
- All peers have optimized routing tables

But:
- The first contact is not necessary near according to the node-ID

1st step
- Copy entries of the first row of the routing table of P
    - good approximation because of the triangle inequality (metric)

2nd step
- Contact fitting peer p‘ of p with the same first letter
- Again the entries are relatively close

Repeat these steps until all entries

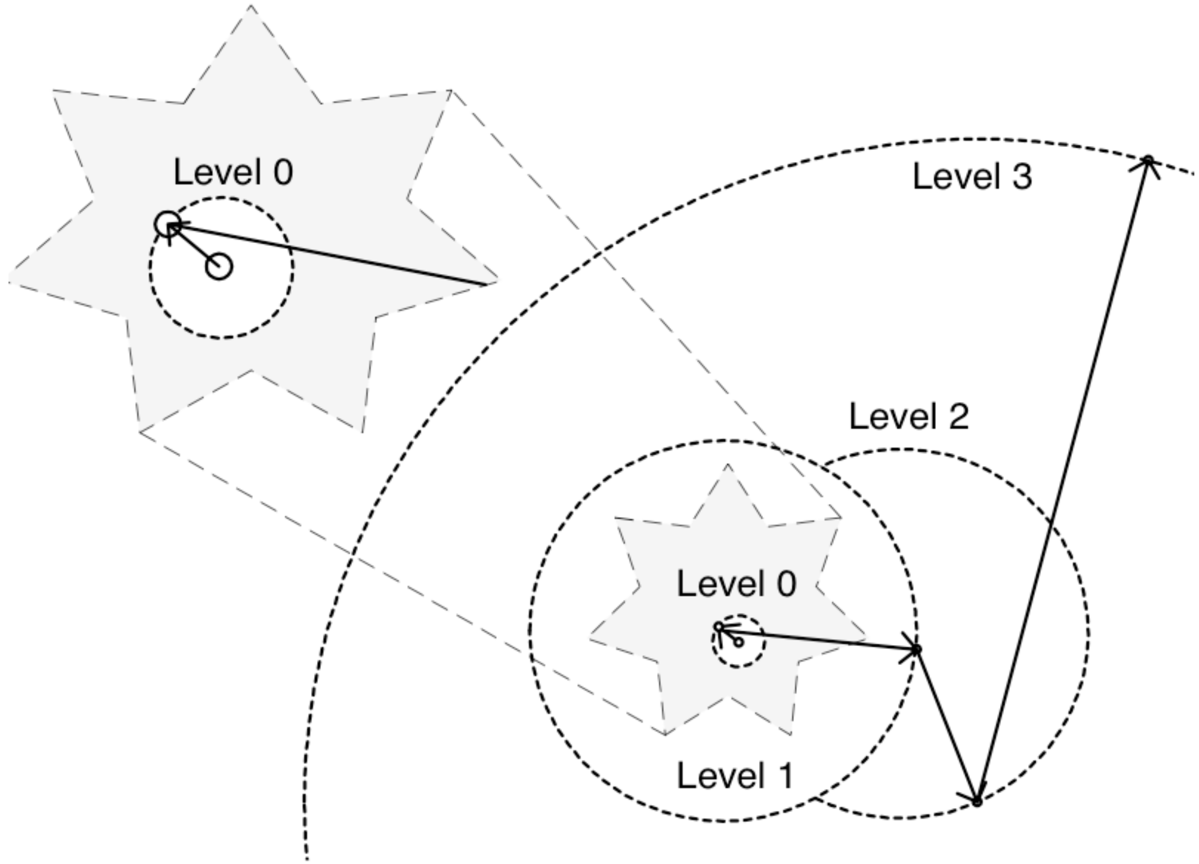


n the best case

- each entry in the routing table is optimal w.r.t. distance metric
- this does not lead to the shortest path

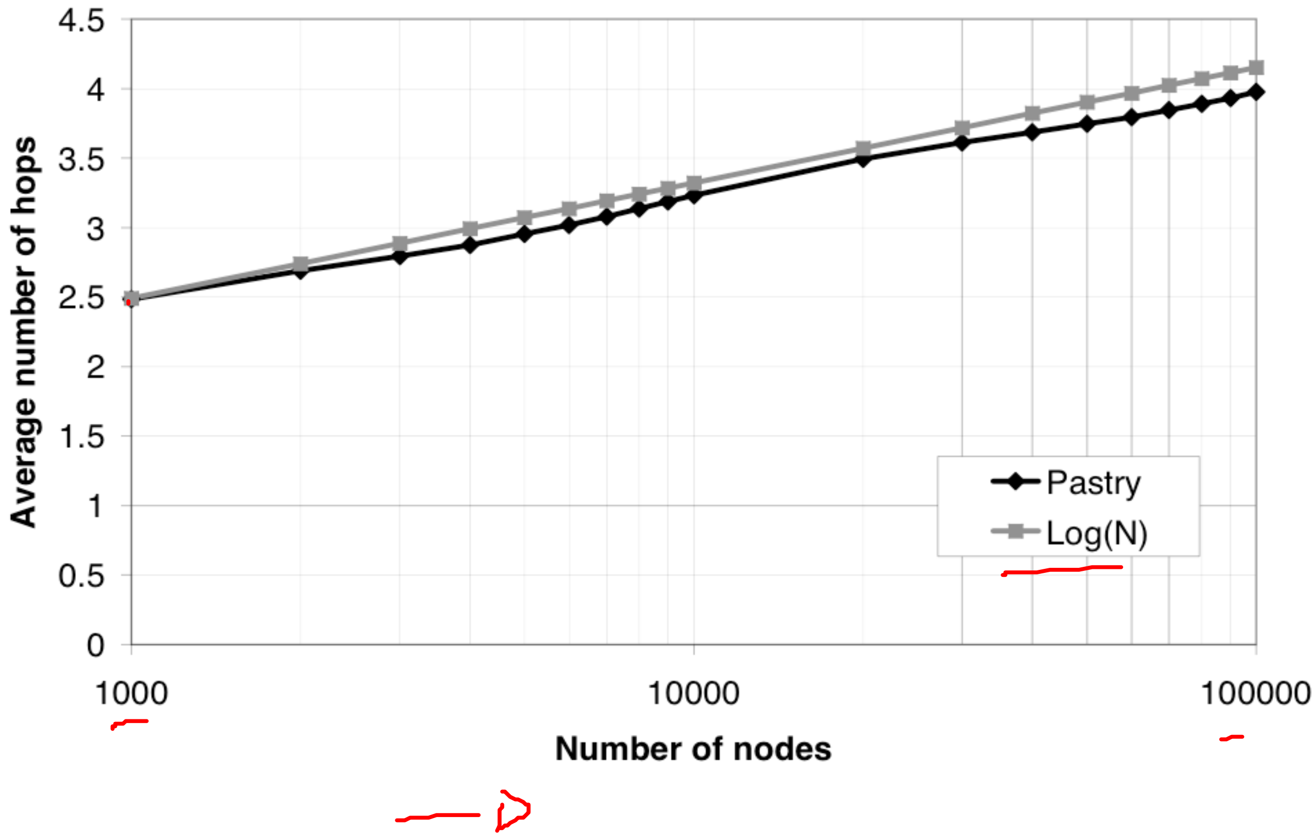
here is hope for short okup times

- with the length of the common prefix the latency metric grows exponentially
- the last hops are the most expensive ones
- here the leaf-set entries help



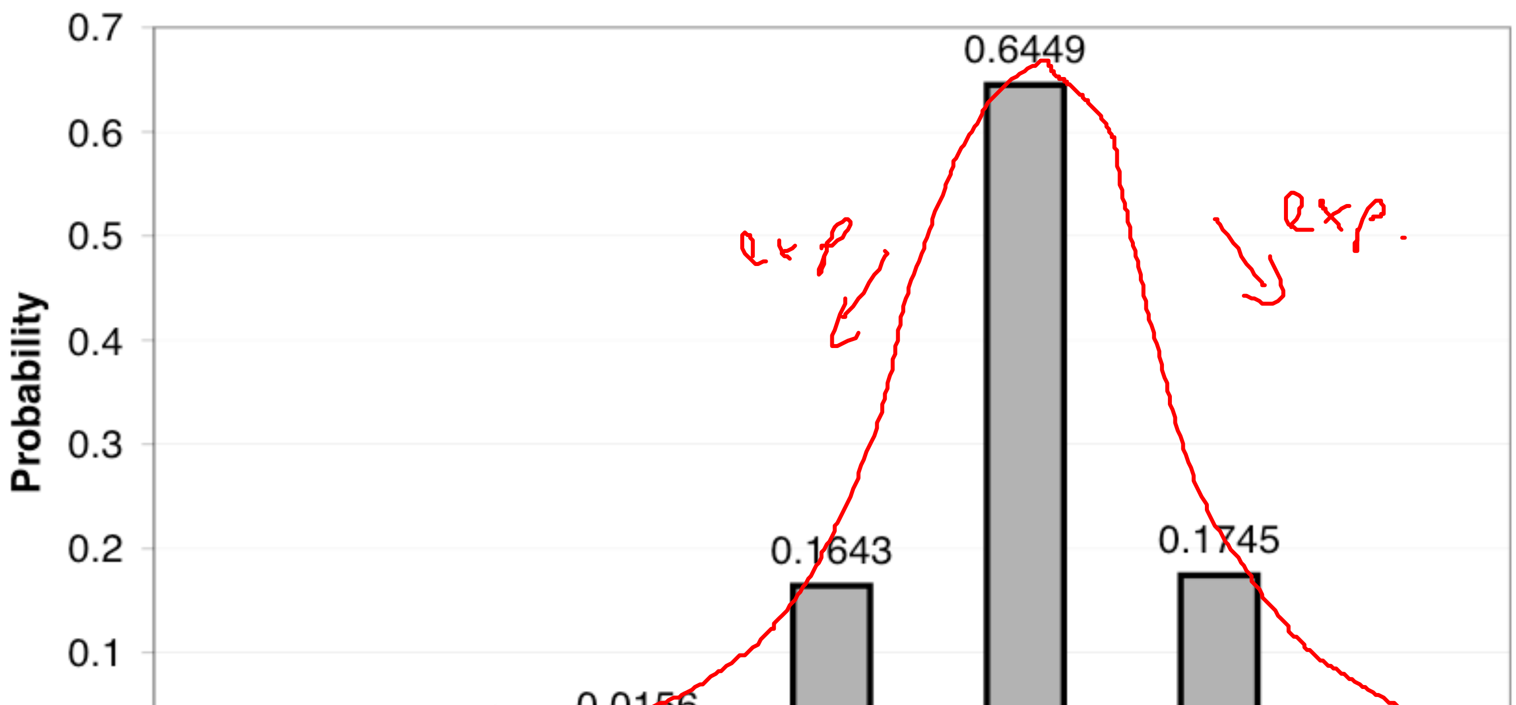
- Node-ID metric and latency metric are not compatible
- If data is replicated on k peers then peers with similar Node-ID might be missed
- Here, a heuristic is used
- Experiments validate this approach

- Parameter b=4, l=16, M=32
- In this experiment the hop distance grows logarithmically with the number of nodes
- The analysis predicts O(log n)
- Fits well

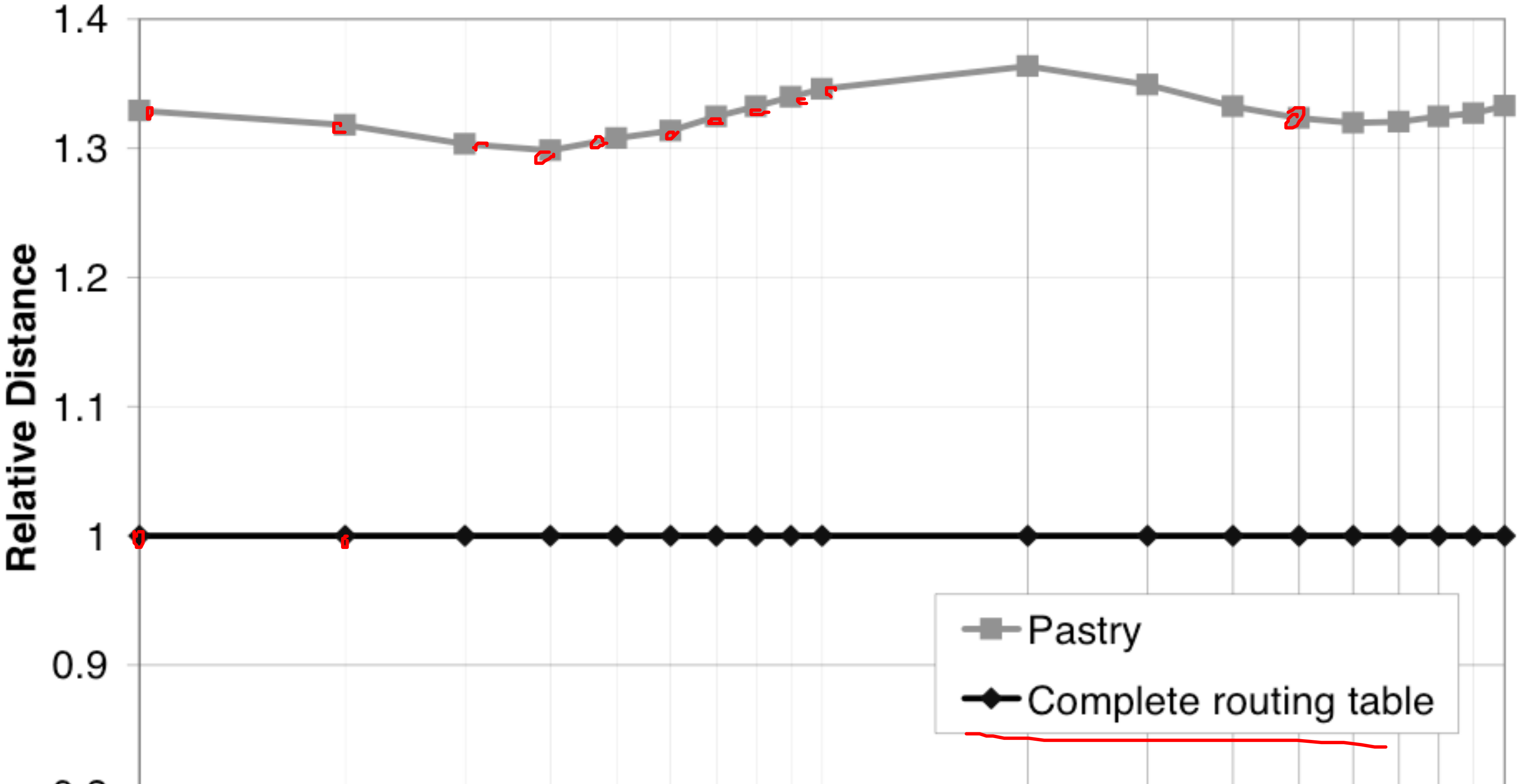


Distribution of Hops

- Parameter b=4, l=16, M=32, n = 100,000
- Result
  - deviation from the expected hop distance is extremely small
- Analysis predicts difference with extremely small probability
  - fits well



- Parameter b=4, l=16, M=3
- Compared to the shortest path astonishingly small
  - seems to be constant



Peer-to-Peer Networks

05 Pastry

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg