Számítógép hálózatok és osztott rendszerek

Peer-To-Peer Hálózatok

Motiváció

- Közösség
  - Szólásszabadság csak akkor lehetséges, ha a felszólalóra az nem járhat negatív következménnyel
  - Így csak egy anonim felszólalónak valódi szólásszabadsága van
- Szerzői jog megsértés
  - Számítógéppel könnyű másolni (legtöbbször számítógéppel történik)
  - Szerzői jogok korlátozzák a másolást
  - File megosztók felhasználói nem akarnak büntethetők lenni
- Diktatúra
  - Elnyomó rendszerek egy előfeltétele az információ és a vélemények kontrollálása tematizáló hatalom, vélemény hatalom
  - Publikálás anélkül hogy félni kelljen a megtorlástól
- Demokráciák
  - bizonyos álláspontok képviselése nem legitim, pl.
  - közösség elleni uszítás (Btk. 332§)
  - bizonyos szexuális tartalmak
  - politikai nézetek (pl. fasiszta, kommunista, szaparatista, …)
- Anonimizáló P2P hálózatoknak biztosítani kell minden felhasználó személyi jogainak védelmét és anonimitását más felhasználók veszélyeztetése nélkül

Definíció

„Egy Peer-to-Peer hálózat egy kommunikációs hálózat számítógépek között, melyben minden résztvevő mind client, mind server feladatokat végrehajt.“

- Megfigyelés
  - Az Internet (tuladonképpen szintén) egy Peer-to-Peer hálózat
- Másik definíció a Peer-to-Peer Working-Group-tól

  „Egy Peer-to-Peer hálózatban elosztott számítási erőforrásokat közösen használnak direkt kommunikáció által.“

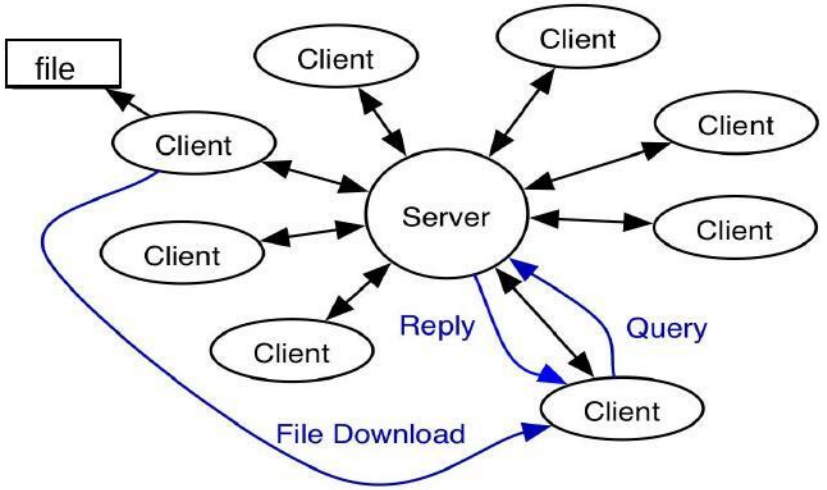
- Mi nem Peer-to-Peer hálózat?
  - Egy Peer-to-Peer hálózat nem Client-Server hálózat!

Terminológia

- Összekapcsolhatatlanság (Unlinkability)
  - Abszolut (ISO15408)
    - „biztosítja, hogy egy felhasználó több erőforrást vagy szolgáltatást használjon, anélkül, hogy mások képesek legyenek összekapcsolni ezeket a használatokat“
  - Relatív
    - Egy támadó nem tudhat meg többet a használatok közötti kapcsolatról a rendszer megfigyelésével
    - a-priori knowledge = a-posteriori knowledge

Hogy működik Napster?

- Kliens-Szerver struktúra
- Szerver tárol
  - Indexet meta-adatokkal
    - filenév, dátum, stb.
  - A résztvevő kliensek kapcsolatainak táblázatát
  - A résztvevő kliensek file-iank a táblázatát
- Lekérdezés (query)
  - Kliens kérdezi a file-nevet
  - Szerver kikeresi a megfelelő résztvevőket
  - Szerver válaszol, kinek van meg a file (tulajdonos kliensek)
  - Kérdező-Kliens letölti a file-t a tulajdonos-klienstől

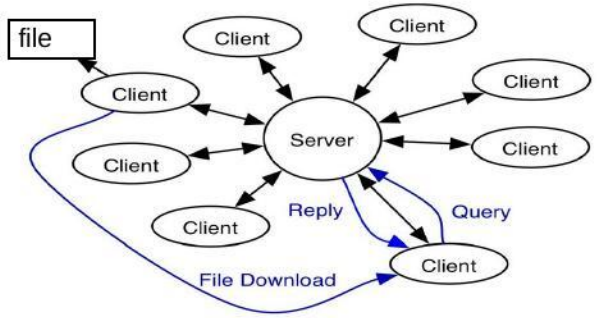


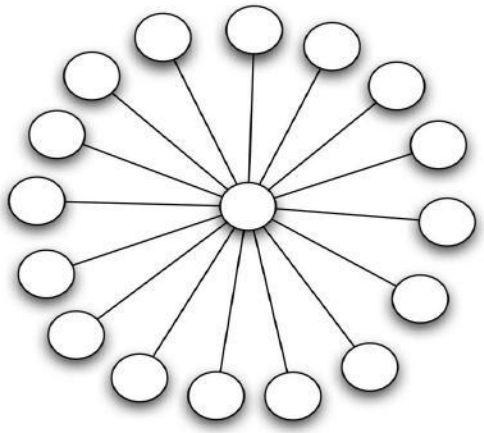
Támadások

- Denial-of-Service támadások (DoS)
  - vagy elosztott DoS (DDoS)
  - egy vagy sok peer kér egy dokumentumot
  - A peer-ek lelassulnak vagy teljesen leállnak
- Sybil támadás
  - Egy támadó sok „fake” peer-t hoz létre új IP címekkel
  - Vagy a támadó egy bot-net-et irányít
- Protokoll gyengeségeinek kihasználása
- Rosszindulatú peer-ek általi elidegenítés (infiltration)
  - Bizánci tábornokok
- Időzítési támadások
  - Üzenetek lelassulnak
  - Kommunikációs vonal lelassul
  - Egy kapcsolat létesíthető küldő és fogadó között
- Megfertőzés támadás
  - Hamis információ rendelkezésre bocsájtása
  - Rossz routing tábla, rossz index file, stb…
- Eclipse támadás
  - Egy peer környezetének megtámadása
  - Egy peer szeparálása
  - Egy „fake” környezet felépítése

Miért nem skálázható Napster?

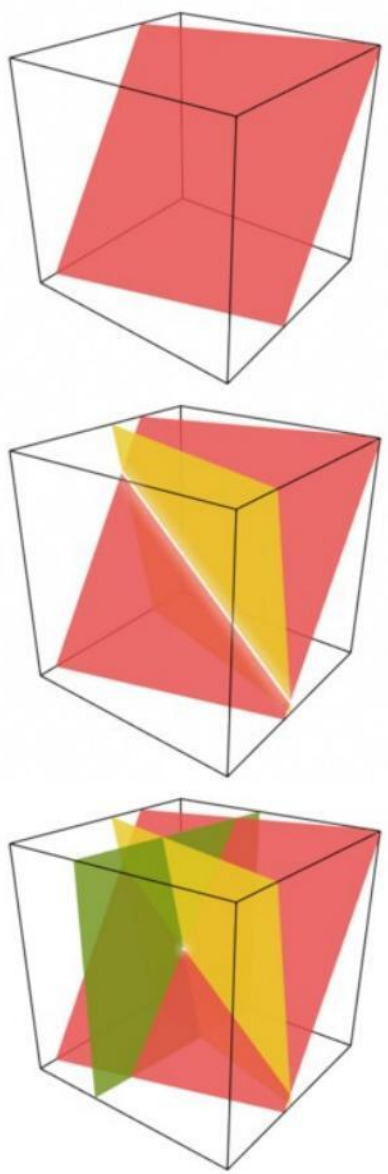
- Napster
  - Client-Server struktúra csillag topológiának felel meg
  - A gráf foka n-1
    - n a Peer-ek száma
  - A csillag csak 1-szeresen összefüggő
  - Egy gráf k-szorosan összefüggő, ha
    - bármely k-1 csomópont eltávolítása után még mindig összefüggő marad
    - van olyan k csomópont, amely eltávolítása után nem marad összefüggő
- Napster nem skálázható, mert
  - a gráf foka nagy
    - szűk keresztmetszet okoz a kommunikációban
  - öszzefüggőség kicsi
    - nem robusztus a konstrukció





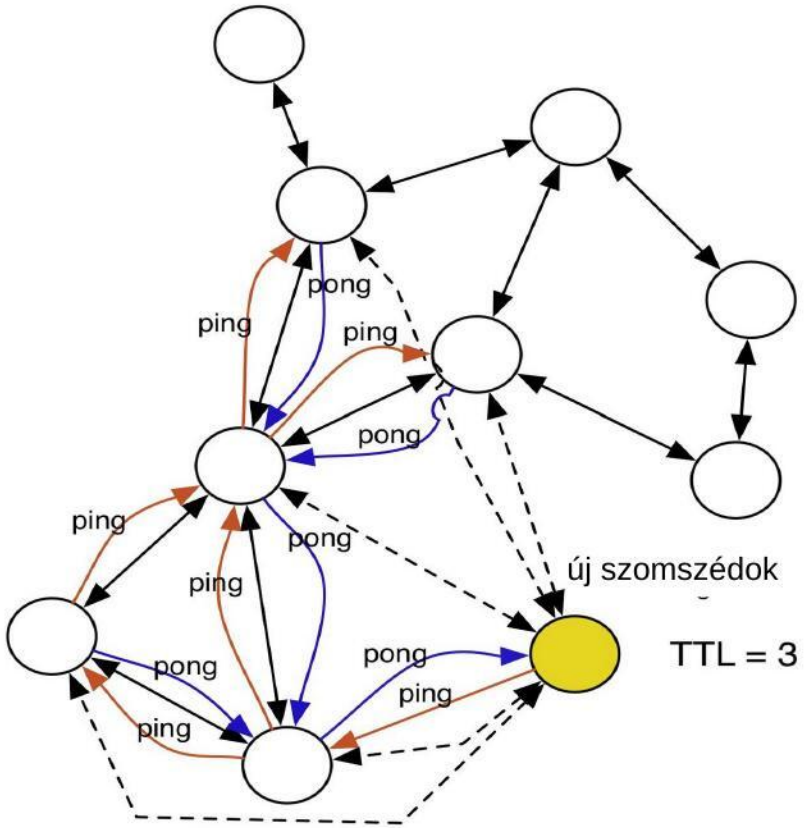
Blakley titok megosztás sémája

- George Blakley, 1979
- Feladatok
  - n személy megoszt egy titkot
  - csak akkor lehet megtudni a titkot, ha k személy jelen van
- Blakley sémája
  - A k-dimenziós térben k (lineárisan független) k-1-dimenziós hipersík metszete definiál egy pontot
  - ez a pont a titkos információ
  - k-1 hipersík egy egyenest határoz meg
- Konstrukció
  - Egy harmadik (megbízható) instancia generál egy $x \in R^k$ ponthoz n hipersíkot, melyek közül bármely k lineárisan független



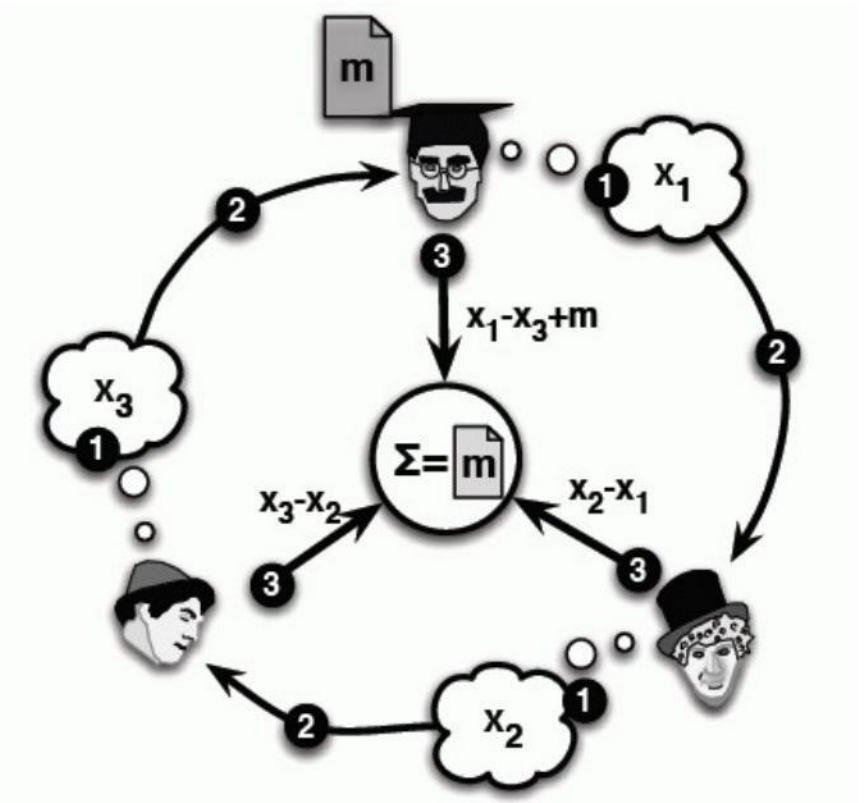
Gnutella – eredeti változat - kapcsolódás

- Szomszédság-listák
  - Gnutella kliens közvetlenül kapcsolódik más kliensekhez
  - Belépésnél az új kliensnek találni kell legalább egy másik csomópontc (bootstrap nodes, gnutella web caches)
  - Ezek kipróbálásra kerülnek, amíg egy aktív jelentkezik
  - Egy aktív kliens ekkor továbbadja a szomszédság-listáját
  - A szomszádság-listákat a kliens mindig tovább hosszabbítja és eltárolja
  - Az aktív szomszédok száma korlátozva van (tipikusan 10-re)



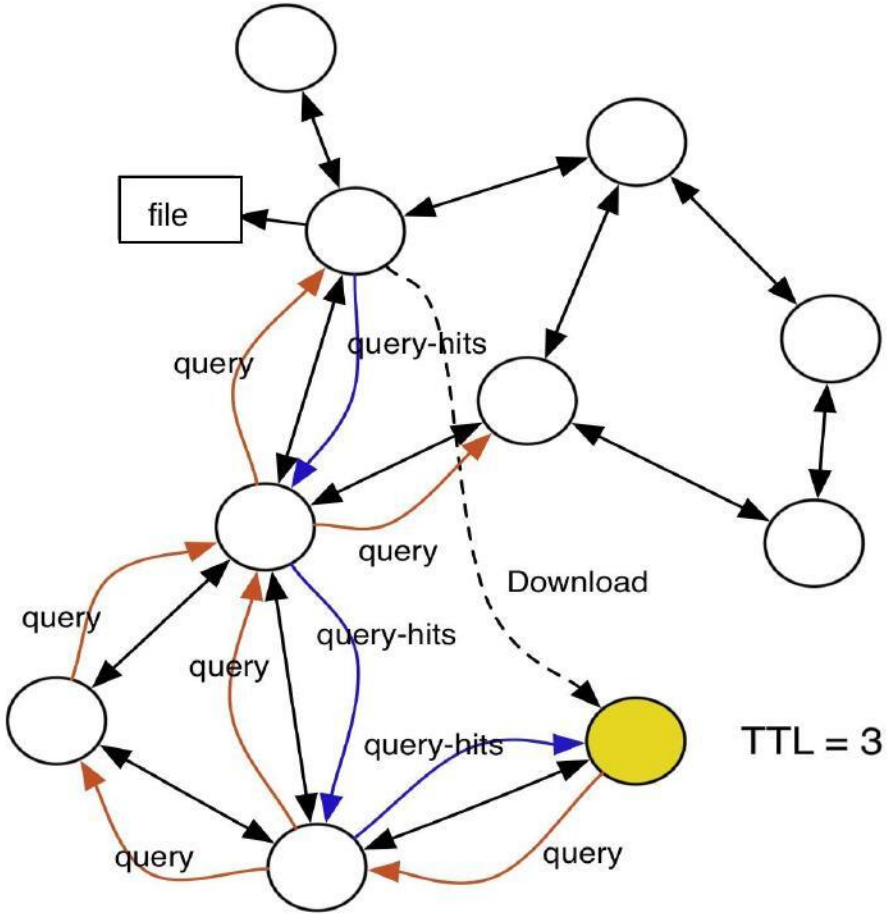
Az vacsorázó kryptográfusok

- Anonim publikálás visszakövethetőség nélkül
- n ≥ 3 kryptográfus ül egy asztalnál
  - Szomszédos kriptográfusok titkosan tudnak egymással kommunikálni
- Minden peer i választ magának egy titkos számot: x_i és ezt elküldi a jobboldali szomszédjának ((i mod n)+1-nek)
- Ha i egy m üzenetet akar publikálni, akkor nyilvánosságra hozza
  - $s_i = x_i - x_{i-1} + m$
- egyébként pedig
  - $s_i = x_i - x_{i-1}$
- Minden peer kiszámítja: $s=s_1+...+s_n$
  - ha s=0, akkor nincs üzenet
  - különben pedig s = m az üzenet



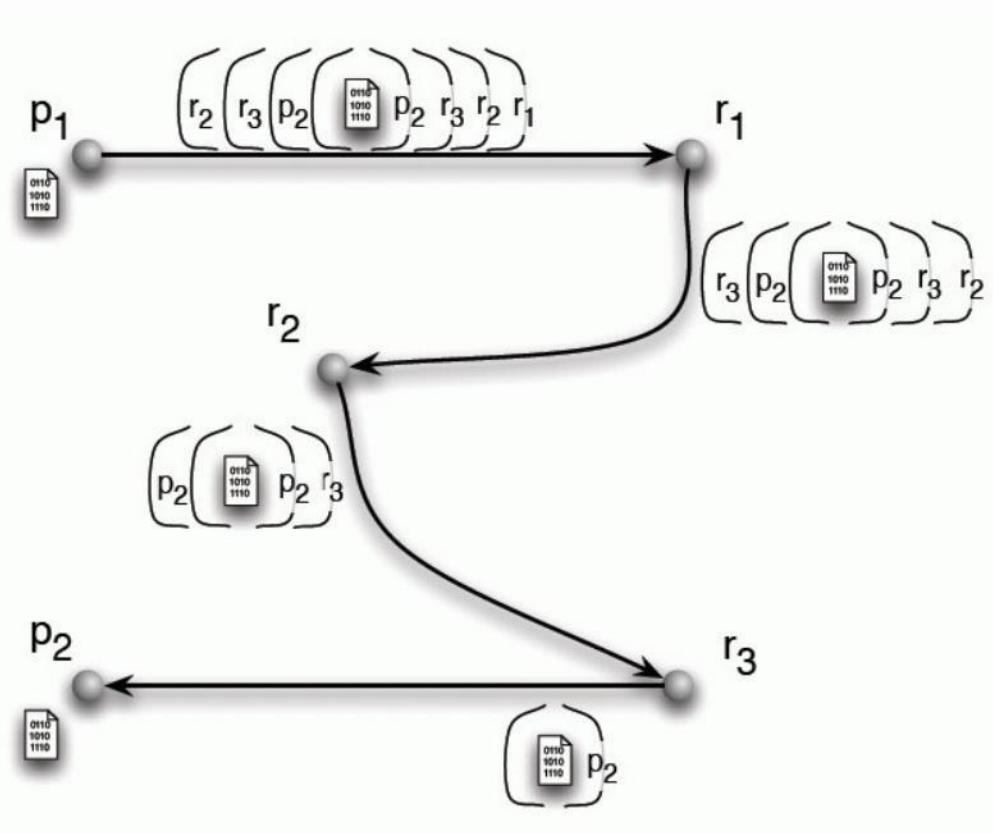
Gnutella – eredeti verzió – file lekérése

- File kérés
  - a kérő elküldi minden szomszédjának
  - a szomszédok továbbküldik a saját szomszédjaiknak
  - egy adott mélységig (#hops)
    - TTL-Feld (time to live)
- Protokoll
  - `Query`
    - A file kérése TTL-ben adott link mélységig továbbítódik
  - `Query-hits`
    - Válasz ugyanazon az útvonalon visszafelé
- Ha file-t megtaláltuk, direkt letöltés



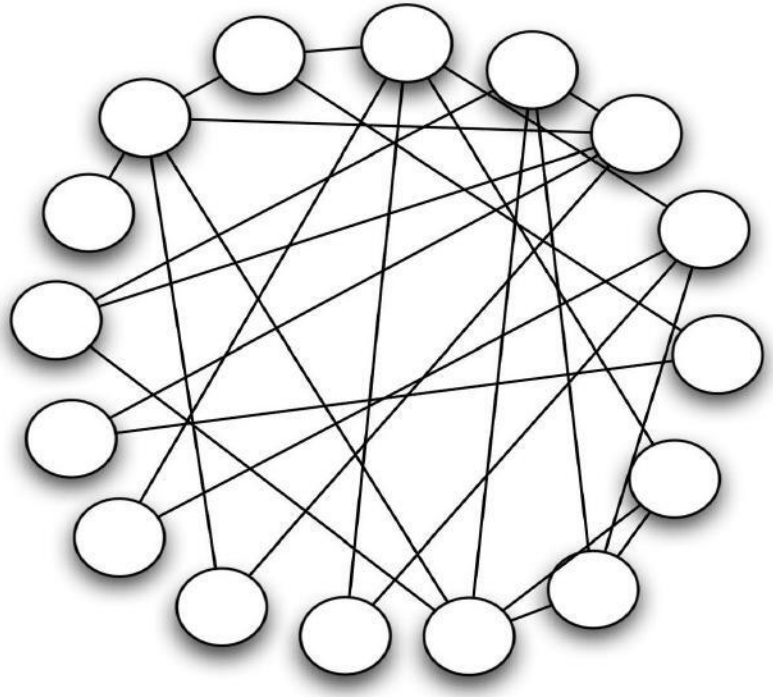
Chaum „mix cascades” módszere

- Az úton egyik peer se tudja
  - hogy hanyadik állomás az úton
  - visszafejteni az eredeti üzenetet
  - ki a végső állomás
- A fogadó nem tudja
  - ki a küldő
- Ezen felül a peer-ek önkéntesen hozzáadhatnak a kerülőutakat a továbbítandó üzenethez
- Chaum Mix Cascades módszere
  - vagy Mix Networks
  - biztonságos mindenféle támadás ellen,
  - de forgalom elemzés ellen nem



Miért nem skálázható Gnutella?

- Gnutella
  - Gráf- struktúra egy véletlen kapcsolat-gráf
  - A gráf foka kicsi
  - A gráf átmérője kicsi
  - Összefüggősége nagy
- A keresés azonban erőforrásigényes
  - ahhoz, hogy biztosan megtaláljunk egy adatot, az egész hálózatot át kell keresni
- Gnutella nem skálázható, mert
  - nics struktúrája az adatok tárolásának

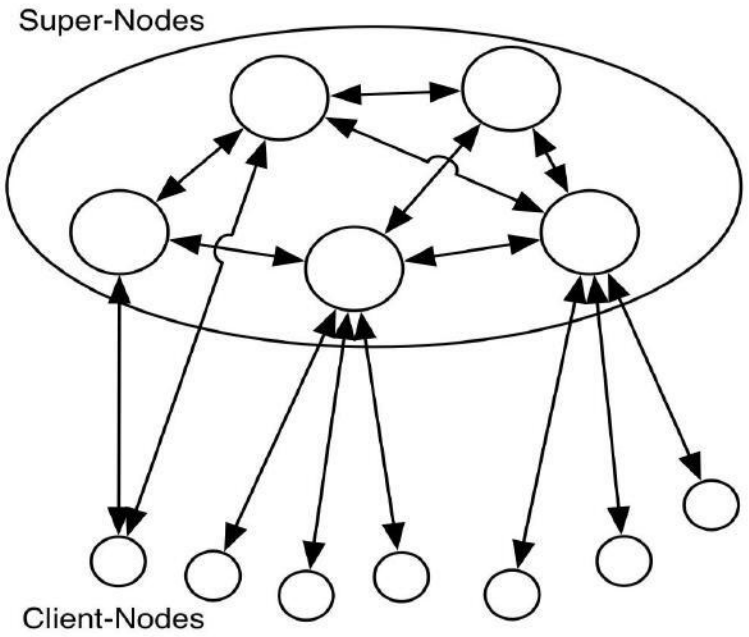


További hagyma routereken alapuló munkák

- Crowds
  - Reiter & Rubin 1997
  - Anoním web-böngészés, amely onion-routereken alapul
- Hordes
  - Shields, Levine 2000
  - Alcsoportokat használ az onion-routing javításához
- Tarzan
  - Freedman, 2002
  - Egy P2P anonimizáló hálózat réteg
  - UDP üzeneteket használ és Chaum Mix-et csoportokban az Internet forgalom anonimizálásához
  - „fake” forgalmat is generál időzítés támadások ellen

Miért nem skálázható Kazaa és Co.?

- Hibrid struktúra
  - Átmérő kicsi
  - Az összefüggőség választható nagyra
    - a klienseknél a szuper-csomópontok száma által
  - A fokszám kicsi
- Skálázhatóság
  - nem olyan rossz, mint Gnutella-nál vagy Napster-nél
  - Nem jó, mert minden szuper-csomópont, megkapja a kliensek kéréseit



Freenet

- Hálózati struktúra
  - Gnutella-hoz hasonló
  - Freenet fokszámeloszlása Pareto (hasonlóan Gnutella-hoz)
- File-ok tárolása
  - Minden file megkereshető, visszakódolható és olvasható a kódolt cím-sztringet használva és az aláírt altér-kulcsot
  - Minden file együtt van tárolva az index kulcsával, de a kódolt cím-sztring nélkül
  - A tároló peer nem tudja ezt a file-t olvasni
    - Hacsak nem próbálja ki az összes lehetséges jelszót (szótár támadás)
- Index file-ok tárolása
  - A cím-string egy kryptografikus hash függvénnyel van kódolva, amely ahhoz a peer-hez vezet, amely tárolja
    - az indexet
    - a cím-sztringet
    - az aláírt altér kulcsot
  - Ezt az index file-t használva megtalálható az eredeti file

Content Addressable Network (CAN)

Két kérdés az információ keresésénél

- Hol van?
- Hogy jutunk oda?

- Napster:
  - Hol?
    - A szerveren ☺
  - Hogy jutunk oda?
    - torlódás/dugó a szerveren ☹
- Gnutella
  - Hol?
    - Nem tudjuk ☹
  - Hogy jutunk oda?
    - Mindenkit megkérdezünk ☹

- Egy jobb ötlet:

- Hol van az x adat?
  - Az f(x) helyen

  - Mi az az f(x)?
    - x egy minden résztvevő számára ismert leképezése egy térre

- Hogy jutunk oda?
  - Egy jól definiált útvonalon, amely a kérdező helyétől f(x) helyréhez vezet.

Freenet

- Keresés
  - Legnagyobb lejtés - hegymászás
    - A keresés ahhoz a peer-hez továbbítódik, melynek az ID-je legközelebb van a kereső indexhez
  - TTL mező (azaz hop korlát)
- A file-ok új peer-ekre kerülnek áthelyezésre
  - Ha a kulcsszó hasonló a szomszéd ID-jához
- Új linkek
  - keletkeznek, ha egy keresés közben peer Id-k közötti hasonlóságot derítünk ki

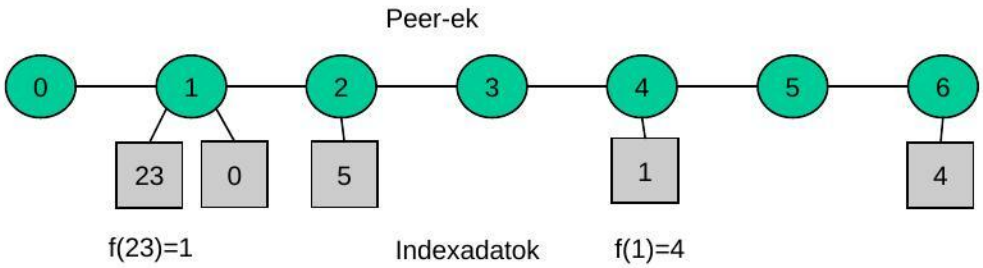
Hash-táblától az elosztott hash-táblához (DHT)

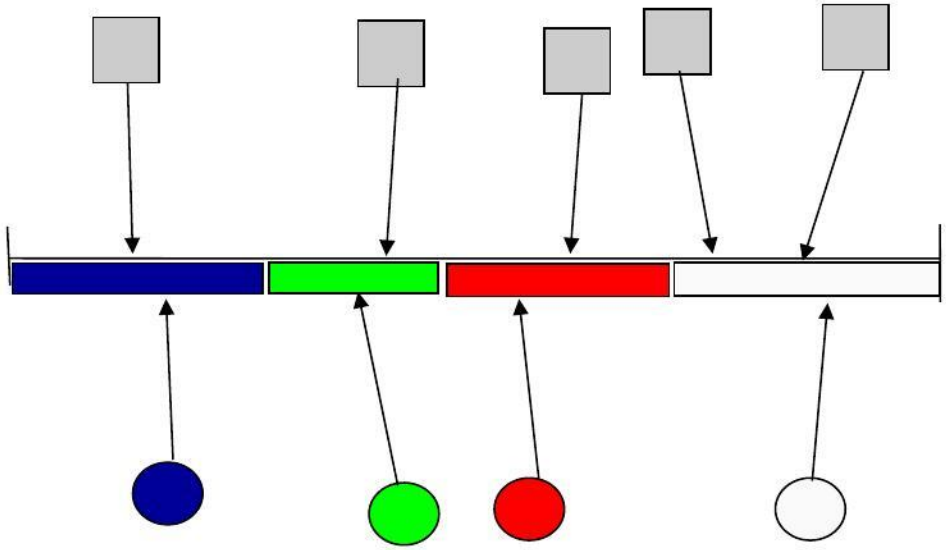
Hash-Táblák

- Előny
  - A keresés egyszerű
- Hátrány
  - Egy új Peer kapcsolódásakor új hash-függvényt kell választani
  - Hosszú utak, nagy élterhelés

Elosztott Hash-Tábla
(Distributed Hash Table, DHT)

- A Peer-eket leképezzük (hashing által) egy helyre és minden Peer-hez hozzárendeljük a hash-függvény értéktartományának egy részét
- Az adatokat is hash-eljük
  - A hash-függvény értéké alapján a tartományért felelős Peer-en tároljuk (azon a peeren, amelyik képe a legközelebbi az adat képéhez.)



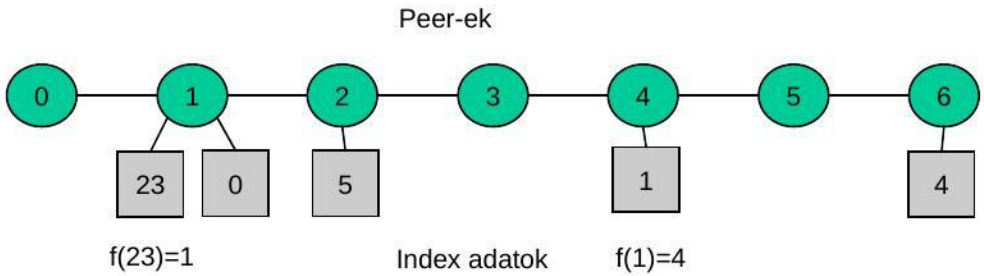


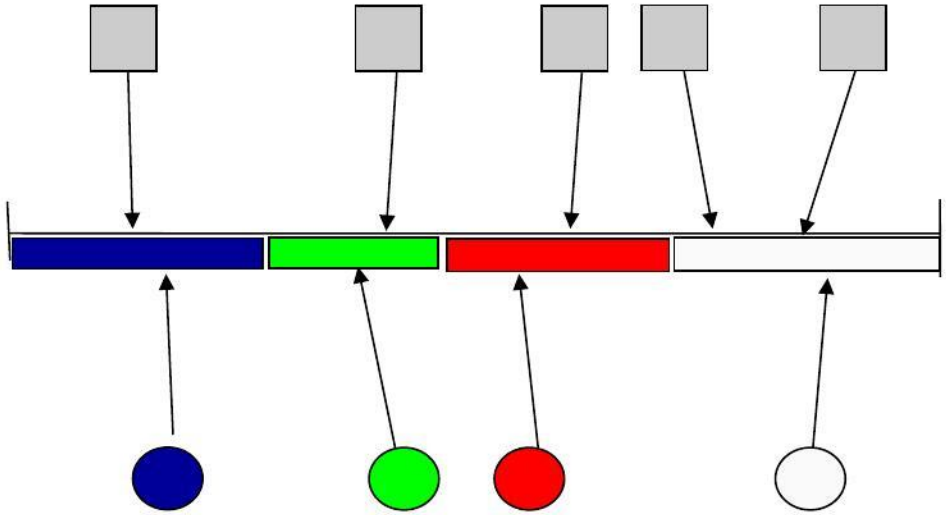
Dark Net & Friend-to-Friend

- Dark-Net egy privát Peer-to-Peer hálózat
  - A tagok megbízhatnak minden más tagban
  - pl.
    - barátok (a valódi világban)
    - sport klub
- Dark-Net a hozzáférés kontroll:
  - titkos címekkel,
  - titkos szoftverrel,
  - jelszó általi autentikálással, vagy
  - központi autentikálással
- Példa:
  - WASTE
    - P2P-Filemegosztás 50 tagig
    - Nullsoft (Gnutella) által
  - CSpace
    - Kademlia használatával

DHT tulajdonságai

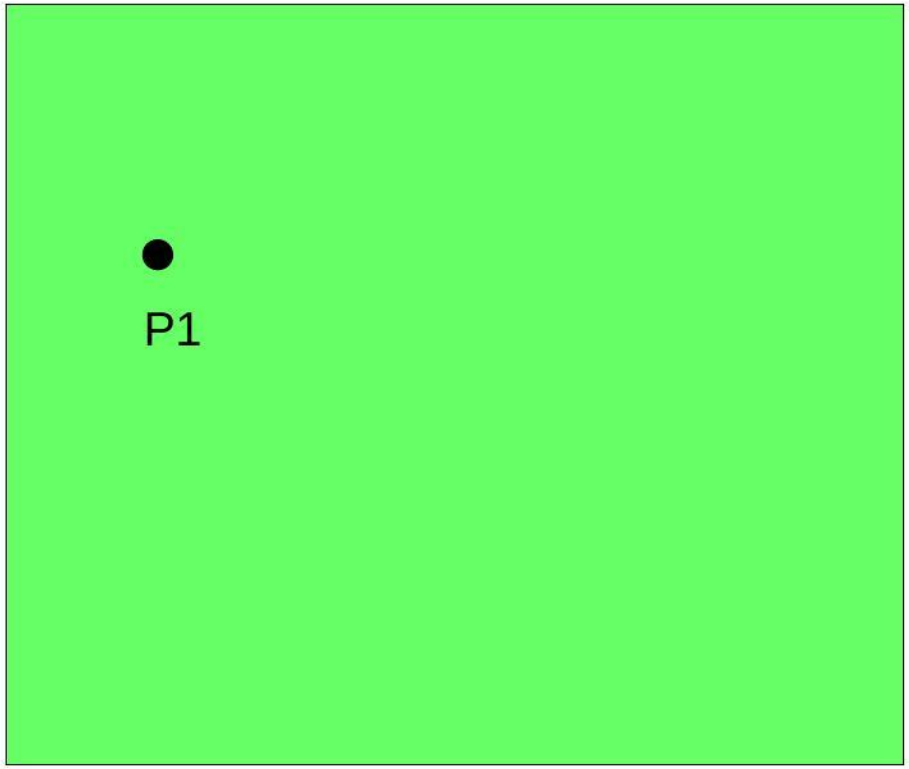
- Előny
  - Minden adatot egyértelműen hozzá lehet rendelni egy Peer-hez
  - Egy Peer csatlakozása vagy kilépése a hálózatból csak a szomszédainál okoz változást
- DHT-t sok P2P hálózat használ
- Még tisztázni kell:
  - Az kapcsolódások struktúráját





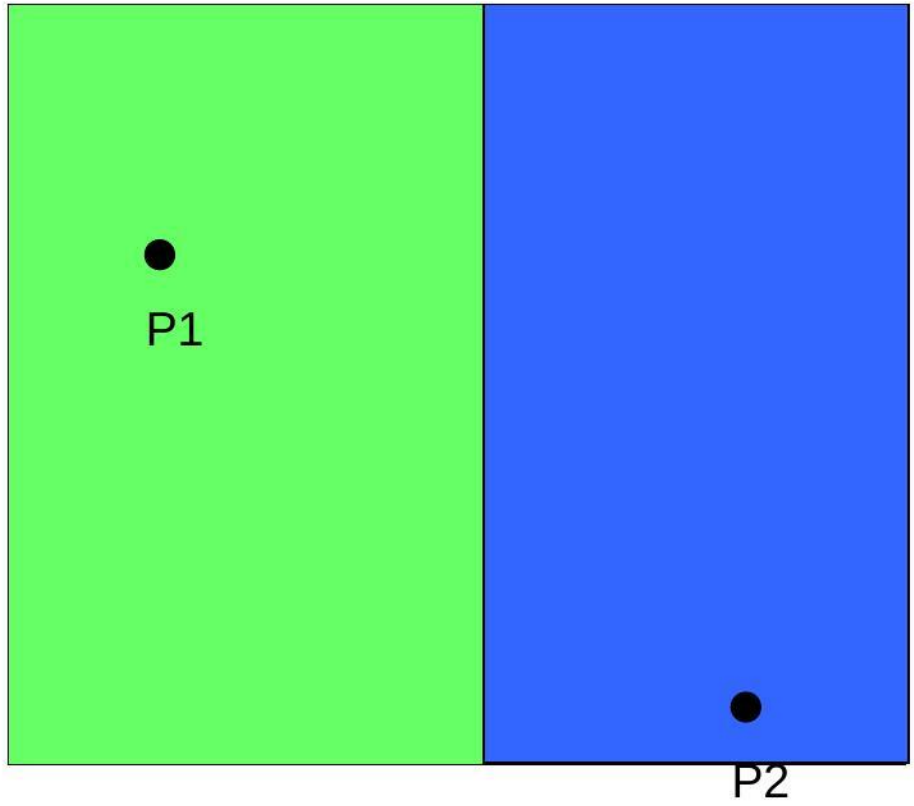
Content Addressable Network (CAN)

- A Peer-ek és a file-ok egy (kétértékű) hash-függvénnyel az egységnégyzetbe képeződnek
- Kezdetben egy üres négyzet és egyetlenegy Peer mint tulajdonos
  Ez a négyzet a Peer zónája (tartománya)



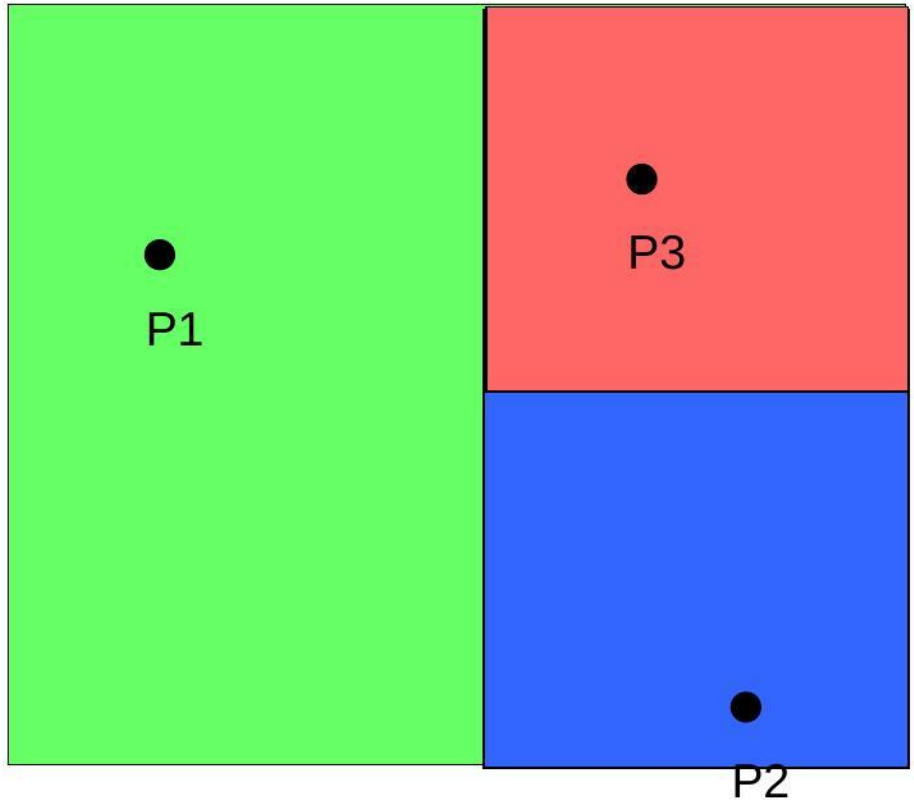
Content Addressable Network (CAN)

- A Peer-ek és a file-ok egy (kétértékű) hash-függvénnyel az egységnégyzetbe képeződnek
- Kezdetben egy üres négyzet és egyetlenegy Peer mint tulajdonos
  Ez a négyzet a Peer zónája (tartománya)
- Egy tartomány tulajdonosa minden adatot tárol, amely arra a tartományra képeződik le
- Egy Peer választ egy véletlen pontot a négyzetben (hash-függvény)
  - A megfelelő négyszög tulajdonosa kettéosztja a négyszöget és
  - átadja a felét az új Peer-nek



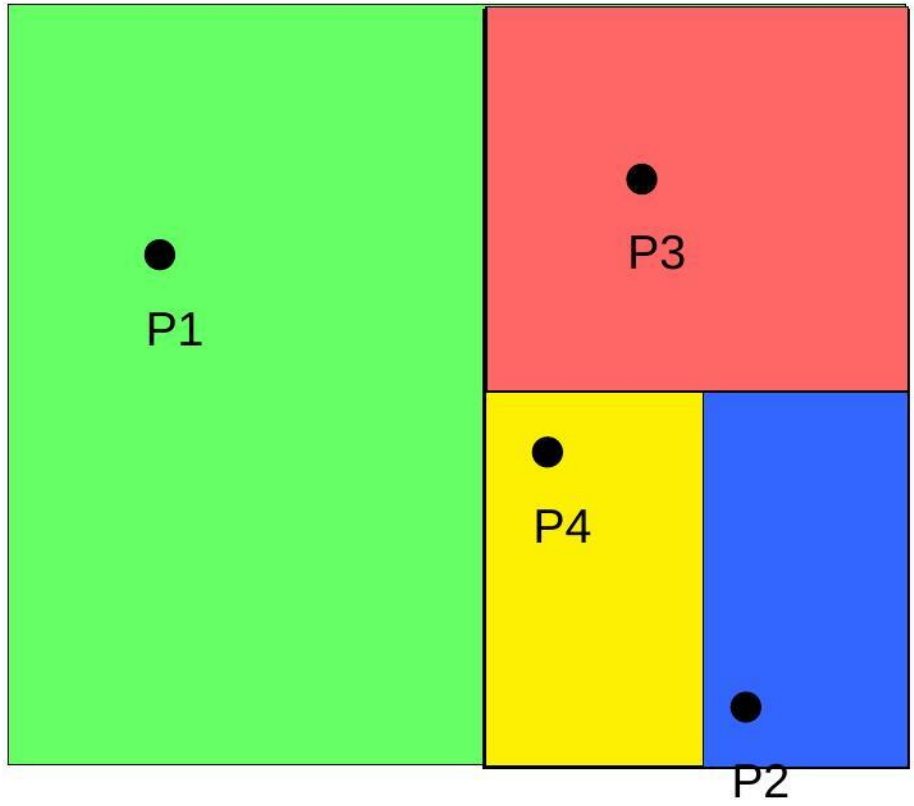
Content Addressable Network (CAN)

- A Peer-ek és a file-ok egy (kétértékű) hash-függvénnyel az egységnégyzetbe képeződnek
- Kezdetben egy üres négyzet és egyetlenegy Peer mint tulajdonos
  Ez a négyzet a Peer zónája (tartománya)
- Egy tartomány tulajdonosa minden adatot tárol, amely arra a tartományra képeződik le
- Egy Peer választ egy véletlen pontot a négyzetben (hash-függvény)
  - A megfelelő négyszög tulajdonosa kettéosztja a négyszöget és
  - átadja a felét az új Peer-nek



Content Addressable Network (CAN)

- A Peer-ek és a file-ok egy (kétértékű) hash-függvénnyel az egységnégyzetbe képeződnek
- Kezdetben egy üres négyzet és egyetlenegy Peer mint tulajdonos
  Ez a négyzet a Peer zónája (tartománya)
- Egy tartomány tulajdonosa minden adatot tárol, amely arra a tartományra képeződik le
- Egy Peer választ egy véletlen pontot a négyzetben (hash-függvény)
  - A megfelelő négyszög tulajdonosa kettéosztja a négyszöget és
  - átadja a felét az új Peer-nek



Content Addressable Network (CAN)

- A Peer-ek és a file-ok egy (kétértékű) hash-függvénnyel az egységnégyzetbe képeződnek
- Kezdetben egy üres négyzet és egyetlenegy Peer mint tulajdonos
  Ez a négyzet a Peer zónája (tartománya)
- Egy tartomány tulajdonosa minden adatot tárol, amely arra a tartományra képeződik le
- Egy Peer választ egy véletlen pontot a négyzetben (hash-függvény)
  - A megfelelő négyszög tulajdonosa kettéosztja a négyszöget és
  - átadja a felét az új Peer-nek