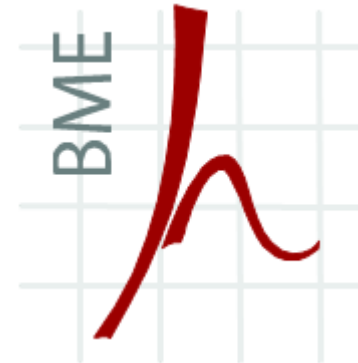




# HÁLÓZATI ALKALMAZÁSOK

Dr. Simon Vilmos  
docens

BME Hálózati Rendszerek és Szolgáltatások Tanszék  
[svilmos@hit.bme.hu](mailto:svilmos@hit.bme.hu)

2017.november 29.

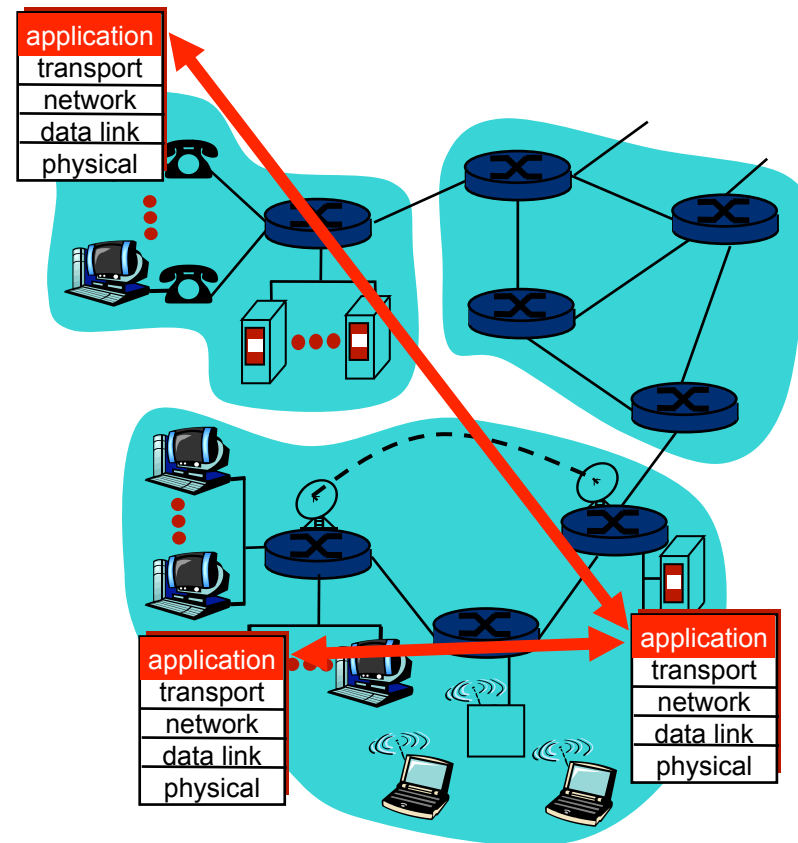
- Hálózati alkalmazások
- Alkalmazásprotokollok
- Infrastrukturális szolgáltatások
  - Névfeloldási szolgáltatás
  - Címkonfiguráció
- Levelezési rendszerek
- Webes rendszerek

## Olyan programot kell írni

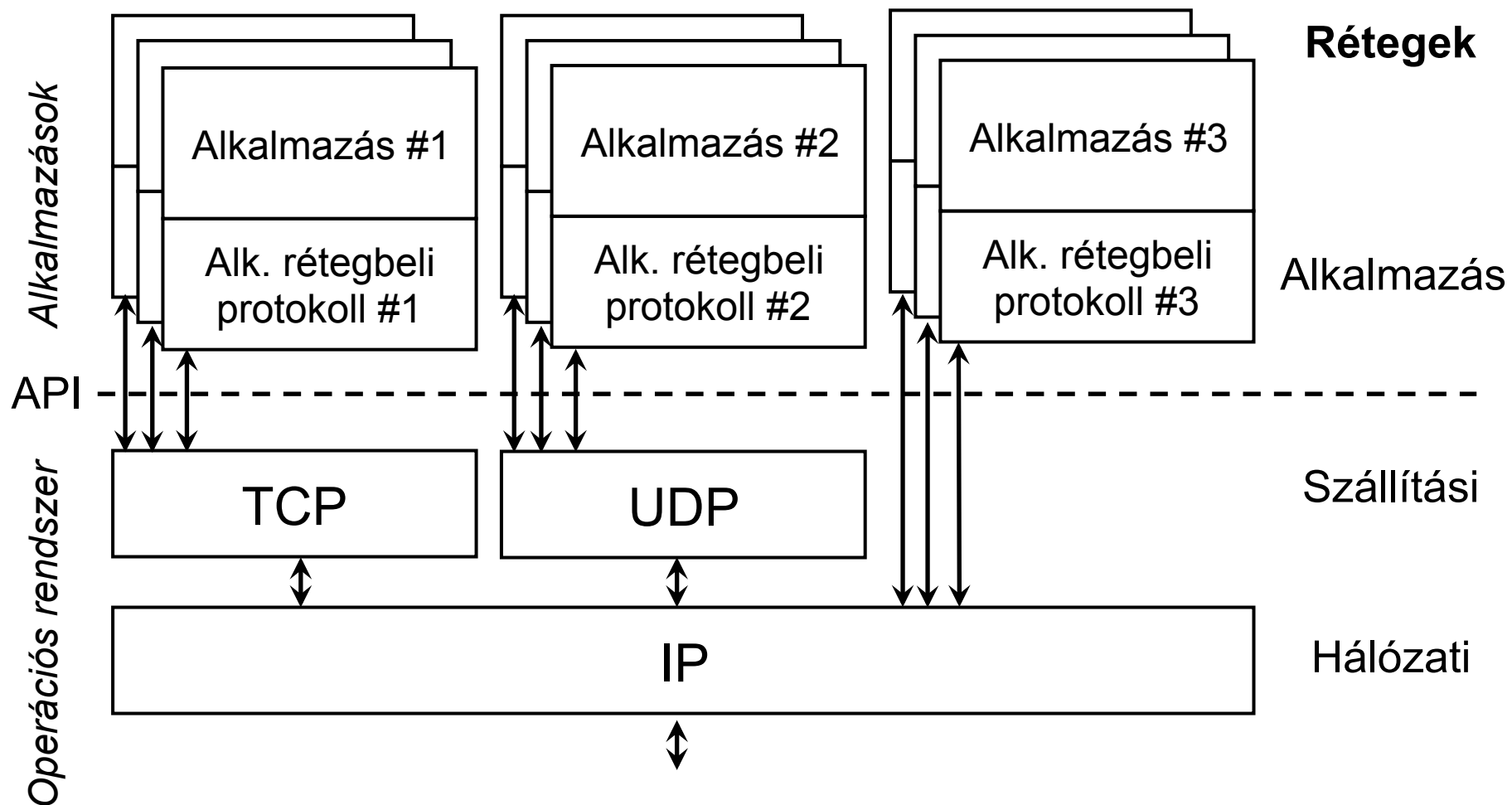
- ami különböző végfelhasználói rendszereken fut és
- a hálózaton keresztül kommunikálnak.
- pl. Web: web server sw-e kommunikál a webböngésző sw-ével

## A hálózati gerincen

- nem fut ez a kód (alacsonyabb rétegek vannak csak megvalósítva)
- gyors alkalmazásfejlesztés és terjesztés ennek köszönhetően

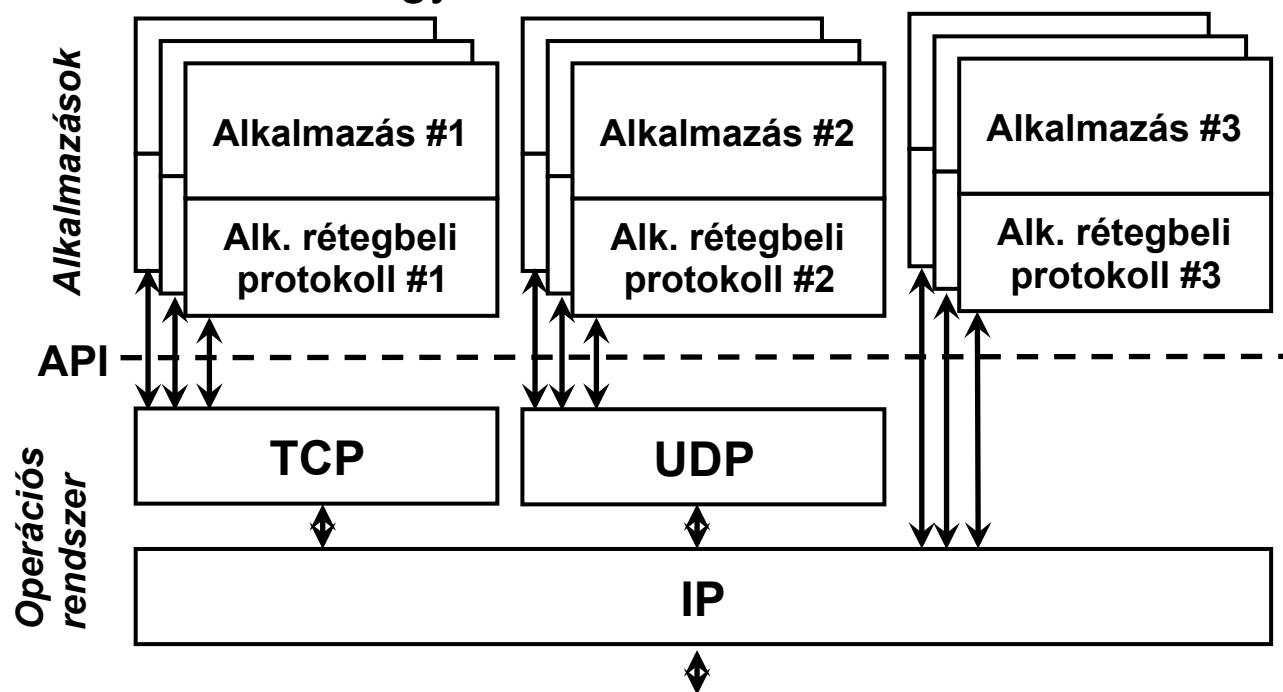


# Alkalmazások kapcsolata az alsóbb rétegekkel

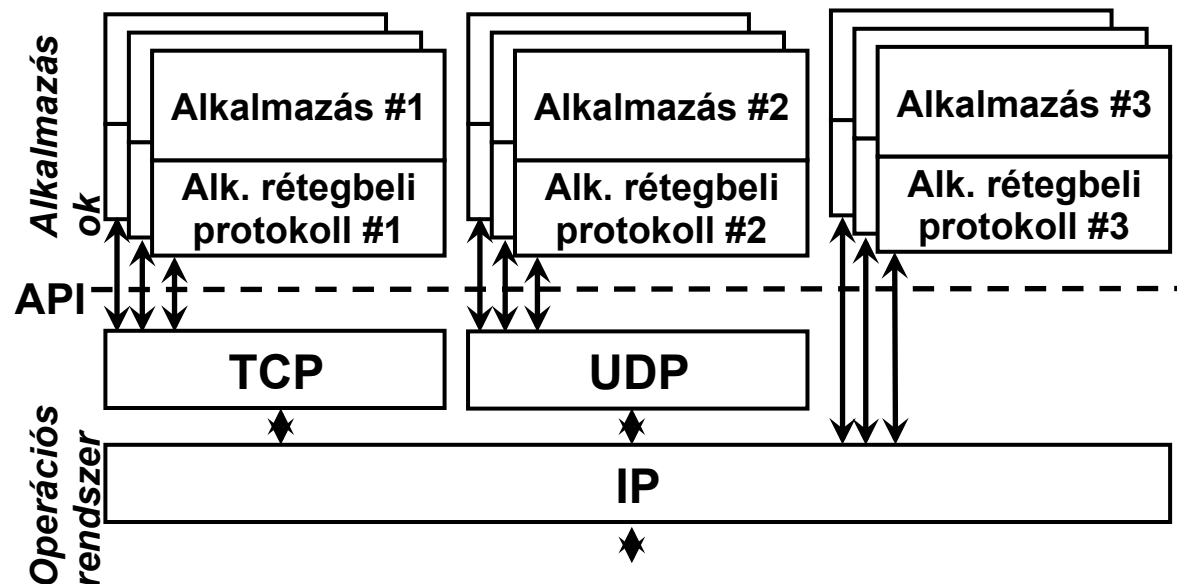


# Alkalmazás-rétegbeli protokollok

- Legtöbbször az alkalmazásban kerül implementálásra
  - Alkalmazás logikájához szorosan kapcsolódik
  - Egy alkalmazás-rétegbeli protokollt tipikusan kevés alkalmazás használja
- Mégis szükséges szabványosítani (lásd RFC-k)
  - Alkalmazások együttműködése

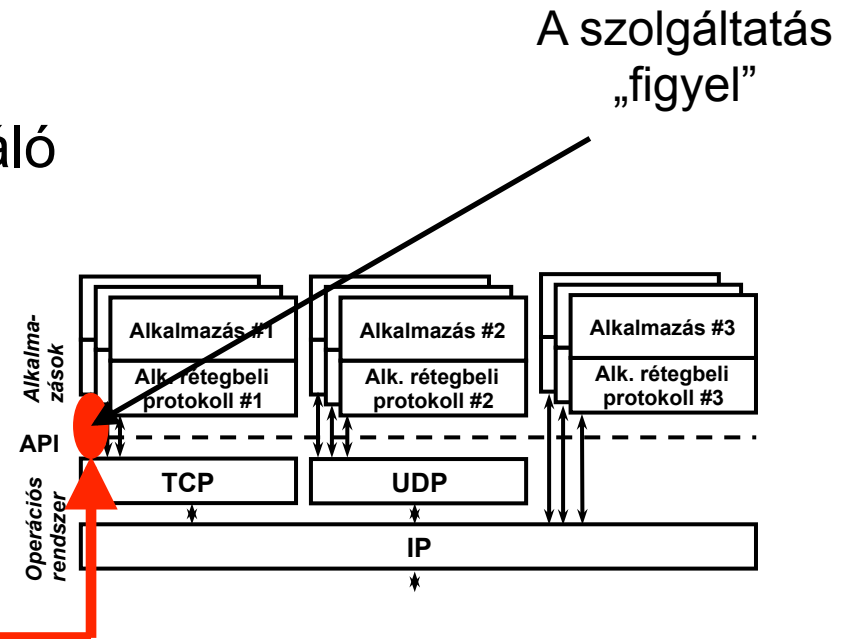
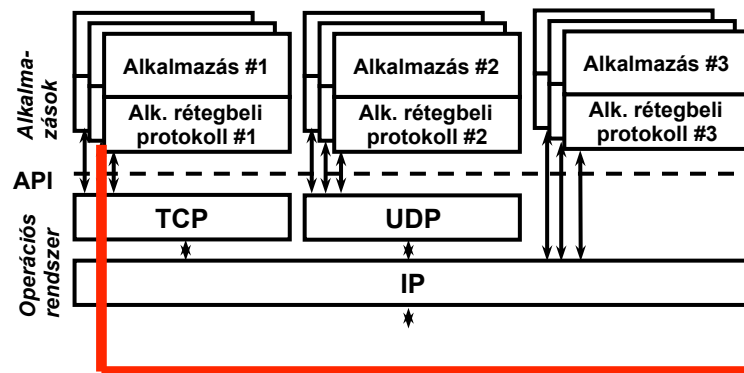


- **Alsóbb rétegeket** – mint szolgáltatásokat – az **operációs rendszer** biztosítja
- **Elfedi** a tényleges rétegeket
- Csak egy interfészt (API: Application Programming Interface) biztosít
  - SAP (Service Access Point)
- Ennek rendszerhívásait használva létrehozható a kívánt kommunikációs csatorna
  - az alkalmazás által használható végződése: **socket**



# Kliens-szerver architektúra

- Kliens
  - Kapcsolatot kezdeményező fél
- Szerver
  - Szolgáltatást nyújtó kiszolgáló

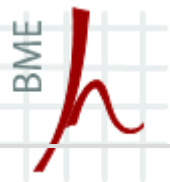


- Kliensnek a szolgáltatást meg kell címeznie
  - IP-cím (vagy DNS név) + szállítási protokoll + portszám

- Szerveren
  - Szolgáltatást azonosítja
    - egy port maximum egy szolgáltatáshoz lehet hozzárendelve
  - **Statikus**
  - 1-65536 tartományból tipikusan 1-1023-ig
    - **System Ports** (régi neve “*well-known ports*”)
    - privilegizált szerver programok kapcsolódhatnak csak
- Kliensen
  - Dinamikusan kerül kiosztásra a még nem használtak közül
  - 1-65536 tartományból 1024-65535-ig
    - **User Ports** (regisztrált portok, 1024-49151) illetve **dinamikus portok** (49152-65535)

# Milyen alkalmazás milyen szállítási rétegbeli protokollon?

|                        | Natív IP                                            | UDP                                | TCP                                |
|------------------------|-----------------------------------------------------|------------------------------------|------------------------------------|
| Kapcsolatorientált     | ✗                                                   | ✗                                  | ✓                                  |
| Megbízható             | ✗                                                   | ✗                                  | ✓                                  |
| Üzenetméret (tipikus)  | rövid                                               | rövid                              | hosszú                             |
| Adattovábbítás jellege | datagram                                            | datagram                           | bitstream pipe                     |
| Portkezelés            | ✗                                                   | ✓                                  | ✓                                  |
| Overhead               | minimális                                           | kicsi                              | nagy                               |
| Alkalmazások           | vezérlési és menedzsment<br>•ICMP, IGMP<br>•Routing | multimédia-átvitel,<br>névfeloldás | fájllátvitel,<br>web,<br>levelezés |



# Alkalmazások, protokollok és portszámok – natív IP felett

- Natív IP szolgáltatást igénybevevő protokollok és az IP Protocol mező értéke\*:
  - Vezérlés:
    - 1: ICMP (Internet Control Message Protocol)
  - Routing:
    - 89: OSPF (Open Shortest Path First)
  - Valódi szállítási rétegbeli protokollok:
    - 6: TCP (Transmission Control Protocol)
    - 17: UDP (User Datagram Protocol)

\* kezdeti lista az RFC 790-ben

\* a mező értékeit **az IANA** (Internet Assigned Numbers Authority) felügyeli

# Alkalmazások, protokollok és portszámok – UDP felett

- UDP szolgáltatásait igénybevevő protokollok és a tipikus **szerveroldali** UDP portszám értéke\*:
  - 53 DNS (Domain Name System)**
    - névfeloldás
  - 67 BOOTP (Bootstrap Protocol) (Server)**  
**DHCP (Dynamic Host Configuration Protocol)**
    - szerver oldal
  - 68 BOOTP (Bootstrap Protocol) (Client)**  
**DHCP (Dynamic Host Configuration Protocol)**
    - kliens oldal
  - 123 NTP (Network Time Protocol)**
    - időszinkronizáció
  - 161 SNMP (Simple Network Management Protocol)**
    - hálózatmenedzsment
  - 520 RIP (Routing Information Protocol)**
    - routing

\* a portszámokat az IANA (Internet Assigned Numbers Authority) felügyeli

# Alkalmazások, protokollok és portszámok – TCP felett

- TCP szolgáltatásait igénybevevő protokollok és a tipikus szerveroldali TCP portszám értéke\*:

**20 és 21 FTP (File Transfer Protocol)**

- Ugyanahhoz a protokollhoz két port is!

**22 SSH (Secure Shell)**

**23 Telnet**

**25 SMTP (Simple Mail Transfer Protocol)**

**53 DNS (Domain Name System)**

- Ugyanaz a protokoll UDP-n és TCP-n is!

**80 HTTP (HyperText Transfer Protocol)**

**110 POP3 (Post Office Protocol version 3)**

**143 IMAP4 (Internet Message Protocol version 4)**

**443 HTTPS (HTTP Secure)**

**465 SMTPS (SMTP Secure)**

**993 IMAP4S (IMAP4 Secure)**

**995 POP3S (POP3 Secure)**

\* a portszámokat az IANA (Internet Assigned Numbers Authority) felügyeli

DNS – Domain Name System

DHCP – Dynamic Host Configuration Protocol

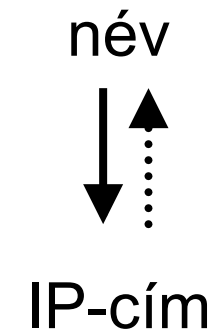
# INFRASTRUKTURÁLIS SZOLGÁLTATÁSOK

DNS – Domain Name System

# NÉVFELOLDÁSI RENDSZER

# A névfeloldás szerepe és követelményei

- Eltérő reprezentációk közötti megfeleltetés
  - Humán: könnyen megjegyezhető, beszédes nevek
  - Gépi: IP-címek→ névfeloldás
- Követelmények:
  - Jó skálázhatóság
  - Hibatűrés
  - Aktuális (friss) információk



# DNS (Domain Name System)

---

Két szerepe van:

- *elosztott adatbázis*: hierarchikus *névszerverekben* implementálva
- *alkalmazás rétegbeli protokoll*: hosztok, névszerverek kommunikálnak, hogy feloldják a *neveket*
  - alapvető Internet funkció, az alkalmazási rétegben implementálva
  - komplexitás a hálózat “szélén”

### DNS szolgáltatások

- Hosztnév IP címmé való fordítása
- Hoszt aliasing
  - Kanonikus és alias nevek
- Levelező szerver aliasing
- Terhelés elosztás
  - Replikált web szerverek: IP címek halmaza egy kanonikus névhez

### Miért nem centralizált a DNS?

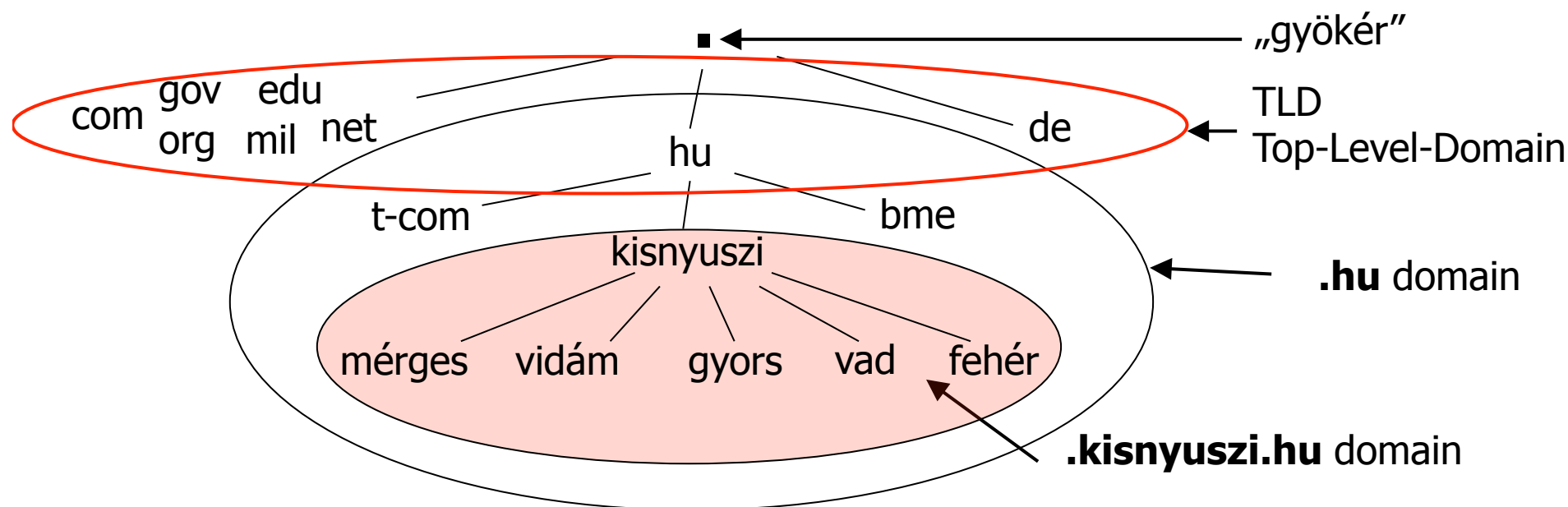
- single point of failure
- forgalom nagysága
- nagy távolságok: késleltetés
- adatbázis frissítése

Nem skálázódik!

# A DNS (Domain Name System) névtere

- Hierarchikus
- Állomások azonosítása: FQDN (Fully Qualified Domain Name)

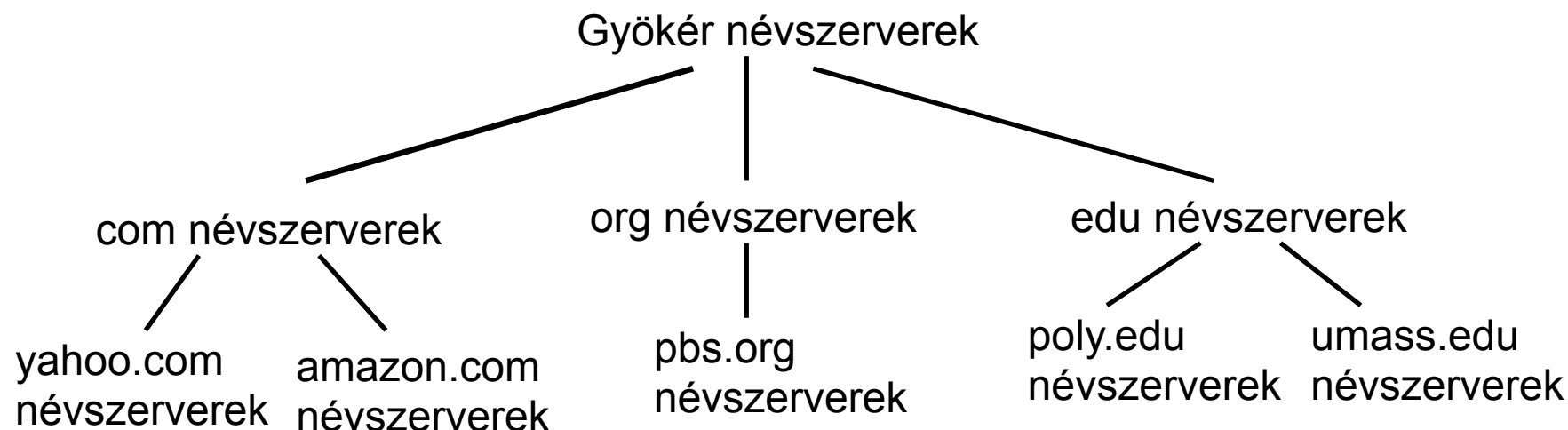
vad.kisnyuszi.hu.  
host neve tartomány neve  
(FQDN suffix)



# A DNS névtér „csúcsa”

- Root (gyökér):
  - „.” – ponttal jelölik
  - Elméletileg minden FQDN emiatt ponttal zárul
- Top level domain-eket a gyökér zóna tartalmazza:
  - Az IANA adminisztrálja  
(Internet Assigned Numbers Authority)
  - country code top-level domains (ccTLD): pl. .hu
  - generic top-level domains (gTLD):  
pl. .org, .edu, .net, .com, .gov, .mil
    - új kategóriák: sponsored gTLD, brand gTLD, geographic gTLD
  - infrastructure top-level domains: egy van, az .arpa

# Elosztott, hierarchikus adatbázis

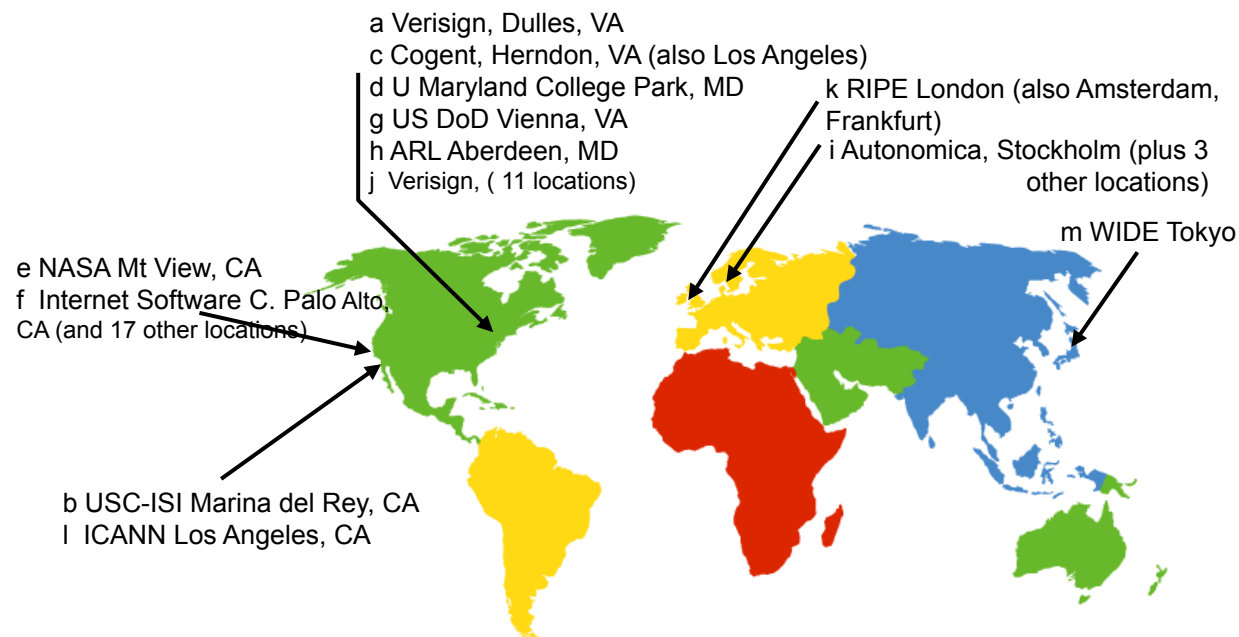


## Kliens az amazon.com IP címét szeretné megtudni:

- Kliens névfeloldási kérelmet küld a gyökér névszervernek, hogy megtudja a com névszerver címét
- Kliens névfeloldási kérelmet küld a com névszervernek, hogy megtudja az amazon.com névszerver címét
- Kliens névfeloldási kérelmet küld az amazon.com névszervernek, hogy megtudja az amazon.com címét

# Gyökér névszerverek

- **Gyökérzóna:** legmagasabb szintű DNS zóna
  - Valójában világszerte 13 DNS server klaszter szolgáltatja, több mint 600 szerverrel
    - Budapest: K és E
  - *a.root-servers.net*-től *m.root-servers.net*-ig terjednek
- **Anycast címzés** előnye!
- <http://root-servers.org>



# TLD és autoritatív névszerverek

---

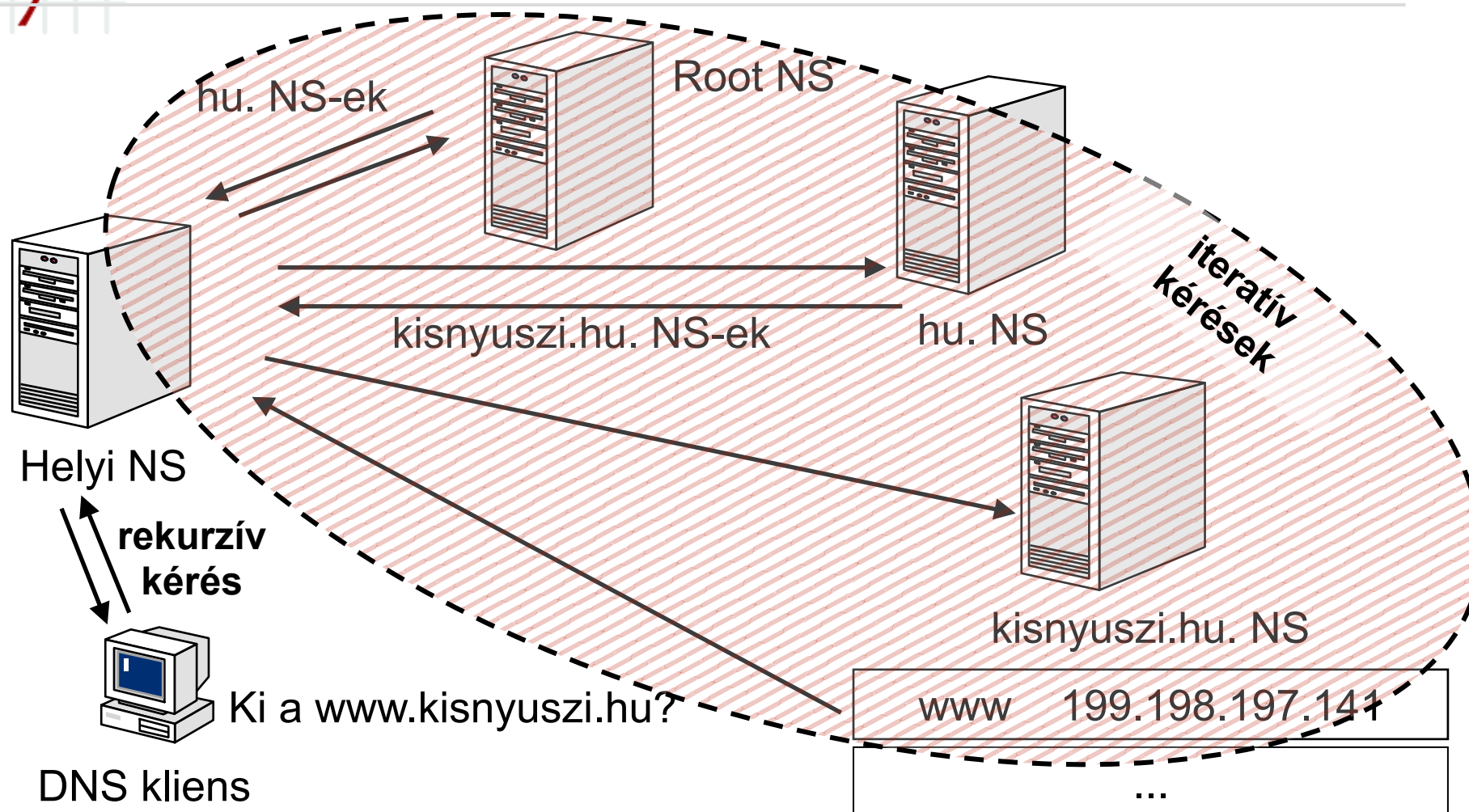
- **Top-level domain (TLD) névszerverek**
  - felelősek a com, org, net, edu, stb. és minden top-level country domain-ért (pl. uk, fr, ca, jp)
  
- **Autoritatív névszerverek**
  - az adott szervezet névszervere
  - autoritatív hosztnév-IP cím megfeleltetést nyújt a szervezet szervereire (pl. web és levelezőszerver).

# Helyi névszerverek

---

- Nem tartozik a hierarchiához
- Minden szolgáltatónak (lakossági internetszolgáltató, cég, egyetem) van
  - “default name server”
- Amikor a hoszt DNS névfeloldási kérelmet küld: a helyi névszerver kapja meg
  - Proxyként továbbítja a névfeloldási kérelmet a hierarchiának

# Névfeloldás menete



DNS kliens

- **Rekurzív kérés**: a konkrét (vagy a negatív) választ várja, a címzettnek a feladata a névfeloldás
- **Iteratív kérés**: a minél közelebbi felelős megtalálása, arra egy hivatkozást ad vissza: **referral válasz**

# A DNS zóna és névszerverek

- Zóna
  - Minden tartomány csúcsának vagy egészének adatait tároló adatbázis
  - A DNS nevével azonosítjuk (pl. kisnyuszi.hu.)
- DNS szerver
  - Egy vagy több zónát tárol, szolgál ki
- **Elsődleges/másodlagos** DNS szerver
  - Master/Slave server
  - Egy adott zónára vonatkozóan
    - Elsődleges: írható és olvasható
    - Másodlagos: csak olvasható
  - Minden esetben pontosan 1 elsődleges és legalább 1 másodlagos kell
    - → hibatűrés
    - → terhelésselosztás és skálázhatóság
  - Szinkronizálás monoton növekvő verziószám alapján
    - Egyre elfogadottabb a verziószám: YYYYMMDDnn



# A zóna elemei: rekordok

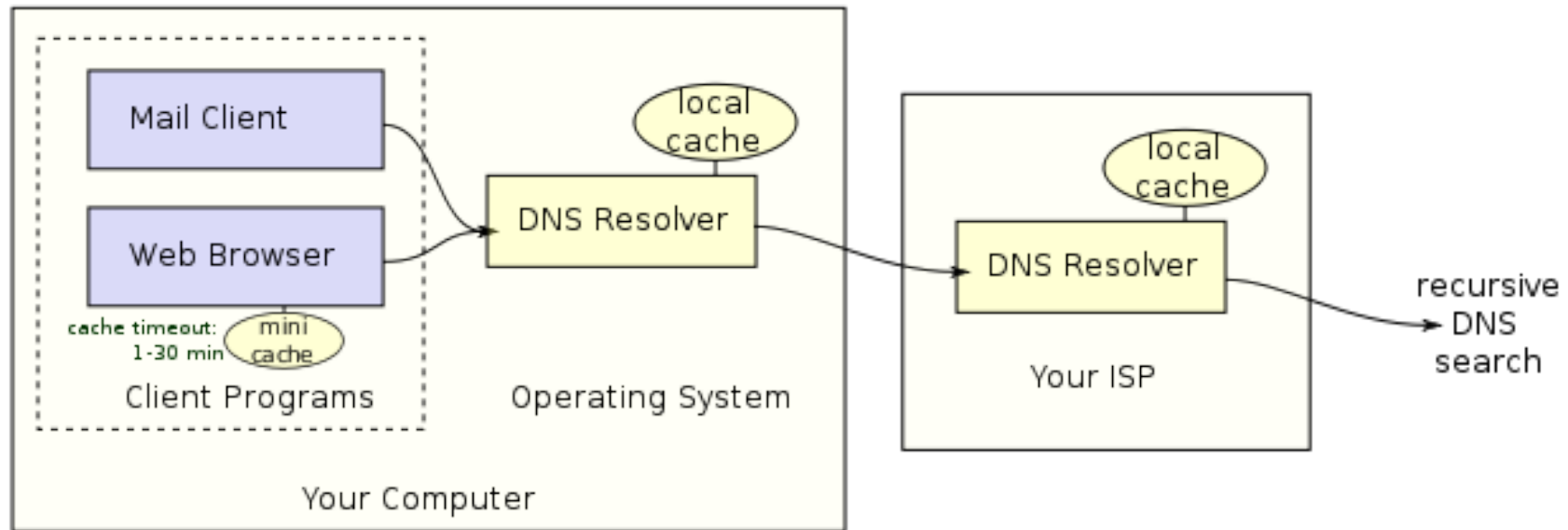
Erőforrásrekordok (**Resource Records**) formátuma: (name, value, type, ttl)

Gyakori típusai:

- **SOA (Start of Authority)**
  - adminisztratív adatok
    - az elsődleges DNS szerver neve
    - zóna verziószáma (ez alapján a szinkronizálás)
    - kapcsolattartó e-mail címe
- **A (Address)**
  - név – IP-cím  
(relay1.bar.foo.com, 145.37.93.126, A)
- **CNAME (Canonical Name)**
  - más néven „alias”
  - név – név összerendelés  
(foo.com, relay1.bar.foo.com, CNAME)
- **NS (Name Server)**
  - az adott zónát kiszolgáló DNS szerverek
  - legalább kettő kell  
(foo.com, dns.foo.com, NS)
- **MX (Mail Exchange)**
  - SMTP kiszolgálót azonosít
  - Több is megadható **preferenciával** (prioritással)  
(foo.com, mail.bar.foo.com, MX)

# Névfeloldás gyorsítótárral

- A DNS-kéréseket a helyi gépen az operációs rendszer oldja fel egységesen
- **DNS gyorsítótár (cache)** a helyi gépen és a DNS szerveren



- Minden rekordnak **TTL-je (Time To Live)** valódi másodpercben megadva → elévülés
- **Autoritatív válasz:** ha a rekordért felelős szerverek valamelyikétől származik a válaszol
- **Nem autoritatív:** ha gyorsítótárból származik

# DNS mint protokoll: kapcsolatok

- A DNS protokoll felhasználási területei:
  - Lekérdezés:
    - DNS kliens  $\Leftrightarrow$  DNS szerver
    - UDP 53  $\leftarrow$  rövid, gyors üzenetváltás  
(TCP 53)
  - Zónaletöltés
    - Elsődleges DNS szerver  $\Rightarrow$  másodlagos DNS szerver
    - TCP 53  $\leftarrow$  hosszabb, megbízhatóbb

# DNS mint protokoll: üzenetek

- **Kérés** elemei
  - Kért rekord típusa(i)
  - Feloldandó név vagy IP-cím
  - **Rekurzív kérés esetén** RD (Recursion Desired) bit beállítva
- **Válasz**
  - Pozitív válasz: egy vagy több elemű lista
    - Ebből „véletlenszerűen” (round-robin) választ
  - **Authoritatív válasz esetén** az AA (Authoritative Answer) bit beállítva
  - **Referral** válasz (egy vagy több elemű lista)
    - Kiegészítő hivatkozásokat tartalmaz, mely közelebb visz a feloldáshoz
      - pl.: illetékesebb NS; A rekord helyett azonos nevű CNAME rekord
  - Negatív válasz: nem található bejegyzés, nem oldható fel
    - Nincs válasz  $\neq$  negatív válasz

# Rekordok felvitele a DNS-be

- Példa: új cég “Network Utopia”
- Regisztrálni kell a networkutopia.com címet a **regisztrárnál** (pl. Network Solutions)
  - meg kell adni a registrarnak az (elsődleges és másodlagos) autoritatív névszerver nevét és IP címét
  - Registrar két RR-t helyez el a com TLD szervernél, pl. elsődlegesre:

```
(networkutopia.com, dns1.networkutopia.com, NS)  
(dns1.networkutopia.com, 212.212.212.1, A)
```

- A saját autoritatív névszervernél fel kell vinni az A és MX rekordokat

## Dynamic Host Configuration Protocol

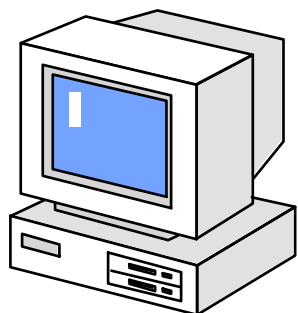
# DHCP

# Általában a DHCP-ről

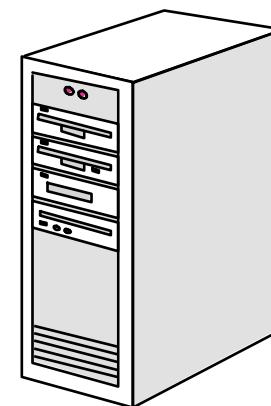
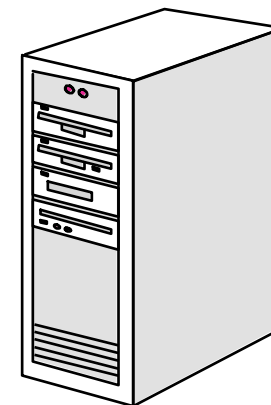
---

- Mi a DHCP?
  - IP-beállításokat oszthatunk ki vele dinamikusan
- Előnyei:
  - Kliensek egyszerű beállítása
  - Módosítások központilag
  - Mobilitás hálózatok között (eltérő beállítások)

Igénylő



DHCP szerver



1. DISCOVER

Bérlet kérése

2. OFFER

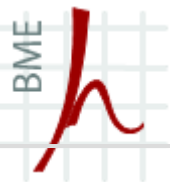
Bérleti ajánlat

3. REQUEST

Bérlet kiválasztása

4. ACK

Bérlet nyugtázása



# DISCOVER és OFFER

## Bérlet kérése és felajánlása

---

- Kérés:  
0.0.0.0-tól → 255.255.255.255-nek  
(még nincs IP-címe) (bárkinek)
- Ajánlat:  
255.255.255.255-nek ← 192.168.1.1-től  
(bárkinek; nem címezhető) (egy DHCP szerver címe)
- Ajánlat tartalma
  - IP-cím
  - Alhálózati maszk
  - Bérleti idő
  - DHCP szerver IP címe
  - **Igénylő MAC címével!**



# REQUEST és ACK

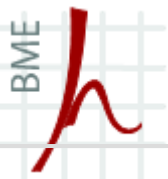
## Bérlet kiválasztása és nyugtázása

- Kiválasztás
  - az első ajánlatot (pl. ha több DHCP szerver)
  - Kiválasztási üzenet:
    - 0.0.0.0-tól 255.255.255.255-nek
    - Az üzenet tartalma
      - **kért IP-cím**
      - **DHCP szerver IP-címe**
- Nyugta
  - 192.168.1.1-től, 255.255.255.255-nek
  - Nyugtázó üzenet tartalma
    - **kiosztott IP-cím**
    - **alhálózati maszk**
    - **bérleti idő**

# Bérleti idő (TTL)

---

- Hosszú vagy rövid TTL?
  - Rövid mellett
    - Ha a kliens szabálytalanul távozik a hálózathoz
      - a bérletét nem adja vissza
    - Ha a kliens szabálytalanul újraindul
      - nem adja vissza a bérletét, és még újat is igényel
    - A beállításváltozások gyorsan életbe lépjenek
  - Hosszú mellett
    - Ne legyen nagy hálózati forgalom



# **Ami DHCP-vel beállítható...**

## **DHCP opciók, paraméterek**

---

- **0x01 Subnet Mask (alhálózati maszk)**
- **0x0F Domain Name (FQDN suffix)**
- **0x03 Router (alapértelmezett átjáró(k))**
- **0x06 DNS (DNS szerver(ek))**
- **0x0C Host Name (gép neve is kiosztható)**
- **0x1F Router Discovery**
- **0x21 Static Route**
- **0x2B Vendor Specific (gyártófüggő beállítások)**
- **0x2C WINS**
- **0x2E NBT**
- **0x2F Node Type**
- **0x32 Requested Address (igényelt IP-cím)**
- **0x33 Lease Time (TTL)**
- **0x36 DHCP Server (DHCP szerver IP-címe)**
- **0x37 Parameter Request List (igényelt paraméterek listája)**
- **0x3A Renewal Time (megújítási idő)**
- **0x3B Rebinding Time**
- **0x3C Client Class Information**
- **0x4D User Class Information**
- **0xF9 Static Route CIDR**

**Stb...**

- DHCP hibatűrés
  - Több DHCP használata egy hálózatban, de **diszjunkt IP-címtartományok osztása!**
- DHCP kiterjesztése több hálózati szegmensre
  - A routerek nem engedik át a DHCP üzeneteket
  - A routerekre ún. „**DHCP Relay Agent**”-et telepítve az továbbítja a DHCP forgalmat a DHCP szerverek és kliensek között

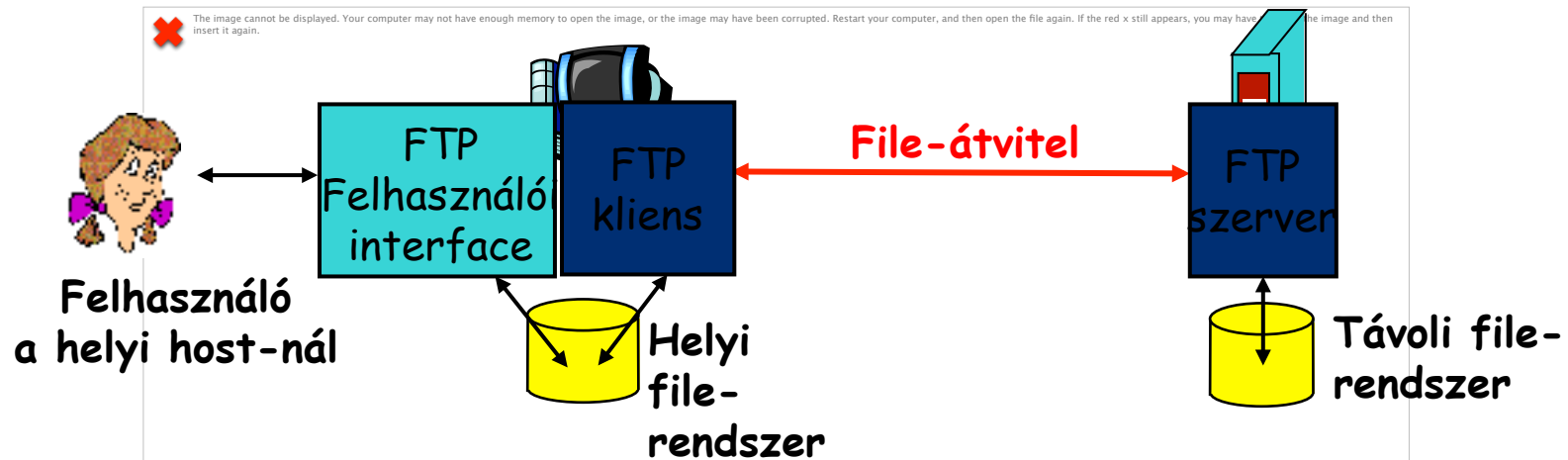
Telnet

FTP – File Transfer Protocol

# **SZÖVEG- ÉS FÁJLÁTVITEL**

- Egyik legrégebbi alkalmazás
- Távoli parancssor
  - Parancsok elküldése
  - Visszajelzések megjelenítése
- Még ma is alkalmazzák, főként hálózati eszközök egyszerű hálózati adminisztrációjára
- Nem biztonságos (jelszavak védelem nélkül)
  - SSH (Secure Shell) helyette

# FTP: a file-átviteli protokoll



- File-átvitel a távoli hostra/hostról
- Kliens/szerver modell
  - *kliens*: az átvitelt indító oldal (akár felmásol, akár leszed adatot a távoli hostra/hostról)
  - *server*: távoli host

# FTP – File Transfer Protocol

---

- Az egyik legelső fájlátvitelre tervezett protokoll
  - RFC 959
- 2 TCP kapcsolat épül fel: 21-es port vezérlés, 20-as port adat
- **out-of-band** vs. in-band protokoll pl. HTTP
  - külön TCP kapcsolat minden fájl átvitelére (nem perzisztens)
  - vezérlési TCP kapcsolat fennmarad (perzisztens)

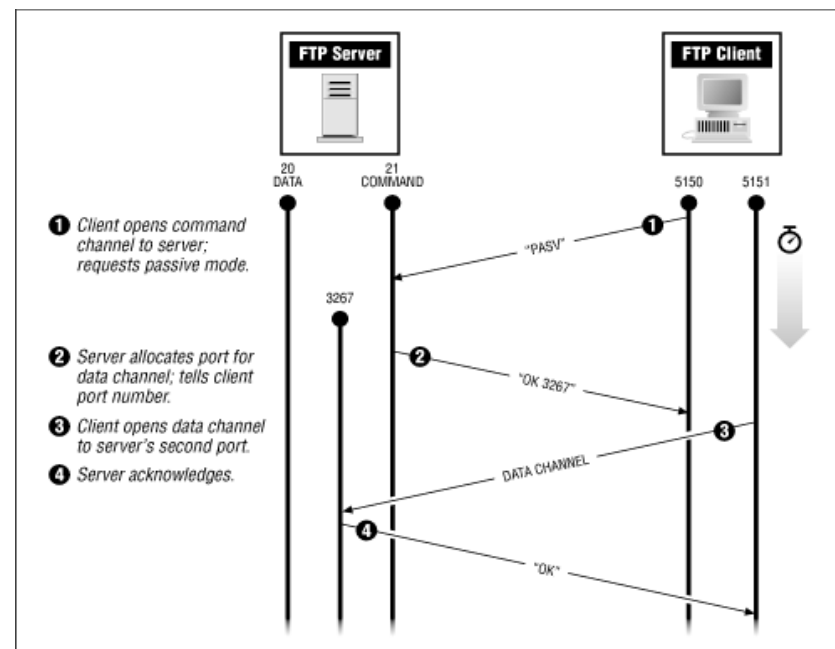
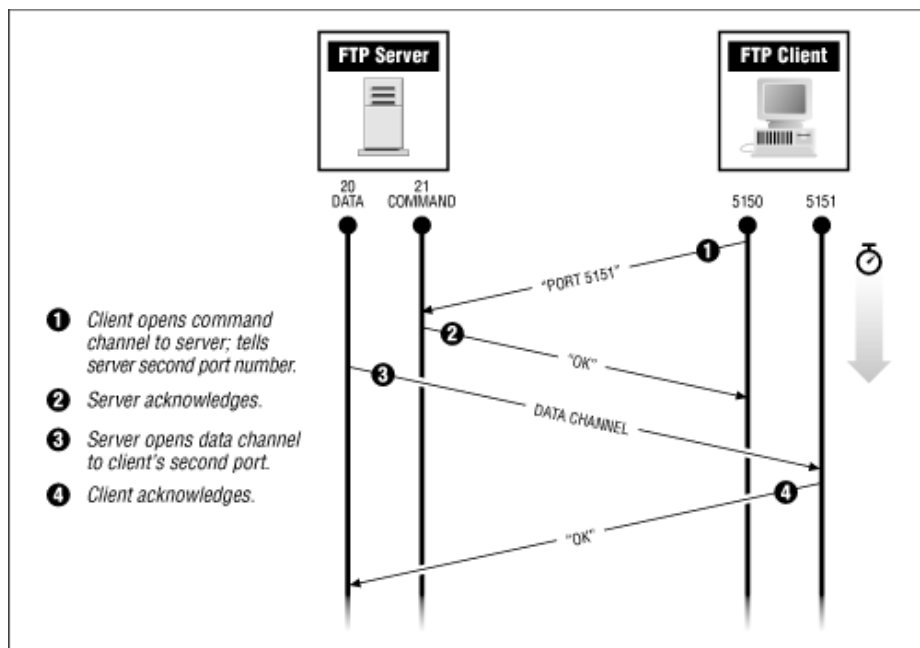
- Parancsok

- USER *felh. név* – felhasználói név elküldése a szervernek
- PASS *jelszó* – jelszó elküldése a szervernek
- LIST – kérés a szerverhez, küldje át a fájlok listáját
- RETR *fájl neve* – elkéri a fájlt a távoli gépről (get)
- STOR *fájl neve* – felmásolja a fájlt a távoli gépre (put)

- Válaszok:

- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

- Aktív/passzív adatátvitel:

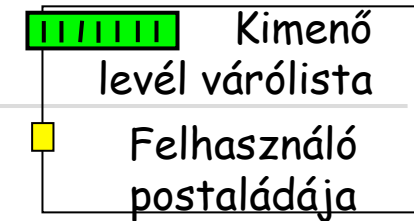


SMTP – Simple Mail Transfer Protocol

POP3 – Post Office Protocol v3

IMAP4 – Internet Message Protocol v4

## **LEVELEZŐ RENDSZEREK**



## Három fő alkotóelem:

### ■ User Agent

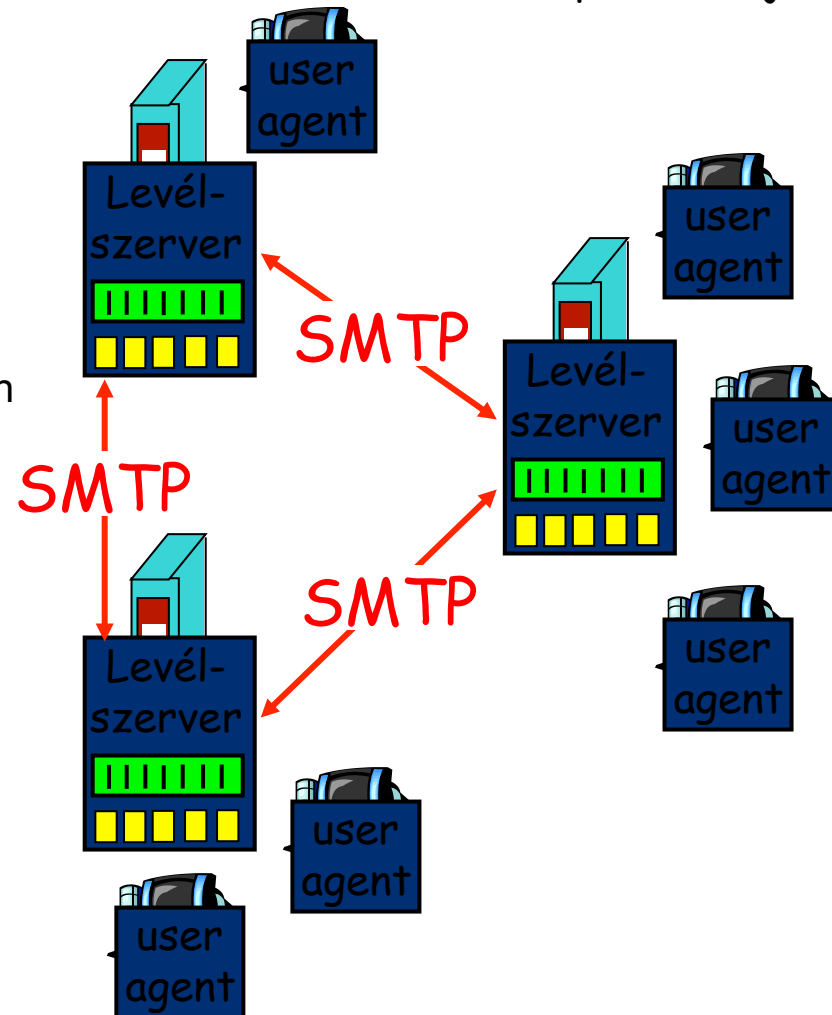
- Levelek írása, szerkesztése, olvasása
- GUI (pl. Mozilla Thunderbird, Outlook, Apple Mail), szöveges (pine), web alapú (Hotmail, Gmail)
- A kimenő és bejövő levelek a szerveren vannak tárolva

### ■ Levélszerverek

- A **postaláda** tartalmazza a felhasználó bejövő leveleit
- Kimenő **levelek várólistája** a szerveren

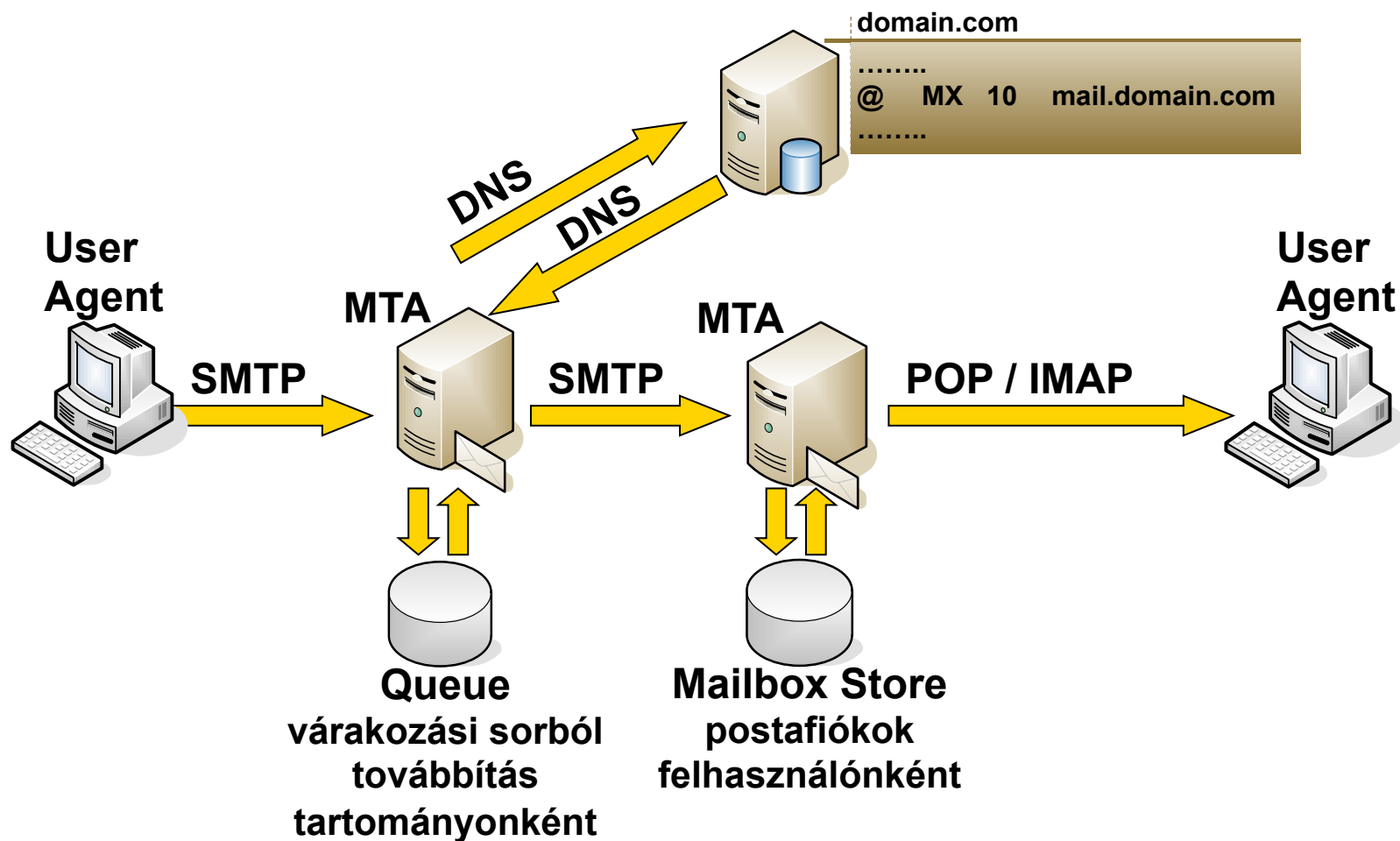
### ■ Levelező protokoll (SMTP) a levelező szerverek között az e-mailek továbbítására

- Kliens: küldő
- Szerver: fogadó levélszerver



- Komponensek
  - (Mail) User Agent ((M)UA) (levelező kliens)
  - Mail Transfer Agent (MTA) (SMTP szerver)
  
- Használt protokollok
  - SMTP: levél továbbításra
  - POP3: levelek lekérdezése
  - IMAP4: levelek lekérdezése
  
- Címzett meghatározása
  - DNS segítségével (MX rekord)

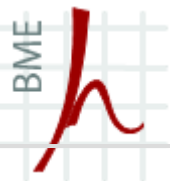
# Levelező rendszerek



# SMTP – Simple Mail Transfer Protocol

---

- RFC 821, 822 (első RFC még 1982-ből!)
  - első „killer application”
- Küldő szervertől fogadó serverig közvetlen TCP kapcsolat
  - nincs közbeeső levelezőszerver
  - perzisztens kapcsolat
- Parancs/válasz interakció
  - parancsok: ASCII szöveg
  - válaszok: állapotkód és szöveg
- TCP 25-ös port, megbízható
- SMTPS (SMTP Secure)
  - SMTP TLS csatornában
  - TCP 465



# Példa egy SMTP kommunikációra

S: 220 lappfold.fi  
C: HELO bme.hu  
S: 250 Hello bme.hu, pleased to meet you  
C: MAIL FROM: <jogyerek@bme.hu>  
S: 250 jogyerek@bme.hu... Sender ok  
C: RCPT TO: <mikulas@lappfold.fi>  
S: 250 mikulas@lappfold.fi ... Recipient ok  
C: DATA  
S: 354 Enter mail, end with "." on a line by itself  
C: Kedves Mikulás!  
C: Jó gyerek voltam. Hozzál sok csokit!  
C: Köszönöm,  
C: Jógyerek Jóska  
C: .  
S: 250 Message accepted for delivery  
C: QUIT  
S: 221 lappfold.fi closing connection

3 szakasz:

- SMTP“kézfogás”: bemutatkozás
- üzenet átvitele
- kapcsolat lezárása

## POP3

- Post Office Protocol version 3
- RFC 1939, 1957, 1725
- Kliens szerver felé TCP kapcsolatot nyit, 110-es port
- POP3S
  - POP3 TLS titkosítással
  - TCP 995
- “Letölt és töröl” mellett van “letölt és megtart” üzemmód is
- De nem viszi át az állapotokat a különböző session-ök között

## IMAP4

- Internet Message Protocol version 4
- RFC 2060, 1731, 1730
- TCP 143-as port
- IMAP4S
  - IMAP4 TLS titkosítással
  - TCP 993
- Intelligensebb a POP3-nál:
  - Postafiókok: mappák
  - Szerveroldali keresés támogatása
  - Állapotok átvitele IMAP session-ökön keresztül
  - Levél egyes részeit is le lehet tölteni (pl. csak fejléc)

# POP3 protokoll

“Letölt, töröl” példa

## autentikáció fázis

- kliens parancsok:
  - **user**: felh.név megadása
  - **pass**: jelszó
- szerver válaszok
  - **+OK**
  - **-ERR**

## tranzakciós fázis

- kliens parancsok:
  - **list**: üzenetek listázása (méret)
  - **retr**: üzenetek lekérése
  - **dele**: törlés
  - **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

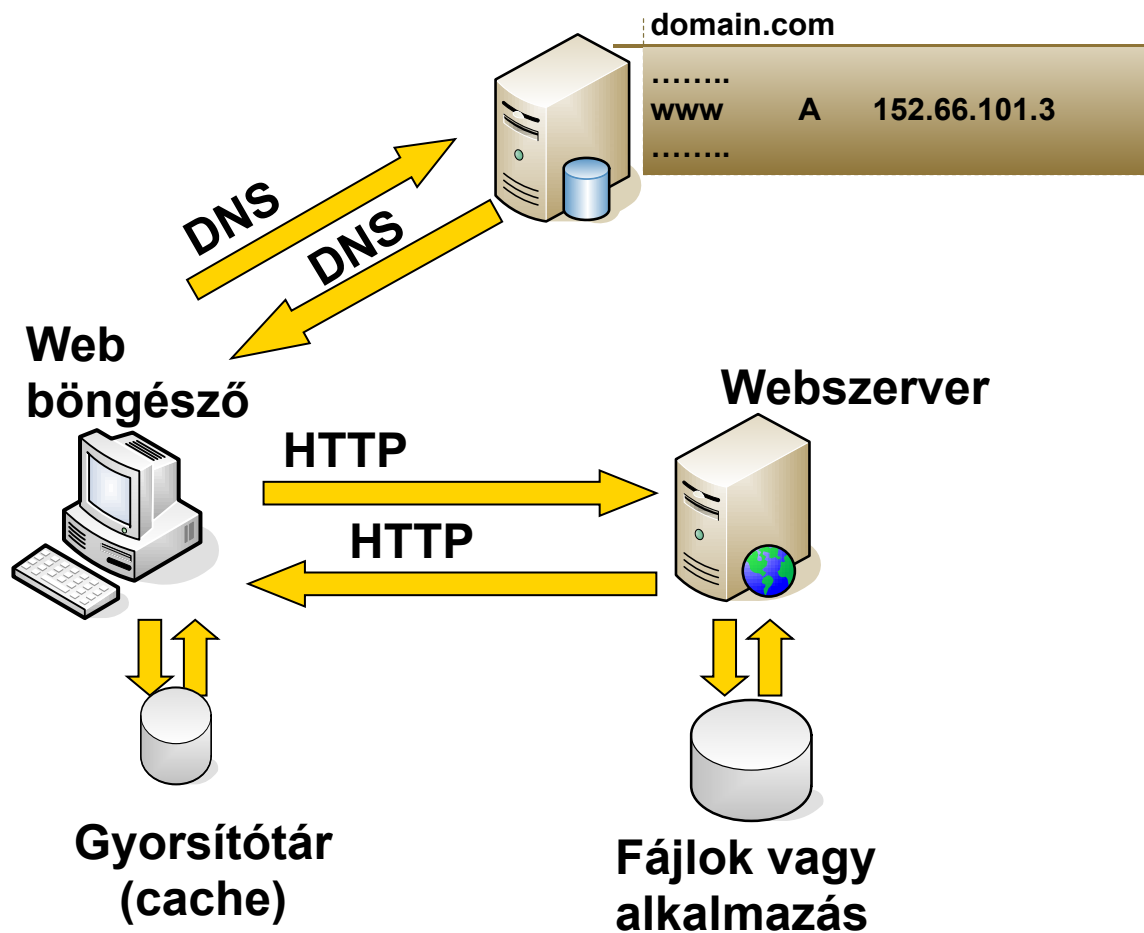
```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

- User agent: web böngésző, HTTP-val kommunikál a postafiókjával
- Levelek küldésénél HTTP a küldő böngészőjéből a levelező szerveréig
- Levelek lekérdezésénél is HTTP (POP3 és IMAP helyett)
- De két levelezőszerver között: SMTP!

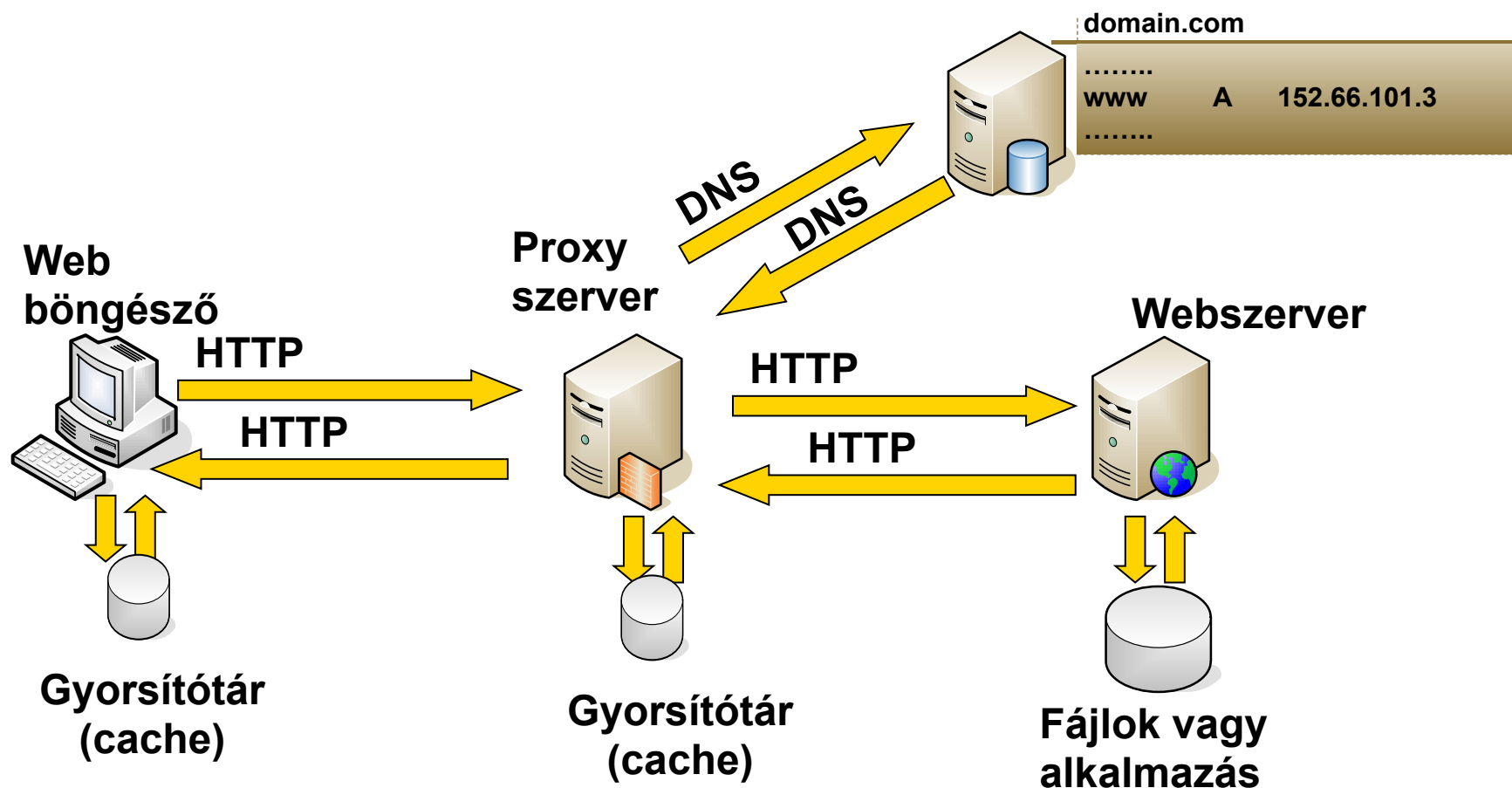
HTTP – HyperText Transfer Protocol

# WEBES RENDSZEREK

# Webes rendszerek



# Webes rendszerek proxyval



# HTTP – HyperText Transfer Protocol

- RFC 2068
- Web
  - Tim Berners-Lee, CERN
  - a második „killer application”
- Parancsorientált állapotkódokkal
- Speciális fejlécek
- Állapotmentes protokoll
- TCP 80
- Proxy
  - Kliens nevében jár el
  - Főként a hatékony gyorsítótárazás miatt



# Perzisztens HTTP

---

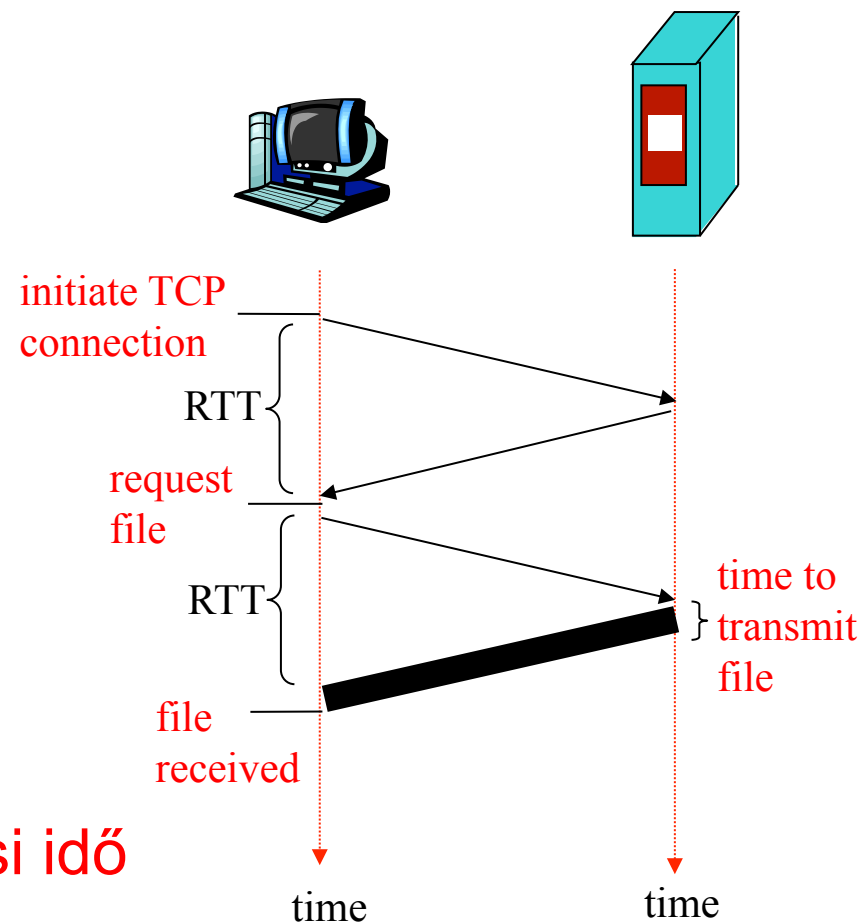
- *Nem-perzisztens:*

- minden kérés/válasz pár egy **külön** TCP kapcsolaton keresztül
- Legfeljebb egy objektumot visz át egy TCP kapcsolaton
- HTTP/1.0

- *Perzisztens:*

- **egy TCP kapcsolaton belül** több kérés/válasz pár
- Alapbeállítás HTTP1.1-nél

# Késleltetés nem perzisztens kapcsolatnál



Teljes késleltetés =  $2RTT + \text{adási idő}$

## Nonperzisztens HTTP:

- 2 RTT/objektum
- + TCP kapcsolat overhead
- Böngészők használhatnak **párhuzamos TCP kapcsolatokat** a különböző objektumok letöltésére

## Perzisztens HTTP

- Szerver **nyitva hagyja a kapcsolatot** a válasz elküldése után
- HTTP üzenetek sorozata ugyanazon kliens és szerver között **egy kapcsolaton belül**

## Perzisztens pipelining nélkül:

- Kliens csak akkor küld új kérést, ha előzőre **megjött a válasz**

## Perzisztens + pipelining:

- Alapbeállítás HTTP/1.1-nél
- Kliens elküldhet több kérést is egymás után **anélkül, hogy megvárná a választ**

# Példa egy HTTP kérésre

Kérés (request line)

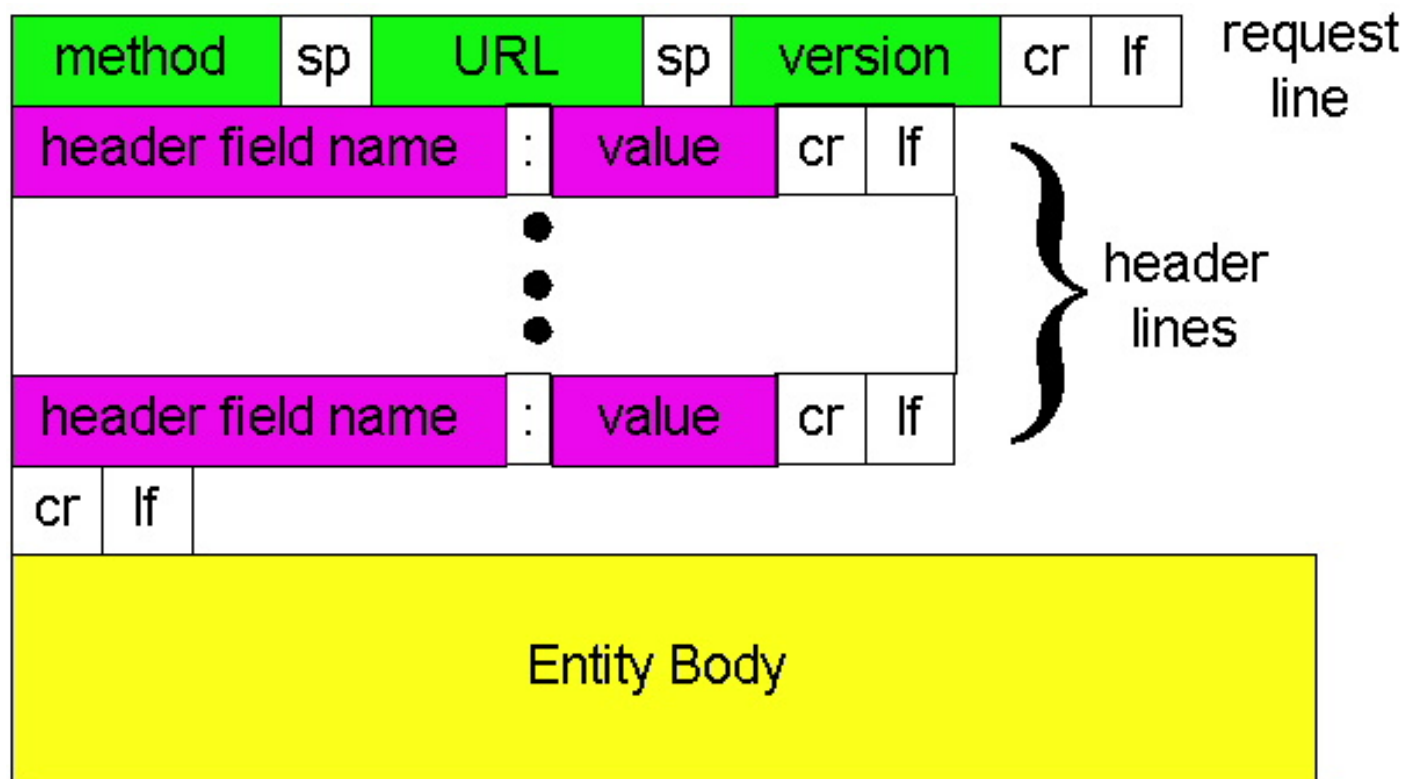
(GET, POST,  
HEAD parancs)

fejlécek  
(header lines)

üzenet végét  
jelző soremelés

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: hu
<CR><LF>
```

# Általános HTTP kérés üzenet formátum



**Entity Body: POST-nál van értelme!**

# Gyakori HTTP parancsok

---

- GET <URL> HTTP/1.1
  - adott URL tartalmának lekérése
- HEAD
  - mint a GET, de csak a metaadatokat adja vissza
- POST
  - a kliens ezzel tud adatokat küldeni a szervernek
- PUT
  - a POST-hoz hasonló, fájlfeltöltésre alkalmas
- DELETE
  - adott URL tartalmának törlése

# Példa egy HTTP válaszüzenetre

állapotkód

HTTP/1.1 200 OK

fejlécek

Connection close

Date: Thu, 04 May 2017 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 01 May 2017...

Content-Length: 6821

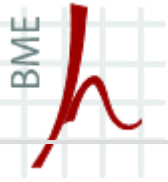
Content-Type: text/html

adat  
(pl. a kért  
HTML fájl)

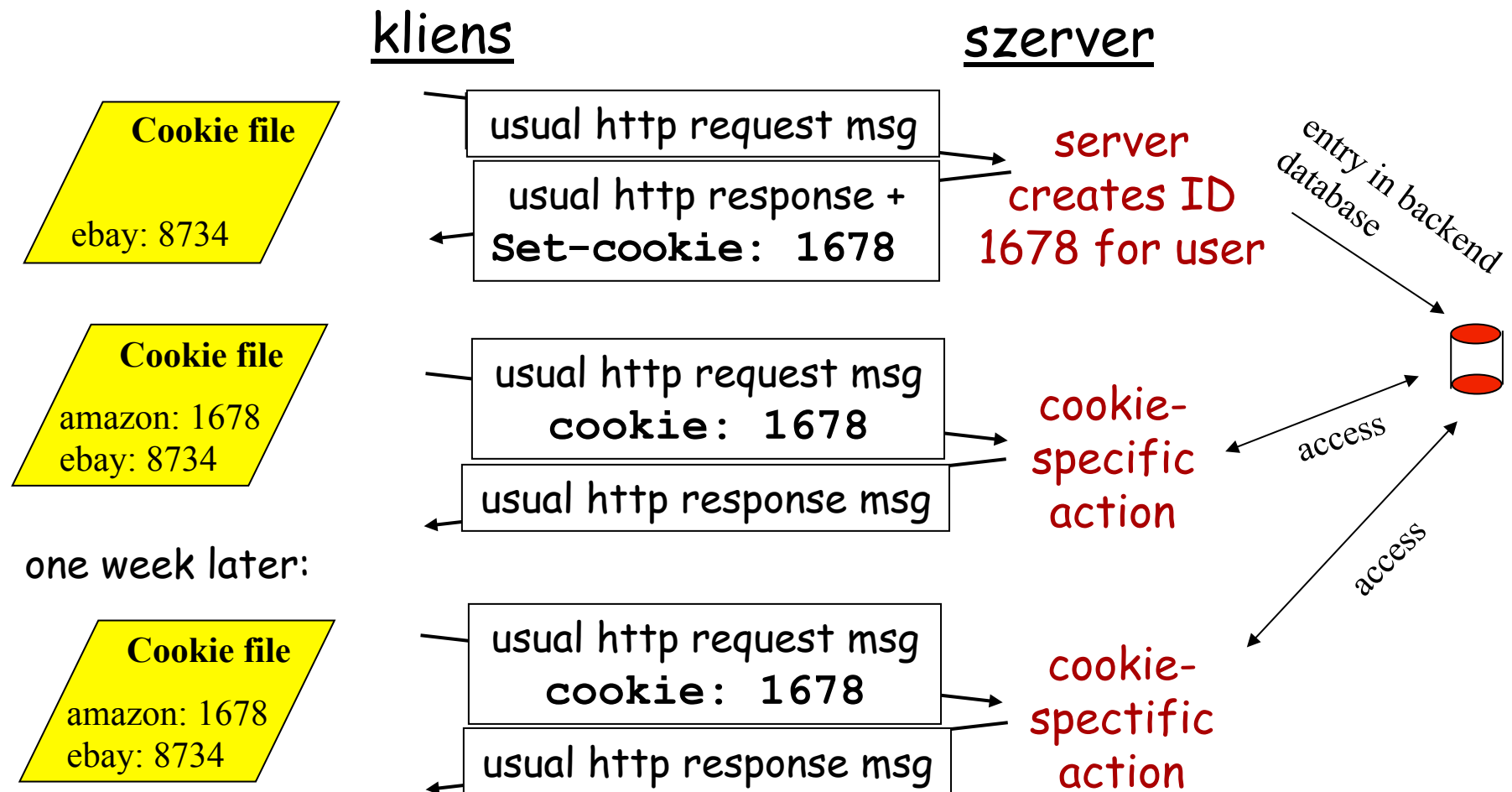
data data data data data ...

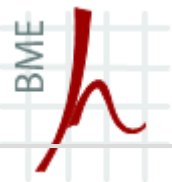
# Gyakori HTTP állapotkódok

| Kód | Jelentés              | Leírás                            |
|-----|-----------------------|-----------------------------------|
| 200 | OK                    |                                   |
| 201 | Created               | POST sikeres                      |
| 202 | Accepted              | Kérés elfogadva                   |
| 204 | No content            | Nincs semmi a kliensnek           |
| 400 | Bad request           | Hibás kérés                       |
| 401 | Unauthorized          | Hitelesítés szükséges             |
| 403 | Forbidden             | Hozzáférés megtagadva             |
| 404 | Not found             | Nem található                     |
| 500 | Internal Server Error | Belső szerver hiba                |
| 503 | Service Unavailable   | Pillanatnyilag nem szolgálható ki |



# Cookie-k használata





# HTTP jövője

---

- Következő verzió: HTTP/2, IETF fejlesztésben
- A Google SPDY protokolljának lényeges elemeit használja
  - De facto szabvány?
- Csökkenti a web oldalak betöltési idejét:
  - multiplexált stream-ek egy TCP kapcsolaton belül
  - kérések priorizálása
  - kérés és válasz fejrészek tömörítése
    - a statikusok nincsenek újraküldve
  - szerver push funkció

- Alkalmazások hálózati kapcsolata
  - Portok és socketek használata
- Infrastrukturális szolgáltatások
  - DNS – Névfeloldási szolgáltatás
  - DHCP
- Szöveg- és fájlátvitel
  - Telnet, FTP
- Levelezési rendszerek
  - SMTP, POP3, IMAP4
- Webes rendszerek
  - HTTP

# Kérdések?

## **KÖSZÖNÖM A FIGYELMET!**

Dr. Simon Vilmos  
BME Hálózati Rendszerek és Szolgáltatások Tanszék  
[svilmos@hit.bme.hu](mailto:svilmos@hit.bme.hu)