Számítógép hálózatok, osztott rendszerek

5: Pastry

Pastry

- **A. Rowstron and P. Druschel, "*Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pages 329-350, 2001 .**
- Antony Rowstron (Microsoft Research, Cambridge, GB)
- Peter Druschel (Rice University, Houston, Texas)
- Fejlesztés: Microsoft Research, Cambridge

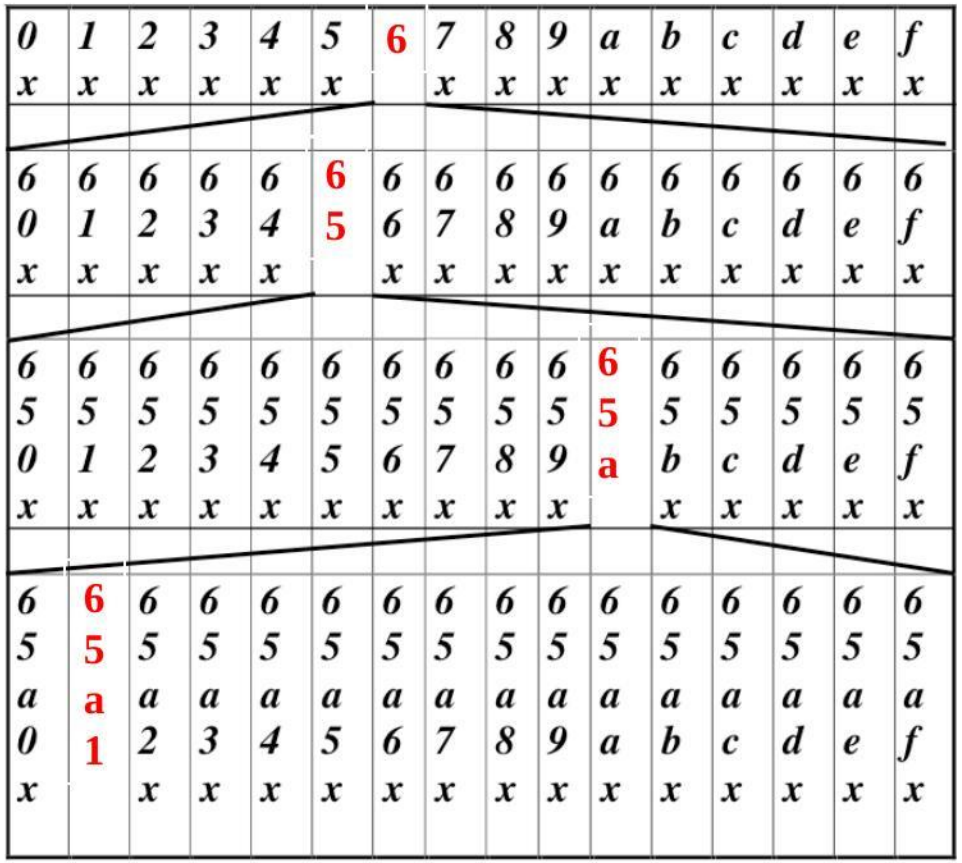
- Pastry
  - Skálázható, decentralizált objektum lokalizálás és routing nagy P2P hálózatokhoz
- PAST
  - Nagy skálájú, persistens P2P adat tárolási utility
- PAST egy Pastry felhasználás, amely lehetővé teszi a teljes P2P adattárolási funkcionalitást
- Pastry-re koncentrálunk

Pastry áttekintés

- Minden peer-nek van egy 128 bites azonosítója: nodeID
  - Egyedi és egyenletes eloszlású
  - Cryptografikus hash függvény IP címekre alkalmazva
- Routing
  - A kulcsok leképeződnek $\{0,1\}^{128}$-ra
  - Egy metrikának megfelelően az adatok a célhoz legközelebbi peer-en tárolódnak
- A routing tábla mérete:
  - $O(2^b(\log n)/b) + k$ bejegyzés
  - n: peer-ek száma
  - L: konfigurációs paraméter
  - b: számrendszer alapjának hossza (bitben)
    - tipikusan: b = 4 (bázis 16),
  - k = 16
  - Az üzenetek megérkezése garantált addig amíg legfeljebb k/2 szomszédos peer esik ki
- Egy peer beszúrásához $O((\log n)/b)$ üzenet szükséges

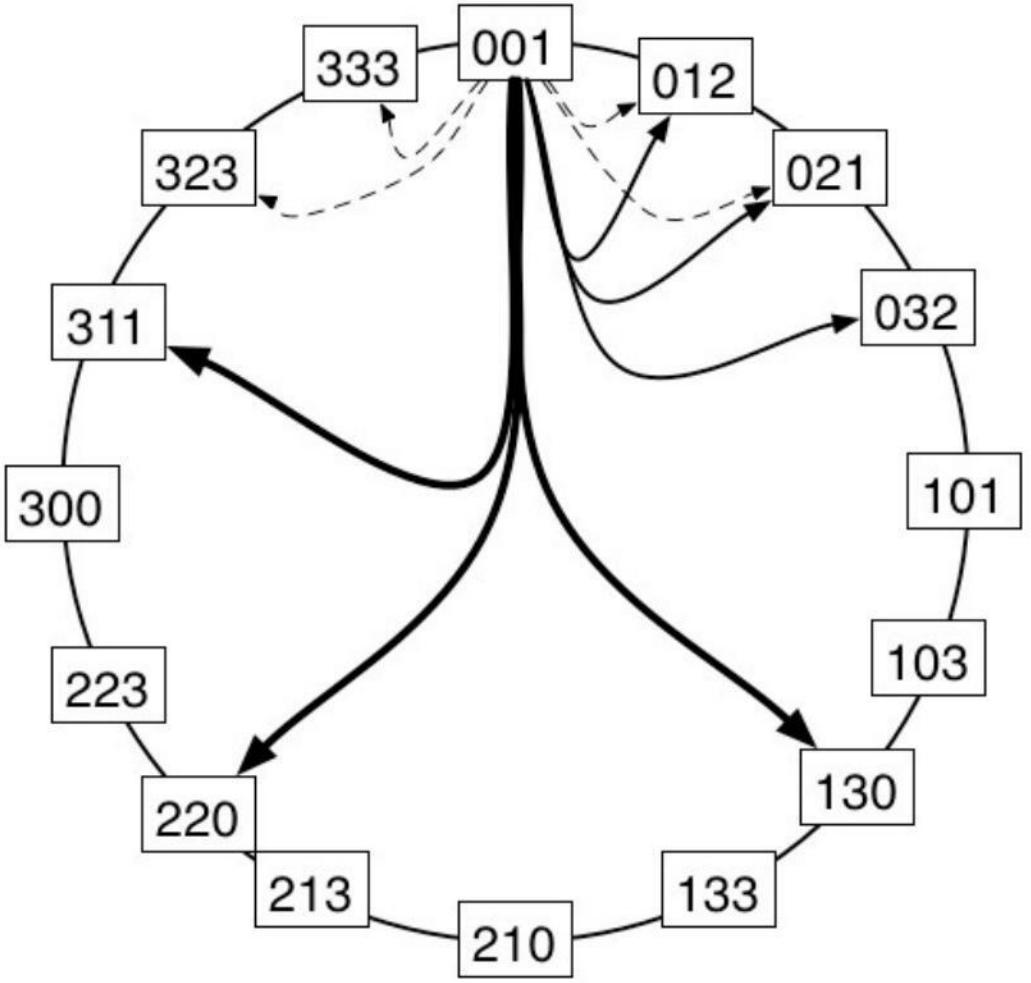
Routing tábla

- NodeID 2^b alapú rendszerben megadva
  - Pl. NodeID: 65A1BA13
- A NodeID minden p prefixéhez és $x \in \{0,\ldots,2^b-1\}$ számhoz adjunk egy peer-t a routing táblához, melynek prefixe px*, pl.
  - b=4, 2^b=16
  - 15 bejegyzés 0*,1*,..., F*
  - 15 bejegyzés 60*, 61*,..., 6F*
  - 15 bejegyzés 650*, 651*,..., 65F*
  - 15 bejegyzés 605A*,..., 65AF*
  - ...
  - Ha nincs ilyen prefixű peer, akkor a bejegyzés üres marad
- A következő szomszédot egy távolság metrikának megfelelően választjuk
  - A metrika a RTT (round trip time)
- Ezen kívül válasszunk k szomszédot
  - a NodeID-ra következő k/2 ID-vel
  - a NodeID-t megelőző k/2 ID-val



Routing tábla

- Példa b=2, k=4
- Routing tábla
  - A NodeID minden p prefixéhez és x ∈ {0,...,2^b-1}-hez adjunk egy peert a routing táblához, melynek prefixe
- Ezen kívül válasszunk k szomszédot
  - a k/2 rákövetkező ID-val
  - a k/2 megelőzőID-val
- Észrevétel
  - A levelek halmaza használható a cél megtalálására
- **Tétel:** Nagy valószínűséggel
  - O(2^b (log n)/b) bejegyzés van minden routing táblában...



Routing tábla

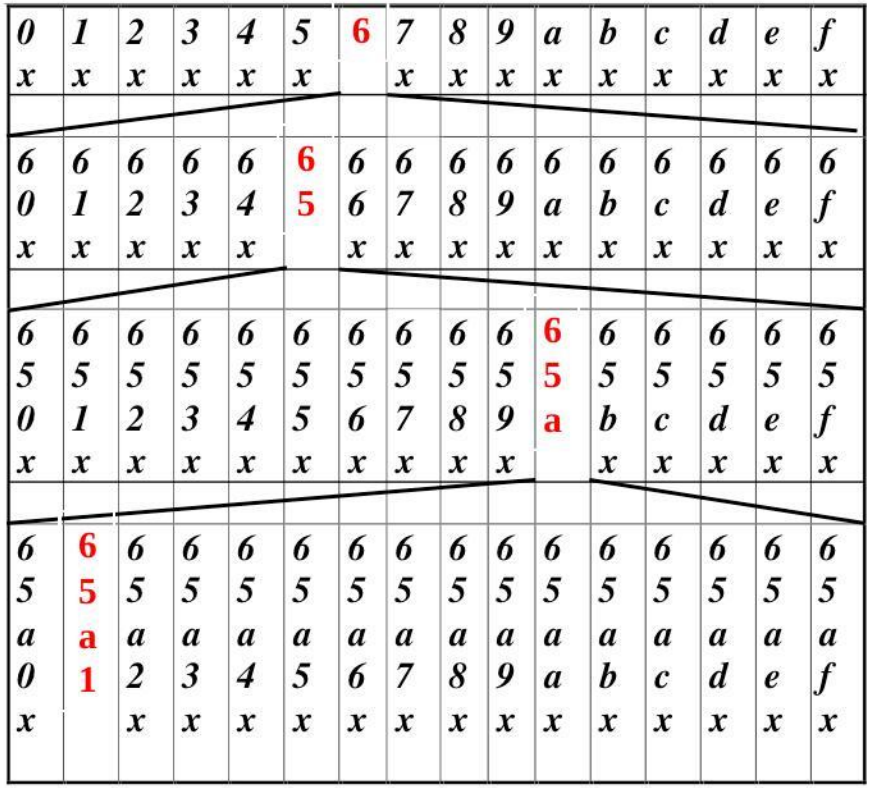
- **Tétel:** Nagy valószínűséggel
  - $O(2^b (\log n)/b)$ bejegyzés van minden routing táblában
- **Biz.:**
  - A valószínűség, hogy egy peer m hosszú perfixe megegyezik egy adott ID-vel: 2^{-bm}
  - A valószínűség, hogy egy m hosszú prefix nincs jelen:

$$(1- 2^{-bm})^n \approx e^{-n/2^{bm}}$$

  - m=c (log n)/b, c>1 esetén azt kapjuk (nagyon) nagy valószínűséggel, hogy

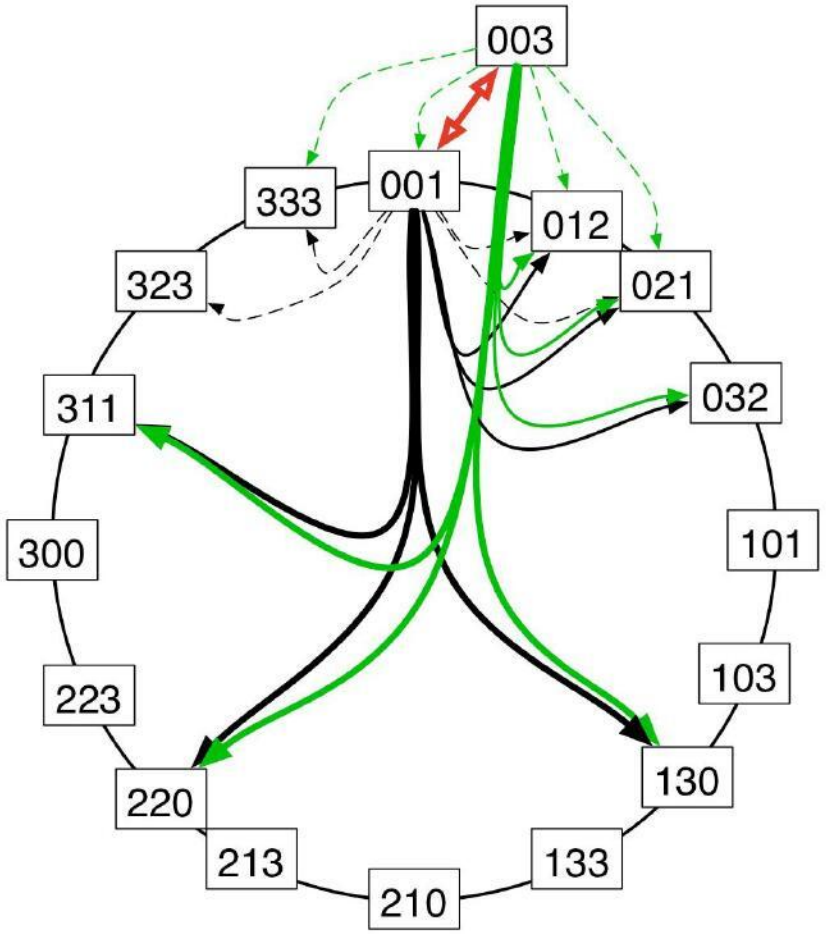
$$e^{-n/2^{bm}} = e^{-n/2^{c \log n}} = e^{-n/n^c} = e^{-n^{1-c}}$$

  - nincs olyan peer, amelynek a (1+ε)(log n)/b hosszú prefixe megegyezik az adott ID-vel
  - Tehát legfeljebb(1+ε)(log n)/b sor van a routing táblában,
    minden sorban 2^b-1 bejegyzés



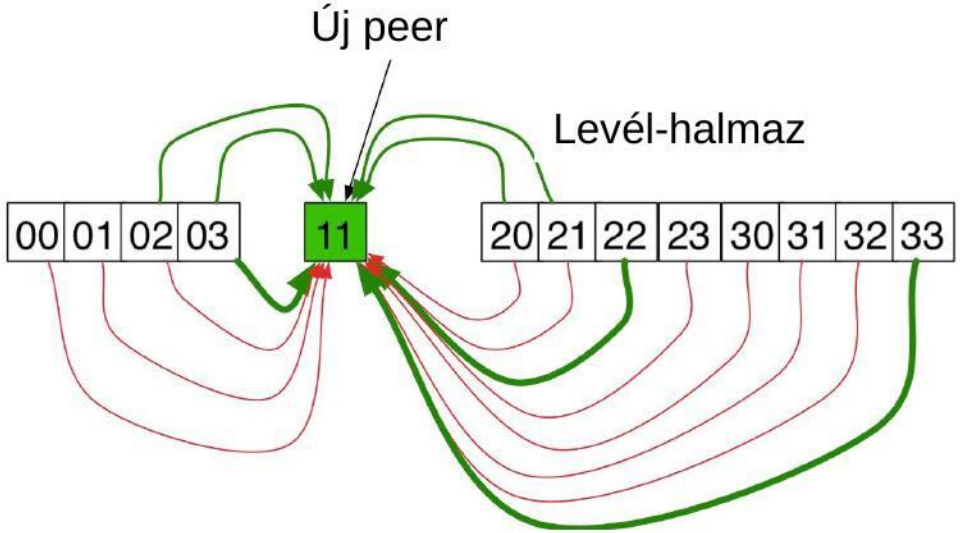
Peer belépése

- Az új peer x küld egy üzenetet annak az y peernek, melynek az ID-je a leghosszabb prefixben egyezik x ID-jével.
- x megkapja
  - y routing tábláját
  - y levél halmazát
- y aktualizálja a levél halmazát
- x informálja a k-levélből álló levél halmazát
- x informálja a peer-eket a routing táblában
  - melyeknek azonos a prefixe x prefixével (ha $k/2 < 2^b$)
- Az üzenetek száma belépéskor
  - k üzenet a levél halmaznak
  - várhatóan ($2^b - k/2$) üzenet a peer-eknek közös prefixszel
  - egy üzenet y-nak a válasszal



Ha a belépés operációban hiba történik

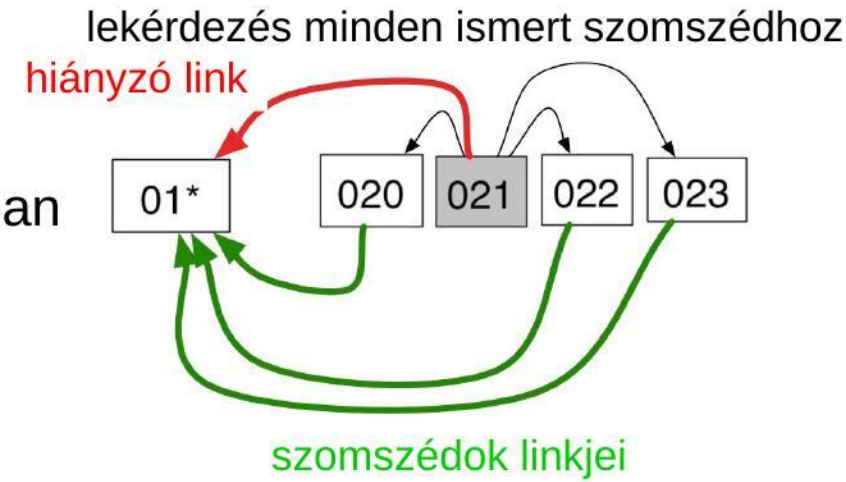
- A legközelebbi szomszéd routing táblájának öröklése nem működik tökéletesen
- Példa
  - Ha nincs peer 1* prefixszel, akkor minden más peer-nek az új peer-re kell mutatni amikor
  - belép 11
  - 03 a routing táblából ismeri:
    - 22,33
    - 00,01,02
  - 03 a levél-halmazból ismeri
    - 01,02,20,21
  - 11 nem tudja minden szükséges linket hozzáadni a routing táblához



Szükséges bejegyzések a routing táblában
hiányzó bejegyzések

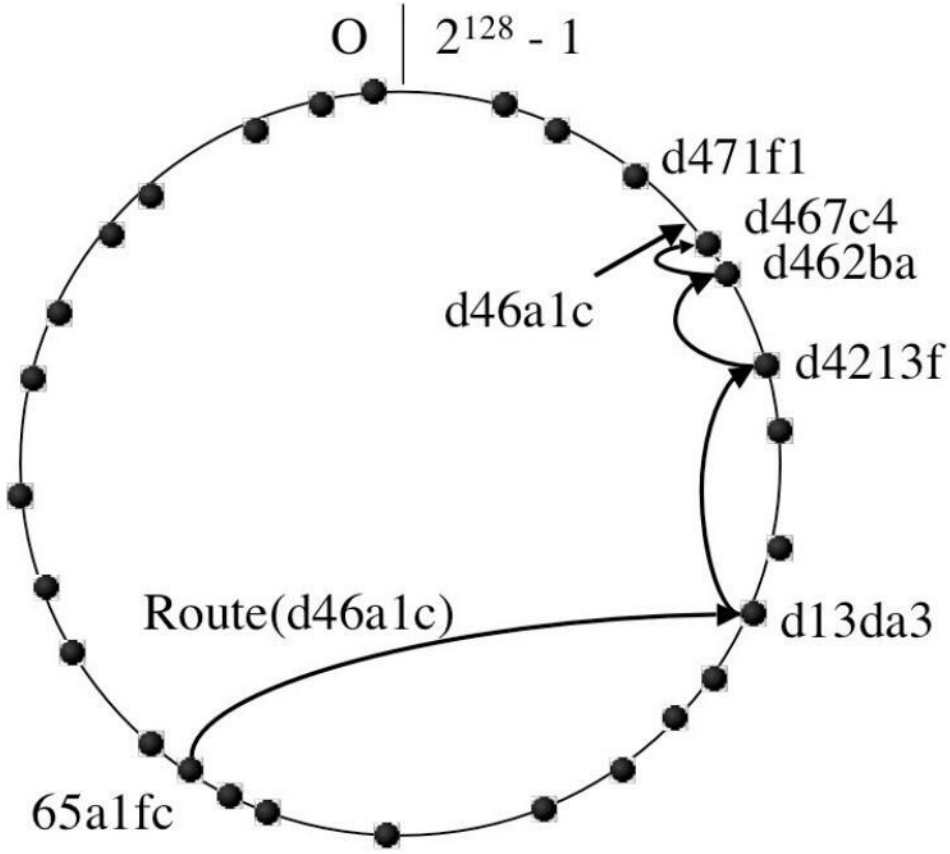
Hiányzó bejegyzések a routing táblában

- Tegyül fel, a D peer-nél az R_{ij} bejegyzés hiányzik
  - i-edik sor j-edik oszlop a routing táblában
- Ez értesítést kap, ha egy peer-től érkezik üzenet, melynek ilyen a prefixe
- Ez olyankor is történhet, amikor egy peer elhagyja a hálózatot
- Értesítjük a peer-eket ugyanabban a sorban
  - ha ezek ismernek ilyen peer-t, akkor ez a bejegyzés másolódik
- ha ekkor hiba lép fel, indítsunk routingot a hiányzó bejegyzéshez



Keresés

- Számítsuk ki a cél címet a hash függvénnyel
- Ha cím a levél-halmazban van
  - küldjük el az üzenetet direkt
  - vagy kiderítjük, hogy a cél nincs jelen
- Egyébként használjuk a címeket a routing táblában az üzenet továbbítására
- Ha ez nem sikerül, továbbítsuk a legjobban illeszkedő címre



A keresés részletesen

- L: k-levél-halmaz
- R: routing tábla
- M: peer-ek D közelében
- (RTT alapján)
- D: kulcs
- A: az aktuális NodeID of current peer
- R_j^i: j-edik sor i-edik oszlop a routing táblában
- L_i: a levél-halmaz számozása
- D_i: D i-edik számjegye
- shl(A,D): A és D leghosszabb közös prefixének hossza (shared header length)

```
(1)  if (L_{-⌊|L|/2⌋} ≤ D ≤ L_{⌊|L|/2⌋}) {
(2)      // D is within range of our leaf set
(3)      forward to L_i, s.th. |D − L_i| is minimal;
(4)  } else {
(5)      // use the routing table
(6)      Let l = shl(D, A);
(7)      if (R_l^{D_l} ≠ null) {
(8)          forward to R_l^{D_l};
(9)      }
(10)     else {
(11)         // rare case
(12)         forward to T ∈ L ∪ R ∪ M, s.th.
(13)             shl(T, D) ≥ l,
(14)             |T − D| < |A − D|
(15)     }
(16) }
```

Routing - diszkusszió

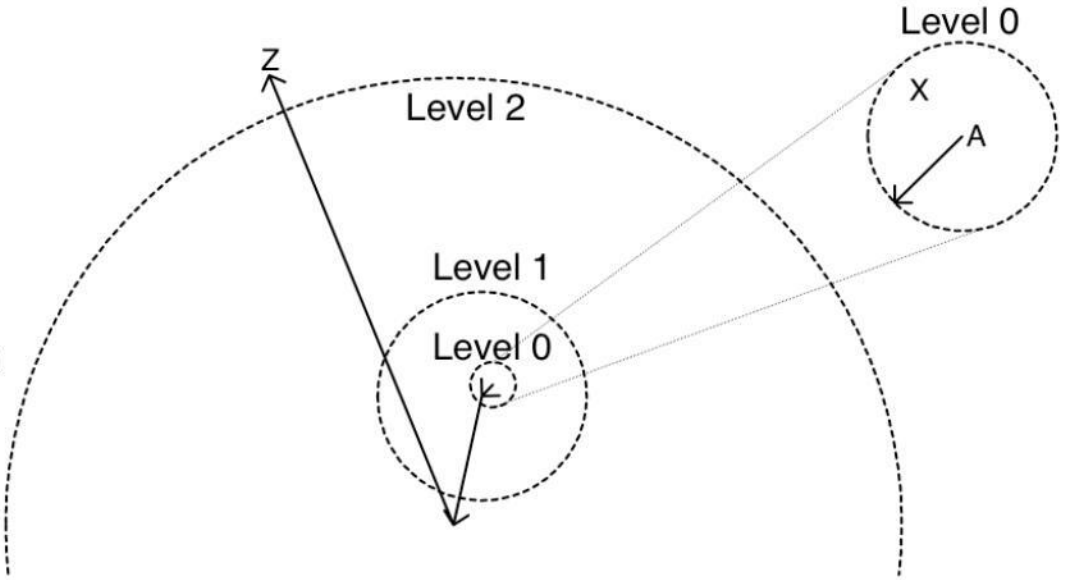
- Ha a routing tábla korrekt
  - a routinghoz O((log n)/b) üzenet továbbítás szülséges
- Amíg a levél-halmaz korrekt
  - a routinghoz O(n/k) üzenet továbbítás szükséges
  - nem reális worst case, még sérült routing táblák is hatalmas gyorsítást eredményeznek
- Routing nem használ földrajzi vagy RTT távolságot
  - M-et csak akkor használjuk, ha a routing tábla hibás
  - lokalitást ki lehet használni a gyorsításhoz
- Pastry heurisztikákat használ a keresési idő gyorsításához

Az m legközelebbi peer lokalizálása

- A levél-halmaz peer-jei nincsenek közel, pl.
  - New Zealand, California, India, ...
- TCP nyilvántartja a késést
  - a késés (RTT) definiálhat egy metrikát
  - ez lesz az alapja, hogy hogyan találjuk meg a legközelebbi peer-eket
- Pastry minden módszere heurisztikán alapul
  - azaz nincs rigorózusan (matematikailag) bizonyítva a hatékonyság
- Feltevés: a metrika Euklídeszi

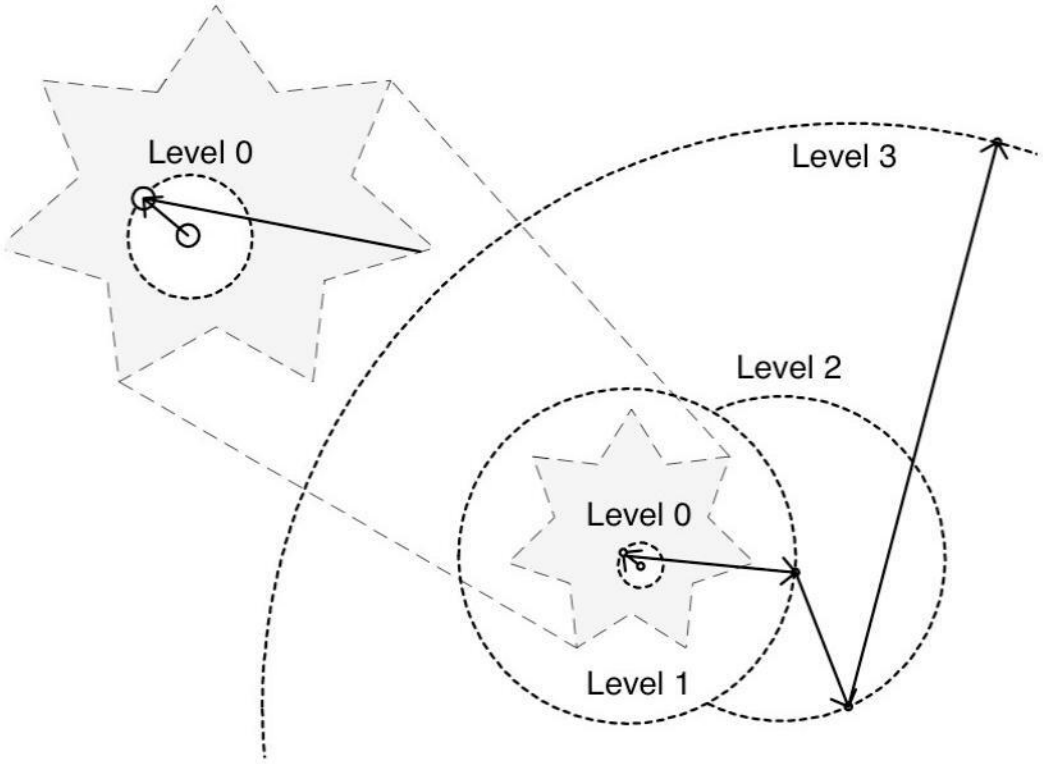
Lokalitás a routing táblában

- Feltesszük:
  - Ha egy peer belép, egy közeli p peer-rel lép kapcsolatba
  - Minden peer routing táblája optimalizált
- De az első kapcsolat nem feltétlenül közeli a NodeID szerint
- 1. lépés
  - másoljuk át p routing táblájának az első sorát
  - jó approximáció a háromszög egyenlőtlenség miatt (metrika)
- 2. lépés
  - lépjünk kapcsolatba egy p' peer-rel, amely ID-je ugyanolyan számmal kezdődik, mint p ID-je
  - Ismét, minden bejegyzés relatív közel van
- Ezt addig ismételjük, amíg minden bejegyzést aktualizáltunk



Lokalitás a routing táblában

- A „best case”:
  - Minden bejegyzés a routing táblában optimális a távolság metrika szerint
  - Ez nem a legrövidebb úthoz vezet
- Van remény rövidíteni keresési időt
  - A közös prefix hosszával exponenciálisan nő a távolság
  - Az utolsó hop-ok a legdrágábbak
- Itt a levél-halmazok segítenek

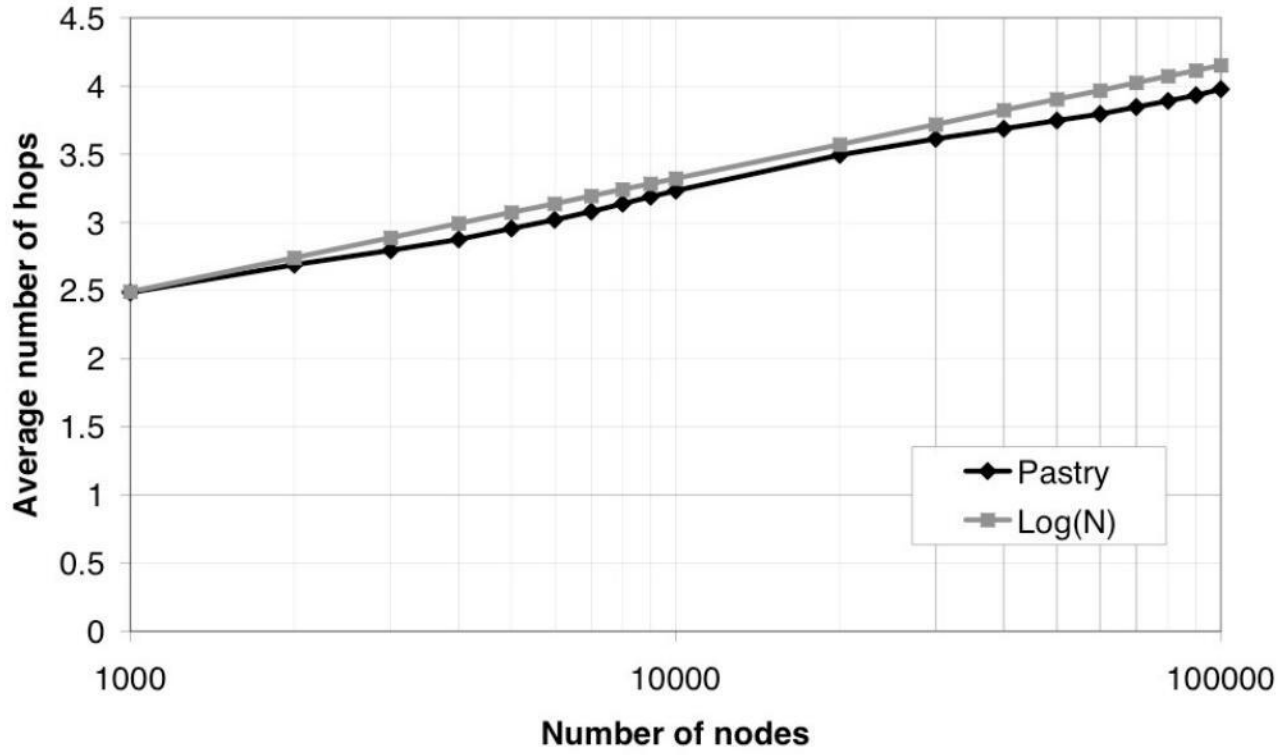


Legközelebbi peer-ek megtalálása

- NodeID metrika és a késés metrika nem kompatibilisek
- Ha az adatokat m peer-en replikáljuk, akkor peer-ek hasonló ID-vel hiányozhatnak
- Itt heurisztikát használunk
- Ezt a megközelítést kisérletekkel validálták

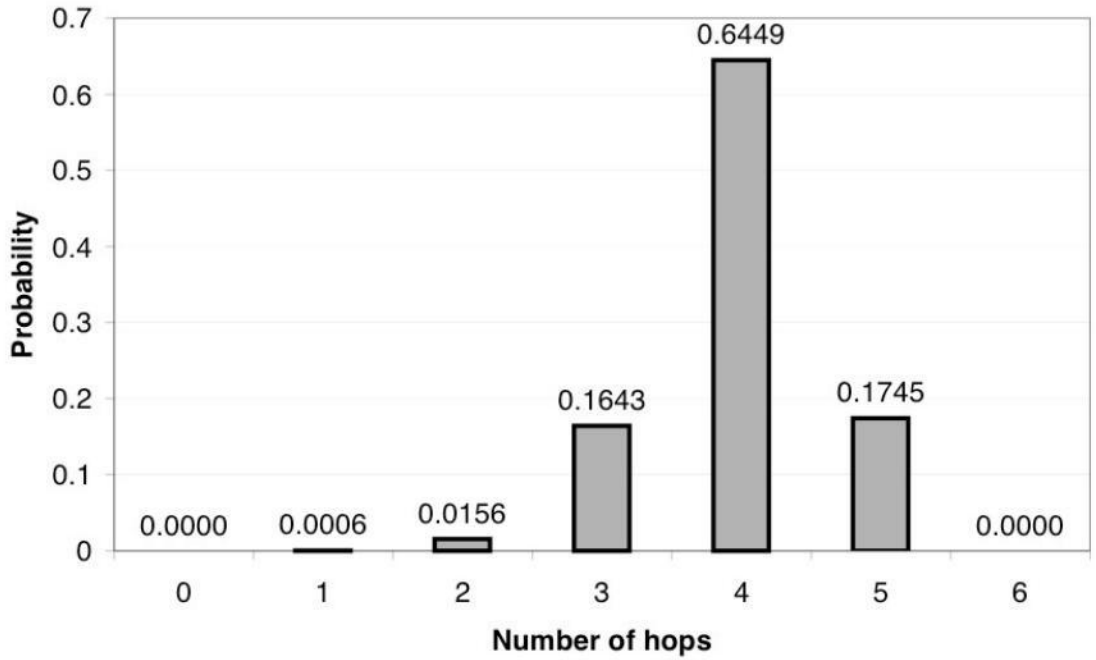
Legközelebbi peer-ek megtalálása

- Paraméter b=4, k=16, M=32
- Ebben a kisérletben a hop-távolság logaritmikusan nő a peer-ek számával
- Az elemzés szerint a hopok száma: O(log n)
- Jól illeszkedik



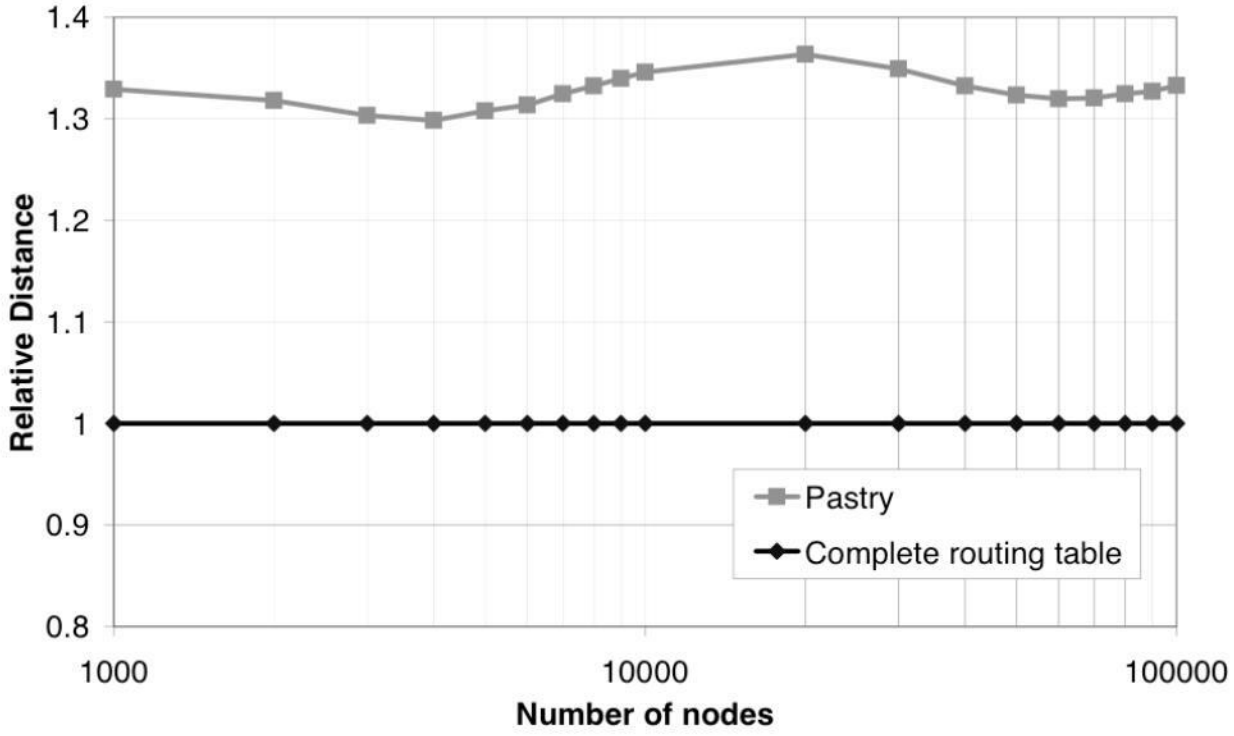
Kisérleti eredmények – hop távolság eloszlása

- Paraméter b=4, k=16, M=32, n = 100,000
- Eredmény
- Az eltérések a várt hop távolságtól nagyon kicsik
- Az elemzés nagyon kicsi valószínűséget ad az eltérésekre
- A kisérlet eredménye jól illeszkedik



Kisérleti eredmények – késés

- Paraméter b=4, k=16, M=3
- Összehasonlítva a legrövidebb úttal, meglepően kicsi
- A kisérletek alapján konstansnak tűnik



Kritikus észrevételek a kisérletekhez

- A kisérleteket egy jól viselkedő szimulációs környezetben végezték
- b=4, k=16 esetén a linkek száma elég nagy
  - $2^b/b = 4$ faktor erősen befolyásolja a kisérletet
  - Példaképpen: n = 100 000 esetén
    - $2^b/b \log n = 4 \log n > 60$ link van a routing táblában
    - Ezen felül 16 link a levél-halmazban, 32 link M-ben
- Összehasonlítva pl. a Chord protokollal, a linkek száma elég nagy
- A feltevés az Euklideszi metrikáról elég önkényes