

## Számítógépes Hálózatok 2013

### 11. Szállítói réteg – TCP hatékonyság, fairness Felhasználói réteg – DNS, email, http, P2P

## Torlódás elkerülési elv: AIMD

- A TCP a „fast recovery” mechanizmussal lényegében a következőképp viselkedik:

$x$ : csomagok száma per RTT

- Kapcsolatfelépítés:

$$x \leftarrow 1$$

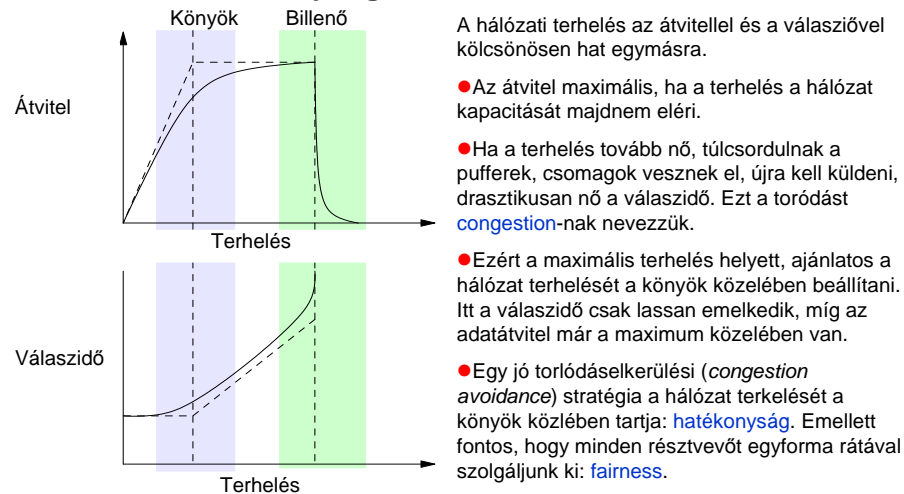
- Csomagvesztésnél, MD: multiplicative decreasing

$$x \leftarrow x/2$$

- Nyugtázott szegmenseknél, AD: additive increasing

$$x \leftarrow x+1$$

## Additive Increase Multiplicative Decrease (AIMD): Fairness és Hatékonyság



## AIMD Fairness és Hatékonyság – Egy egyszerű modell

- $n$  résztvevő, forduló-modell
- résztvevő  $i$  adatrátája a  $t$ -edik fordulóban  $x_i(t)$
- Kezdeti adatráták:  $x_1(0), \dots, x_n(0)$
- A visszacsatolás (feedback) forduló  $t$  után:  $y(t) = 0$ , ha  $\sum_{i=1}^n x_i(t) \leq K$

$$y(t) = 1, \text{ ha } \sum_{i=1}^n x_i(t) > K$$

- Minden résztvevő aktualizálja az adatrátáját a  $t+1$ -edik fordulóban:

$$x_i(t+1) = f(x_i(t), y(t))$$

- Increase-stratégia  $f_0(x) = f(x, 0)$
- Decrease-stratégia  $f_1(x) = f(x, 1)$

- Tekintsük a következő lineáris függvényeket:

$$f_0(x) = a_I + b_I x, \quad f_1(x) = a_D + b_D x$$

## AIMD Fairness és Hatékonyság– A Modell

- A következő lineáris függvényeket vizsgáljuk:

$$f_0(x) = a_I + b_I x, \quad f_1(x) = a_D + b_D x$$

- Érdekes speciális esetek:

- MIMD: Multiplicative Increase/Multiplicative Decrease

$$f_0(x) = b_I x, \quad f_1(x) = b_D x, \quad \text{ahol } b_I > 1, b_D < 1.$$

- AIAD: Additive Increase/Additive Decrease

$$f_0(x) = a_I + x, \quad f_1(x) = a_D + x, \quad \text{ahol } a_I > 0, a_D < 0.$$

- AIMD: Additive Increase/Multiplicative Decrease

$$f_0(x) = a_I + x, \quad f_1(x) = b_D x, \quad \text{ahol } a_I > 0, b_D < 1.$$

## AIMD Fairness és Hatékonyság

- Hatékonyság

- Terhelés:  $X(t) := \sum_{i=1}^n x_i(t)$

- Mérték:  $|X(t) - K|$

- Fairness:  $x=(x_1, \dots, x_n)$  esetén:

$$F(x) = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2}.$$

- $1/n \leq F(x) \leq 1$
- $F(x) = 1 \leftrightarrow$  abszolút Fairness
- skálázástól független
- Folytonos, differenciálható
- Ha  $n$  közül  $k$  fair, a többi 0, akkor  $F(x) = k/n$

## Konvergencia

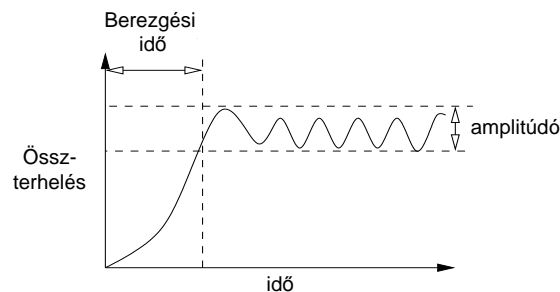
- Konvergencia nem lehetséges
- Legjobb esetben oszcilláció az optimális érték körül

- Az oszcilláció amplitúdója  $A$

$$A = \inf_{t_0 \geq 0} \sup_{t \geq t_0} |X(t) - K|.$$

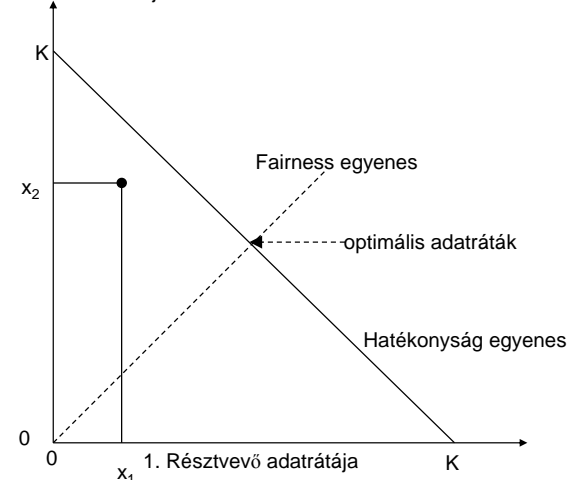
- Berezgési idő  $T$

$$T = \min\{t_0 \mid \forall t \geq t_0 : |X(t) - K| \leq A\}.$$

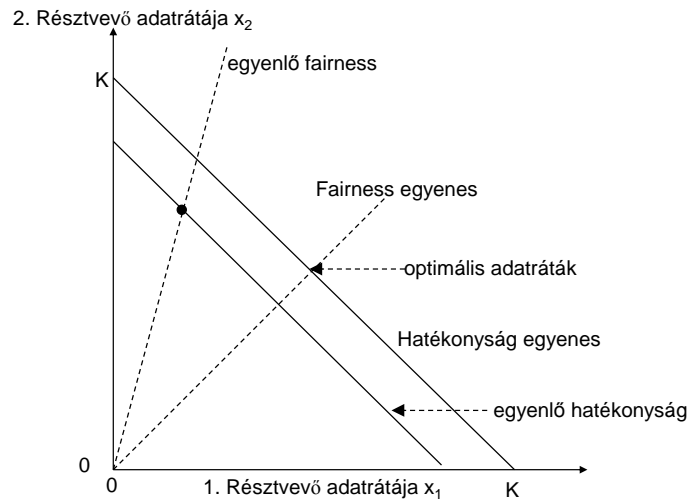


## Vektor Ábrázolás (I)

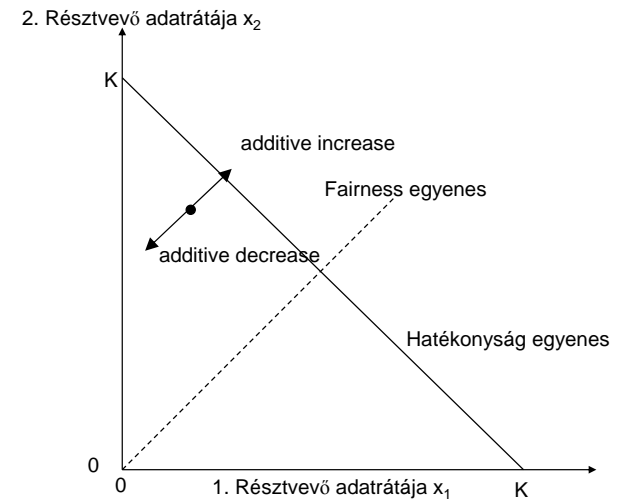
2. Résztevő adatrátája



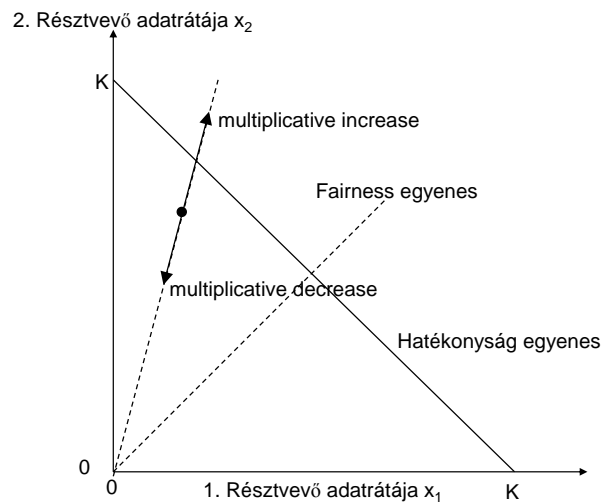
## Vektor Ábrázolás (I)



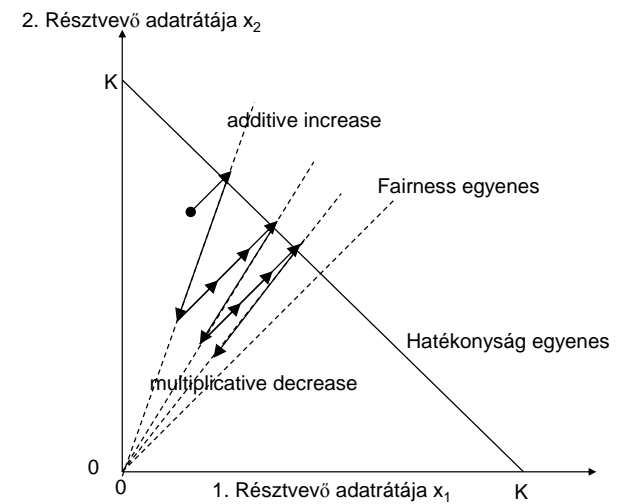
## Vektor Ábrázolás (I)



## Vektor Ábrázolás (I)



## Vektor Ábrázolás (I)



## TCP összefoglalás

- TCP egy megbízható byte-folyamot hoz létre
  - Hibafelügyelet "Go-Back-N" által
- Congestion control
  - Ablak alapú
  - AIMD, Slow start, *Congestion Threshold*
  - Folyamfelügyelet *Window* által
  - Kapcsolatfelépítés
  - Nagle algoritmus

## TCP fairness & TCP friendliness

- TCP
  - Dinamikusan reagál a rendelkezésre álló sávszélességre
  - A sávszélesség fair felosztása
    - Ideális esetben:  $n$  TCP-kapcsolat mindegyike  $1/n$  részt kap
- TCP más protollokkal
  - Reakció más szállítói protollok terhelésétől függ
    - pl. UDP-ben nincs congestion control
  - Más protollok mindenkor felhasználhatók
  - UDP és más protokoll el tudja nyomni a TCP kapcsolatokat
- Véggövetkeztetés
  - A szállítói protolloknak TCP-kompatibilisnek kell lenni (TCP friendly)

## Felhasználói réteg

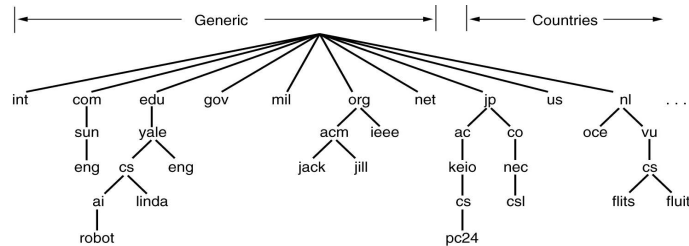
- Domain Name System
- Példák a felhasználói rétegre:
  - E-Mail
  - WWW
  - Content Delivery Networks
  - Peer-to-Peer-Networks
- A forgalom az Interneten

## Domain Name System (DNS)

- Az emberek számára 4 byte IPv4 cím nehezen kezelhető:
  - 209.85.135.99 google.com-hoz
  - 157.181.151.154 az ELTE-hez
  - Mit jelent?
    - 207.46.19.30
    - 157.181.35.45
- Jobb: Természetes szavak az IP-címekhez
  - Pl. www.google.com
  - vagy www.elte.hu
- A Domain Name System (DNS)
  - lefordítja ezeket a címeket IP-címekre (és fordítva)
  - elosztott adatbázis

## DNS – Felépítés

- DNS neveket képez le IP-címekre
  - Pontosabban: neveket erőforrás-bejegyzésekre
- A nevek hierarchikusan struktúráltak egy névtérben
  - Max. 63 jel komponensenként, összesen max. 255 jel
  - Minden domain-en belül, a domain tulajdonosa ügyeli fel a névteret a domain alatt



## DNS Resource Record

- **Erőforrás bejegyzés** (resource record RR): a domain-ekről, egyes host-okról, stb... adnak információt

- RR formátum: (name, ttl, class, type, value)

- name: pl. domain név vagy host név
- ttl (time to live): érvényesség (másodpercben)
- class: Internet esetén mindig "IN"
- type: lásd a táblázatot
- value: pl. IP-cím

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

- RR Példa:  
pandora.inf.elte.hu. 43200 IN A 157.181.161.52

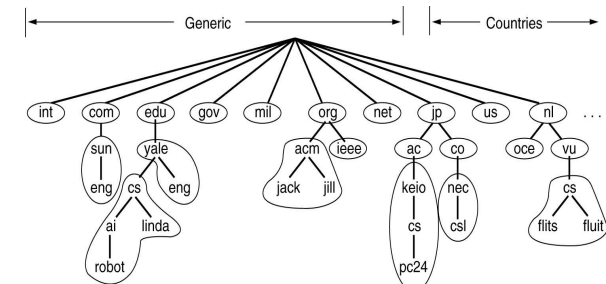
## DNS Resource Records -- Példák

Példák RR típusokra

- Type=A
  - name: egy végrendszer (host) neve
  - value: egy IP-cím
- Type=NS
  - name: egy domain (pl elte.hu)
  - value: a domain authoritative name server-jének az IP-címe
- Type=MX
  - value: a name-hez tartozó mail server neve
- Type=CNAME
  - name: egy alias név egy kanonikus névhez
  - value: a kanonikus név
- Type = SOA (start of authority)
  - name: a domain neve
  - value: szerverek neve, melyek a zónához tartozó mérvadó információkat rendelkezésre bocsátják, paraméterek a zónához
    - a zóna sorszám, a zóna sorszáma,
    - frissítési intervallum a másodlagos szervernek,...

## DNS Name Server

- A névtér **zónákra** van osztva
- Minden zónához tartozik egy **Authoritativ Server** a mérvadó információval
  - Egy **Primary Name Server**
  - Továbbá egy vagy több **Secondary Name Server** a megbízhatóság miatt
- Minden Name Server ismeri
  - a saját zónáját
  - a gyermek-zónák Name-Server-jeit

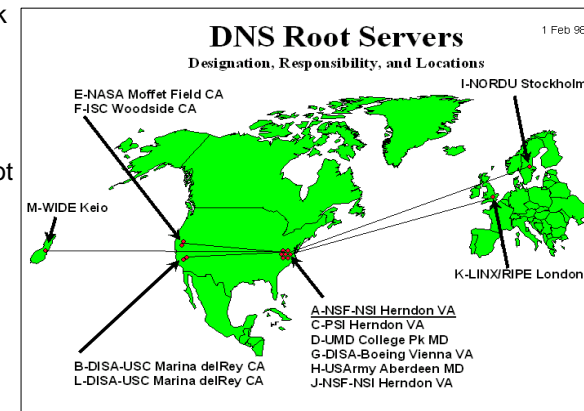


## Servers/Resolvers

- Minden végrendszernek van egy „feloldója” (resolver)
  - Tipikusan egy könyvtár, amit felhasználásokhoz kapcsolhatunk
  - Lokális name-server-ek kézzel konfigurálva (pl. /etc/resolv.conf)
- Name servers
  - Tipikusan egy zónáért felelősek
  - Lokális szerverek
    - A lokális végrendszereknek végznek lekérdezéseket távoli végrendszer nevekről
    - Megválaszolják a lekérdezéseket a lokális zónáról

## DNS: Root Name Servers

- A “root” zónáért felelősek
- Jelenleg 13 root name server világszerte
  - A-M „számozva”
- Lokális szerverek kapcsolatba lépnek a root szerverrel, ha ők nem tudják megválaszolni a lekérdezést
  - Jól ismert root szerverekkel konfiguráltak



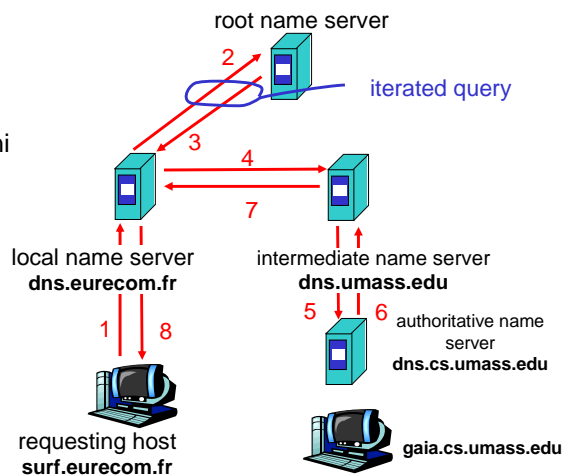
## DNS lekérdezések

### Iteratív lekérdezés:

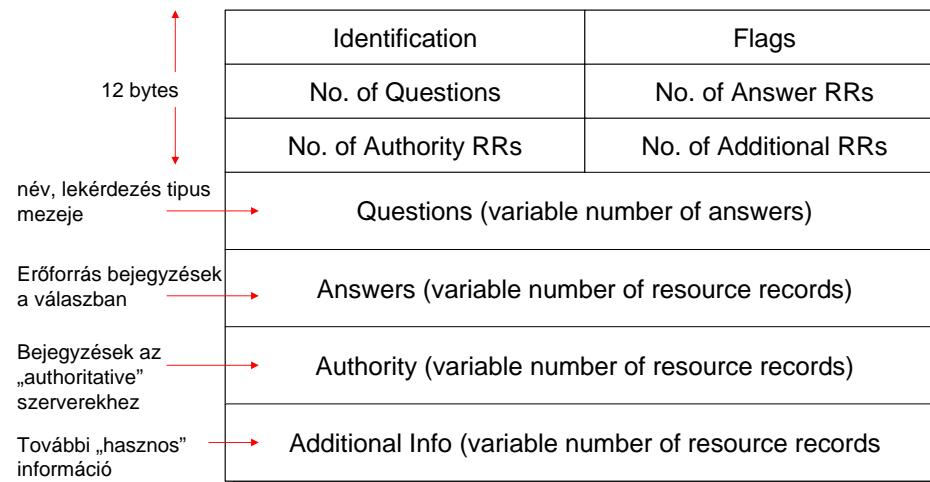
- A megkérdezett szerver annyi információt ad a válaszban, amit ő maga tud
- Pl. annak a szervernek a nevét, akit meg kell kérdezni

### Rekurzív lekérdezés:

- A megkérdezett szerver rekurzív „kideríti” a hiányzó információt
- A lokális szerverek tipikusan rekurzív lekérdezési módban dolgoznak
- Root vagy távoli szerverek iteratívban



## DNS üzenet formátum



## Tipikus feloldási folyamat

- A `www.inf.elte.hu` név feloldásának lépései
  - A felhasználás hívja a `gethostbyname()` függvényt
  - A végrendszer lekérdezi a lokális name server-t ( $S_1$ )
  - $S_1$  lekérdezi a root server-t ( $S_2$ ) a `www.inf.elte.hu` névvel
  - $S_2$  válaszol a `elte.hu`-hoz ( $S_3$ ) tartozó NS bejegyzéssel
  - Honnan tudjuk meg az A bejegyzést  $S_3$ -hoz
    - Erre való az „additional information section”
  - $S_1$  lekérdezi  $S_3$ -t a `www.inf.elte.hu` névvel
  - $S_3$  válaszol a `www.inf.elte.hu`-hoz tartozó A bejegyzéssel
- Több A bejegyzés is érkezhethet a válaszban → mit jelent ez?

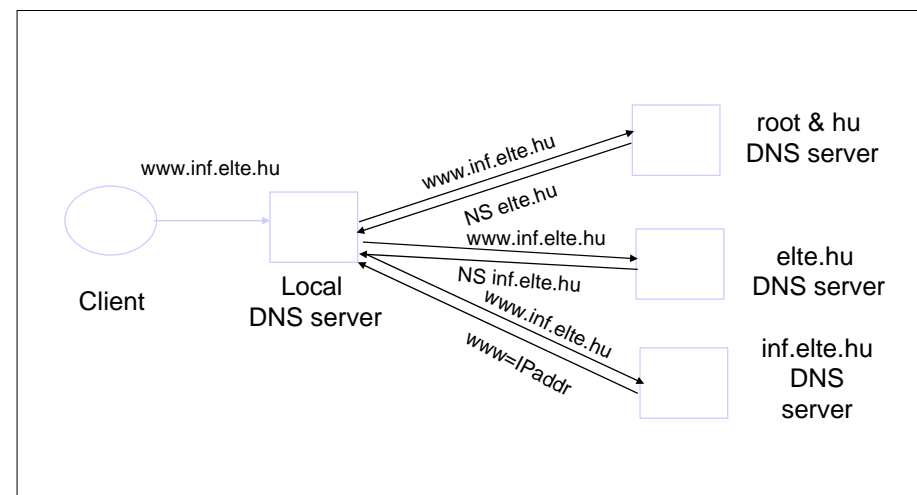
## Caching

- DNS válaszok tárolódnak az érintett szervereken (caching)
  - Gyors válasz ismételt lekérdezés esetén
  - Más lekérdezések bizonyos részeket újra felhasználhatnak a válaszból
    - PI. NS bejegyzéseket a domain-ekhez
- DNS negatív lekérdezések tárolódnak a cache-ben
  - Ne kelljen megismételni a kudarcot
  - PI. elgépelés
- A cache-ben tárolt adatok érvényessége egy idő után lejár
  - Az érvényesség idejét (TTL) az adat tulajdonosa határozza meg
  - Minden bejegyzés tartalmaz TTL-t

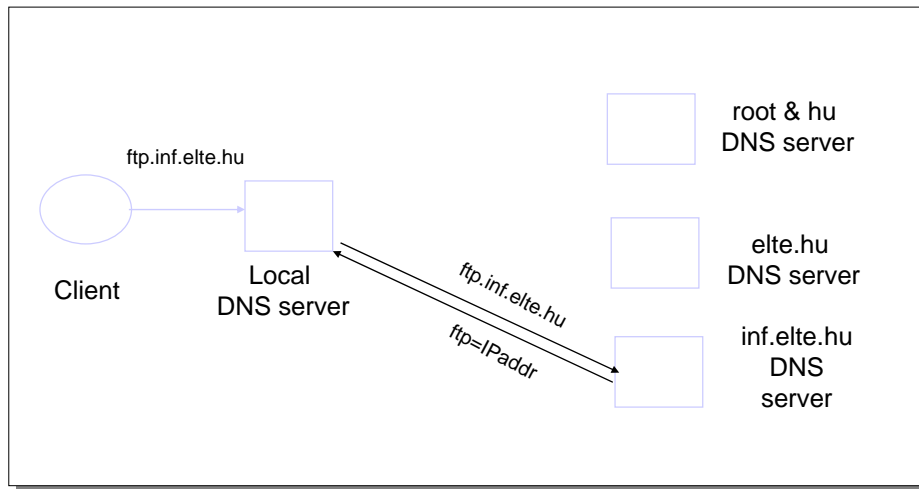
## Prefetching

- Name server minden válaszhoz adhat további adatokat
- Tipikusan prefetching-hez használják
  - CNAME/MX/NS tipikusan más végrendszer nevére mutat
  - Válaszok tartalmazzák a végrendszerek címét, amelyekre mutatnak az „additional section” részben

## DNS lekérdezés példa



## Példa egy későbbi lekérdezésre

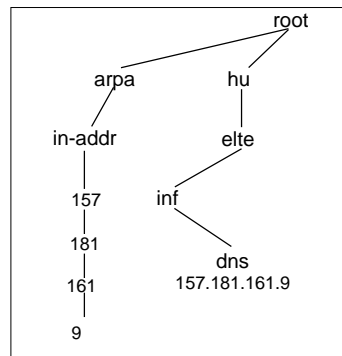


## Megbízhatóság, rendelkezésre állás

- DNS szerverek replikáltak
  - A name service működik, ha egy replika működik
  - A lekérdezések kiegyensúlyozhatók a replikák között (load balancing)
- UDP-t használ a lekérdezéshez
  - Megbízhatónak kell lenni → Miért nem TCP?
  - Timeout esetén alternatív szervert próbál
  - „Exponential backoff”, ha visszatér ugyanahhoz a szerverhez
  - Ugyanaz az azonosító minden lekérdezéshez
    - Mindegy melyik szerver válaszol

## Reverse Name Lookup

- Melyik számítógéphez tartozik az 157.181.161.9 IP-cím?
  - Lekérdezés: 9.161.181.157.in-addr.arpa
  - Miért van megfordítva a cím?
  - dns.inf.elte.hu



## Dinamikus DNS

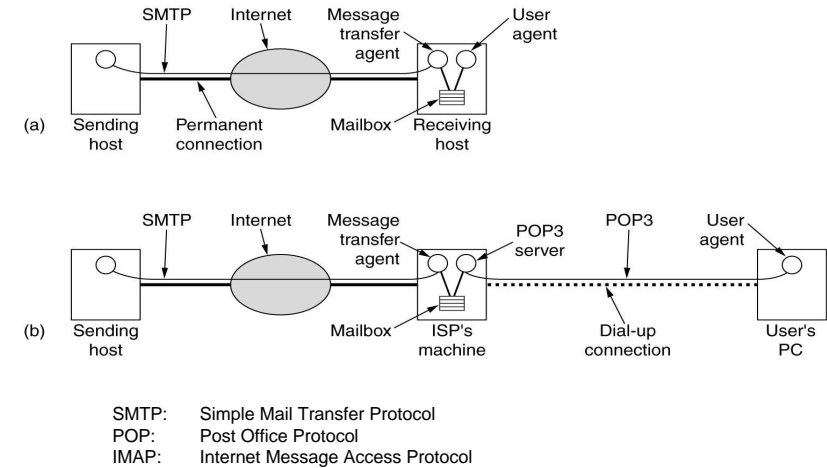
- Probléma
  - Időlegesen hozzárendelt IP-címek
  - Pl. DHCP által
- Dinamikus DNS
  - Amint egy csomópont egy új IP-címet kap, regisztrálja azt azon a DNS-szerveren, amely őérte felelős
  - Rövid TTL bejegyzések biztosítják azt, hogy a bejegyzések gyorsan aktualizálódjanak
    - egyébként a lekérdezések rossz számítógépre irányítódnának
- Felhasználás
  - Egy privát domain regisztrálása
  - lásd [www.dyndns.com](http://www.dyndns.com)



## Email (RFC 821/822)

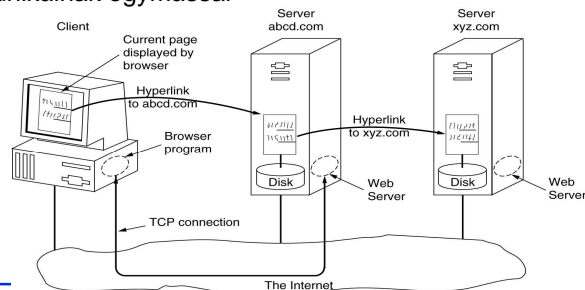
- **Komponensei:**
  - user agents (UA)
  - message transfer agents (MTA)
- **Szolgáltatások**
  - kompozíció, küldés, értesítés, megjelenítés, rendelkezés (disposition)
- **További szolgáltatások**
  - továbbküldés, auto-válasz, szabadság-funkciók, levelező listák, ...
- **Struktúra:**
  - Boríték – a szállításhoz szükséges információ, a MTA használja
  - Tartalom
    - Fejléc – kontroll információ a UA-nek
    - Törzs – a valódi tartalom

## E-Mail: SMTP és POP



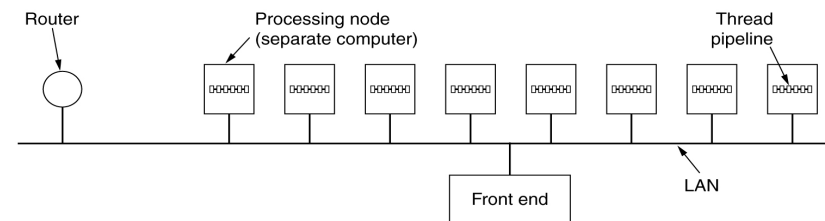
## World Wide Web

- **Client-Server-Architektúra**
  - **Web-Server** web-oldalakat bocsát rendelkezésre
  - Formátum: **Hypertext Markup Language (HTML)**
  - **Web-Browser** oldalakat kérdez le a web-server-től
  - Server és browser **Hypertext Transfer Protocol (HTTP)** által kommunikálnak egymással



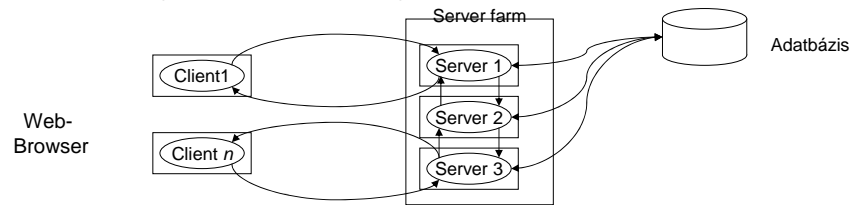
## Szerver-Farm

- A szerver oldal teljesítményének növeléséhez
  - több web-server dolgozik
- **Front end**
  - Fogadja a lekérdezéseket
  - Továbbítja a lekérdezéseket egy különálló csomóponthoz további feldolgozásra



## Web-Server-ek és adatbázisok

- Web-Server-ek nem csak statikus web-oldalakat bocsátanak rendelkezésre
  - Web-oldalakat automatikusan is létre lehet hozni
  - Ehhez egy adatbázisból kérdeznék le adatokat
  - Ez az adatbázis nem szükségszerűen statikus, interakció által megváltoztatható lehet
- Probléma:
  - **Konzisztencia**
- Megoldás
  - Web-szolgáltatás és adatbázis egy 3-fokú architektúrája



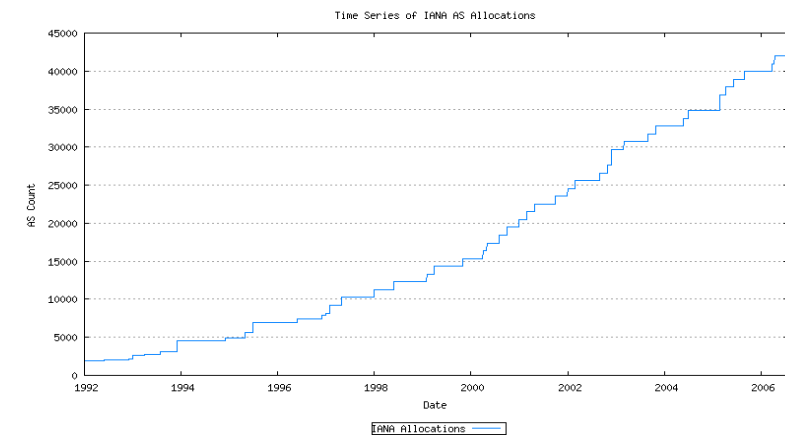
## Web-Cache

- Server-Farm ellenére a várakozási idő gyakran kritikus
- Megoldás:
  - Cache (Proxy)
- Helye
  - A kliens oldalon
  - A lokális hálózatban (egy Proxy-n)
  - Az Internet-Service-Provider-nél
- Kérdések
  - Adatok elhelyezése, nagysága, aktualitása
  - Érvénytelenítés Time-Out által

## Content Distribution Networks (CDN)

- Cache-ek koordinált halmaza
  - Nagy web-helyek terhelését elosztja globálisan elosztott szerver-farmon
  - Lehetőleg különböző szervezetek web-oldalainak kezelése
    - pl. hírek, szoftver-gyártók, kormányok
  - Példák: Akamai, Digital Island
  - A Cache-lekérdezések regionálisan és terhelést tekintve a leginkább megfelelő helyre kerülnek átirányításra
- Példa Akamai:
  - Elosztott hash-tábla által lehetséges az oldalak/adatok elosztása hatékonyan és lokálisan

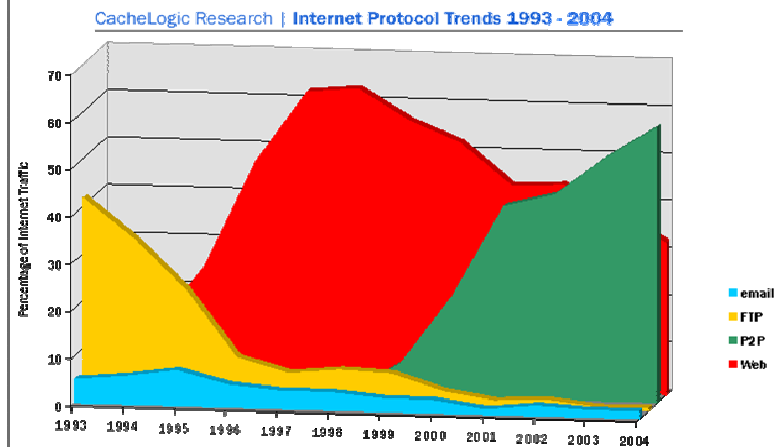
## Az Internet exponenciális növekedése



<http://www.potaroo.net/tools/asns/>

## Forgalom az Interneten

• [http://www.cachelogic.com/research/2005\\_slide07.php#](http://www.cachelogic.com/research/2005_slide07.php#)



## Mi az hogy Peer-to-Peer hálózat?

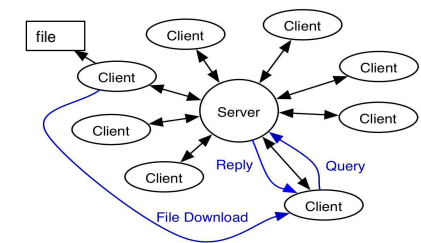
- Mi nem Peer-to-Peer hálózat?
  - Egy Peer-to-Peer hálózat nem kliens-szerver hálózat!
- Definíció
  - **Peer-to-Peer**
    - egyenértékű partnerek közötti kapcsolatot jelenti
  - **P2P** = Peer-to-Peer (Internet slang)
  - Egy **Peer-to-Peer hálózat** egy
    - számítógépek közötti kommunikációs hálózat az Interneten
    - melyben nincs központi irányítás
    - és megbízható partner sem.

## Napster

- Shawn (Napster) Fanning
  - 1999 júniusában adta közre az azóta legendás P2P hálózat beta verzióját
  - Cél: File-sharing rendszer
  - Valójában: Zene cserebörze
  - 1999 őszén Napster volt az „év download-ja”
- A zene ipar szerzői jog pere 2000 júniusában
- 2000 végére kooperációs szerződés
  - Fanning és Bertelsmann Ecommerce között
  - jogilag is biztosított
- 2001 óta Napster egy kommerciális file-sharing rendszer

## Hogy működik Napster?

- Kliens-szerver struktúra
- A szerver tárolja
  - Indexet meta-adatokkal
    - File-név, dátum, stb...
  - Táblázatot a résztvevő kliensek közötti kapcsolatokról
  - Táblázatot a résztvevő kliensek minden file-járól
- Lekérdezés (query)
  - Kliens a file-nével kérdezi le a szert
  - A szerver megkeresi a megfelelő résztvevőket, akik tárolják a file-t
  - A szerver válaszol, ki tárolja a file-t
  - A lekérdező kliens a file-t a tulajdonos klientsól tölti le

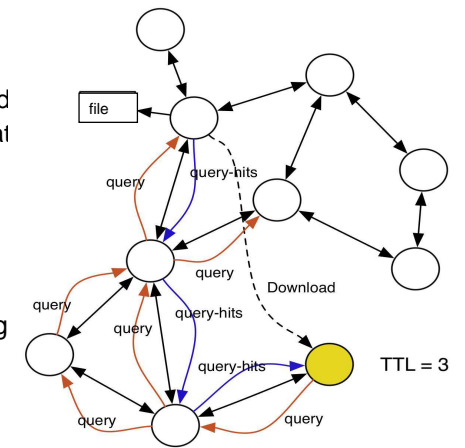


## Gnutella - Történet

- Gnutella
  - 2000 márciusában tette közzé Justin Frankel és Tom Pepper a Nullsoft-tól
  - Nullsoft 1999 óta AOL tulajdona
- File-Sharing rendszer
  - Cél: mint Napster-nél
  - De teljesen központi struktúrák nélkül dolgozik

## Gnutella

- File lekérdezés:
  - a szomszédoknak küldi a kliens
  - azok a saját szomszédjaikhoz küld
  - amíg hop-ok egy megadott számát nem lépi túl
    - TTL mező (time to live)
- Protokoll
  - Query
    - A file lekérdezése TTL hop-ig továbbítódik (restricted flooding)
  - Query-hits
    - A válasz a fordított útvonalon
- Ha file-t megtalálta, direkt letöltés a tulajdonos klienstől



## Peer-to-Peer összefoglalás

- Peer-to-Peer hálózatok forgalmának túlnyomó része szerzői jogokat sért
- De vannak legális felhasználások:
  - Internet-telefon, pl. Skype
  - Szoftver elosztás (pl. Suse disztribúció BitTorrent által)
    - Gyorsabb letöltés, szerverek tehermentesítése
  - Group Ware
    - néhány Group Ware rendszer Peer-to-Peer-t használ
  - GNU-licence alatti szoftver cseréje
  - Privát filmek, fényképek, dokumentumok cseréje
- Peer-to-Peer hálózatok illegális hasznélvezőit az utóbbi időben egyre inkább büntetőjogilag üldözik