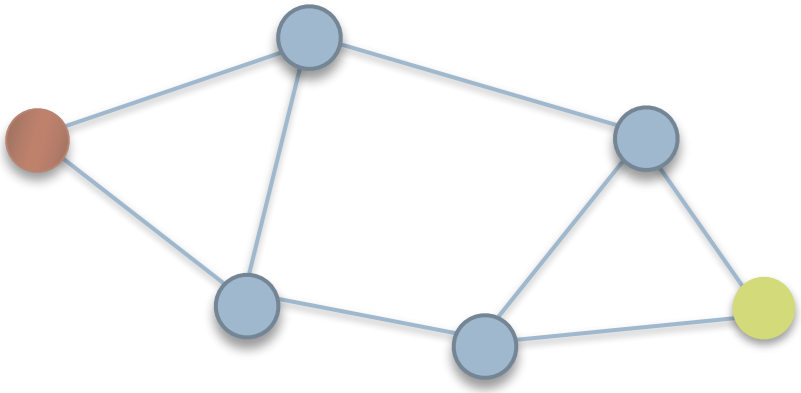
Hálózatok építése és üzemeltetése

Többutas átvitel

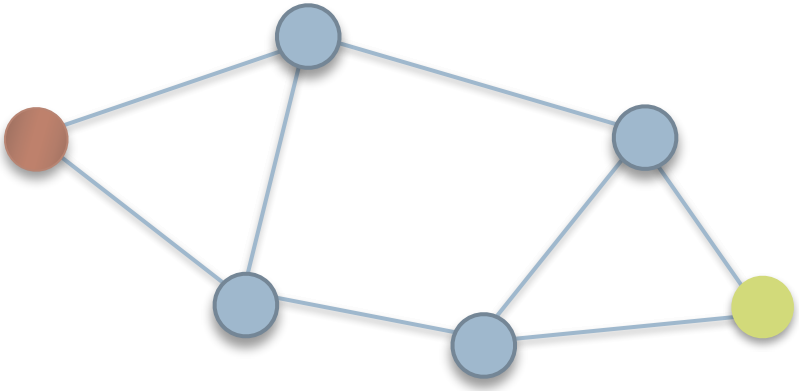
Mai téma

- Adatok továbbítása párhuzamosan több útvonalon



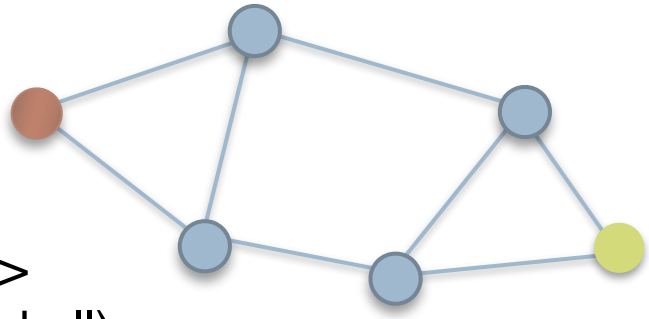
Többutas átvitel előnyei

- Gyors
  - több út nagyobb sebesség
- Hibatűrő
- Biztonságos
  - egy ponton figyelve nem lehet a forgalmat lehallgatni
- Hatékony
  - a hálózat kihasználtsága növelhető



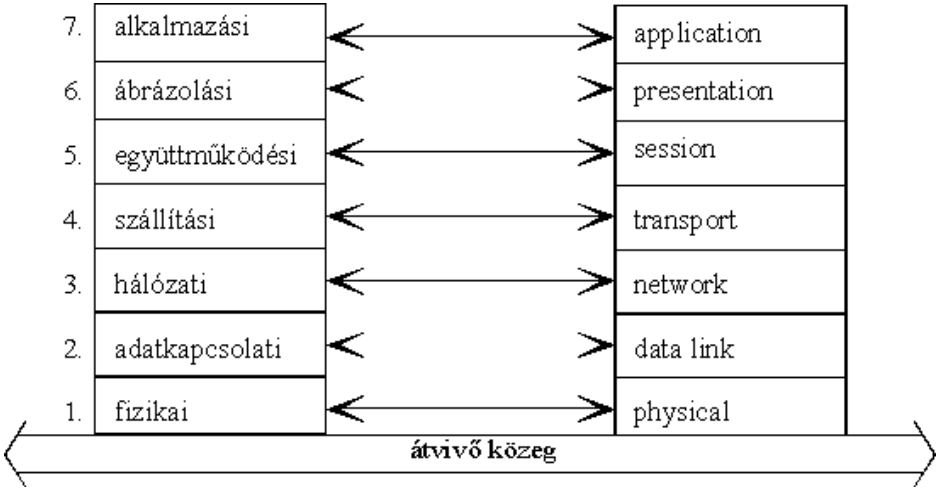
Többutas átvitel nehézségei

- Ha csak 1 út kell jó a fa topológia
  - fán minden egyszerűbb
  - egyszerűbb protokollok, gyors és olcsó kapcsolóeszközök
- Körök a gráfban -> bonyolultabb protokollok-> drágább eszközök (hurokvédelemhez állapot kell)
- Fejlett útválasztás
- Fejlett hálózatmenedzsment
- Fejlett hálózatmonitorozás
- Hagyományos architektúrában ezek pont nem járnak az élen, de majd az SDN-el



Többutas átvitel és rétegek

- OSI rétegek

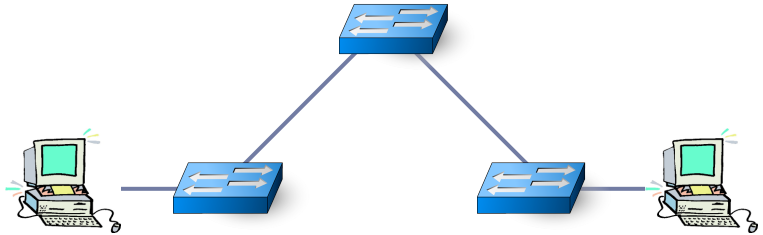


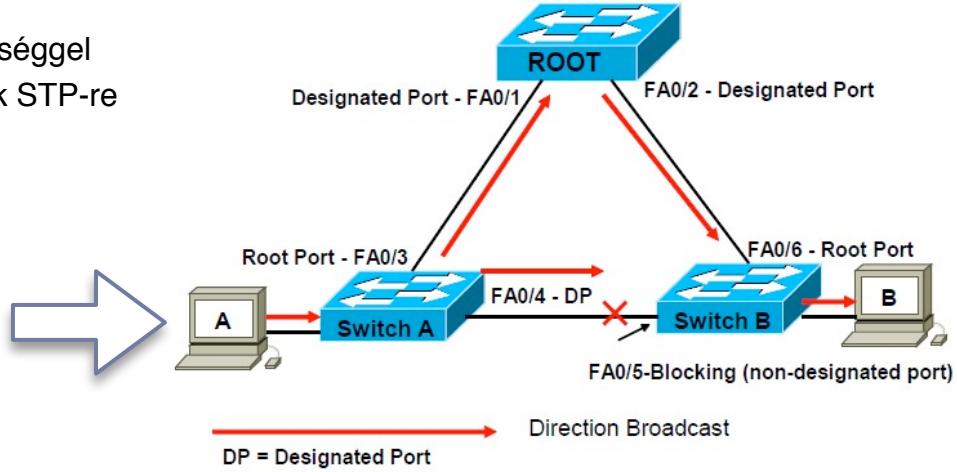
Többutas átvitel a 2-3 rétegekben

Többszörös feszítőfák

Feszítőfa protokoll (Spanning Tree Protocol STP)

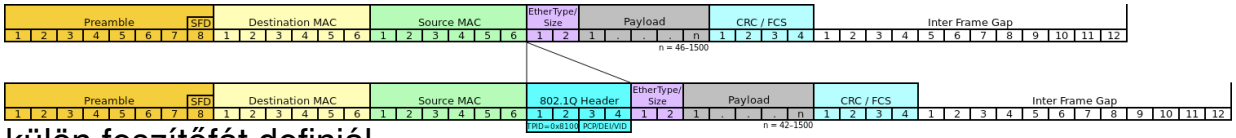
- Nincs kör nincs, olcsó egyszerű, de nem túl hatékony hálózat
  - nincs hibatűrés, alacsony átviteli sebesség, alacsony kihasználtság
- Ha lehet, akkor minimális extra művelettel valami extra szolgáltatás
- Root switch kiválasztása
- Root-switch-től BPDU (Bridge Protocol Data Units) 0-s költséggel mindenhova
- BPDU-t minden switch továbbküldi megnövelve a link költséggel
- Több útvonal csak időbeli eltolással: pl. átkapcsolás másik STP-re hálózati hiba esetén



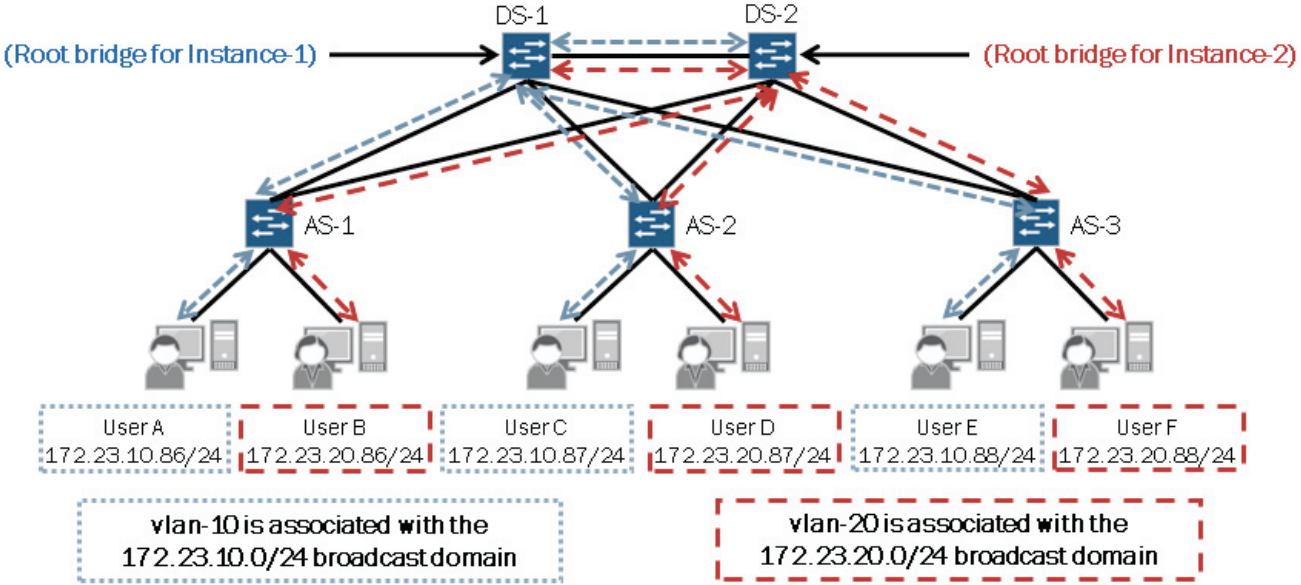


Többszörös feszítőfa protokoll (MSTP)

- VLAN 802.1Q

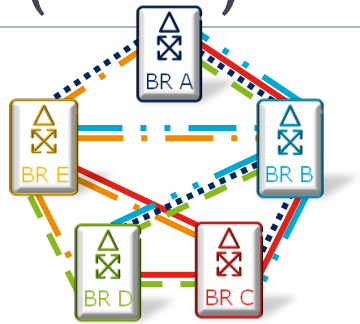


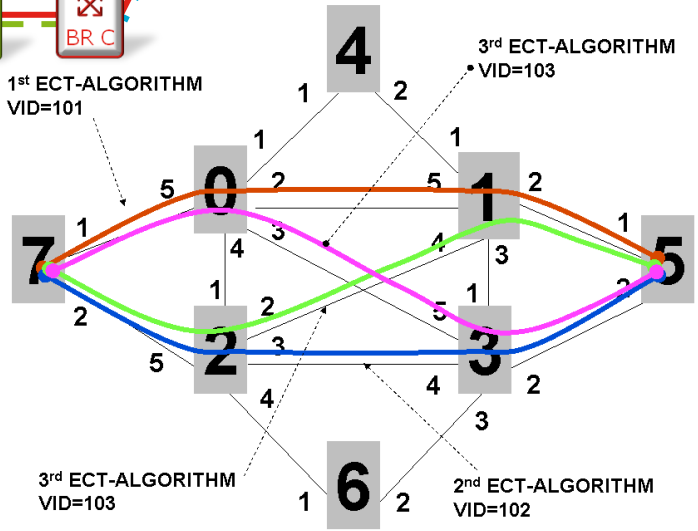
- Minden VLAN-hoz külön feszítőfát definiál
- Felhasználó szintű granularitást már tud: különböző VLAN-hoz tartozó csomagok különböző utakat járnak be



Shortest path bridging (SPB)

- IS-IS minden switch-en
- Minden IS-IS példányhoz saját feszítőfa
- Service ID->Vlan ID leképzés
  - így lehet dinamikusan szolgáltatásokat VLAN-hoz rendelni
- ECMP algoritmussal több útvonal használata azonos VLAN-on belül
- Determinisztikus multipath: több útvonal közül hash kód alapján választ
- Granularitás: folyamonként más útvonal



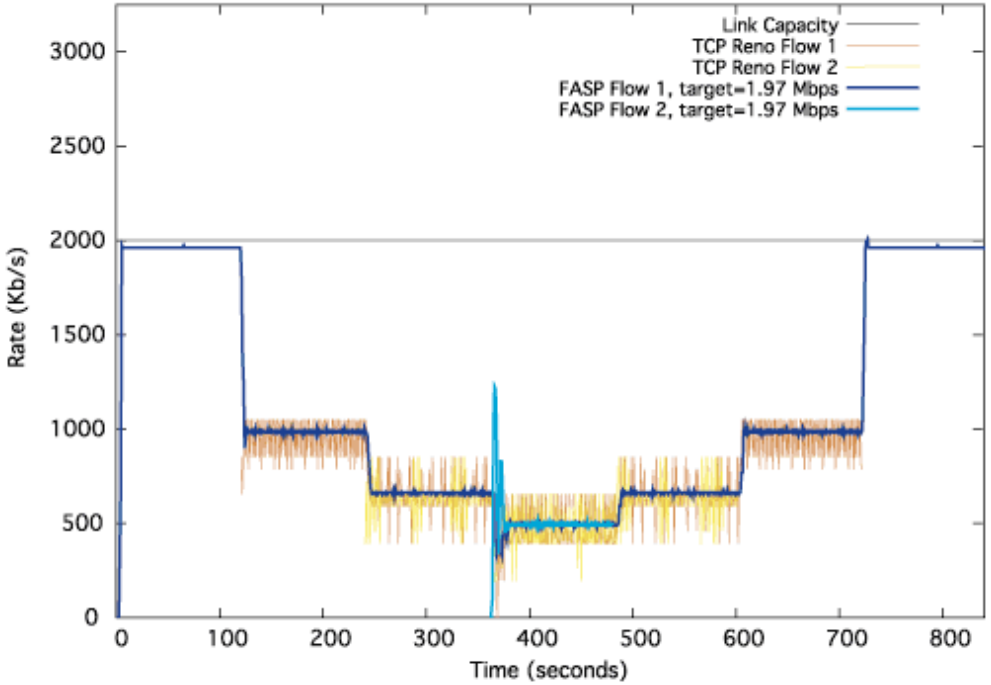


Többutas átvitel a 4. rétegben

Multipath TCP

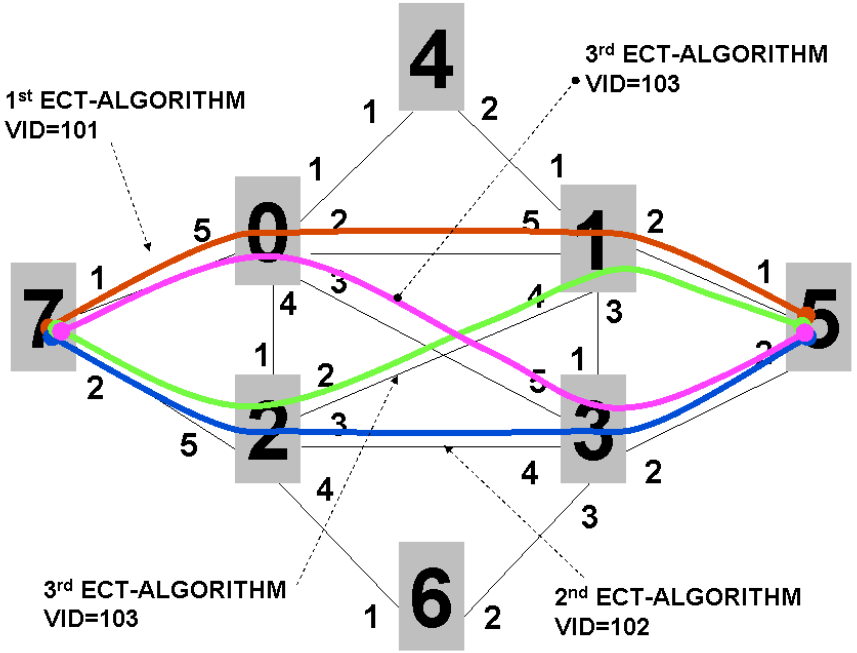
TCP protokoll áttekintés

- Elosztott önszerveződő torlódáskezelés
- A felhasználók valósítják meg
- A TCP kliensek a torlódási ablak állításával saját magukat szabályozzák vissza, így a felhasználók fair módon férhetnek hozzá a hálózati erőforrásokhoz
- Nem kellene drága bufferek, és csomagütemezők
- Olcsóbb hálózat



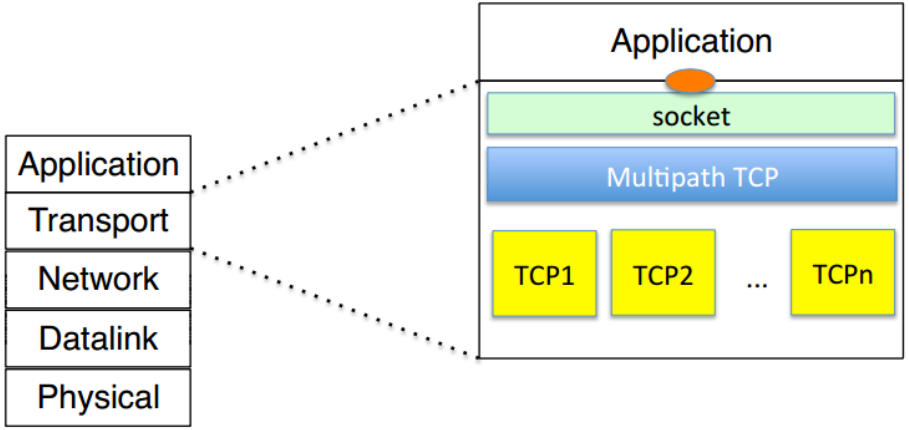
Hálózatok, többszörös útvonalkezelés

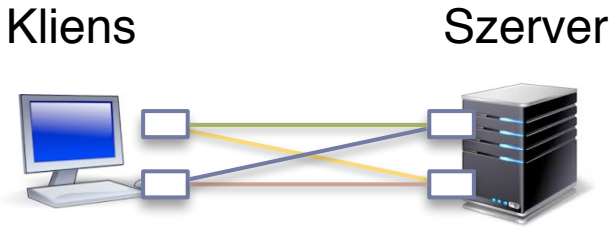
- A hálózat aktívan kell, hogy monitorozza magát és az aktuális viszonyokhoz kell, hogy igazítsa a bejövő forgalmat
- Többféle hardver: monitorozó eszközök, load balancer, útvonalfigyelés stb.
- Drágább hálózat



Lehet-e a többszörös útvonalak kezelését a felhasználókra bízni?

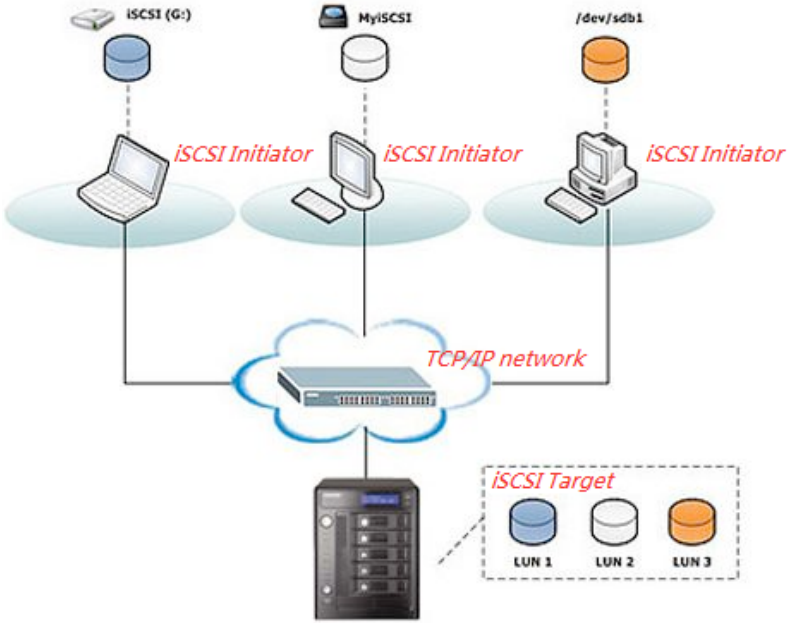
- Multipath TCP
- Egy TCP folyamot több alfolyamra bont szét és ezeket együtt kezeli
- subflow szintű granularitás egy adott felhasználói session is hibatűrőbb és gyorsabb lehet tőle
- Linux implementáció (https://www.multipath-tcp.org/) egyszerűen telepíthető, de saját kernelverzió kell
- alapból a kliens minden interfészéről a szerver minden interfészére indít egy alfolyamot
- Ha a routing megfelelően van beállítva a címek között akkor kihasználja a többszörös útvonalakat

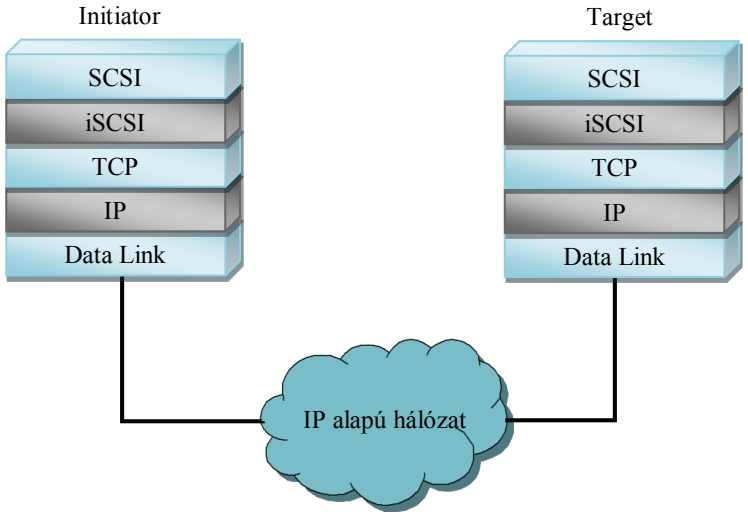




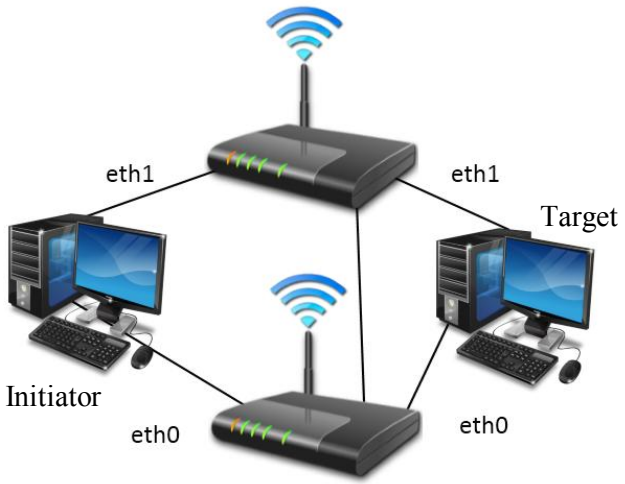
Alkalmazás specifikus példa: iSCSI multipath

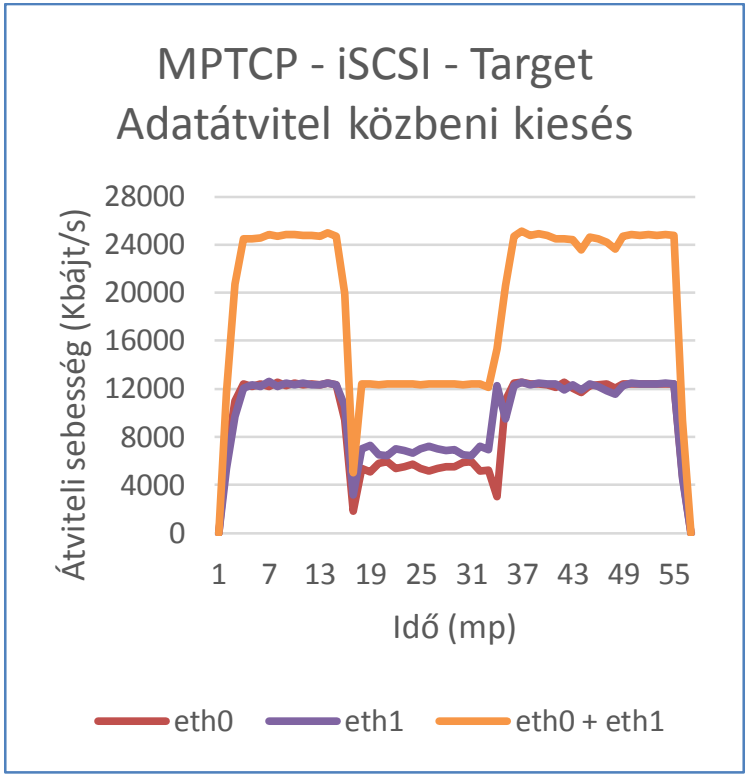
- SCSI protokoll üzenetek átvitele IP csomagokban: olyan mintha helyi tároló lenne





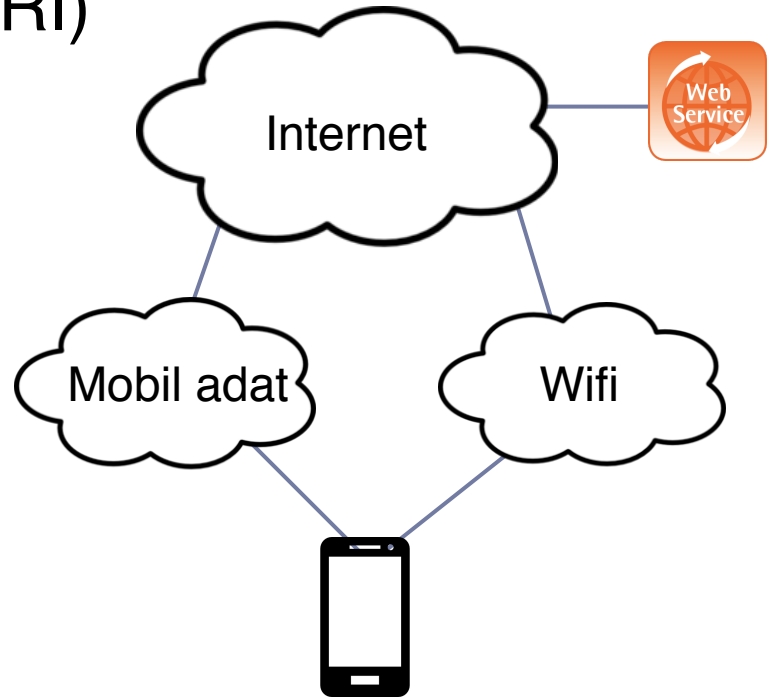
Multipath TCP mérés



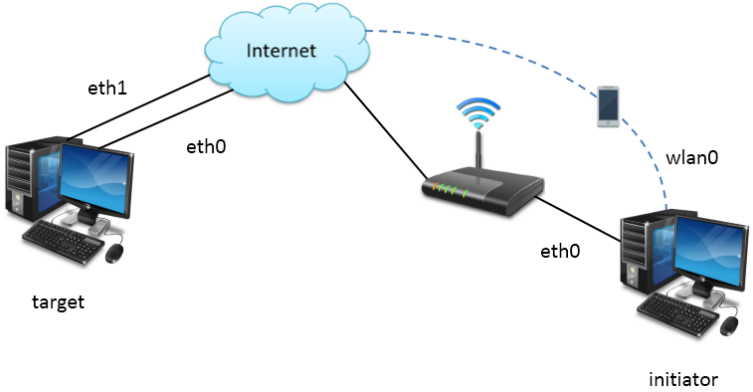


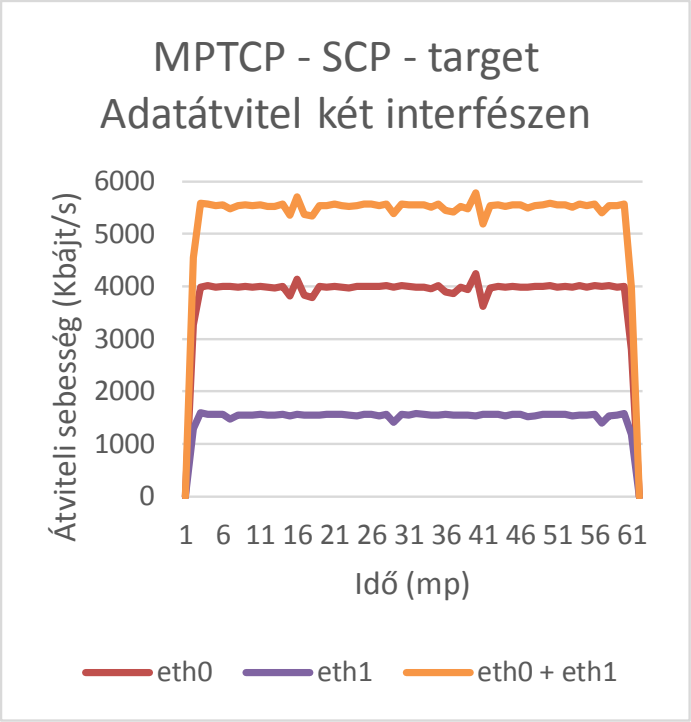
Multipath TCP

- Felhasználások (SIRI)



Multipath TCP valós mérés



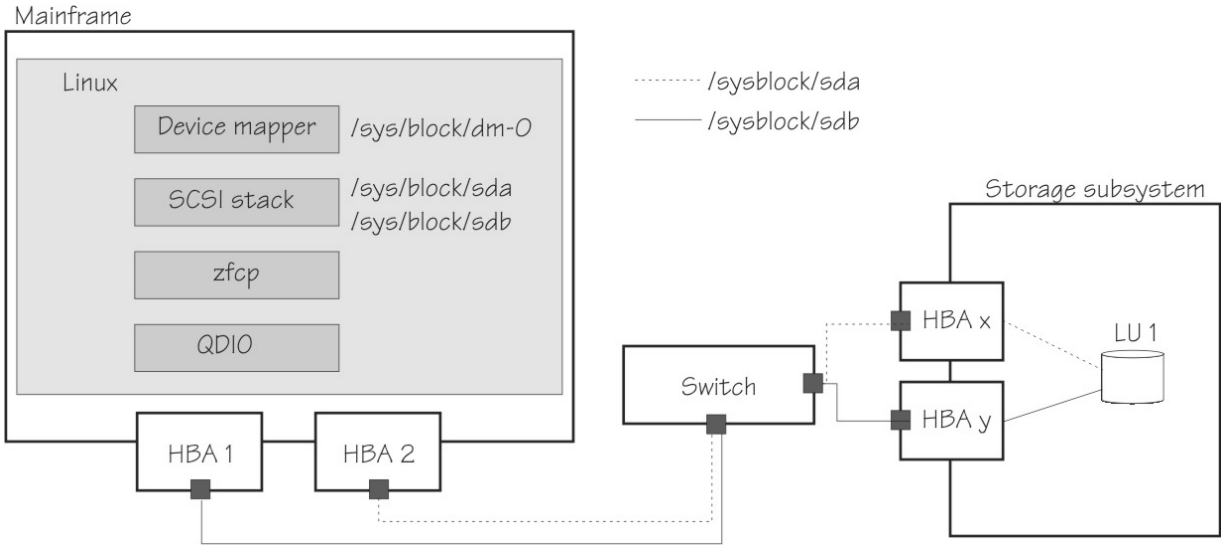


Többutas átvitel a felsőbb rétegekben

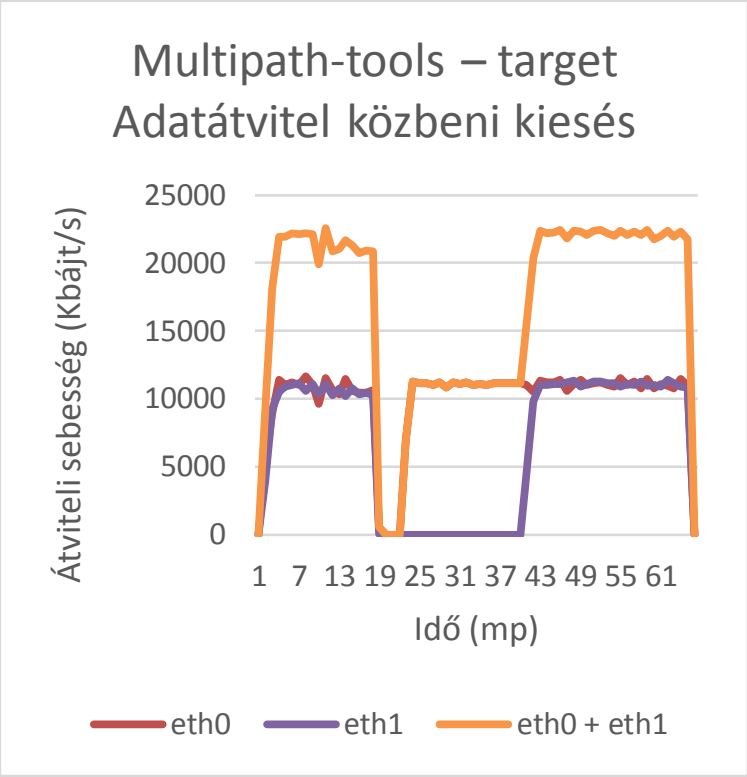
Példa: iSCSI multipath

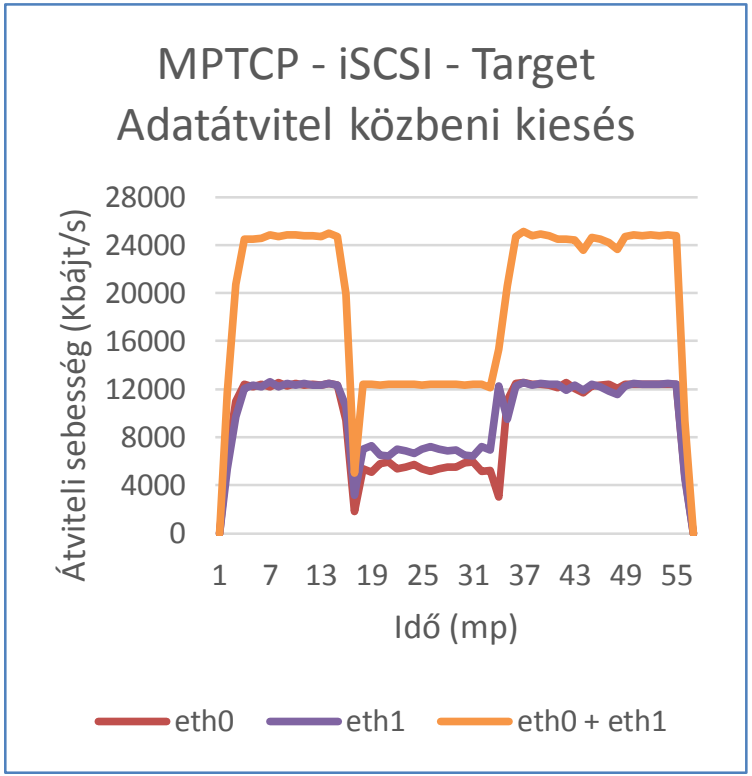
iSCSI multipath

- Multipath tools linux csomag SAN (storage area network) többutas elérésére



iSCSI multipath mérések



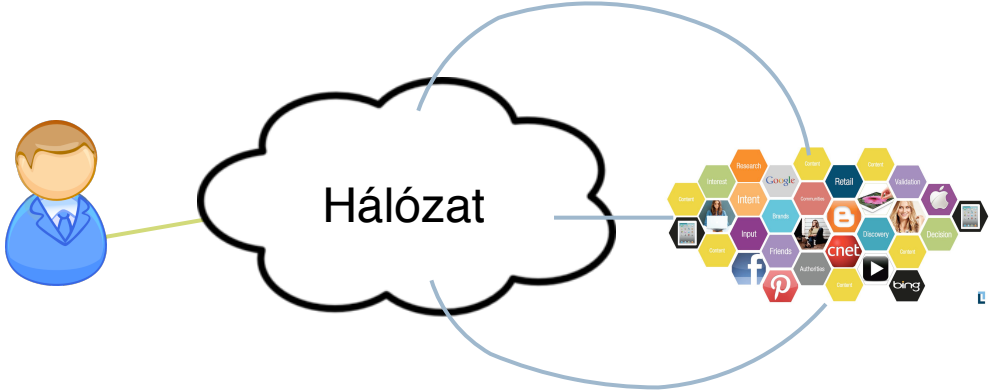


Tartalomelosztó hálózatok (CDN)

Felhasználó-tartalom többutas elérés

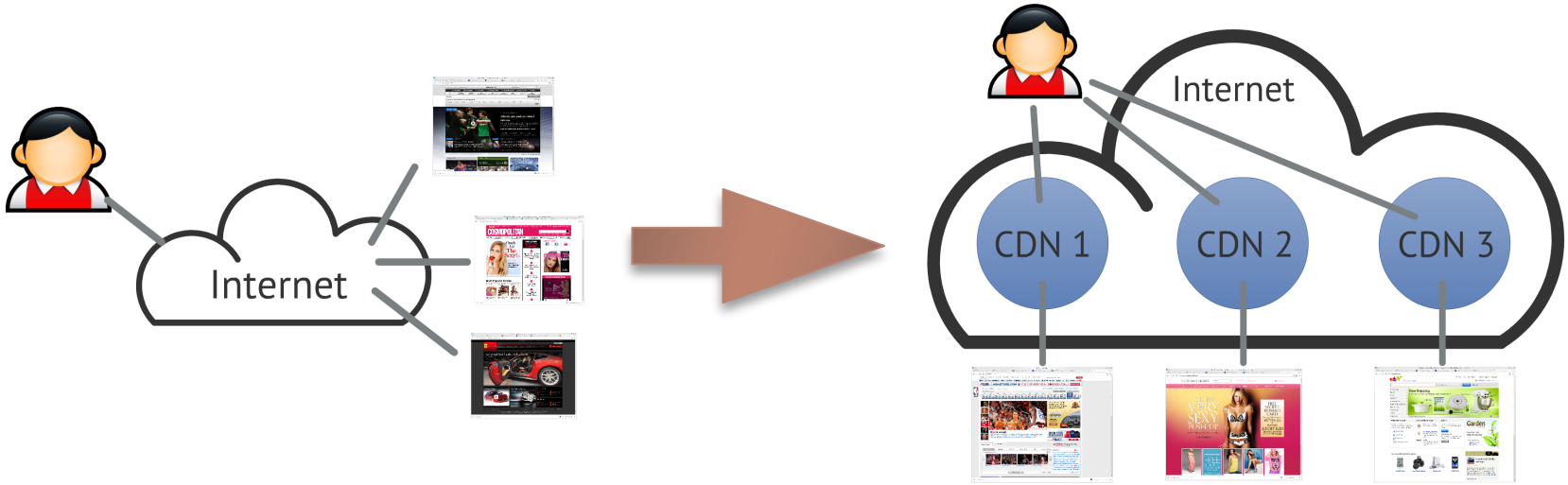
Összeköttetési filozófiák

- Számítógép hálózatok:
  - kliens-szerver architektúra
- Tartalom hálózatok:
  - felhasználó-tartalom összeköttetés



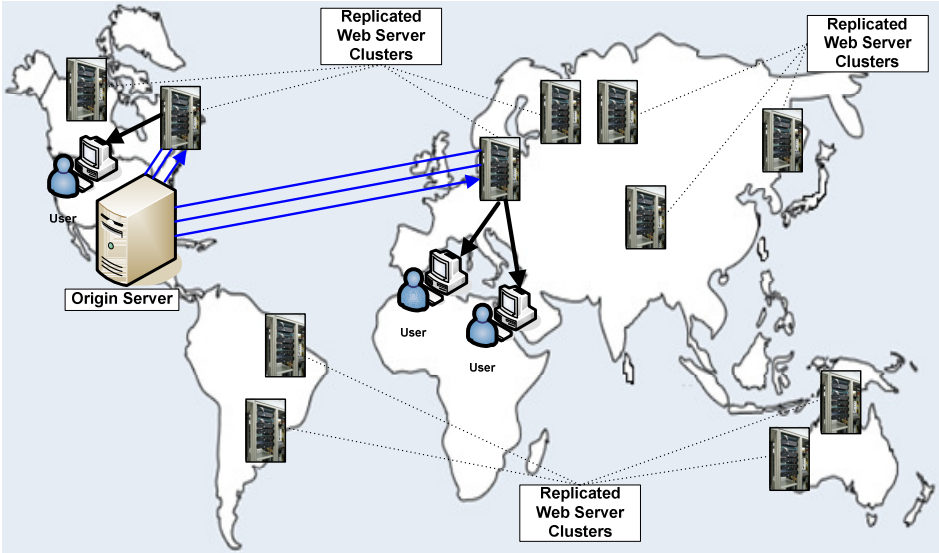
CDN: új tartalomelosztási modell

- Az internet pont-pont filozófiája tartalomelosztáshoz nem illeszkedik jól.



CDN áttekintés

- CDN feladata:
  - tartalom elosztása
  - kérések irányítása
  - monitorozás és számlázás
- Inter domain CDN
  - Pl. nyilvános felhő szolgáltatók (Amazon, Azure stb.)
- Intra domain: saját felhő
  - Regionális szolgáltatók



Kérésirányítás módszerek

- Round robin
  - operációs rendszerekből ismert ütemező eljárás
  - feltételezi, hogy a helyettesítő szerverek képességei hasonlóak
  - bármelyik képes bármilyen kérést kiszolgálni
  - fizikailag közel (egy klaszterben) vannak
  - minden kérést egyenrangúnak kezel (számítási igényben jelentősen eltérhetnek)
  - földrajzilag elszórt szerverek esetén nem hatékony
- Terheltség alapú súlyozásos round robin
  - terheltség mérése a szervereken (pl. eddig kiszolgált igények száma alapján)
  - terheltség mérése a hálózatban
  - súlyok beállítása ennek megfelelően

Kérésirányítás mechanizmusok

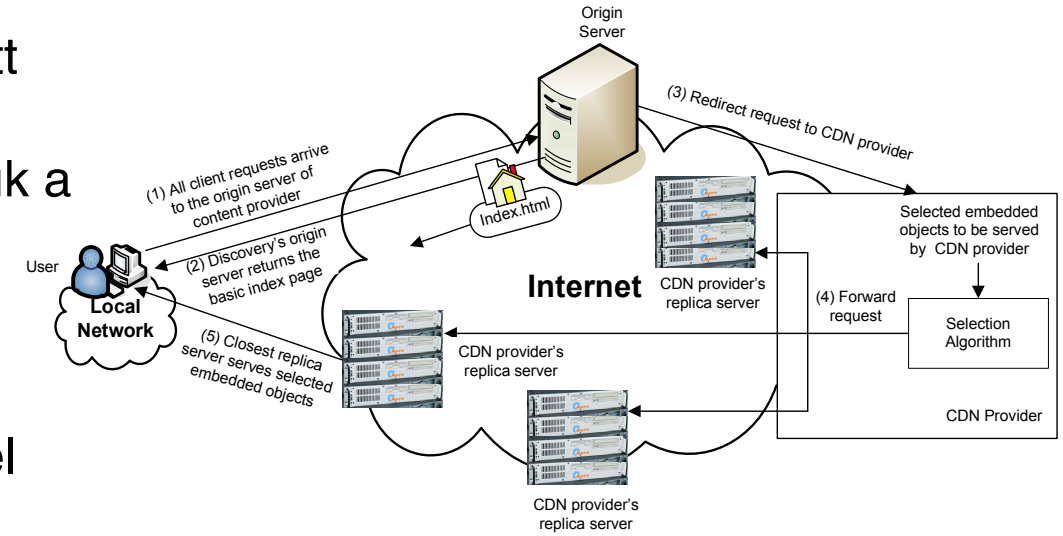
- DNS alapú
- URL átírás
- HTTP redirect
- Anycast

DNS alapú kérésirányítás

- Működése:
  - adott domain névhez több IP is regisztrálva van a DNS-ben
  - ezt a listát adja vissza a DNS lekérdezés
  - innen a helyi resolver választ egyet (random)
  - teljesen transzparens
  - a DNS szolgáltatás mindenütt jelen van, ez optimalizációra ad lehetőséget
  - nagyon egyszerű, szinte semmi plussz hardver/szoftver nem kell hozzá
- Hátrányai
  - DNS névfeloldási mechanizmus késleltetése
  - a felhasználó IP címét nem veszi figyelembe (a DNS kiszolgáló választ IP-t)
  - a DNS cache-ek miatt nem mindig használható előfordul, hogy a kérések 5%-a irányítódik át
  - nem hibatűrő (helyettesítő szerverek kiesése esetén)

URL átírás

- Az eredetszervertől visszakapott html kódban a beágyazott objektumok elérési adatait átírjuk a helyettesítő szerverekre
- Proaktív: az URL-eket előre legeneráljuk
- Reaktív: URL generálás script-el valós időben
- Több helyettesítő szerver párhuzamos használata



- Pl: beágyazott jpg file-ok

  <img src="http://cnn.com/af/foo.jpg>

  may be modified as follows:

  <img src=http://a73.g.akamai.net/7/23/cnn.com/af/foo.jpg>

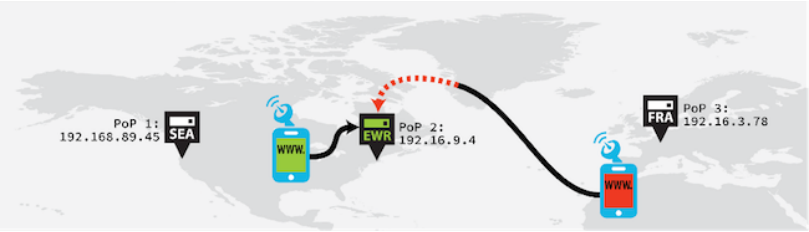
HTTP redirection

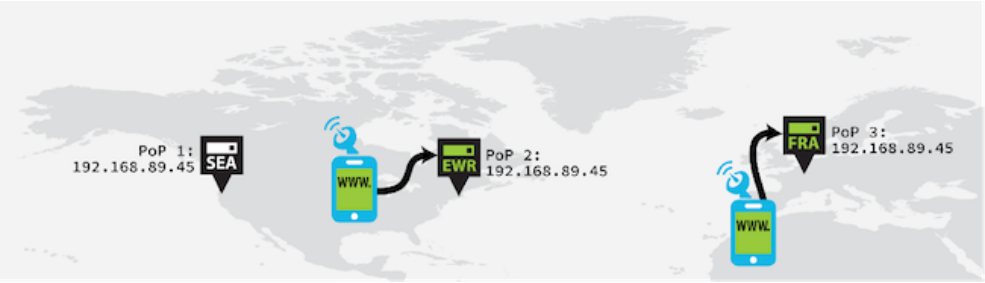
- Elosztó web szerver, ami csak a HTTP refresh fejlécet generálja
- egyszerű és rugalmas
- nem transzparens (felhasználó látja mi történik)
- nagy overhead, két http kérés generálódik minden esetben, magasabb késleltetés

```
<html>
<head>
<title>Your Page Title</title>
<meta http-equiv="REFRESH"
content="0;url=http://www.redirection-
domain.com"></HEAD>
<BODY>
Optional page text here.
</BODY>
</HTML>
```

Anycast

- ugyanazt az IP címet rendeljük több géphez is
- az IP routerek tárolják az útvonalat a hozzájuk levő legközelebbi géphez
- sima unicast címekkel megy



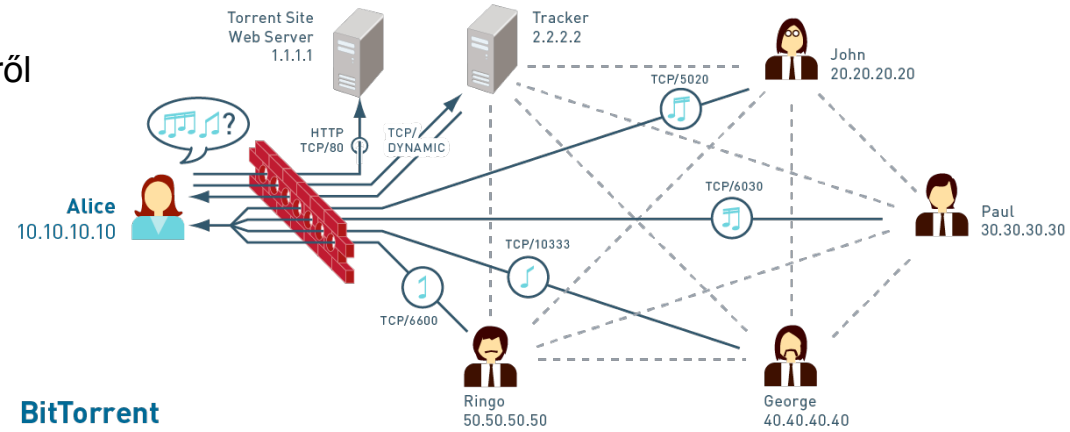


Peer-to-peer rendszerek

- egy alkalmazáscsoport mely kihasználja az Internet peremén
- levő felhasználók erőforrásait:
- Tárolás – merevlemez kapacitás
- CPU – számítási kapacitás
- Tartalom – adatok, információk megosztása
- Bármilyen más megosztható erőforrás, szolgáltatás, funkció

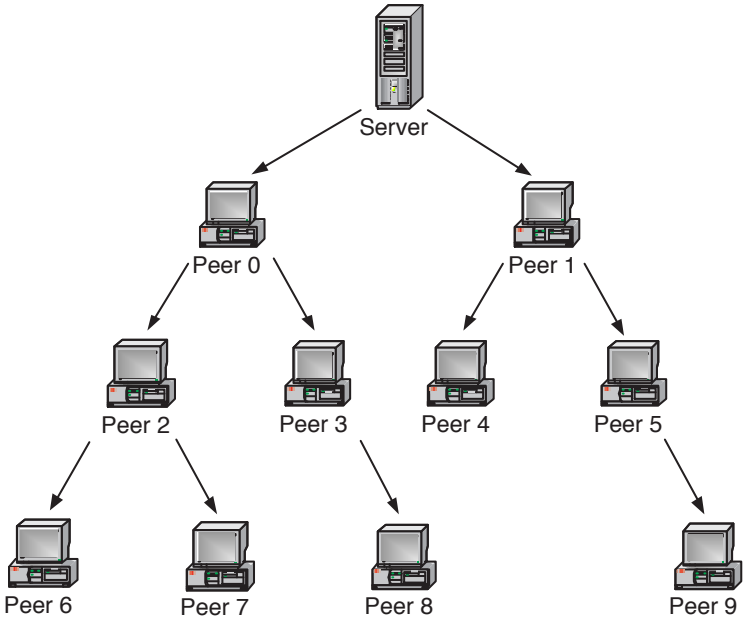
Peer-to-peer: BitTorrent

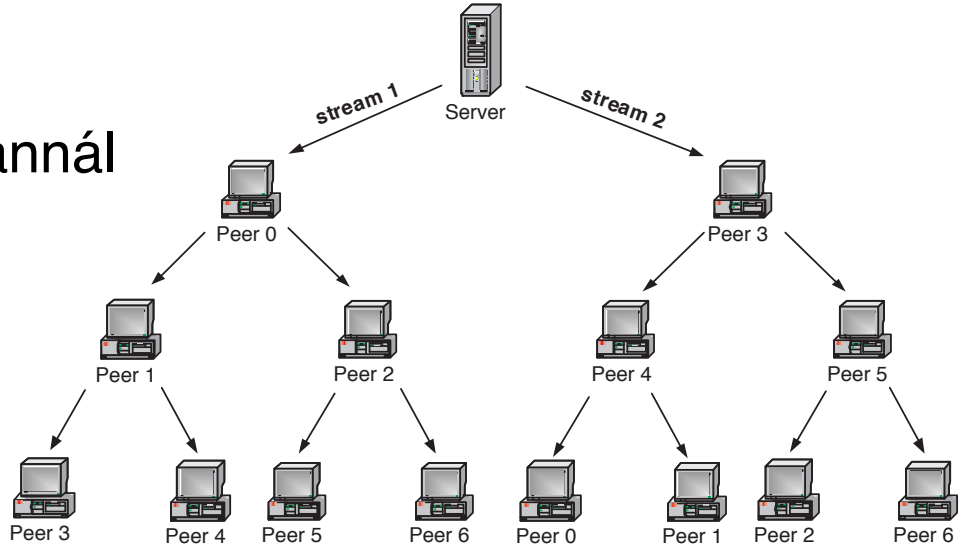
- Hagyományos webes keresés
  - .torrent fájl
  - egy „tracker szerver” címe
- Tracker
  - A csatlakozó peer-nek megad egy véletlenszerű listát a lehetséges peer-ekről
  - Default 50
- Általában egy peer 20-40 peer-el tart kapcsolatot (peer lista)
- Ha 20 alá csökken, újra a trackerhez fordul
- A Peer-ek időközönként jelentenek a Trackernek
  - Hány letöltés
  - Hány peer-nél van a teljes fájl
  - Hány peer-nél vannak fájl részek



P2P streaming

- A felhasználók gépei fába rendeződnek
- Nagyon egyszerű
- A csomagok ugyanazon az úton mennek
- Push jellegű átvitel
- Videó minősége számítható (korlátos késleltetés) a fa mélysége alapján
- fokszám korlátozott a feltöltési sebesség alapján
- a levél peer-ek nem töltenek fel (pedig sokan vannak)



P2P streaming - többszörös fa

- MDC Multiple description coding
  - több videófolyamot csinál különböző kódolással
  - minél többet vesz a vevő annál jobb a minőség
- n folyam
- n-db fa létrehozása az n folyamhoz
- egy peer csak 1 fában lehet közbenső csomópont