



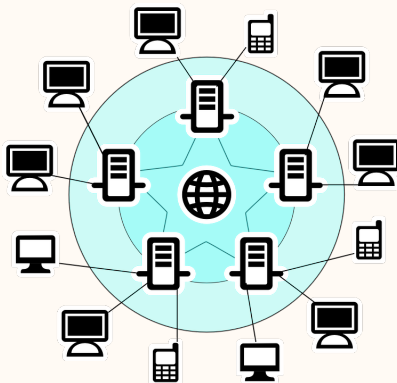
Hálózatok, protokollok & HTTP

Webprogramozás – 1. előadás

Sulyok Csaba

csaba.sulyok@gmail.com

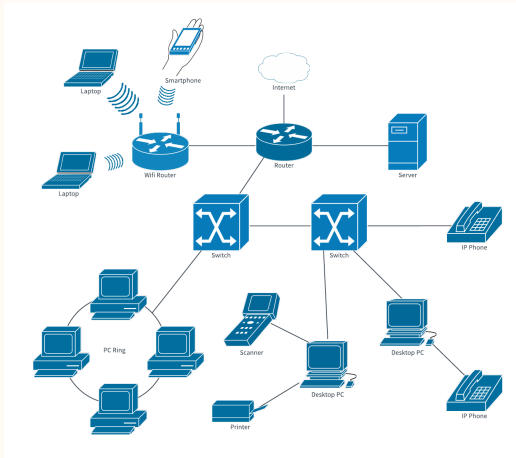




- ▶ *Az Internet egy globálisan összekötött számítógépes hálózat, melyben a gépek jól meghatározott protokollok segítségével kommunikálnak egymással.*

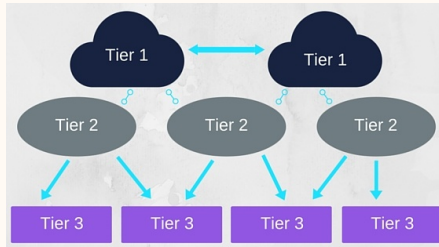
- ▶ Az Internet nyújtotta néhány alapszolgáltatás:
 - ▶ weboldalak böngészése
 - ▶ e-mail (elektronikus levelezés)
 - ▶ állományok átvitele
 - ▶ VoIP (Voice over IP) – online telefonálás (Skype, Google Hangouts, stb.)
 - ▶ peer-to-peer hálózatok – pl. torrent
 - ▶ felhőalapú szolgáltatások – pl. Amazon Web Services, Azure, etc.

- ▶ Összeköttetésben lévő számítógépek milliói:
- ▶ **host-ok = végrendszerek**
- ▶ *hálózati alkalmazásokat futtatnak*
- ▶ **Kommunikációs vonalak**
- ▶ optikai szál, réz, rádió, műhold sb.
- ▶ szállítási gyorsaság = *sávzsélesség*
- ▶ **útvonalválasztók (router):**
csomagokat irányítanak (adat darabok)



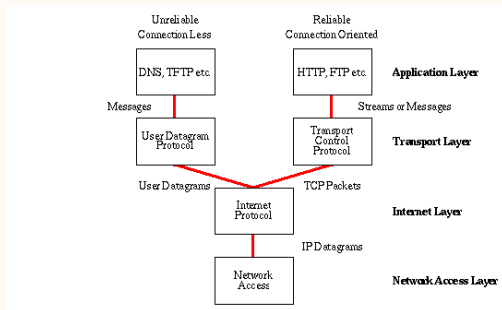
Infrastruktúra és ISP-k

- ▶ Az Internet hálózatok hálózatából áll
- ▶ Rétegzett (tier-based) megközelítés
- ▶ A végfelhasználó egy **ISP**-n (Internet Service Provider) keresztül csatlakozik (alsóbb réteg, általában *Tier 3*)
- ▶ A *Tier 1* szolgáltatók nemzetközi infrastruktúrát biztosítanak; pl. AT&T, Verizon, stb.
- ▶ További részletek

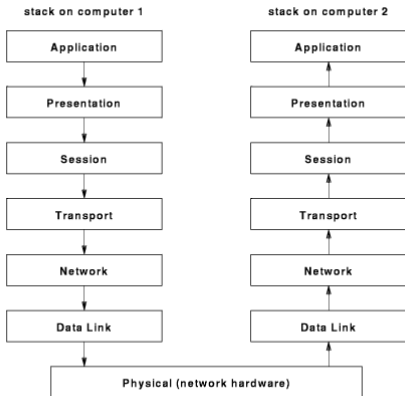


- ▶ A végrendszerek azonosítója:
 - ▶ az *IP címük* (lokális vagy globális, formátum szerint)
 - ▶ a *hostnevük* (belső hálózatban)
 - ▶ a *domain nevük* (globális – www.)
- ▶ Ha a globális hálózaton keresztül csatlakozunk egy géphez (nem intraneten keresztül, hanem egy Tier 1 szolgáltatáson keresztül), elérjük a *világhálót*
- ▶ Bármilyen bináris információt küldhetünk ezeken keresztül, így használatuk *instabil és sok hibalehetőséggel ellátott.*
- ▶ A válasz: **protokollok.**

- ▶ Hálózati kommunikációra vonatkozó standardok és szabályok.
- ▶ Tartalmaz üzenetekre vonatkozó formázási, strukturálási, encoding, stb. megkötések.
- ▶ Mindkét félnek muszáj ismernie a használt protokollt.



- ▶ A protokollok *rétegzett megközelítést* alkalmaznak – számunkra a legfontosabb rétegek:
 - ▶ **Hálózati réteg** – azonosítja a célpont gépet (IP, hostname)
 - ▶ **Adatátviteli réteg** – TCP & UDP – érdekeli-e a csomagvesztés?
 - ▶ **Alkalmazás réteg** – konvencionális kulcsszavak és formátumok a Socketeken küldött üzeneteknek – ezeket mindkét fél érti. *Példák:*
 - ▶ *HTTP* – webfejlesztésben használatos; HyperText-et közvetít – dinamikus szöveg kiegészítve linkekkel, formázási, kép/hang s más média elemekkel
 - ▶ *FTP* – fájlok átvitelére
 - ▶ *SMTP/IMAP/POP* – e-mailek küldésére/fogadására
 - ▶ *SSH* – külső gép termináljának elérésére
 - ▶ *DNS* – keresési protokoll; domain neveket társít IP-címekhez
 - ▶ *peer-to-peer protokollok* – nincs dedikált szerver/kliens, minden gép egyenlő. Példa: torrentek

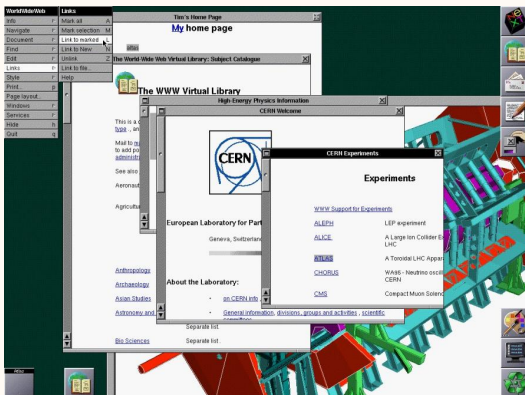


- ▶ minden küldő réteg hozzátesz információt a kimenő csomaghoz
- ▶ minden fogadó réteg “lehámoz” valamennyi információt fordított csomagból
- ▶ routerek/switch-ek s más köztes eszközök csak az alsó rétegeket érintik

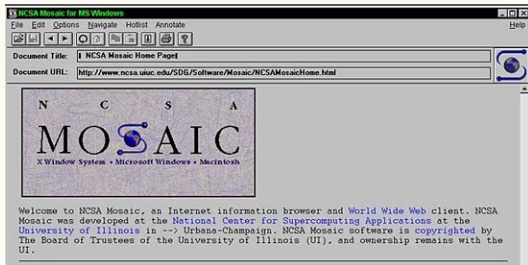


- ▶ 1989 – **Tim Berners-Lee** (CERN – Európai Részecskegyorsító Intézet) javaslata:
 - ▶ *hiperszöveg-rendszer* alkalmazása kutatók közti kommunikáció javítására
 - ▶ szöveges információ, képek s hanganyag hiperhivatkozások (*hyperlink*) segítségével összekapcsolt rendszere
 - ▶ ingyenes standard, bárki által használható
- ▶ Berners-Lee egy NeXT számítógépen (25MHz CPU, 256MB merevlemez, 8MB RAM) írta az első böngészőt és ezen használt először webszervert.

- ▶ Az egyik legrégebben fennmaradt oldal egyike a Tim Berners-Lee által készített oldalaknak: [World Wide Web](#)
- ▶ A legelső böngésző Tim Berners-Lee gépén:

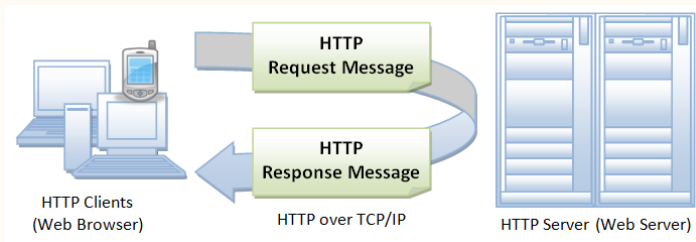



- ▶ 1992 január, Genf – a **web** első nyilvános használata
- ▶ 1993, Illionois-i egyetem – megjelenik az első ingyenes böngésző, az **NCSA Mosaic** – egy éven belül már kb. 2 millióan használják
- ▶ 1994 – Megalakul a **World Wide Web Consortium (W3C)** – WWW-vel kapcsolatos standardokat általánosító intézet



Kliens-szerver architektúra

- ▶ a Web **kliens-szerver architektúrát** alkalmaz az információk megosztására és terjesztésére.
- ▶ a rendszer működése:
 - ▶ kliens program (web-böngésző) → Internetre rákapcsolt web-szerver
 - ▶ web-szerver: értelmezi az üzenetet → kliens válasz = a kért dokumentum + kiegészítő információk
 - ▶ böngésző: értelmezi a kapott állományt, majd megjeleníti/lementi/futtatja azt a felhasználói gépen
- ▶ az információ-csere a **HTTP protokoll** segítségével történik





SEO Cheat Sheet: Anatomy of a URL

1 Protocol
2 Subdomain
3 Domain
4 Top-Level Domain (TLD)
5 Folders / Paths
6 Page
7 Named Anchor

Keyword Priority¹
Observed Google priority of keyword placement:

1. Domain
2. Subdomain
3. Folder
4. Path/Page

SEO Tips for URLs

- Use subdomains carefully. They may be treated as separate entities, splitting domain authority.
- Separate path & page keywords with hyphens ("-").
- Anchors may help engines understand page structure.
- Keyword effectiveness in URLs decreases as URL length and keyword position increases.¹

¹ Moz correlational data (2015)

1 Protocol
2 Subdomain
3 Domain
4 Top-Level Domain (TLD)
5 Page / File Name
6 File Extension
7 CGI Parameters

Popular TLDs

- .com - commercial
- .org - non-profit
- .net - infrastructure
- .edu - schools
- .info - information
- .co - commercial
- .biz - small business
- .io - technology
- .gov - government

Popular ccTLDs

- .cn - China
- .de - Germany
- .fr - France
- .uk - United Kingdom
- .ru - Russian Federation
- .nl - Netherlands
- .br - Brazil
- .eu - European Union
- .au - Australia

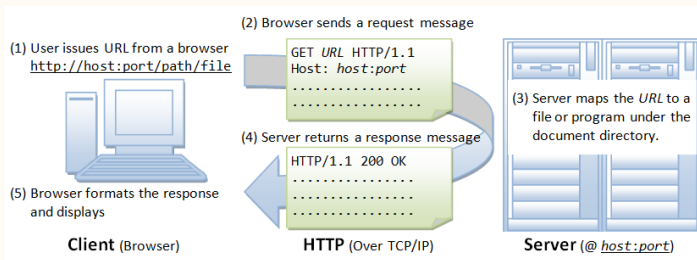
Popular Extensions

- .htm or .html - Static HTML
- .php - PHP code
- .aspx - ASP.NET
- .xml - XML file
- .jpg or .jpeg - JPEG image file
- .pdf - Adobe PDF file
- .js - JavaScript Source file
- .rb or .rhtml - Ruby
- .py - Python

¹Technically a GccTLD

- ▶ Uniform Resource Locator
- ▶ Egyedi azonosítója egy a hálózatban található erőforrásnak.
- ▶ Formátum: `<protokoll>://<host>[:<port>][/<útvonal>]`
- ▶ Alapértelmezett portok: HTTP: **80**, SSH: **22**, FTP: **21**, stb.

- ▶ **HTTP:** állapotmentes kliens-szerver alapú kommunikációs protokoll
- ▶ a kliens oldal küld kéréseket, a szerver oldal válaszol.



Forrás: www.ntu.edu.sg

- ▶ **kérésfejléc (header)** – szöveg a köv. tartalommal:
 - ▶ első sor hívásinformációval:
 - ▶ metódus (**GET**, **POST**, **PUT**, **DELETE**, etc.)
 - ▶ az erőforrás útvonala (path)
 - ▶ a használt HTTP protokoll verziója (opcionális)
 - ▶ *fejlécek* – kulcs-érték párosok per sor, pl. forráshost, üzenet hossza, információ a böngészőről és a test tartalmának típusáról
- ▶ **kéréstest (body)** (opcionális – a metódustól függ)
 - ▶ a fejléctől egy üres sor választja el
 - ▶ bármilyen formátumú adatot tartalmazhat – lehet ember által olvasható (paraméterek, JSON, etc.), bináris (kép, hang, stb.), stb.

▶ GET

- ▶ lekérdezést jelképez (**adj** vissza valamit)
- ▶ a paraméterek az URL-ben lesznek elküldve (nem biztonságos)
- ▶ konvencionálisan *idempotens*: többszöri lekérdezés mindig ugyanazt eredményezi

▶ POST

- ▶ feltöltésre, erőforrások létrehozására/módosítására használjuk
- ▶ a paramétereket az üzenet törzsében küldi a böngésző (biztonságosabb) a szervernek
- ▶ nem idempotens

- ▶ A böngészők szinkron kommunikáció esetén általában GET és POST hívásokat intéznek a szerver felé

- ▶ Az aszinkron kommunikáció és REST API-k kiemelkedése kiemeli más metódusok használatát is (PUT, PATCH, DELETE), ezekről később

HTTP kérés példa



```
GET /doc/test.html HTTP/1.1
```

```
Host: www.test101.com
```

```
Accept: image/gif, image/jpeg, */*
```

```
Accept-Language: en-us
```

```
Accept-Encoding: gzip, deflate
```

```
User-Agent: Mozilla/4.0
```

```
Content-Length: 35
```

```
bookId=12345&author=Tan+Ah+Teck
```

Request Line

Request Headers

Request
Message
Header

A blank line separates header & body

Request Message Body

Forrás: www.ntu.edu.sg

- ▶ **válaszfejléc (header)**
- ▶ első sorban a **státusz**:
 - ▶ a HTTP protokoll verziója a szerveren
 - ▶ státuszkód s ennek magyarázata (utóbbi opcionális) – részletek [itt](#)
 - ▶ **2xx rendben** – pl. 200 OK, 201 Created
 - ▶ **3xx átirányítás** – pl. 301 Moved, 304 Not Modified
 - ▶ **4xx kliensoldali hiba** – pl. 400 Bad Request, 403 Forbidden, 404 Not Found
 - ▶ **5xx szerveroldali hiba** – pl. 500 Internal Server Error, 502 Bad Gateway
- ▶ **fejlécek** – kulcs-érték párosok per sor, pl. szerverinformáció s a test tartalmának típusa
- ▶ **választest (body)**

HTTP válasz példa



HTTP/1.1 200 OK

Date: Sun, 08 Feb xxxx 01:11:12 GMT

Server: Apache/1.3.29 (win32)

Last-Modified: Sat, 07 Feb xxxx

ETag: "0-23-4024c3a5"

Accept-Ranges: bytes

Content-Length: 35

Connection: close

Content-Type: text/html

<h1>My Home page</h1>

→ Status Line

} Response Headers

} Response
Message
Header

→ A blank line separates header & body

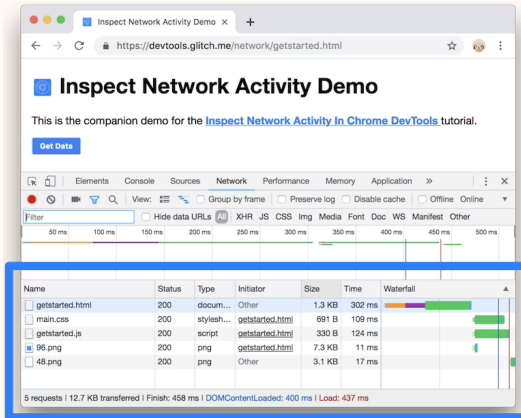
} Response Message Body

Forrás: www.ntu.edu.sg

- ▶ Lokális állományokban a *kiterjesztés* ad útmutatót arról, hogy hogyan kezeljük a tartalmat (pl. **.jpeg**, **.png** – képek, **.txt** szöveges állományok, stb.)
- ▶ Egy HTTP kérés vagy válasz testében bármilyen típusú adat megjelenhet.
- ▶ A **Content-Type** válasz-fejléckulcs megmondja, hogy a kliens (pl. a böngésző) hogyan értelmezze a visszaküldött body tartalmát.
- ▶ Értéke a **MIME** (Multipurpose Internet Mail Extensions) standardot követi.
 - ▶ eredetileg e-mailek formátumának jelzésére lett kifejlesztve
 - ▶ struktúrája: **típus/altípus**
 - ▶ példák:
 - text/plain
 - text/html
 - image/jpeg
 - image/png
 - audio/mp3

HTTP kérések monitorizálása

- ▶ a böngészőben monitorizálható, hogy milyen HTTP kérések mennek végbe
- ▶ **Developer Console (F12): Network tab**

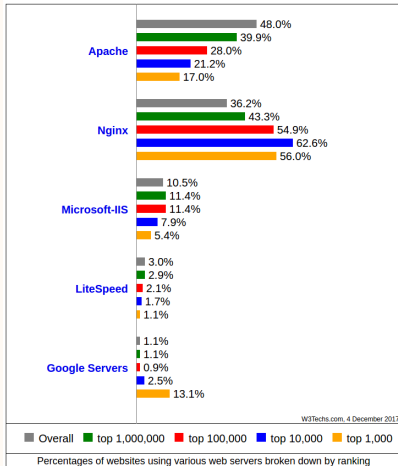


The screenshot shows the Chrome DevTools Network tab. The browser address bar displays `https://devtools.glitch.me/network/getstarted.html`. The page title is "Inspect Network Activity Demo". Below the title, there is a description: "This is the companion demo for the [Inspect Network Activity in Chrome DevTools](#) tutorial." and a "Get Data" button.

The Network tab is active, showing a list of requests. The table below summarizes the data shown in the screenshot:

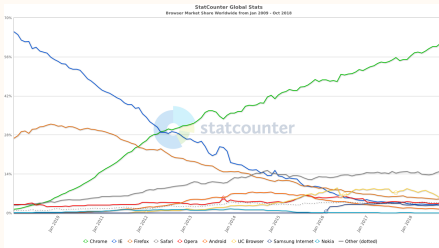
Name	Status	Type	Initiator	Size	Time	Waterfall
<input type="checkbox"/> getstarted.html	200	document	Other	1.3 KB	302 ms	
<input type="checkbox"/> main.css	200	stylesheet	getstarted.html	691 B	109 ms	
<input type="checkbox"/> getstarted.js	200	script	getstarted.html	330 B	124 ms	
<input checked="" type="checkbox"/> 96.png	200	png	getstarted.html	7.3 KB	11 ms	
<input type="checkbox"/> 48.png	200	png	Other	3.1 KB	17 ms	

At the bottom of the Network tab, a summary bar shows: 5 requests | 12.7 KB transferred | Finish: 458 ms | DOMContentLoaded: 400 ms | Load: 437 ms.



- ▶ A legegyszerűbb HTTP szerver: felszolgálja fájlok tartalmát statikusan, hozzáadott logika nélkül.
- ▶ A piacot uralják:
 - ▶ Apache
 - ▶ nginx
- ▶ Dinamikus szerverek (amelyek komplex logikát használnak HTTP kérések kiszolgálására) képesek mímelni egy statikus szert. Pl.: A node.js-re épített `serve` könyvtár.

- ▶ Még mindig a leghasználtabb HTTP kliens a **böngésző**.
- ▶ Böngészők: Internet Explorer, Mozilla Firefox, Google Chrome, Safari, Edge
- ▶ A böngészők a következő nyelveket ismerik: **HTML, CSS, Javascript**.
- ▶ Más eszközöket is alkalmazhatunk HTTP hívások tesztelésére (i.e. ezek a “kliens” szerepet töltik be): pl. [curl](#) vagy [Postman](#).



Forrás: <http://gs.statcounter.com/>

HTTP kliensek: Postman Console



- ▶ A Postman interaktív vizuális felület, de képes kirajzolni a pontos HTTP kéréseket is

```
Postman Console
File Edit View Collection History Help
Filter Messages Clear
GET http://localhost:8080/index.html 11:52:40.599
Pretty Raw 200
20 ms
GET /index.html
cache-control: no-cache
postman-token: 9a13f31b-bba3-4929-9a7e-bc7fa93af096
user-agent: PostmanRuntime/6.4.1
accept: */*
host: localhost:8080
cookie: connect.sid=s%3AKqe71fkDUKvvYjTd-24kVQG01PiThDcV.o088PdM9Jbpv1lFYi%2FwIVz9lIxNgtbQVbxuly%2FAhH4E
accept-encoding: gzip, deflate

HTTP/1.1 200
status: 200
server: nginx/1.10.3 (Ubuntu)
date: Mon, 04 Dec 2017 09:52:40 GMT
content-type: text/html
content-length: 7
last-modified: Mon, 04 Dec 2017 09:45:13 GMT
connection: keep-alive
etag: "5a251929-7"
accept-ranges: bytes

hello
```