

Számítógépes hálózatok

TÍZEDIK ELŐADÁS – Alkalmazási réteg...

KÉSZÍTETTE: ÁCS ZOLTÁN

<http://5gdemocompetition.hu/>

Gondolkodj 5G-ben!

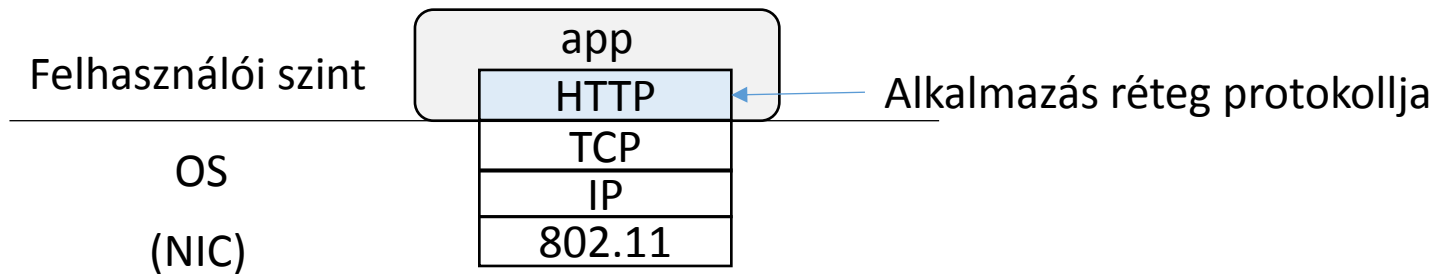
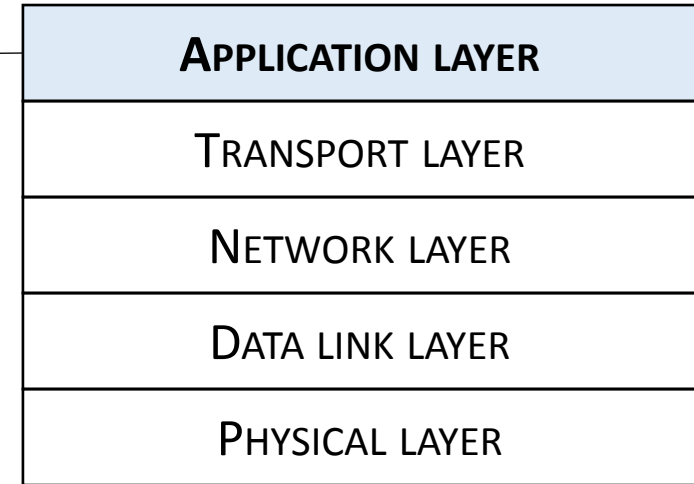
Ericsson Demo Competition

https://www.youtube.com/watch?time_continue=9&v=Ww3sOG8UDfU

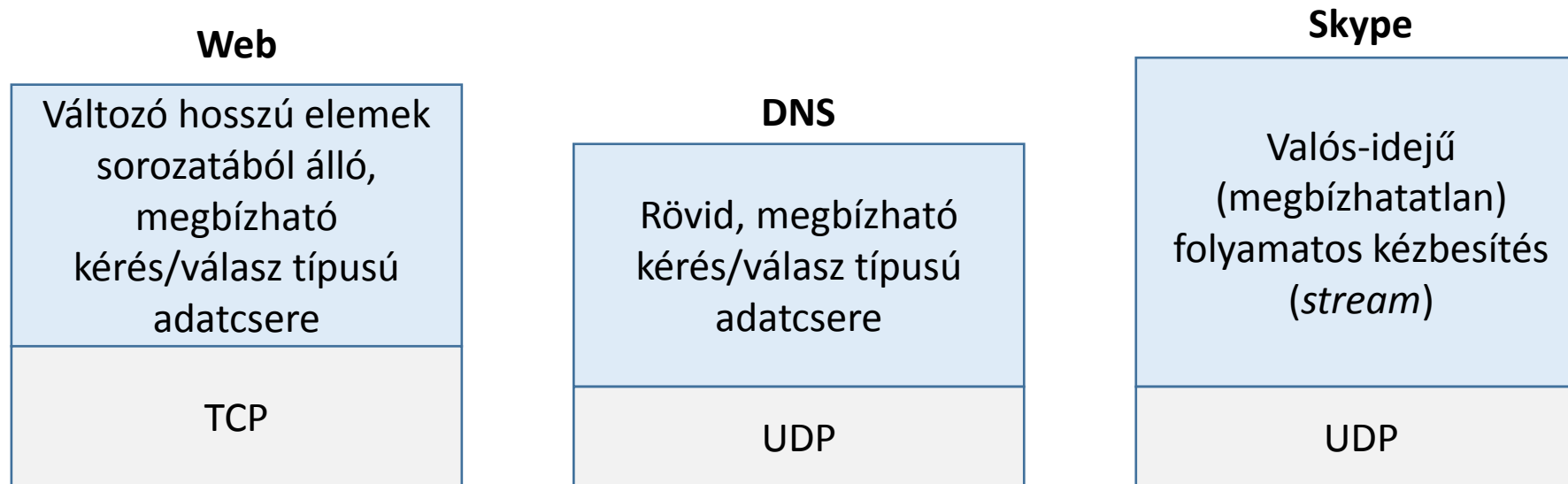
<http://5gdemocompetition.hu/>

Alkalmazási réteg

- **Feladata:** Egy kommunikációs funkciót megvalósító alkalmazással történő interakció.
 - nem mindig szükséges grafikus interfész, például *DNS*;
 - az alkalmazási réteg üzeneteit méretük miatt általában több csomag segítségével továbbítják, például egy sok képet tartalmazó weblap lekérése esetén



Egy alkalmazás kommunikációs szolgálat igényei példákkal



OSI modell (emlékeztető)

A hibrid modellben az alkalmazási réteg részei, de nem szigorúan rétegeként

Application
Presentation
Session
Transport
Network
Data link
Physical

- A felhasználó számára nyújt bizonyos funkciókat.
- A különböző reprezentációs formákat konvertálja.
- Feladat dialógusok kezelése.
- Végpont a végpontnak típusú továbbítást szolgáltat.
- Csomagokat küld több *linken* keresztül.
- Információt kereteken keresztül küldi.
- A biteket szignálok formájában közvetíti.

Munkamenet és megjelenítés koncepciói

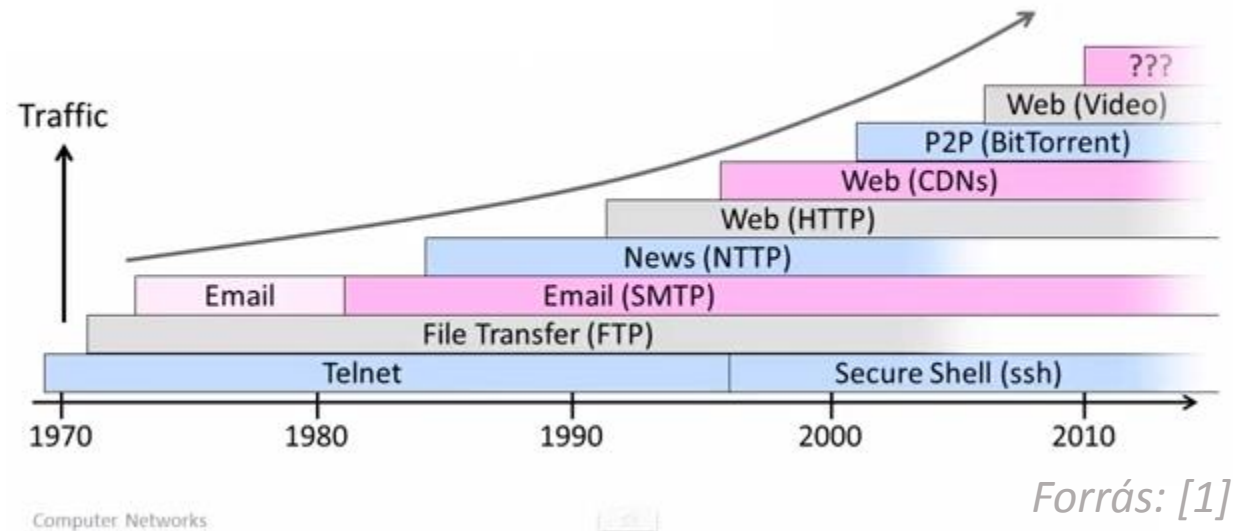
MUNKAMENET/ÜLÉS (*SESSION*)

- **Definíció:** Egy munkamenet a egymással összefüggő hálózati interakciók sorozata egy alkalmazási feladat elvégzése során.
- *Példák*
 - Egy *weblap* több erőforrást is behozhat.
 - Egy *Skype* hívás szintén érinthet képet, hangot, chat-et.

MEGJELENÍTÉS (*PRESENTATION*)

- Az alkalmazásoknak azonosítani kell a tartalom típusát, és kódolnia is kell az átvitelhez.
- *Példák*
 - Média típusa, azaz *MIME* típusok (pl.: *image/jpeg*)
 - Alkalmazott átviteli kódolás a tartalomra (pl.: *gzip*)

Internetes alkalmazások evolúciója



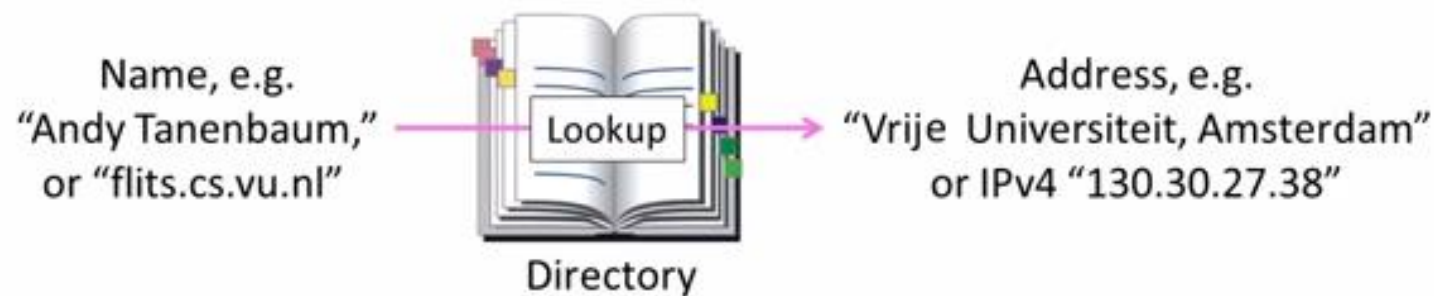
- Folyamatosan változik, növekszik...
 - www.evolutionoftheweb.com

Domain Name System

- olvasható hoszt nevekhez, és egyéb célokra használják.
- Két témáról lesz szó:
 1. az elosztott névtér,
 2. a névfeloldás.

ALAPOK

- A *nevek* az erőforrások magas-szintű azonosítói.
- A *címek* az erőforrások alacsony-szintű lokátorai.
- A *feloldás* egy név leképezése egy címre.



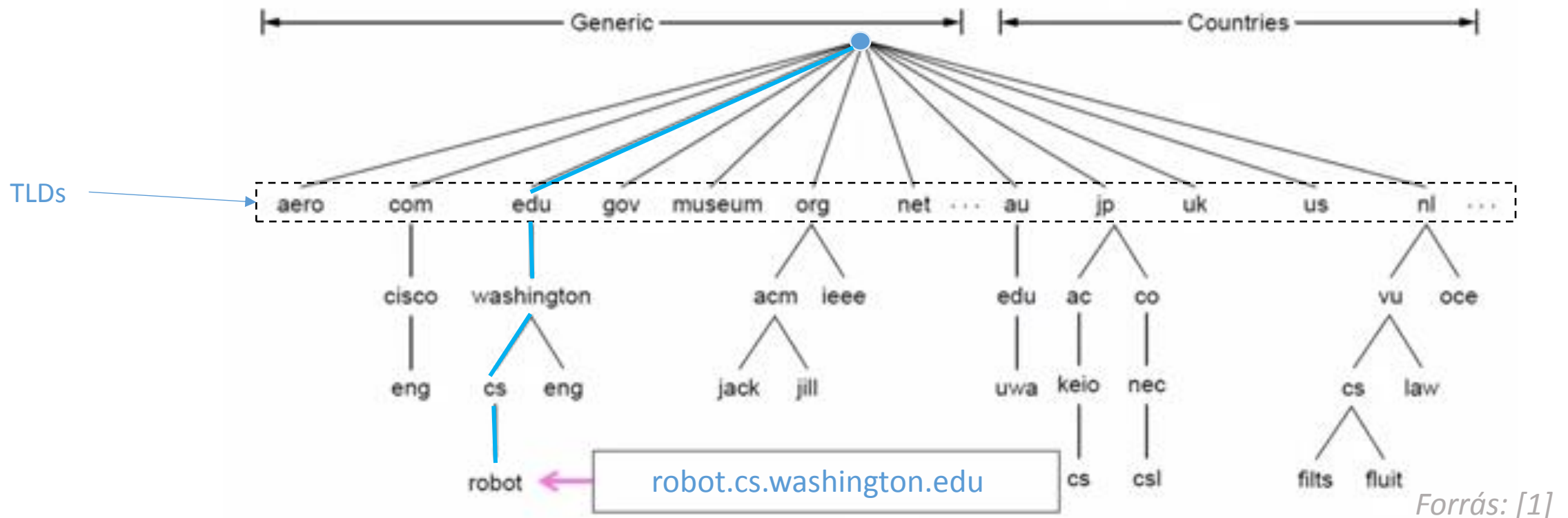
Történeti háttér

- Előzmények: Egy *HOSTS.TXT* állomány használata, amelyet az összes hoszt egy központi gépről (***Network Information Center***) szerzett be.
- 1985 körül az egyszerű nevek helyett el kezdtek hierarchikus neveket használni.
- A hálózat méretének növekedésével ez egyre több problémát jelentett ez a módszer.

DNS alapok 1/5

- Név szolgáltatás a hosztok nevének és IP-címének összerendeléséhez.
 - Például: www.inf.elte.hu → 157.181.161.79
- Célok:
 - Könnyű kezelhetőség és
 - hatékonyság.
- Megközelítés:
 - Elosztott jegyzék egy hierarchikus névtér felett.
 - Automatikus protokoll a darabok összefűzéséhez.

DNS alapok 2/5 – címtér

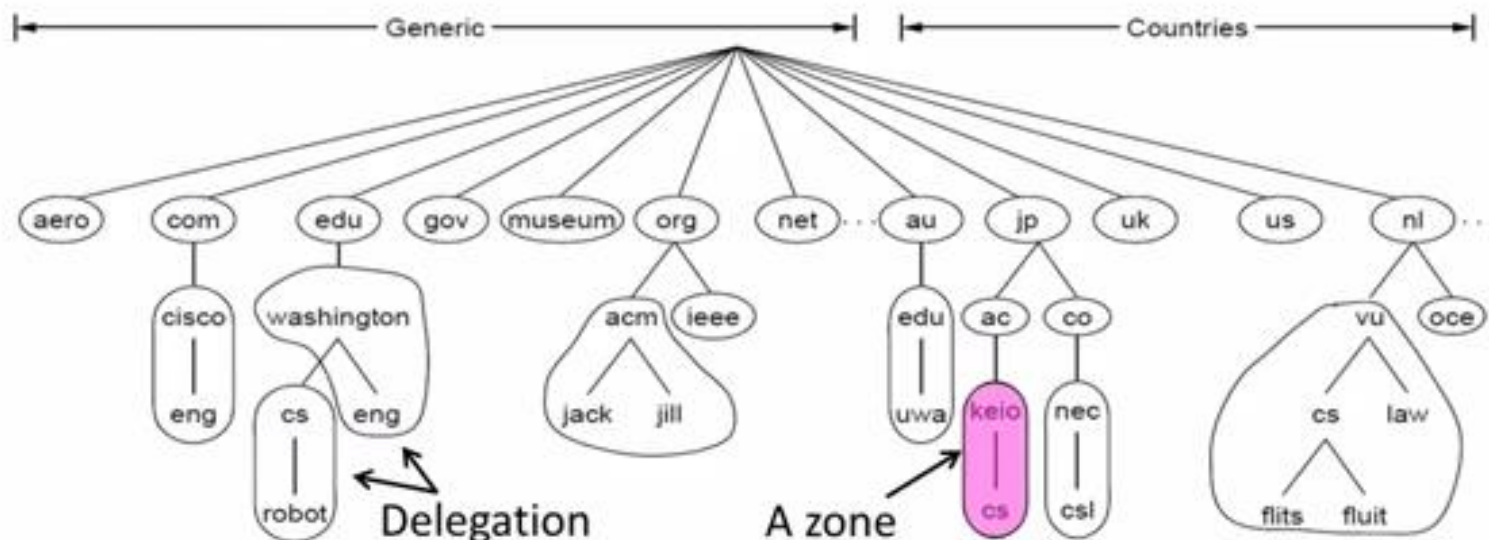


DNS alapok 3/5 – *Top Level Domains*

- Internet Corp. **A**ssigned **N**ames and **N**umbers (1998)
- 22+ **általános TLDs** létezik
 - klasszikusok: *.com, .edu, .gov, .mil, .org, .net*
 - később keletkeztek: *.aero, .museum, .xxx*
- ~250 TLDs a különböző **ország kódok**nak
 - Két betű (mint például *.au, .hu*), 2010-től plusz nemzetközi karakterek (például kínai)
 - Több elüzletisedett, például a *.tv* (Tuvalu)
 - Példa domén hack-ekre: *instagr.am* (Örményország), *goo.gl* (Grönland)

DNS alapok 4/5 – Zónák

- A *zóna* a névtér egy összefüggő része.
- Az elosztás alapjai a zónák lesznek.
- Minden egyes zóna rendelkezik egy *névszerverrel* ezen információkhoz, beleértve a delegációra vonatkozó ismereteket is.



Forrás: [1]

DNS alapok 5/5 – DNS erőforrás rekordok

- DNS erőforrás rekordokba tárolják a zónára vonatkozó információkat.

Típus	Jelentés
SOA	Hatáskör kezdete, zóna kulcsparamétereivel rendelkezik.
A	Egy hoszt IPv4-es címe.
AAAA	Egy hoszt IPv6-es címe.
CNAME	Egy alias kanonikus neve.
MX	A levelező kiszolgáló a doménre nézve.
NS	A domén névszervere vagy delegált aldomén.

DNS erőforrás rekordok minta

; Authoritative data for cs.vu.nl

cs.vu.nl.	86400	IN	SOA	star boss (9527,7200,7200,241920,86400)
cs.vu.nl.	86400	IN	MX	1 zephyr
cs.vu.nl.	86400	IN	MX	2 top
cs.vu.nl.	86400	IN	NS	star

Névszerver

star	86400	IN	A	130.37.56.205
zephyr	86400	IN	A	130.37.20.10
top	86400	IN	A	130.37.20.11

Számítógépek IP címei

www	86400	IN	CNAME	star.cs.vu.nl
ftp	86400	IN	CNAME	zephyr.cs.vu.nl

flits	86400	IN	A	130.37.16.112
flits	86400	IN	A	192.31.231.165
flits	86400	IN	MX	1 flits
flits	86400	IN	MX	2 zephyr
flits	86400	IN	MX	3 top

rowboat		IN	A	130.37.56.201
		IN	MX	1 rowboat
		IN	MX	2 zephyr

Email gateway-k

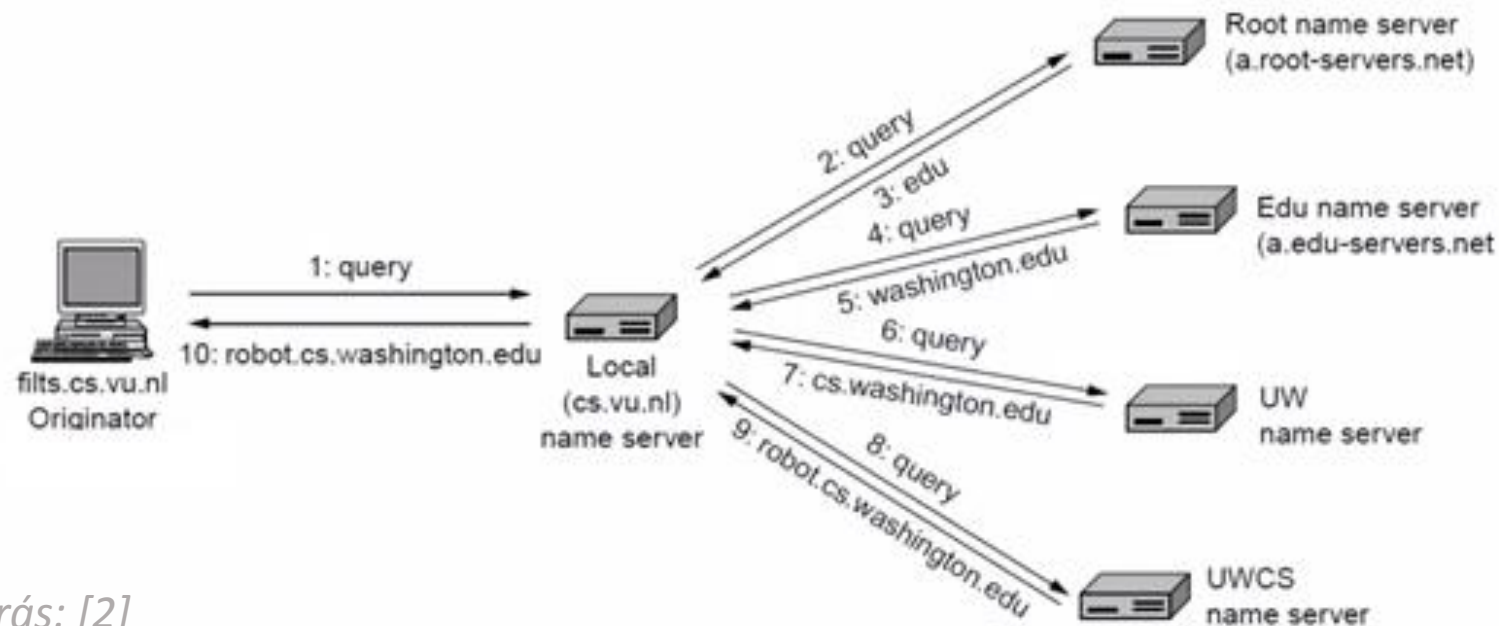
little-sister		IN	A	130.37.62.23
---------------	--	----	---	--------------

laserjet		IN	A	192.31.231.216
----------	--	----	---	----------------

Forrás: [1]

DNS névfeloldás 1/4 – Alapok

- A *DNS* protokoll lehetővé teszi egy hosztnak, hogy bármely hoszt nevét feloldja *IP* címmé.
- Ha ismeretlen, akkor kezdhethük a „gyökér” névszervernél, és onnan haladhatunk lefele.
- Alábbi ábra szemlélteti a feloldás lépéseit a flits.cs.vu.nl hoszt esetén a robot.cs.washington.edu rezolúciójára.



Forrás: [2]

DNS névfeloldás 2/4 – Lekérdezések

- A lekérdezésnek két fajtája van:
 - *Rekurzív lekérdezés* – Ha a névszerver végzi el a névfeloldást, és tér vissza a válasszal.
 - *Iteratív lekérdezés* – Ha a névszerver adja vissza a választ vagy legalább azt, hogy kitől kapható meg a következő válasz.
- Melyik a jobb?
 - *Rekurzív jellemzői*
 - Lehetővé teszi a szervernek a kliens terhelés kihelyezését a kezelhetőségért.
 - Lehetővé teszi a szervernek, hogy a kliensek egy csoportja felett végezzen *cachelést*, a jobb teljesítményért.
 - *Iteratív jellemzői*
 - Válasz után nem kell semmit tenni a kéréssel a névszervernek.
 - Könnyű magas terhelésű szervert építeni.

DNS névfeloldás 3/4 – „Caching”

- A feloldás miatt fellépő késleltetés lehetőleg legyen alacsony.

MEGOLDÁS

- A kérések/válaszok megőrzése a jövőbeli kérdések azonnali megválaszolására.
 - Részleges (iteratív) válaszokat is beleértve,
 - A cache-ben lévő elemekhez élettartam (*TTL*) rendelése szükséges.

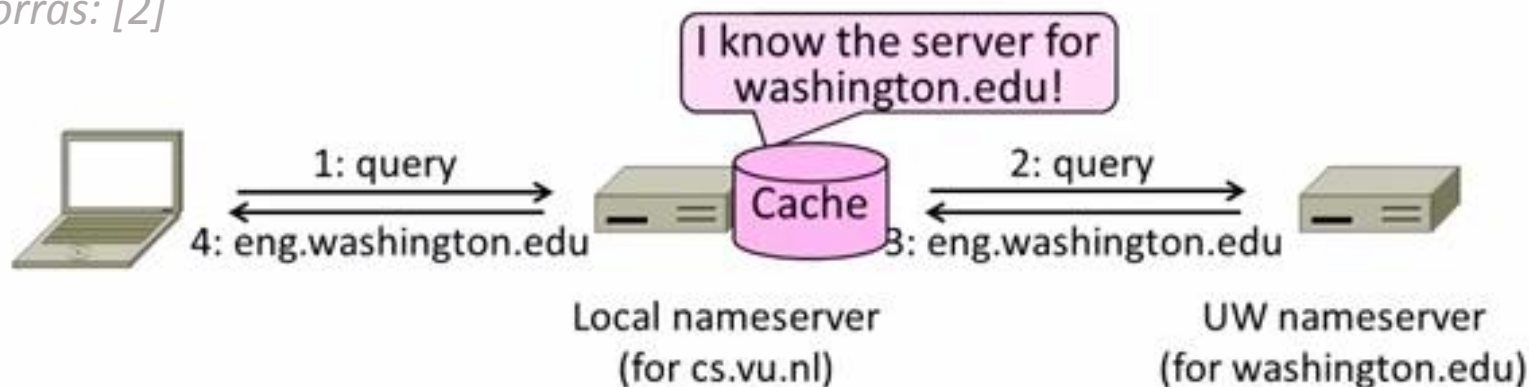


Forrás: [2]

DNS névfeloldás 4/4 – „Caching” példa

Tegyük fel most, hogy a korábbi példában szereplő *flits.cs.vu.nl* hoszt most a *eng.washington.edu* rezolúcióját is el akarja végezni, és azt is, hogy a cache-ben még benne van az előző feloldásból tanult információ.

Forrás: [2]



Lokális névszerverek

- A lokális névszervereket általában az ISP vagy a vállalat IT részlege üzemelteti.
 - a saját számítógép is konfigurálható névszervernek;
 - más alternatívák is léteznek, mint például *Google public DNS*.
- A klienseknek képesnek kell lenni kommunikálni a saját lokális névszerverével.
 - Általában a DHCP-n keresztül konfigurálódik fel.

„Root” névszerverek 1/2

- 13 névszerver szolgálja ki a gyökér elemet
 - az a.root-servers.net-től m.root-servers.net-ig
 - Minden névszervernek szüksége van a „root” névszerverek IP címére.
 - Ez egy konfigurációs állományban van rögzítve. (*named.ca*)
- Több mint 250 elosztott szerver instancia létezik.
 - Magas rendelkezésre állás, megbízhatóság.
 - A legtöbb szerver elérhető *IP anycast* segítségével.
 - A szerverek IPv4 és IPv6 címmel is elérhetőek.

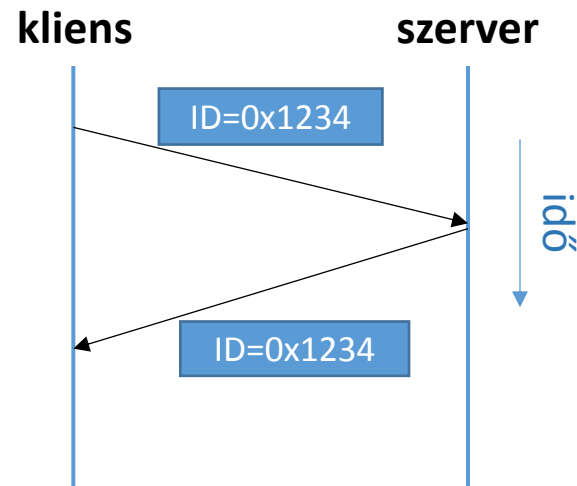
„Root” névszerverek 2/2

(<http://root-servers.org/>)



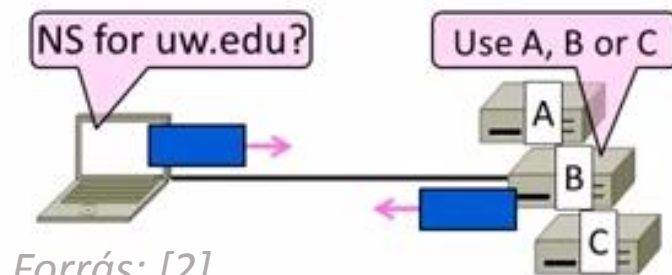
DNS protokoll 1/2

- Kérdés és válasz üzenetek
 - *UDP* üzenetekre épül az 53-as szolgáltatás porton keresztül.
 - *ARQ* használata a megbízhatóság biztosítására.
 - Az üzenetek összekapcsolása egy 16-bites azonosító használatával történik.



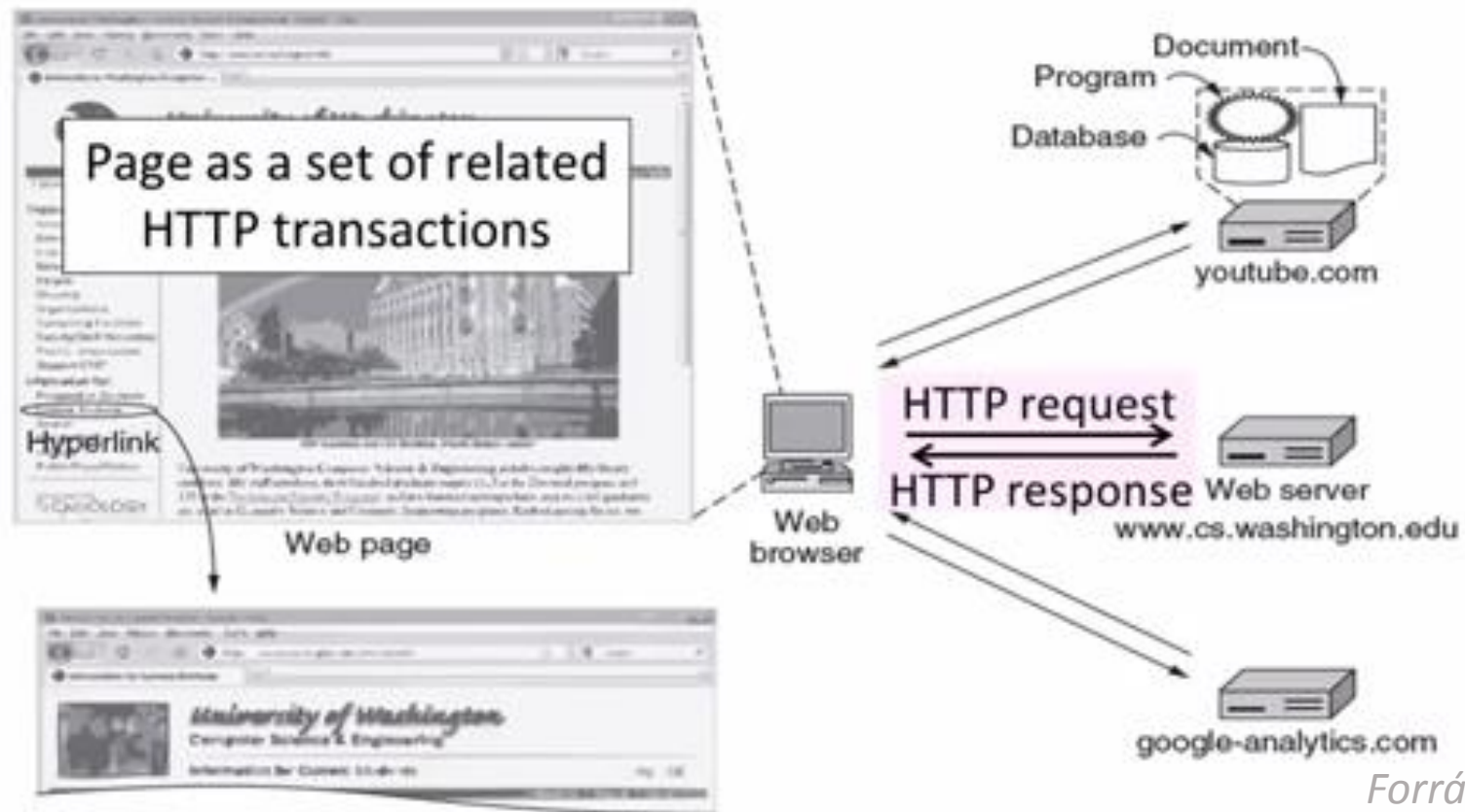
DNS protokoll 2/2

- Szolgáltatás megbízhatósága másolatok által:
 - Több névszerver futtatása az adott domén-en.
 - Lista lesz a válasz, és a kliens csak egy választ használ fel.
 - A terhelés elosztását is segíti.
- Mi a teendő a rosszindulatú átirányításokkal?
 - Eredeti koncepcióban nem szerepelt a biztonság kérdése
 - DNSSEC (később)



Forrás: [2]

Webes tartalmak



Forrás: [4]

HTTP protokoll

- Egy kérés/válasz protokoll webes erőforrások lekérésére. (*szöveg alapú parancsok, fejlécek*)
- TCP protokollt használ, és tipikusan a 80-as szolgáltatás porton fut.

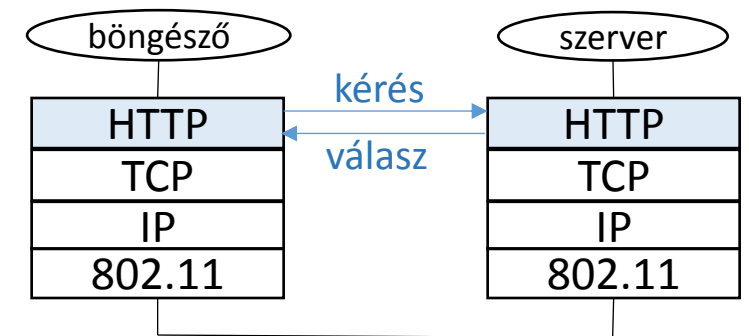
EGY OLDAL LEKÉRÉSE

- Egy oldal *URL*-jének felépítése:

<http://www.inf.elte.hu/szervezet/Lapok/default.aspx>

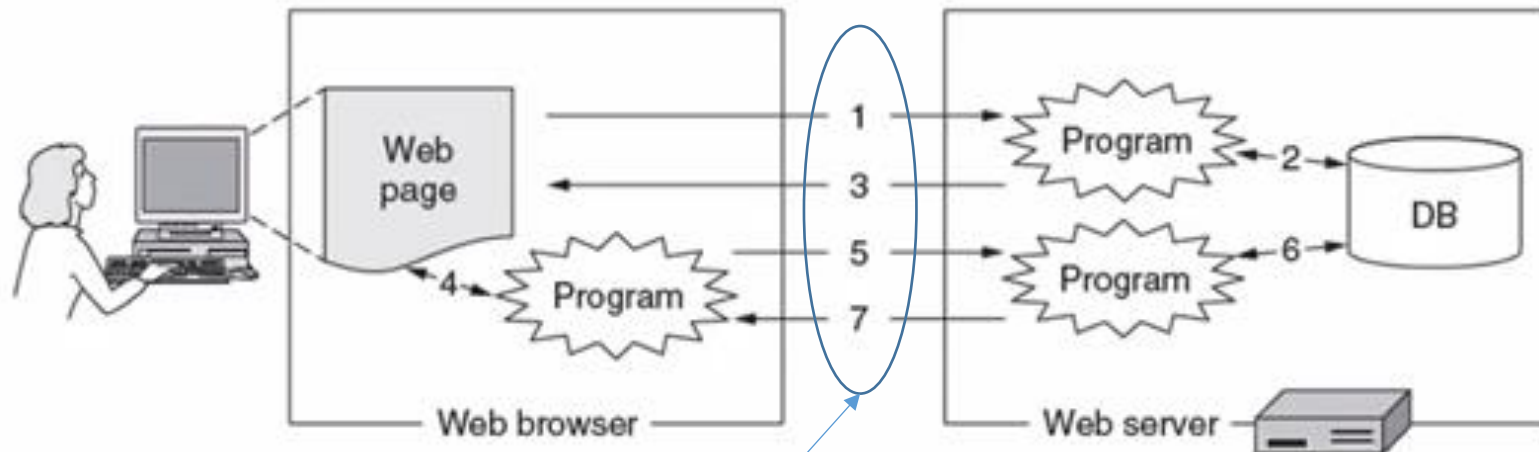
protokoll szerver oldal a szerveren

1. A szerver IP-jének feloldása. (DNS)
2. TCP kapcsolat felépítése a szerverrel.
3. Az oldalra vonatkozó lekérdezés küldése. (*request*)
4. Várakozás a HTTP válaszra az oldaltól. (*response*)
 ** *Végrehajtás/beépülő erőforrások hozzáadása/renderelés*
5. Az üresjáratú TCP kapcsolatok bontása.



Statikus és dinamikus weboldalak

- A *statikus weboldal* tartalma nem változik csak manuális átszerkesztéssel.
- A *dinamikus weboldal* valamilyen kód végrehajtásaként keletkezik, mint például: *javascript*, *PHP*, vagy mindkettő egyszerre.



Forrás: [4]

HTTP

HTTP protokoll parancsok és státusz kódok

Eljárás	Leírás
GET	Egy weboldal olvasása.
HEAD	Egy weboldal fejlécének olvasása.
POST	Egy weboldalhoz csatolás.
PUT	Egy weboldal tárolása.
DELETE	Egy weboldal eltávolítása.
TRACE	A bejövő kérés visszhangozása.
CONNECT	Proxy-n keresztüli csatlakozás.
OPTIONS	Kérés opciók egy oldalhoz.

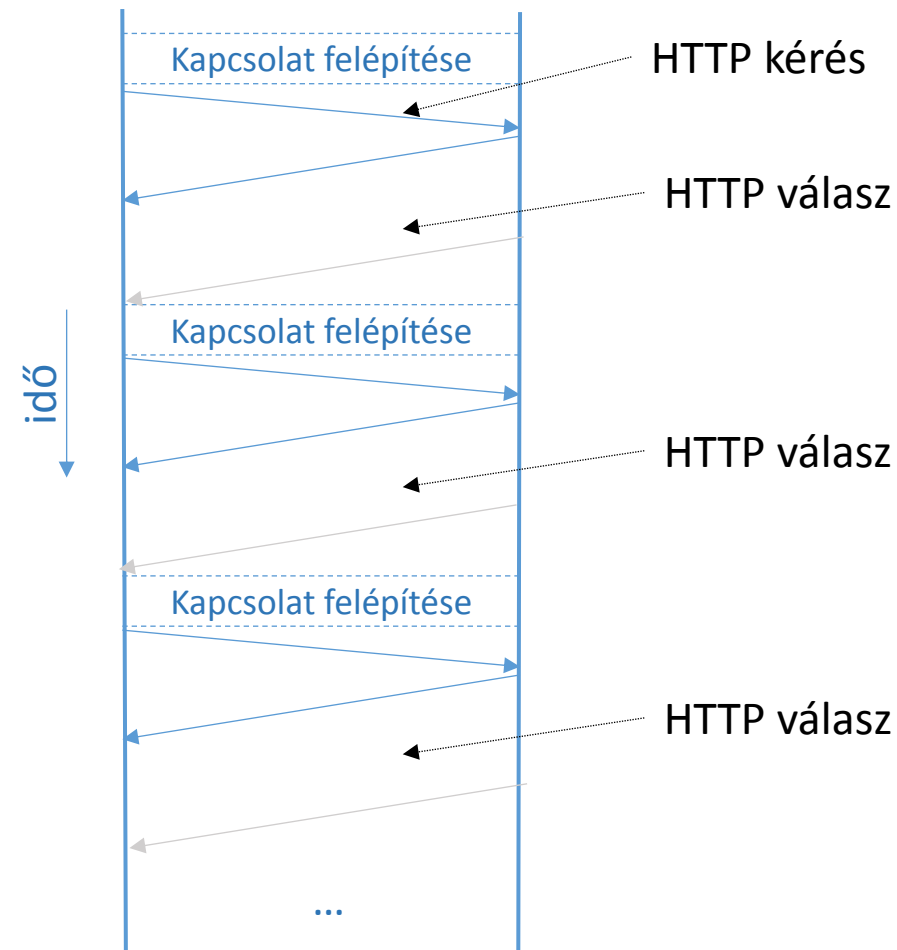
Kód prefix	Jelentés	Példa
1xx	Információ	<i>100 – server agrees to handle client's request</i>
2xx	Sikeres	<i>200 – request succeeded; 204 – no content present</i>
3xx	Átirányítás	<i>301 – page moved; 304 – cached page still valid</i>
4xx	Kliens hiba	<i>403 – forbidden page; 404 – page not found</i>
5xx	Server hiba	<i>500 – internal server error; 503 – try again later</i>

HTTP protokoll fejlécek

Funkció	Példa fejléc
Böngésző lehetőségek (<i>kliens</i> → <i>szerver</i>)	User-Agent, Accept, Accept-Charset, Accept-Encoding, Accept-Language
Tárolással kapcsolatosak (<i>mindkét irány</i>)	If-Modified-Since, If-Non-Match, Last-Modified, Expires, Cache-Control, ETag
Böngésző kontextus (<i>kliens</i> → <i>szerver</i>)	Cookie, Referer, Authorization, Host
Tartalom kézbesítés (<i>szerver</i> → <i>kliens</i>)	Content-Encoding, Content-Length, Content-Type, Content-Language, Content-Range, Set-Cookie

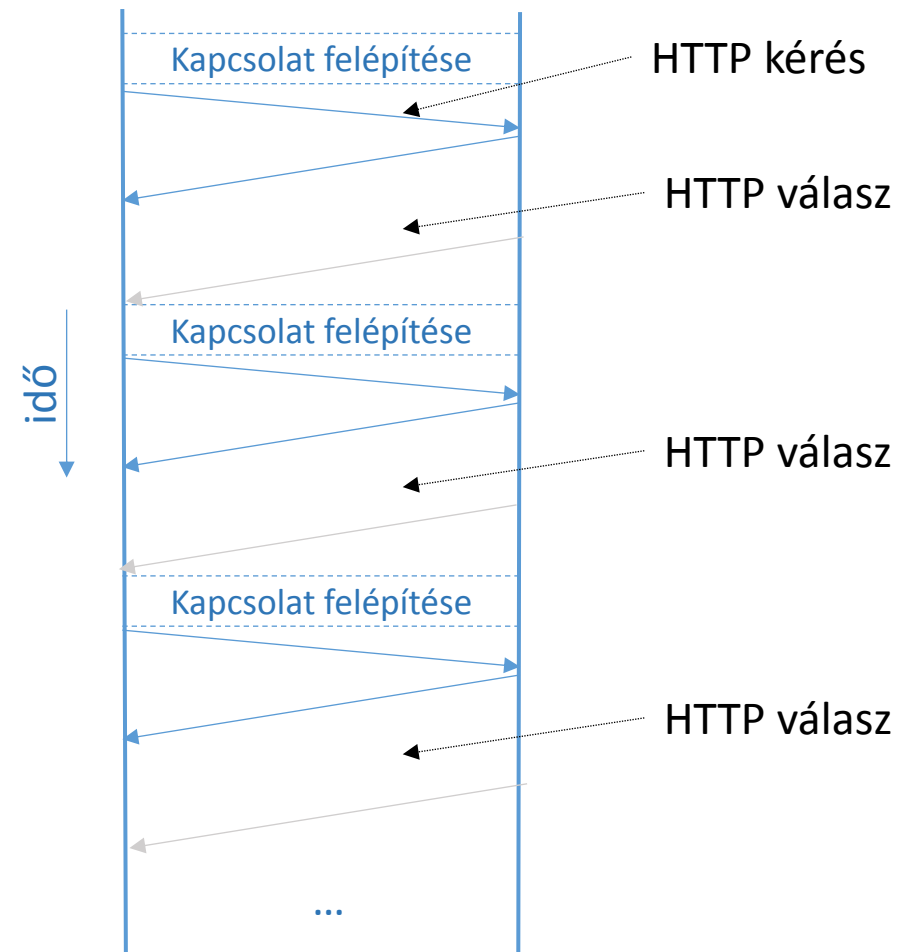
HTTP teljesítménye 1/2

- A HTTP protokoll teljesítmény mérésének egy fő mérőszáma: **Page Load Time**.
 - Azon időtartam, ami a kattintás, és az oldal betöltése között eltelik.
 - Több tényezőtől is függhet: oldal/tartalom struktúrájától, a HTTP (és TCP) protokolltól, a hálózati RTT-től és a sávszélességtől.
- Korai teljesítménye (HTTP 1.0)
 - Egy TCP kapcsolat per web erőforrás.
 - **Konklúzió:** szekvenciális kérés/válasz, több *TCP* kapcsolat ugyanahhoz a szerverhez, többszörös *TCP slow-start* fázis.



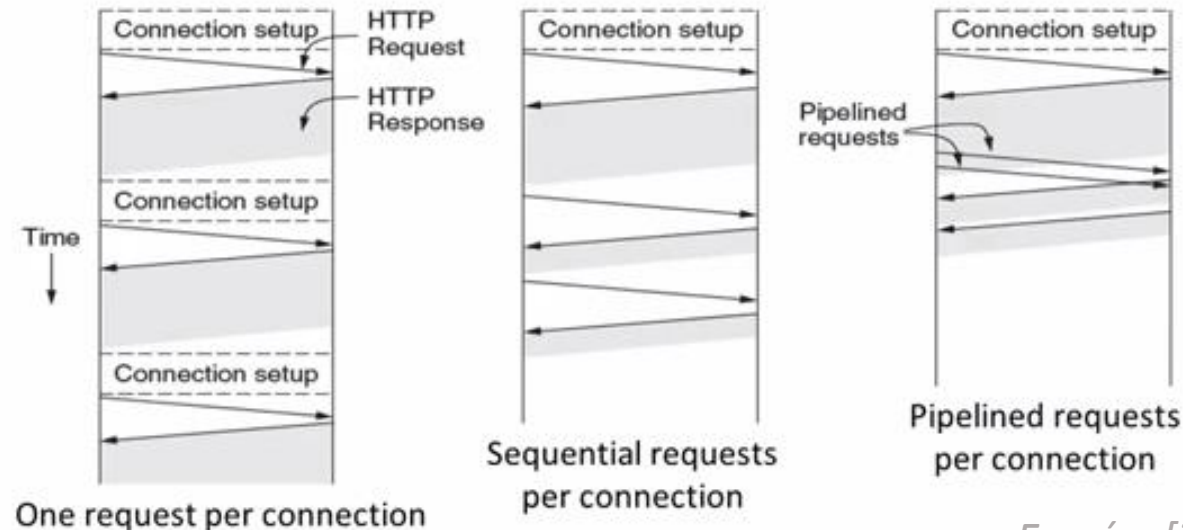
HTTP teljesítménye 2/2

- Korai teljesítményének javítási lehetőségei
 1. A tartalom méretének csökkentése.
 2. HTTP megváltoztatása a hatékony sávszélesség használathoz.
 3. HTTP megváltoztatása ugyanazon tartalom újrakézbítésének elkerülésére (*cache, proxy*)
 4. A tartalom „közelebb” helyezése a klienshez. (*CDNs*)



Párhuzamos és perzisztens kapcsolatok

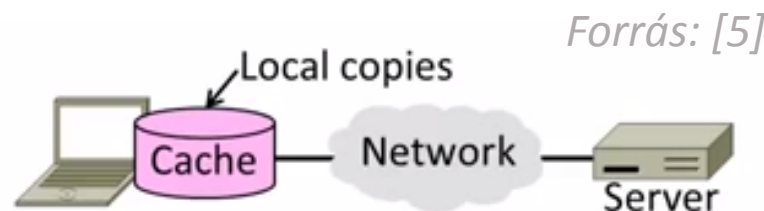
- Egyszerű módszer a PLT leszorítására a párhuzamos kapcsolatok használata.
 - A böngésző több HTTP instanciát futtat párhuzamosan.
 - A szerver viselkedése változatlan marad.
- Másik módszer a PLT leszorítására a perzisztens kapcsolat használata.
 - Egy TCP kapcsolat használata a szerverrel való kommunikációra.
 - Ezen kapcsolat használata többszörös HTTP kérésre.



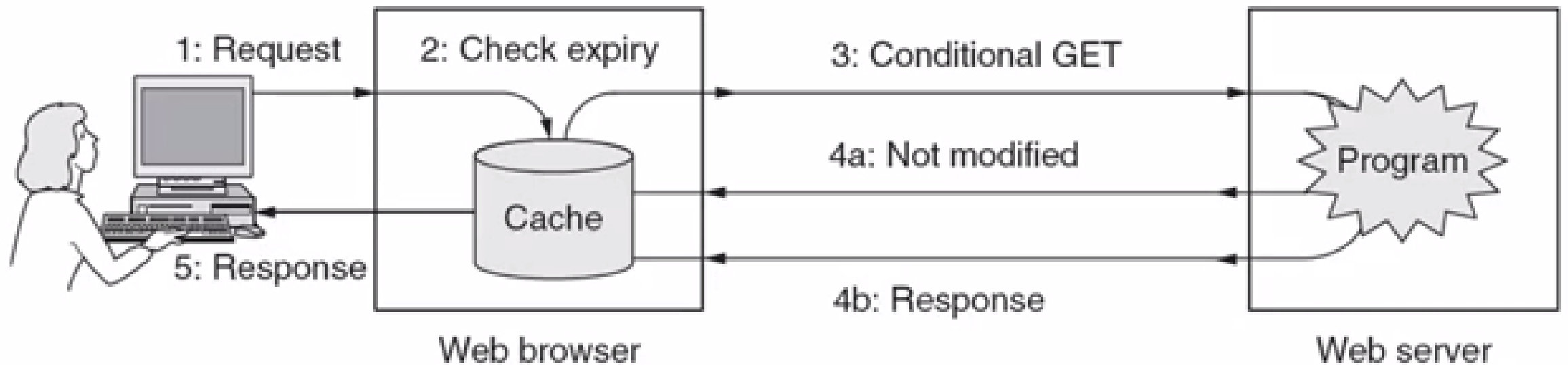
Forrás: [7]

Web „caching” 1/2

- Felhasználók gyakran újra látogatják az oldalakat.
 - Használható a lokális másolata a lapnak. Ezt nevezzük „caching”-nek.
- **Kérdés:** *Mikor használhatjuk a lokális másolatot?*
 - Lokálisan meghatározni, hogy valid-e a kópia.
 - Lejáratási információk alapján, mint például az „Expires” fejléc alapján a szervertől.
 - Heurisztikák használatával megtippelni.
 - **Előny:** A tartalom azonnal elérhető.
 - A másolat újra validálása a szerverrel.
 - Másolatban lévő időbélyegző alapján. (a *Last-Modified* fejléc)
 - A másolat tartalma alapján. (az *Etag* fejléc a szervertől)
 - **Előny:** A tartalom egy RTT-nyi késleltetéssel elérhető.



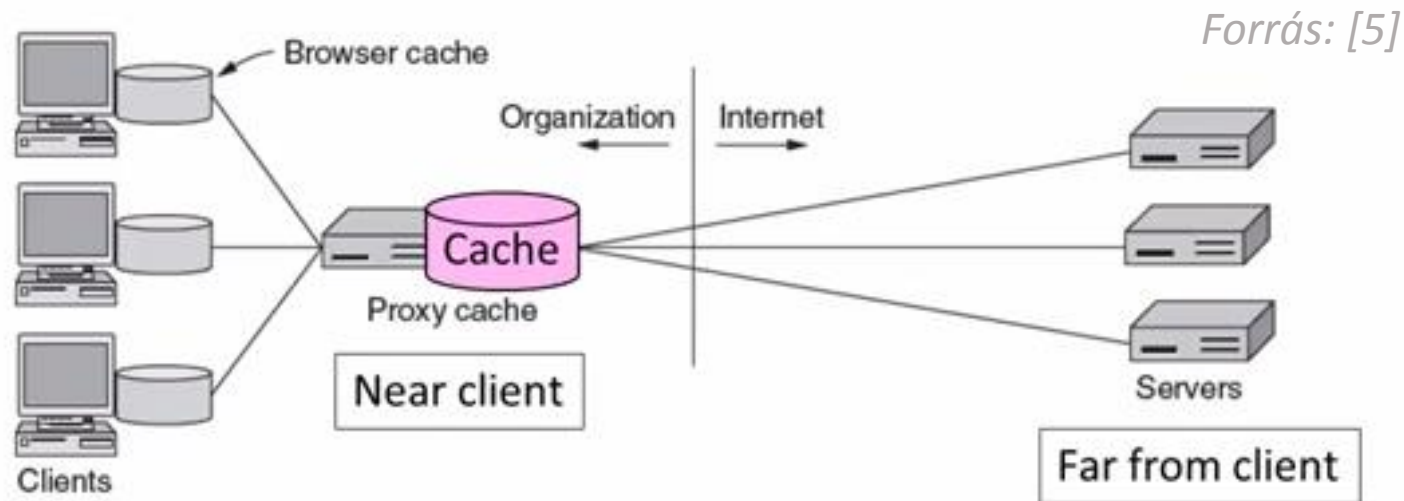
Web „caching” 2/2



Forrás: [5]

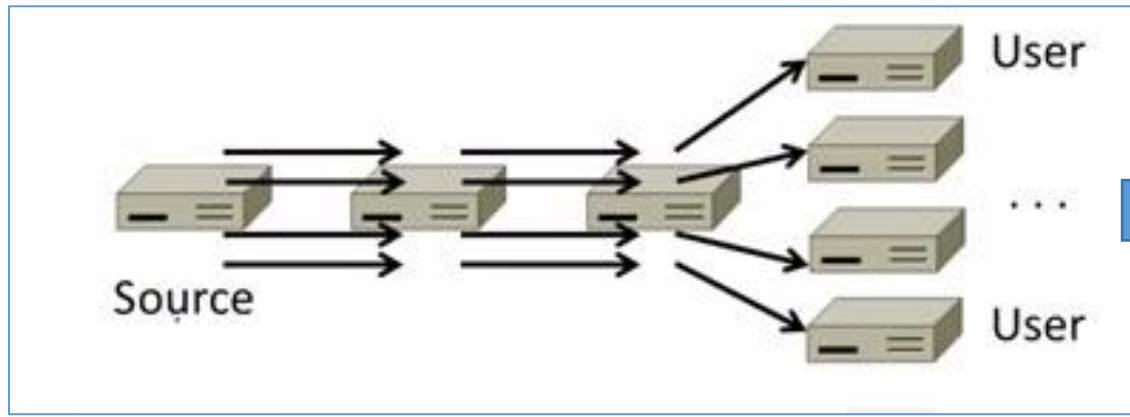
Web proxy

- A kliensek csoportja és a külső web szerverek közé egy közbelső elem elhelyezése, ez lesz a proxy.
 - Előnyök a klienseknek: nagyobb *cache* és biztonsági ellenőrzés.
 - A szervezeti hozzáférés szabályozás.
- *Proxy cache* egy elosztott, nagy gyorsító tárat biztosít a klienseknek.
 - Korlátozások a tárolásra: biztonságos tartalmak, dinamikus tartalmak.

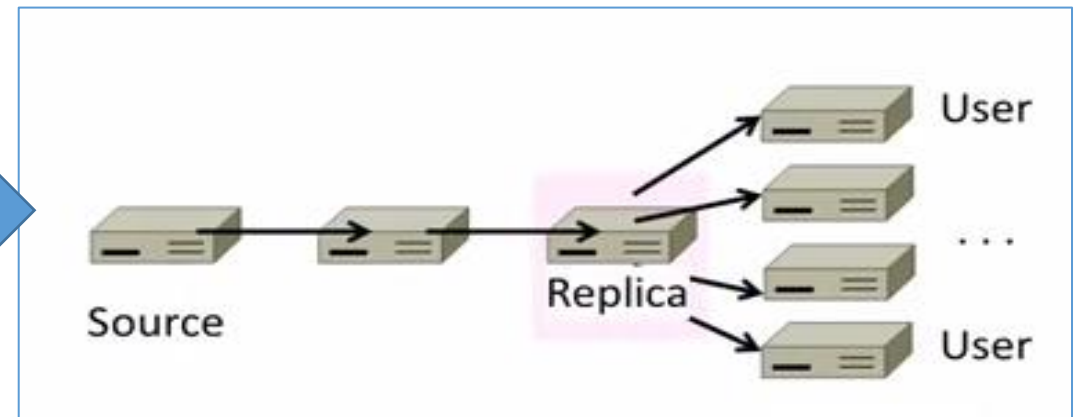


Content Delivery Networks 1/2

- A forgalom növekedése a következő problémákat okozta a weben a 90-es években:
 1. A népszerű szerverek koncentrált terhelése.
 2. Torlódások keletkeztek, és egyre nagyobb sávszél biztosítása vált szükségessé.
 3. A felhasználói élmény nagyon rossz volt.
- Ötlet: A népszerű tartalmak elhelyezése a kliensekhez közel.



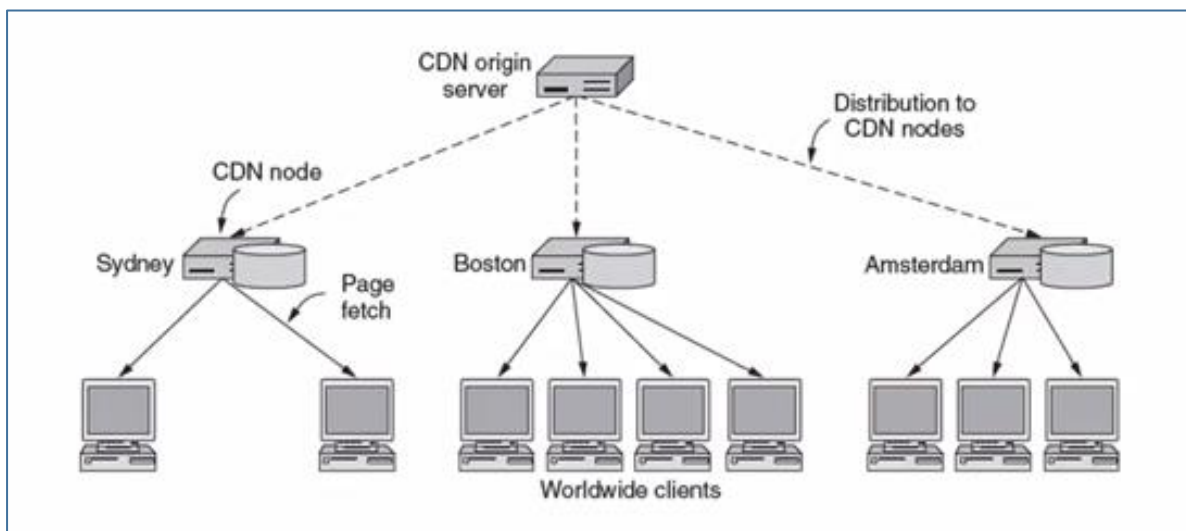
CDNs előtt
(Forrás: [1])



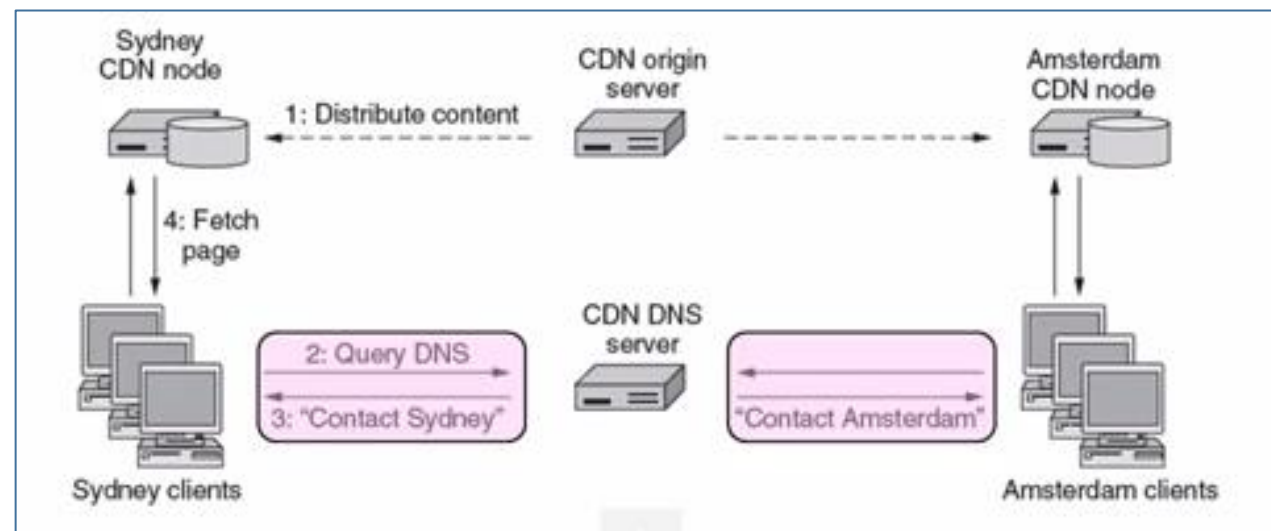
CDNs
(Forrás: [1])

Content Delivery Networks 2/2

- Hogy helyezhetjük a klienshez közel a tartalmakat?
 - Használjuk fel a böngésző és a proxy gyorsító tárat. (Segít, de véges erőforrás.)
 - Másolatok elhelyezése Internet szerte a közeli kliensek részére. (*DNS*)



Másolat képzés
Forrás: [1]



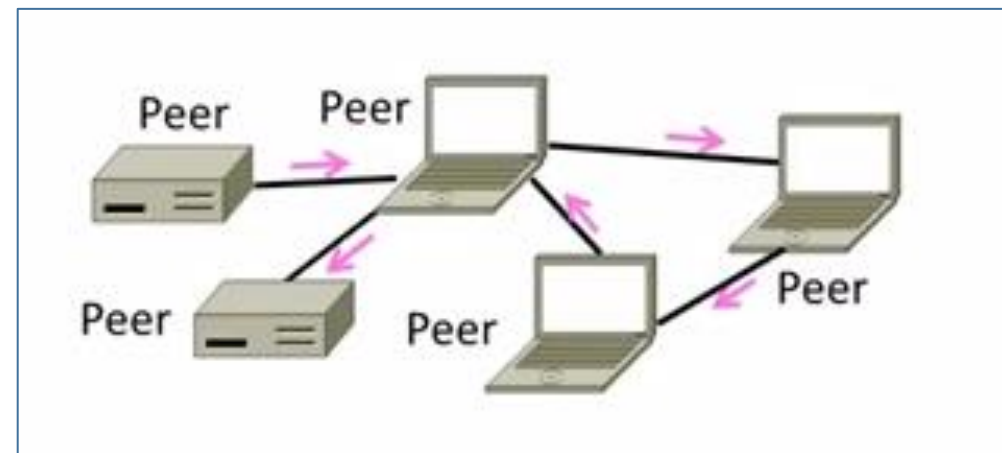
DNS protokoll használata
Forrás: [1]

p2p tartalom megosztás 1/3

- CDNs hátrányai
 - Dedikált infrastruktúrát igényelnek.
 - Centralizált vezérlés/felügyelet kell.
- **Cél:** Olyan dedikált infrastruktúra és központi felügyelet mentes kézbesítés megvalósítása, ami még mindig hatékony és megbízható.
- **Kulcs:** a résztvevők, avagy peer-ek, segítsenek magukon
 - Napster 1999 zenei tartalomra,
 - BitTorrent 2001 bármilyen tartalomra.

P2P HÁLÓZAT JELLEMZŐI

- Nincs szerver.
- A kommunikáció peer-ek között folyik és önszerveződő.
- Skálázási problémák merülnek fel.

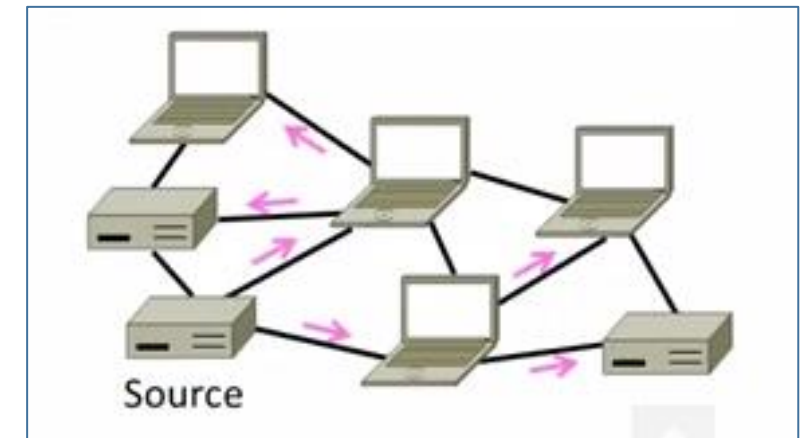


p2p hálózat
Forrás: [2]

p2p tartalom megosztás 2/3

P2P HÁLÓZAT KIHÍVÁSOK

1. Korlátozott lehetőségek
 - Hogyan képes egy *peer* tartalmat továbbítani az összes többi *peer*-nek?
 2. Résztétel ösztönzése
 - Miért segítenének egymásnak a *peer*-ek?
 3. Decentralizálás
 - Hogy találják meg a *peer*-ek a tartalmat?
- Peer-ek kettős szerepe:
 1. Feltöltés a többiek segítésére.
 2. Letöltés saját magának.

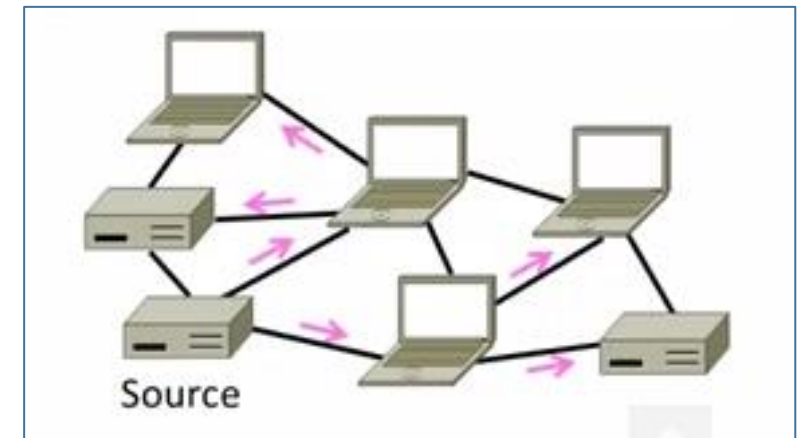


p2p megosztási fa
Forrás: [2]

p2p tartalom megosztás 3/3

P2P DECENTRALIZÁLÁS MEGVALÓSÍTÁSA

- A *peer* meg kell tanulja a tartalom helyét. (**Distributed Hash Tables**)
- A DHTs teljesen decentralizált, hatékony algoritmusok egy elosztott indexhez.
 - Az index a *peer*-ek között terjed.
 - Az index listázza a tartalommal rendelkező *peer*-eket.
 - Bármely *peer* kikeresheti az indexet.
 - Tudományos munkaként indult 2001-ben.



p2p megosztási fa
Forrás: [2]

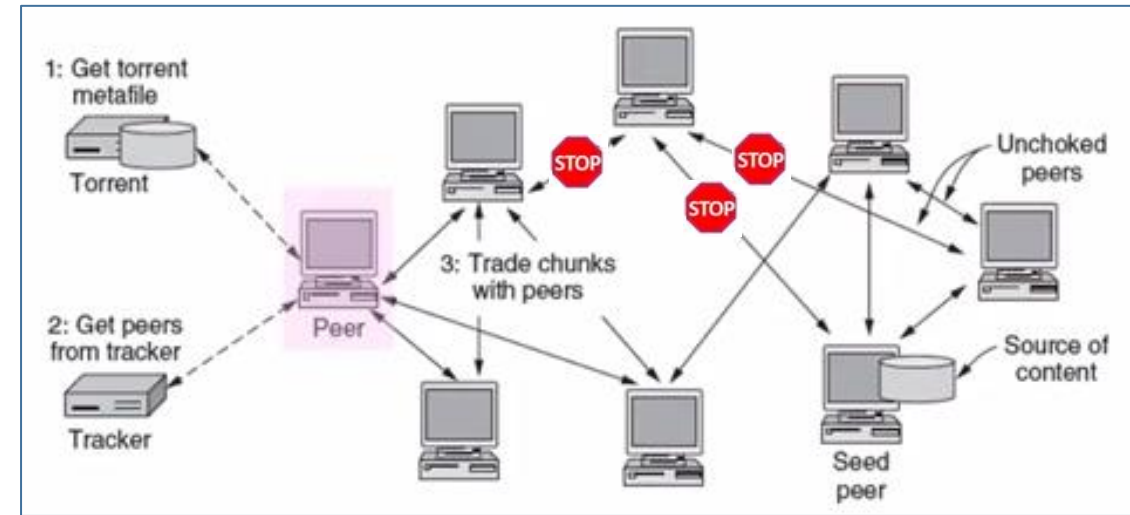
BitTorrent

- A legelterjedtebb p2p rendszer, amit:
 - 2001-ben fejlesztette ki Bram Cohen.
 - Nagyon gyorsan elterjedt, és jelenleg az Internet forgalom nagy részét teszi ki a BitTorrent forgalom.
 - Legális és illegális tartalmak megosztására is használják.
- Az adatkézésítés „torrent”-ek segítségével történik:
 - A fájlok megosztása darabonként történik.
 - Az ösztönzés céljára figyelemreméltó módszert alkalmaz.
 - Tracker-ek vagy decentralizált indexek használata.

BitTorrent protokoll

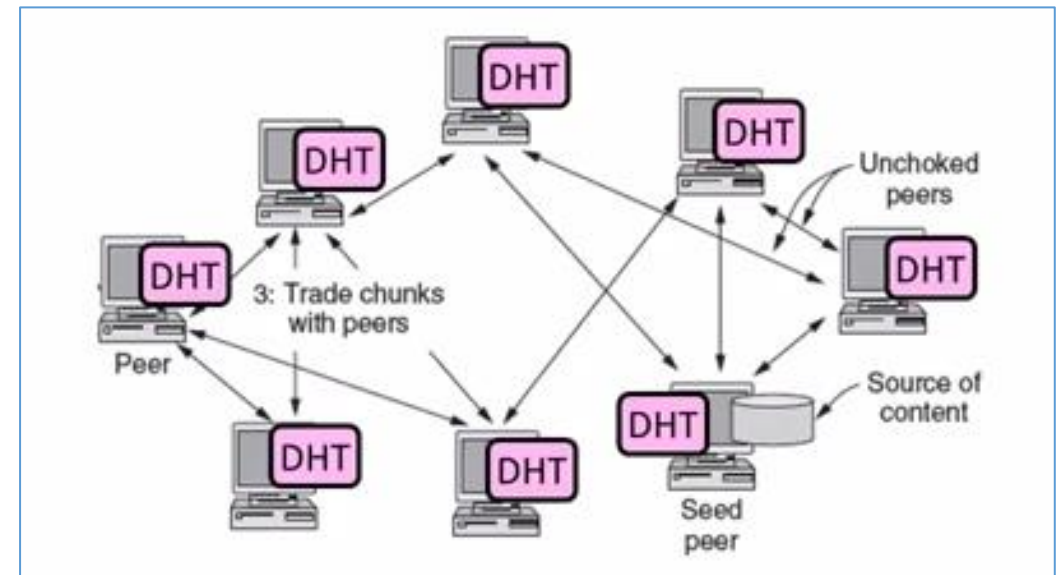
EGY TORRENT LETÖLTÉSÉNEK LÉPÉSEI

1. A *torrent* leírásával kezdődik.
2. Két lehetőség van:
 - a) Kapcsolatba lépni a *tracker*-rel a csatlakozáshoz, és elkérni a *peer*-ek listáját, amelyen legalább egy *seed peer* is van. (rég)
 - b) Vagy *DHT* index használata a *peer*-ekhez. (új)
3. A különböző *peer*-ekkel forgalom lebonyolítása.
4. Előnybe részesítjük azon *peer*-eket, akik gyorsan töltenek fel a részünkre.
 - **choke peer** – Olyan *peer*, aki korlátozza a letöltést más *peer*-ek részére.



BitTorrent tracker-rel (példa)

Forrás: [6]



BitTorrent DHT-val (példa)

Forrás: [2]

Hivatkozások

- [1] <https://www.youtube.com/watch?v=myDU5ChaMN0>
- [2] <https://www.youtube.com/watch?v=4uuggjM3F0I>
- [3] <http://root-servers.org/>
- [4] <https://www.youtube.com/watch?v=UD12J9cmVNo>
- [5] <https://www.youtube.com/watch?v=mCm2JNOcOCs>
- [6] <https://www.youtube.com/watch?v=YdLJhhWfzGM>
- [7] <https://www.youtube.com/watch?v=Zw2bvvZHJp8>

ELŐVIZSGA

Elővizsga – FAQ

- Ki elővizsgázhat?
 - *Csak az tehet elővizsgát az utolsó előadás helyén és idejében, aki az előadások legfeljebb 1/3-áról. **Azaz részvétel legalább 7 előadáson a jelenléti ívek alapján.***
- Mi van, ha valaki nem volt jelen legalább 7 előadáson?
 - Akkor a vizsgaidőszakban tehet vizsgát.
- Az elővizsga vizsga alkalomnak számít?
 - Nem, ez egy extra vizsga lehetőség.
- Mikor lesz az elővizsga?
 - Május 10-én kedden az előadás időpontjában és helyén.

Elővizsga – FAQ

- Hány részből áll vizsga feladatlap?
 - 4 részből:
 1. Definíciók kimondása. 10 kérdés. Kérdésenként egy pont.
 2. Feleletválasztás elméleti jellegű kérdésekből. 10 kérdés. Helyes válasz 1 pont, helytelen válasz -1 pont.
 3. Feleletválasztás gyakorlati jellegű kérdésekből. 10 kérdés. Helyes válasz 1 pont, helytelen válasz -1 pont.
 4. Kifejtős kérdések. Négy téma. Témánként maximum 5 pont szerezhető.

Elővizsga – FAQ

- Van-e minimum követelmény bármely részből?
 - Igen, az első 3 részből legalább 15 pontot kell szerezní a 30 pontból. Ellenkező esetben a dolgozat elégtelen(1) érdemjegyet ér.
- Van-e minimum az utolsó kifejtős résznél?
 - Nincs.
- A minimum teljesítése után mik a pont határok?

Pont határok		Érdemjegy
50-42		Jeles(5)
41-37		Jó(4)
36-30		Közepes(3)
29-25		Elégséges(2)
24- 0		Elégtelen(1)

Elővizsga – FAQ

- Kell jelentkezni elővizsgára külön?
 - Nem.
- Mikor kerül be a jegy Neptun-ba?
 - Betekintéskor jelezni kell, hogy a megajánlott jegyet elfogadja-e a vizsgázó.
- Hogyan kell jelezni az elfogadást?
 - Egy nyomtatványt az utolsó előadáson alá kell írni.

Elővizsga – menete

KONZULTÁCIÓS LEHETŐSÉG: 2016. MÁJUS 3., 16-17:30, 2.EM. 2.506 (NEUMANN NAP)

NAPPALI: 2016. MÁJUS 10., 12 ÓRAI KEZDÉSEL A KITAIBEL TEREMBEN

ESTI: 2016. MÁJUS 10., 16 ÓRAI KEZDÉSEL A MOGYORÓDI TEREMBEN

1. Jogosultság ellenőrzése a jelenléti összesítés alapján. (10-20 perc)
2. Dolgozat megírása. (60 perc)
3. Dolgozatok beszédése (5-10 perc)

Példa feleletválasztós kérdés – elméleti

- Rendelje a BitTorrent protokollt az TCP/IP rétegmodell megfelelő rétegéhez.
 - a) szállítási
 - b) kapcsolati
 - c) hálózati
 - d) egyéb:

Példa feleletválasztós kérdés – elméleti

- Hány bájtos egy *UDP* fejléc?
 - a) 40 bájtos
 - b) 8 bájtos
 - c) 20 bájtos
 - d) egyéb:

Példa feleletválasztós kérdés – gyakorlati

- Hány hosztot képes kezelni maximálisan az a hálózat, amelynek címe 135.46.56.0/23?
 - a) 512
 - b) 256
 - c) 510
 - d) egyéb:

Példa feleletválasztós kérdés – gyakorlati

- Egy végtelen populációjú réselt ALOHA-rendszer mérései azt mutatják, hogy a rések 10%-a tétlen. Mekkora a csatorna terhelés?
 - a) $G = 0.1$
 - b) $G = \ln(0.1)$
 - c) $G = \log(0.1)$
 - d) egyéb:

Vége

- Köszönöm a figyelmet!