HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

HÁLÓZATOK ALAPJAI ÉS ÜZEMELTETÉSE

Alkalmazási réteg

2019. február 11.

Zsóka Zoltán

BME Hálózati Rendszerek és Szolgáltatások Tanszék

zsoka@hit.bme.hu

MŰEGYETEM 1782

MA MIRŐL LESZ SZÓ?

1. A Web és a HTTP
2. Fájlátvitel
3. Az elektronikus levelezés protokolljai
4. Doménnév-szolgáltatás az Interneten

A fóliák elkészítéséhez felhasználtuk Jim Kurose és Keith Ross „Számítógép hálózatok működése” című könyvéhez készült fóliákat.

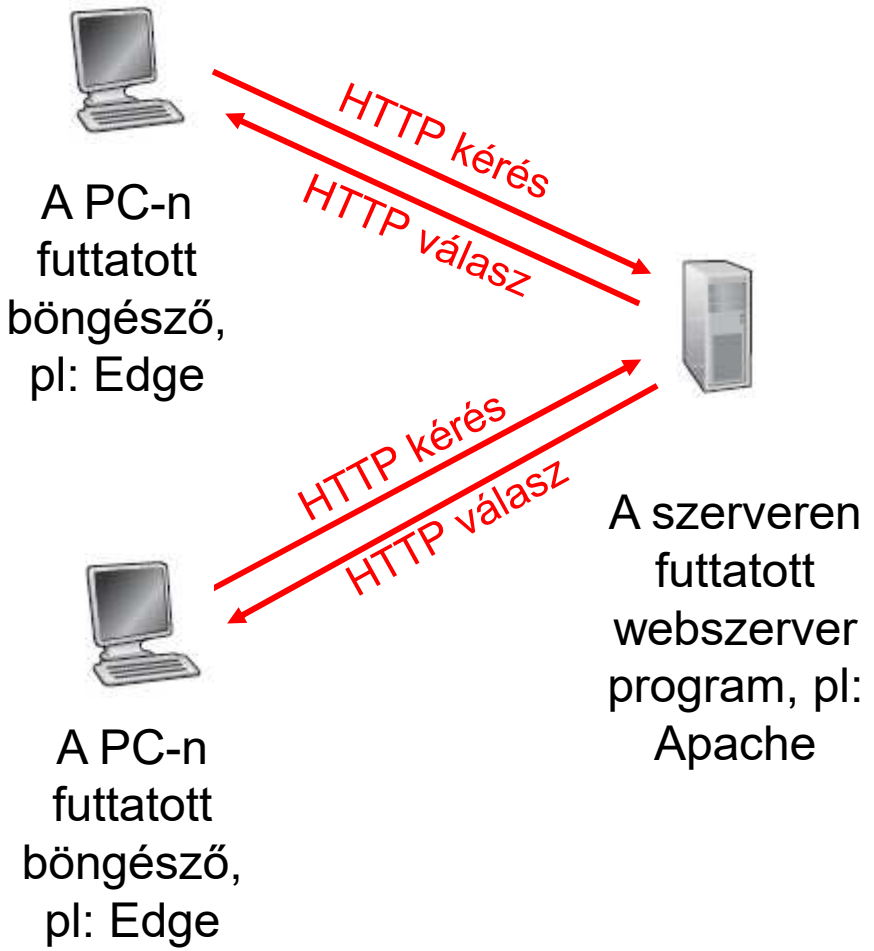
WEB ÉS HTTP

- Weboldal
  - HTML nyelven írt oldal, ami további, hivatkozott objektumokat is tartalmaz
  - Egy objektum lehet például kép, hang, videó, futtatható program vagy szkript
- HTTP – HyperText Transfer Protocol
- **Objektum**
  - Az oldalt alkotó objektumok nem feltétlenül ugyanazon a szerveren vannak
  - Objektum hivatkozása **URL**-lel (Uniform Resource Locator) történik
  - Például: `www.someschool.edu/someDept/pic.gif`

Szerver hosztneve | Elérési út a szerveren

A HTTP PROTOKOLL

- A Web alkalmazási réteg protokollja
  - **kliens**: böngésző, ami objektumokat kér, fogad és “megjelenít”
  - **szerver**: a webszerver, ami objektumokat küld a kérésekre válaszul
- Verziók
  - HTTP/1.0: RFC 1945 (1996)
  - HTTP/1.1: RFC 2068 (1997)
  - HTTP/2: RFC 7540 (2015)
  - HTTP/3: Internet-draft (2018)
  - A korábbi verziókhoz újabb RFC-k is íródtak



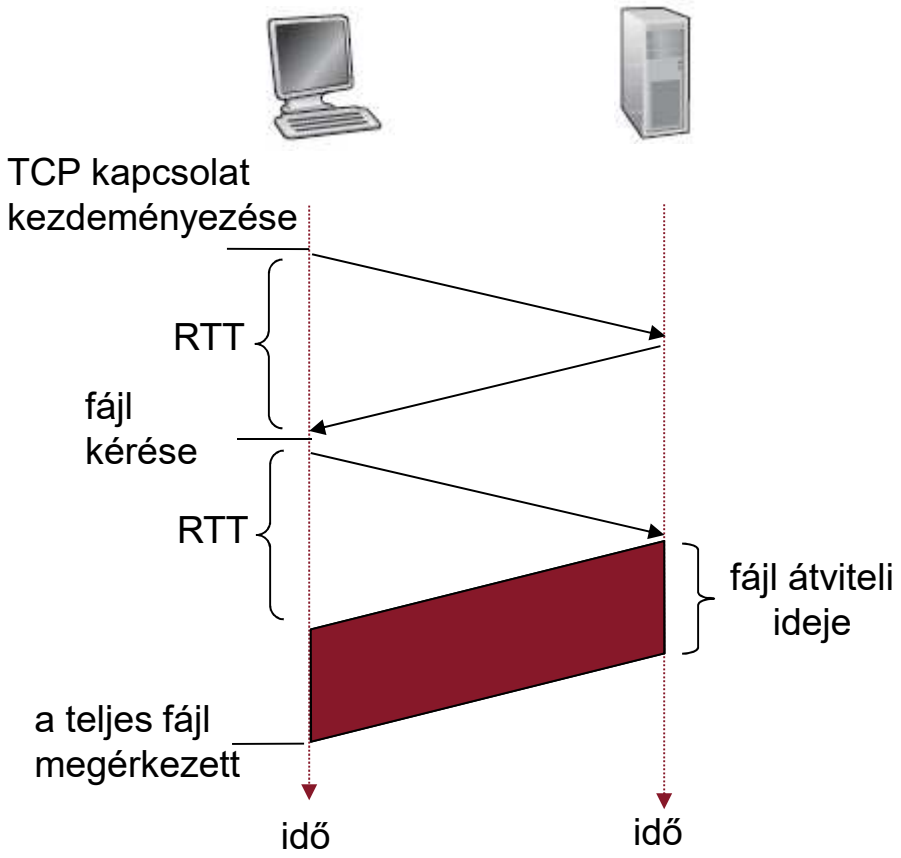
HTTP JELLEMZŐK

- **TCP**-t használ
  - A kliens TCP kapcsolatot kezdeményez a szerverrel, annak 80-as portján
  - a szerver elfogadja a TCP kapcsolatot a klienssel
  - HTTP üzeneteket váltanak
  - Végül lezárják a TCP kapcsolatot
  - Kivétel a HTTP/3
- Két üzenettípus
  - **Kérés** (request)
  - **Válasz** (response)

- **Állapotmentes** protokoll
  - A szerver nem tartja nyilván a kliens(ek) korábbi kéréseit
- Egy kapcsolaton belül küldött adatok
  - **Nem perzisztens** (időleges) eset: egyetlen objektumot/kapcsolat
    - HTTP/1.0
  - **Perzisztens** (állandó) eset: több (az összes) objektum/kapcsolat
    - HTTP/1.1 -től

NEM PERZISZTENS HTTP IDŐIGÉNYE

- Minden egyes objektumra 3·RTT+átviteli idő
  - TCP kapcsolat kezdeményezése (és elfogadása)
  - HTTP kérés és a HTTP válasz első byte-jának megérkezése
  - A fájl átviteli ideje
  - Kapcsolatbontás
- A böngésző esetleg nyithat párhuzamos TCP kapcsolatokat az egyes objektumokhoz
- Az operációs rendszernek kezelnie kell minden egyes TCP kapcsolatot



RTT: Round Trip Time

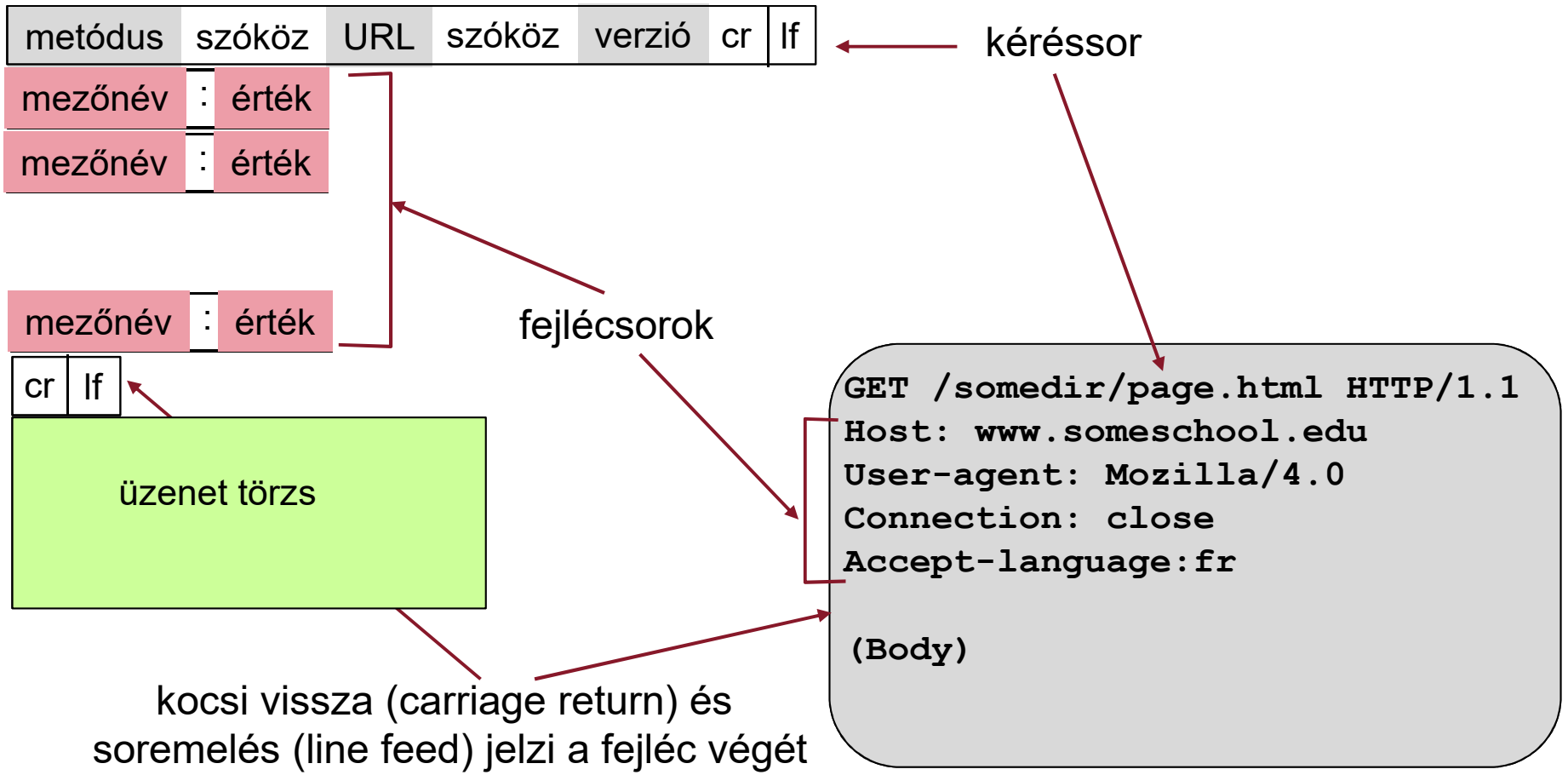
PERZISZTENS HTTP KAPCSOLAT

- A szerver **nyitva hagyja a TCP kapcsolatot** a válasz elküldése után
- Nem kell újra nyitni objektumonként
- Egymást követő HTTP üzenetváltások egy adott kliens-szerver viszonylatban ugyanazon a nyitott TCP kapcsolaton

- Pipelining (csővezetékezés, átlapolás) nélkül
  - A kliens csak akkor küld új kérést, ha az előző kérésre megérkezett a válasz
  - Kérés+válasz időigénye: 1·RTT **minden egyes hivatkozott objektumra**
  - N objektum időigénye: (1+N+1)·RTT + átviteli idő
- Pipelining esetén
  - A kliens azonnal kérést küld amint hivatkozott objektumra bukkan
  - Kérés+válasz időigénye: 1·RTT **az hivatkozott objektumokra összesen**
  - N objektum időigénye: (1+1+1)·RTT + átviteli idő
  - Alapértelmezett a HTTP/1.1-ben és újabbakban

HTTP KÉRÉSÜZENET

ASCII (ember számára olvasható formátum)



HTTP METÓDUSOK

HTTP/1.0

- GET
  - Objektum letöltése az URL alapján
  - Űrlap-adat feltöltése az URL-ben lévő információként

```
www.hit.bme.hu/asearch?monkey&banana
```

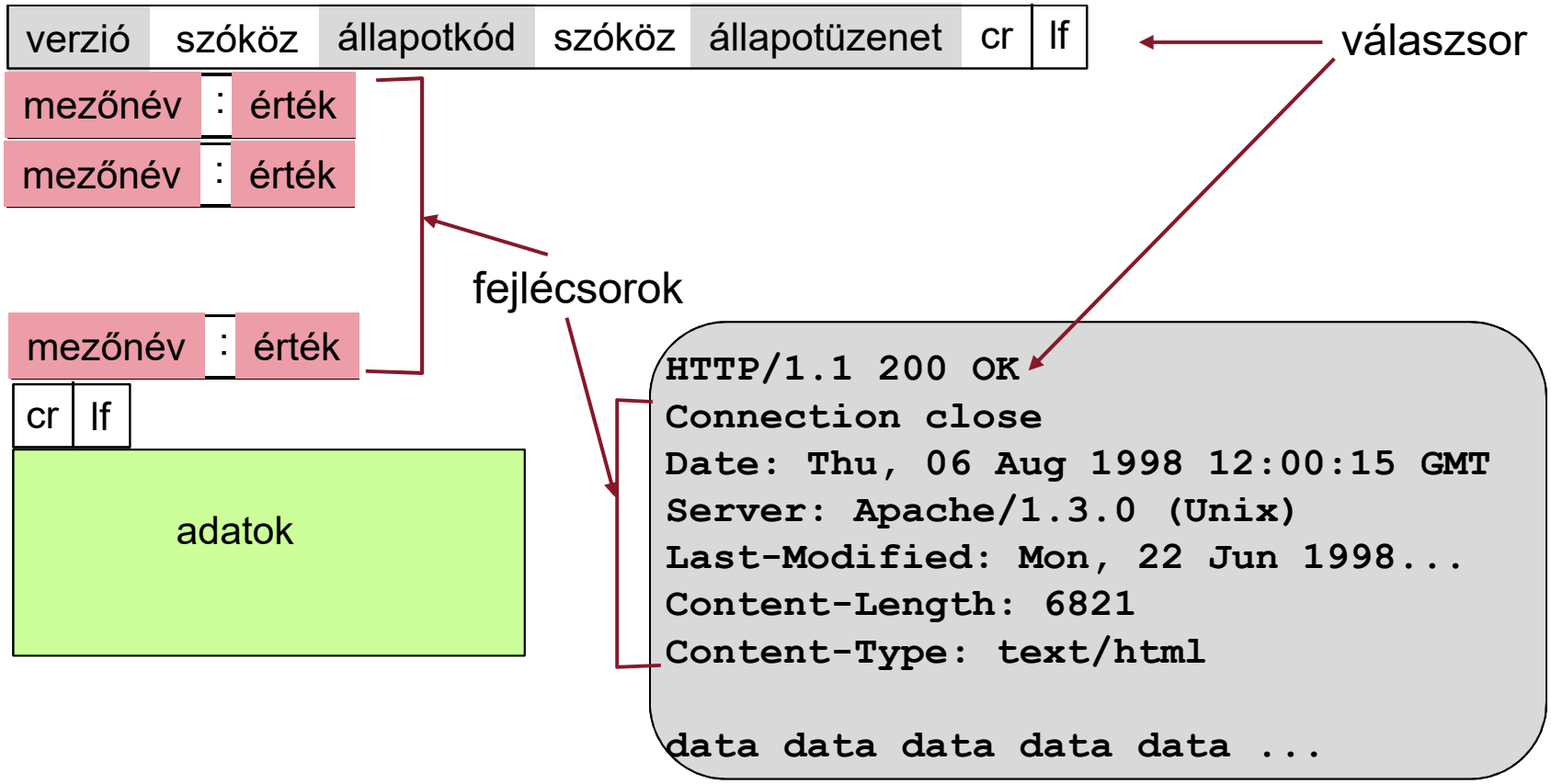
- POST
  - Űrlap-adat feltöltése az üzenet törzsében
- HEAD
  - Csak fejléc (pl. hibakeresésre)

HTTP/1.1

- GET, POST, HEAD
- PUT
  - Fájl feltöltése adott könyvtárba
- DELETE
  - Fájl törlése könyvtárból
- Továbbiak: CONNECT, OPTIONS, TRACE

HTTP VÁLASZÜZENET

Felépítésében nagyon hasonló a kérésüzenethez



HTTP VÁLASZÜZENET ÁLLAPOTKÓDOK

- A kérés teljesüléséről visszajelzést ad a szerver
- A legjellemzőbb állapotkódok és a kapcsolódó szöveges állapotleírások

200 OK
- sikeres kérés, a kért objektum az üzenetben

301 Moved Permanently
- a kér objektum elmozgatva, új helye megadva az üzenetben

400 Bad Request
- a szerver nem érti a kérésüzenetet

403 Forbidden
- az objektum hozzáférése megtagadva

404 Not Found
- A kért objektum nincs ezen a szerveren

505 HTTP Version Not Supported
- A küldéskor jelzett HTTP protokollváltozatot a szerver nem támogatja

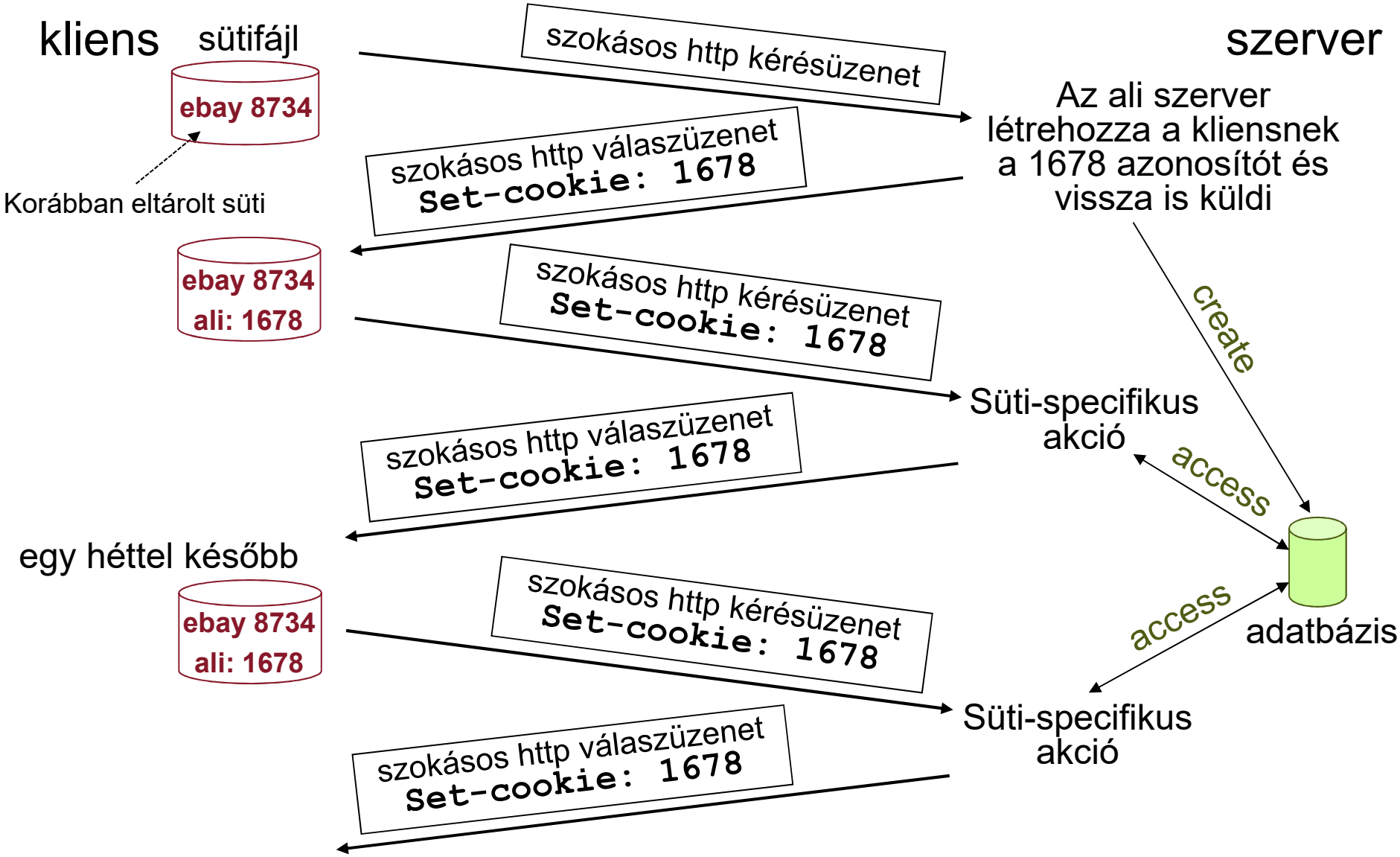
ÚJABB HTTP VÁLTOZATOK

- HTTP/2
  - A HTTP/1.x kiterjesztése
  - Új mechanizmusok
    - **Egyetlen TCP kapcsolat** épül fel
    - Több azonosított adatfolyam (stream) küldözget üzeneteket (message)
    - Adatátvitel **binárisan kódolt elemekben** (frame)
    - A folyamokat **priorizálhatjuk** és **multiplexáljuk** (interleaving)
    - **Szerver oldali PUSH**: kérés nélkül küldhet objektumot
  - Előnyök
    - Gyorsabb
    - Head Of Line blokkolás elkerülhető
    - Prioritások állíthatók be
- HTTP/3
  - Nem kiterjesztés
  - **TCP helyett QUIC** transzport protokoll
  - Gyorsabb kapcsolat-felépítés tanúsítványkezeléssel

ÁLLAPOTOK KEZELÉSE SÜTIKKEL

- A legtöbb weboldal használ sütiket (**cookie**)
- A mechanizmus alkotórészei
  - Süti fejlécsor a HTTP válaszüzenetben
  - Süti fejlécsor a HTTP kérésüzenetben
  - A kliens hoszt tárolja a sütit, amit a felhasználó böngészője kezel
  - Háttéradatbázis a webszerveren
- Mire használható?
  - Felhasználó azonosítása
  - Felhasználói munkamenet vagy folyamat állapotának tárolása, pl. kosár tartalma, webes e-mail
  - Ajánlások
- Személyes adatok védelme
  - A sütik alapján egy sor dolgot megtudnak a szerverek felhasználókról
  - Érzékeny adatokat is tárolhatnak bennük
  - Az EU-ban jóvá kell hagynunk a használatát

SÜTIKEZELÉS ILLUSZTRÁCIÓJA



WEB GYORSÍTÓTÁRRAL

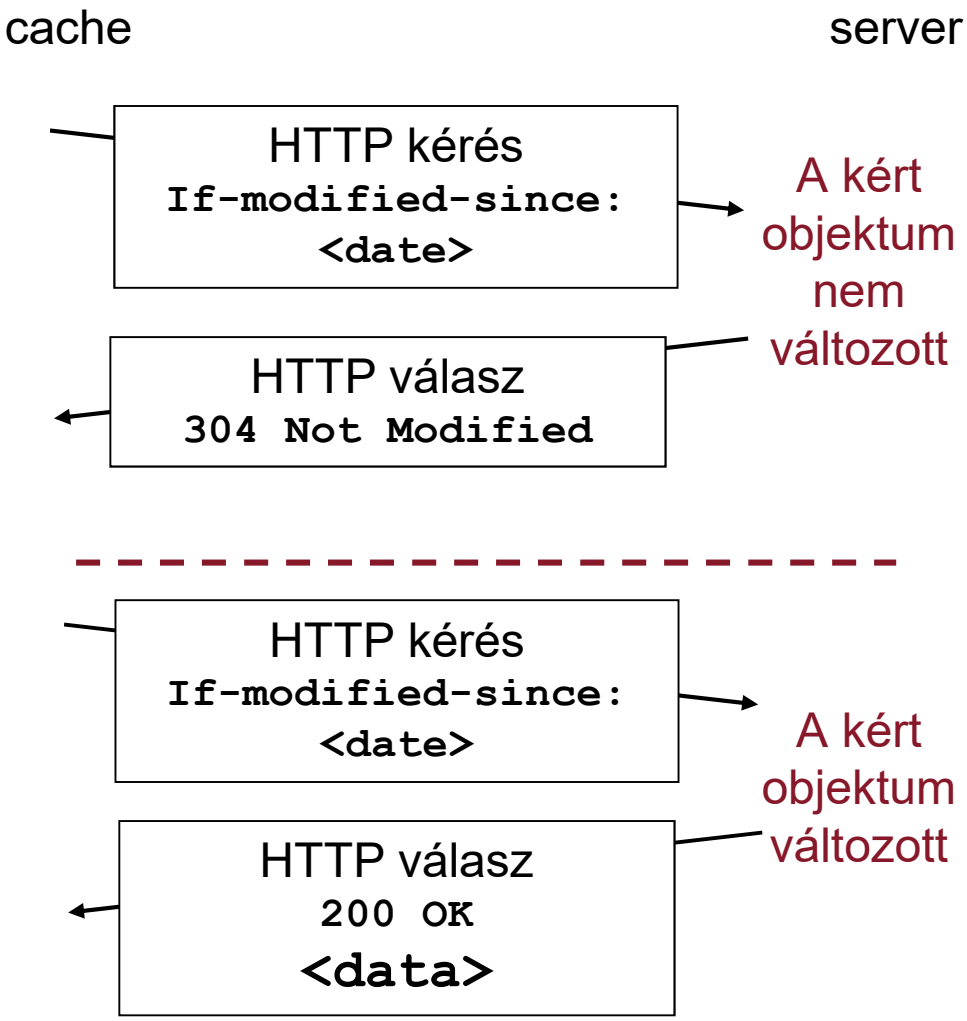
- A gyorsítótár (web-cache) segítségével a kliens kérését a tényleges szerver közreműködése nélkül szolgáljuk ki
- A böngészőben állítható be
- A böngésző minden HTTP kérést a gyorsítótárnak küld
  - Ha megvan a kért objektum a cache-ben: a cache elküldi a kliensnek
  - Ha nincs meg, akkor a cache kéri az objektumot a forrás szervertől, és miután megkapta, elküldi a kliensnek
  - A találati arány nem mindig 100%
- A cache kliensként és szerverként is funkcionál
- A gyorsítótárat tipikusan az ISP telepíti (egyetem, vállalat, lakossági szolgáltatást nyújtó ISP)

A WEB-CACHE ELŐNYEI

- Csökkentheti a kliens kérés kiszolgálásának válaszidejét
- Csökkenti az intézmény nyilvános csatlakozási linkjének forgalmát és kihasználtságát
- A sűrűn telepített cache-ek esetén a “szegényebb” tartalomszolgáltatók is hatékonyan tudják terjeszteni tartalmaikat az Interneten

FELTÉTELES LEKÉRÉS

- Cél, hogy ne küldjön el a szerver egy objektumot, ha annak legfrissebb változata megvan a cache-ben
- A cache a kérés fejlécében megadja a tárolt változat dátumát
- Ha a tárolt változat a legfrissebb, a szerver válasza nem tartalmazza az objektumot, és ezt a státuszban jelzi



A HTTP-N TÚL

- **HTTPS**
  - Biztonságos webelérés
  - Hitelesítés – az oldal tényleg az, aminek mondja magát
  - Titkosítás – egyszerű szöveg helyett titkosított adatátvitel
  - A szerver a 443-as TCP porton hallgatózik
- **REST**
  - Representational State Transfer
  - Nem protokoll, hanem koncepció hálózati applikációkhoz, API-khoz
    - Állapotot csak a kliens tárol
    - Erőforrások azonosítása URI (Uniform Resource Identifier) segítségével
  - A jellegzetes funkciók HTTP kérési metódusokkal és a rájuk kapott válaszokkal megoldhatók
    - CRUD (create, retrieve, update, delete)

MA MIRŐL LESZ SZÓ?

1. A Web és a HTTP
2. Fájlátvitel
3. Az elektronikus levelezés protokolljai
4. Doménnév-szolgáltatás az Interneten

AZ FTP PROTOKOLL

- File Transfer Protocol
- Kliens/szerver modell
  - Kliens: az átvitelt kezdeményező fél, aki távolra, vagy távolról szeretne **fájlokat másolni**
  - Szerver: a távoli hoszt
- **Sávon kívüli (Out-of-band)** vezérlés
  - Külön vezérlési és adatátviteli összeköttetés
- Nem állapotmentes

FTP FÁJLÁTVITEL

- Lépések
  1. Az FTP kliens csatlakozik az FTP szerverhez a 21-es TCP porton
  2. A kliens azonosítása a vezérlési összeköttetésen
  3. A kliens áttekintheti a távoli könyvtárakat megfelelő parancsokat küldve a vezérlési összeköttetésen
  4. A mikor fájlátvitel parancs érkezik a szerver nyit egy **újabb TCP összeköttetést** (a 20-as porton) a kliens felé a fájl átviteléhez
  5. A fájl átvitele után a szerver lezárja ezt a TCP összeköttetést, de a vezérlési kapcsolat tovább él, és ugrunk a 3. lépésre
- Parancsok
  - ASCII kódolású **szövegként** küldve a vezérlési összeköttetésen
- Válaszok
  - **Státuszkódok** és szöveges leírások, hasonlóan mint a HTTP-nél

SCP, SFTP, SMB

- FTP problémák
  - Nem biztonságos
  - Mindent külön paranccsal kell letölteni
- Secure CoPy
  - Titkosított, ssh alapú átvitel: a 22-es TCP porton keresztül
  - Csak fájlátvitel
- SSH File Transfer Protocol
  - Titkosított, ssh alapú átvitel: a 22-es TCP porton keresztül
  - Könyvtárkezelés is elérhető (az ftp-hez hasonlóan)
- Server Message Block
  - Fájlmegosztás Microsoft hálózatokban
  - Kliens-szerver architektúra
  - A 445-ös TCP porton keresztül
  - Szabad hozzáférésű implementáció: Samba

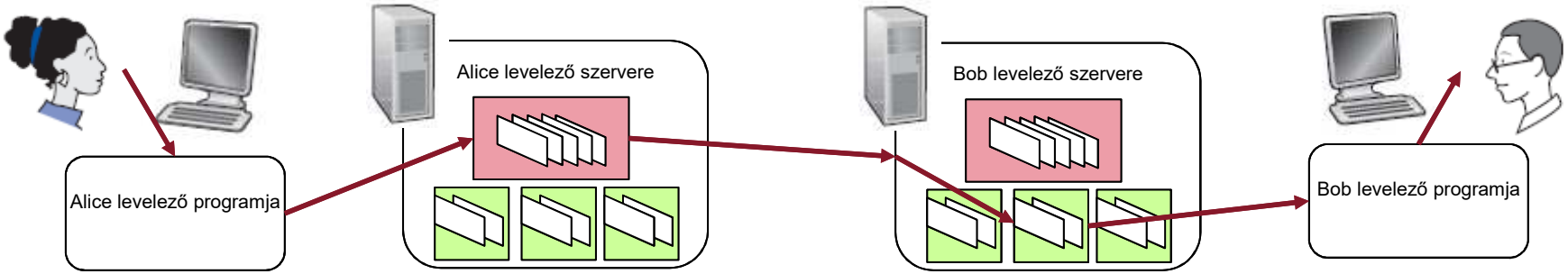
MA MIRŐL LESZ SZÓ?

1. A Web és a HTTP
2. Fájlátvitel
3. Az elektronikus levelezés protokolljai
4. Doménnév-szolgáltatás az Interneten

ELEKTRONIKUS LEVELEZÉS

- Levelezőszerver
  - A **postafiók** (mailbox) a felhasználó bejövő üzeneteit (leveleit) tárolja
  - A **kimenő üzenetsorban** (message queue) az elküldendő levelek vannak
- Felhasználói ügynök program
  - Levél megírása, szerkesztése, olvasása
  - Levél küldése és lekérése a szervernek, ill. szervertől
  - pl. Outlook, Opera Mail, Thunderbird, Firefox
- Azonosított szerepek:
  - Mail User Agent (MUA)
  - Mail Submission Agent (MSA)
  - Mail Transfer Agent (MTA)
  - Mail Delivery Agent (MDA)
- Az egyes hosztokon több szerep megvalósítása is szükséges lehet
- A levélküldési protokoll szempontjából egy levelezőszerver lehet:
  - Kliens – kezdeményező, küldő
  - Szerver – fogadó

LEVÉLKÜLDÉS



1. Alice felhasználói ügynökével (MUA-Alice) levelet ír és megcímzi `bob@someschool.edu`
2. Alice felhasználói ügynöke elküldi a levelet Alice levelezőszerverének (MSA), ami a szerveren a kimenő üzenetsorba teszi
3. A levelezőszerver (MTA-Alice) kapcsolatot létesít Bob levelezőszerverével (MTA-Bob)
4. Az egyik szerver (MTA-Alice) elküldi a levelet a másiknak (MTA-Bob)
5. Bob levelezőszervere elhelyezi a megkapott levelet Bob postafiókjába (MDA-Bob)
6. Bob a felhasználói ügynöke (MUA-Bob) segítségével elolvassa a levelet, levél-hozzáférés

AZ SMTP PROTOKOLL

- TCP-t használ a levelek megbízható továbbítására a kliens és a szerver között, a 25-ös porton
- közvetlen átvitel
  - a küldő és fogadó levelezőszerver között
  - a MUA és az MSA között
- három szakaszból áll
  - kézfogás, üdvözlés (greeting)
  - üzenet (levél) továbbítása
  - lezárás
- parancs/válasz protokollüzenetek
  - ASCII szöveges parancsok
  - állapotkód, szöveges állapotleírás a válaszban
- az üzenetnek (levél) 7-bites ASCII kódolásúnak kell lennie

SMTP KAPCSOLAT ILLUSZTRÁCIÓ

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Sweet but a psycho. . .
C: A little bit psycho
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

SMTP LEVÉLÜZENET

- Eredetileg csak **szöveges** üzenet formátum
  - Fejlécsorok, pl:
    - To:
    - From:
    - Subject:
    - Nem tévesztendők össze az SMTP parancsokkal!
  - Törzs
    - a “levélüzenet”: csak ASCII karakterek
- SMTP problémák
  - Csak szöveg?
  - Azonosítás?
  - Adatbiztonság?

- Kiterjesztés **bináris** adatokra
  - MultImedia Mail Extension
  - További fejlécek jelzik, pl:
    - MIME-Version
    - Content-Transfer-Encoding
    - Content-Type

LEVÉL-HOZZÁFÉRÉSI PROTOKOLLOK

- Közvetlen átvitel az MTA és a MUA között
- **Post Office Protocol (POP3)**
  - Szöveges parancsok és válaszok
  - Azonosítás
  - Tranzakciók
    - Letöltés és törlés mód: a MUA lokálisan tárolja a leveleket
    - Letöltés és megtartás mód: szétosztjuk a leveleket a felhasználó MUA-i között
  - Nem tárol állapotinformációkat az egymást követő munkamenetek (session) között
- **Internet Mail Access Protocol (IMAP)**
  - Minden levélüzenetet a levelezőszerveren tárol
  - Támogatja, hogy a felhasználó mappákba (folder) rendezze tárolt üzeneteit
  - A felhasználói állapotokat is tárolja az egymást követő munkamenetek (session) között:
    - A mappák nevei
    - Mappák és levélüzenet azonosítók összerendelése
- HTTP: Gmail, Hotmail, Yahoo! Mail, stb.

MA MIRŐL LESZ SZÓ?

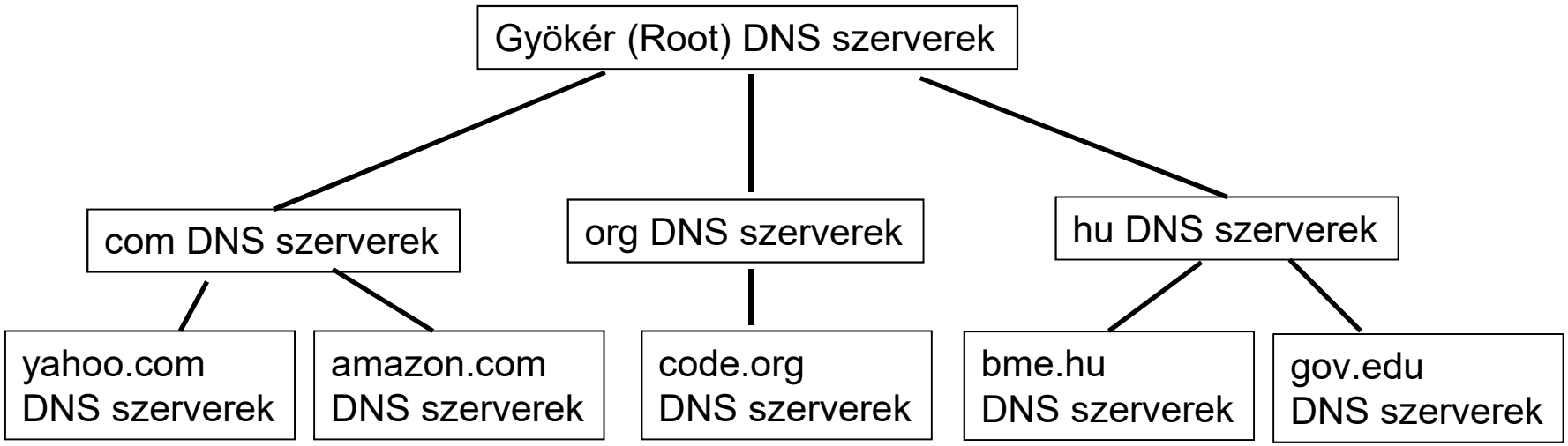
1. A Web és a HTTP
2. Fájlátvitel
3. Az elektronikus levelezés protokolljai
4. Doménnév-szolgáltatás az Interneten

DOMÉNNÉV-SZOLGÁLTATÁS

- Internet hosztok, routerek azonosítása
  - IP cím
  - Emberi használatra is alkalmas hosztnév, pl. www.hit.bme.hu
  - Hogyan párosíthatók az IP címek és a nevek?
- **Domain Name System (DNS)**
  - Elosztott adatbázis hierarchiába szervezett, nagyszámú szerver
  - Alkalmazás rétegbeli protokoll
- DNS szolgáltatások
  - Hosztnév IP címre „fordítása”
  - Valódi (kanonikus, canonical) nevek
  - Hoszt álnevek (alias)
  - Levelezőszerverek neve
  - Terheléselosztás: több IP cím tartozik egy kanonikus névhez, pl megtöbbszörözött web szerverek

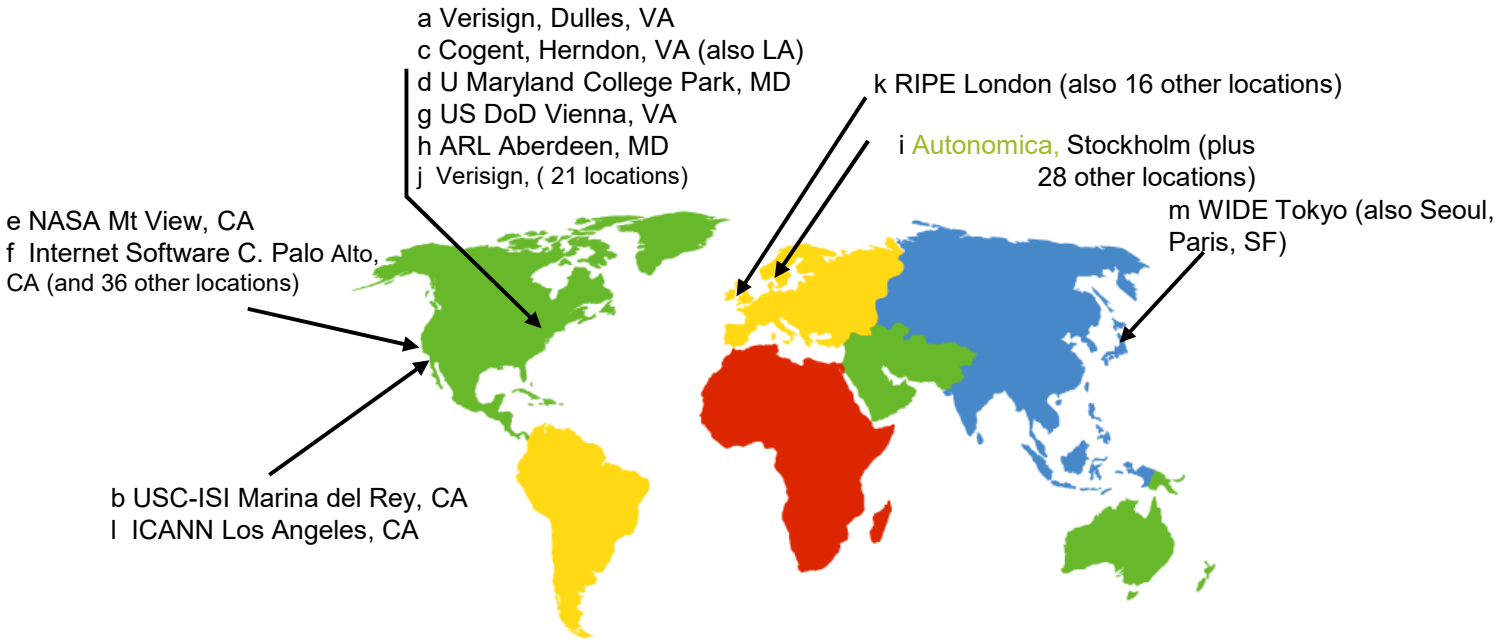
DNS ADATBÁZIS

- Elosztott információ-tárolás
  - Elosztott felelősség
  - Robusztusság
  - Karbantartás
  - Skálázhatóság
- Hierarchia
  - Strukturált keresés
  - Egy szervernek elég az alatta lévő szint szervereit ismerni
  - Skálázhatóság



DNS SZERVEREK

- Gyökér (**Root**) szint
  - Világszerte 13 szerver
  - Ismeri a TLD szerverek címét
  - Hozzájuk fordulnak a helyi szerverek
- Legfelső szint (**Top Level Domain, TLD**)
  - A com, org, net, edu **domének** (körzetek)
  - Országokhoz tartozó domének, pl: hu, it, de

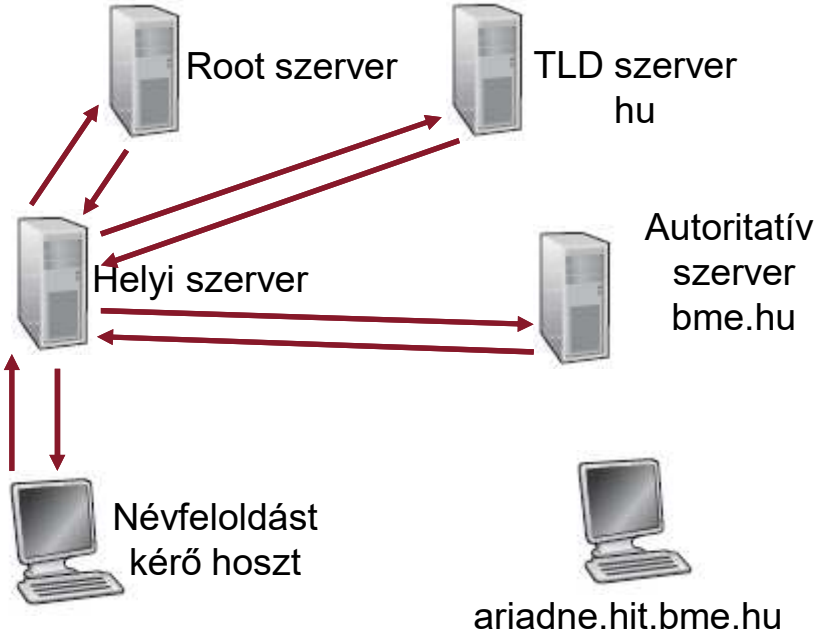


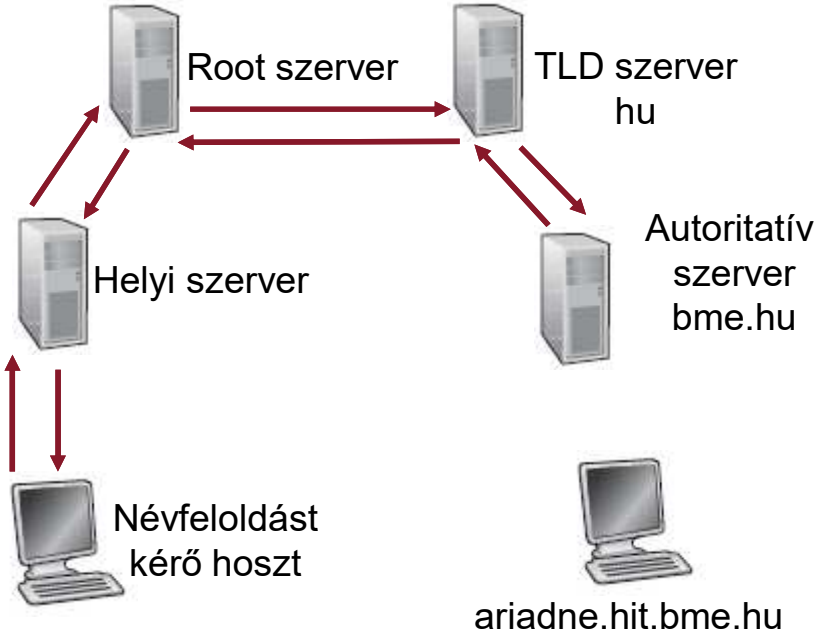
DNS SZERVEREK – FOLYTATÁS

- Hiteles (**Authoritative**) szerverek
  - Internetes hosztokat üzemeltető szervezetek szerverei
  - Gyakorlatilag minden céghez, egyetemhez, stb. tartozik egy
  - A szervezet hosztjainak, szervereinek nevét tudja feloldani
  - Nem biztos, hogy a szervezet üzemelteti
- Helyi (**Local, Default**) szerver
  - Nem tartozik a hierarchiába
  - Minden ISP működtet egyet
  - A hosztok ettől kérik egy ismeretlen doménnév vagy hosztnév feloldását
- **Gyorsítótár (cache)**
  - Egy megismert összerendelést a szerver bizonyos ideig tárol
  - Minden szerver alkalmazhatja
- TLD címek helyi tárolása

NÉVFELOLDÁS

- **Iteratív** lekérdezés
  - a megszólított névszerver megadja, hogy helyette melyik névszerverhez kell fordulni a kéréssel
- **Rekurzív** lekérdezés:
  - A névfeloldás feladatát a megszólított névszerverre bízza





DNS BEJEGYZÉSEK

- **Resource Record (RR)**
  - Típus (type)
  - Név (name)
  - Érték (value)
  - Élettartam (Time To Live)
- **A** típusú RR
  - név: hosztnév
  - érték: a névhez tartozó IP cím
- **NS** típusú RR
  - név: egy domén neve (pl. foo.com)
  - érték: a domén autoritatív szerverének a hosztneve
- **AAAA** típusú RR
  - név: hosztnév
  - érték: a névhez tartozó IPv6 cím
- **CNAME** típusú RR
  - név: egy valódi (kanonikus) névhez tartozó álnév (pl. www.polito.it)
  - érték: a kanonikus név
- **MX** típusú RR
  - név: a levelező szerver álneve
  - érték: levelező szerver kanonikus neve

DNS PROTOKOLLÜZENETEK

- **Lekérdezés (query)** és **válasz (reply)**
  - Azonos üzenetformátum
  - A fejléc jelzőbitjeiből derül ki, hogy melyik
- Azonosítás
  - 16 bites szám a lekérdezéshez
  - A válaszban ugyanez az azonosítószám
- A kérdések név- és típusmezőkkel adottak

azonosítás	jelzőbitek
kérdések száma	válasz RR-ek száma
hiteles RR-ek száma	kiegészítő RR-ek száma
Kérdések (változó számú kérdés)	
Válaszok (változó számú RR erőforrás-bejegyzés)	
Autoritatív szerver adatok (változó számú RR erőforrás-bejegyzés)	
Kiegészítő információk (változó számú RR erőforrás-bejegyzés)	

HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK