
Balogh Zoltán

Senior Software Engineer

Email:

baloghzoltan1977@gmail.com

2003 és 2008 között tanársegéd az Óbudai Egyetem elődjénél.

2008-tól az Andtek GmbH telekommunikációs mérnöke

2015-től Az Enghouse Interactive vezető fejlesztője

Fő fejlesztési területek:

Cisco VoIP technológiákra épülő szoftverek fejlesztése

Linux alapú nagy rendelkezésre állású rendszerek

Backend szolgáltatások fejlesztése

Internetes alkalmazások és WEB technológiák

A félév során érintett témakörök:

- TCP/IP protokollcsalád, Web működése
- HTTP működése: HTTP/1.1, HTTP/2, HTTP3 protokollok
- TLS/SSL, PKI, HTTPS
- MVC, REST, API tervezés
- Kliensoldali webes technikák
- Autentikáció és jogosultságkezelés
- Webbiztonság alapjai
- SOAP, gRPC, WebSocket
- Protokollok alkalmazása a MicroService architektúrában

GitHub repository

A tantárgyhoz kapcsolódó anyagok és példaprogramok elérhetők az alábbi GitHub linken:

<https://github.com/zbalogh/oe-internetes-alkalmazasok>



Internet alapjai, TCP/IP protokollcsalád, a Web működése



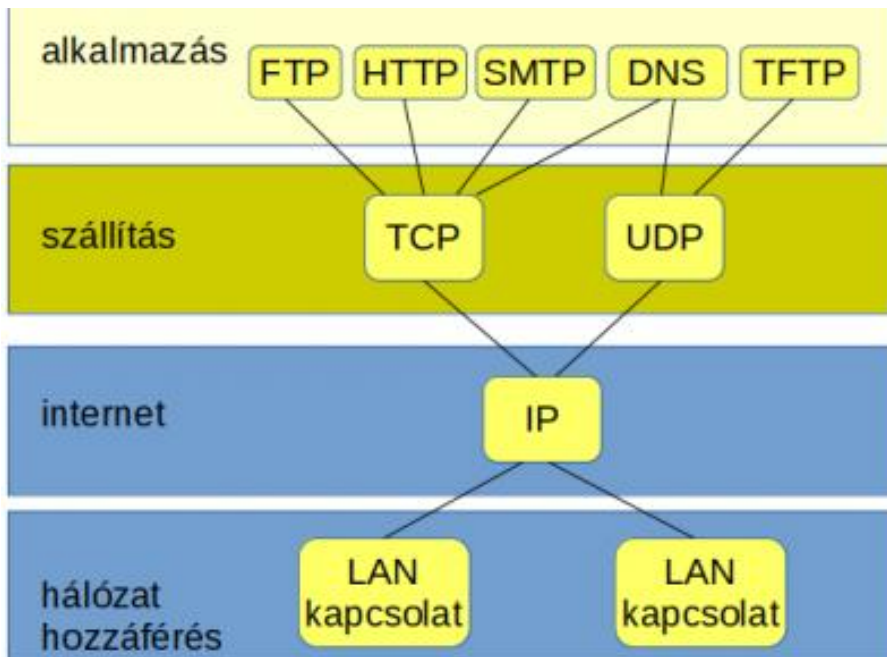
Mi az Internet?

-
- Hálózatok hálózata, **globális** infrastruktúra
 - **Elosztott, decentralizált**, nincs központi irányítója
 - **IP** címezés alapja
 - **Nyílt protokollok** (TCP/IP, HTTP, DNS, stb) használata
 - Ezekre épülnek azok az átviteli protokollok, amelyekkel a félév során megismerkedünk.

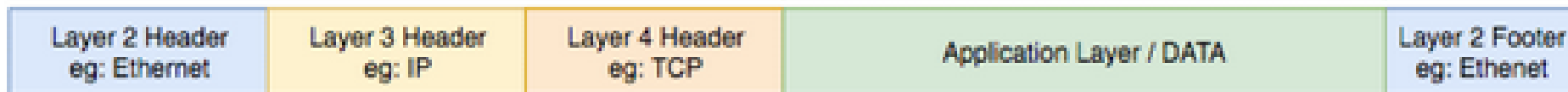
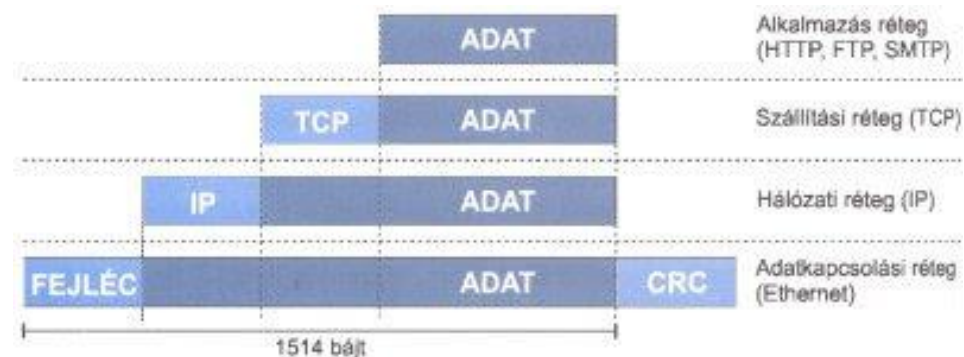
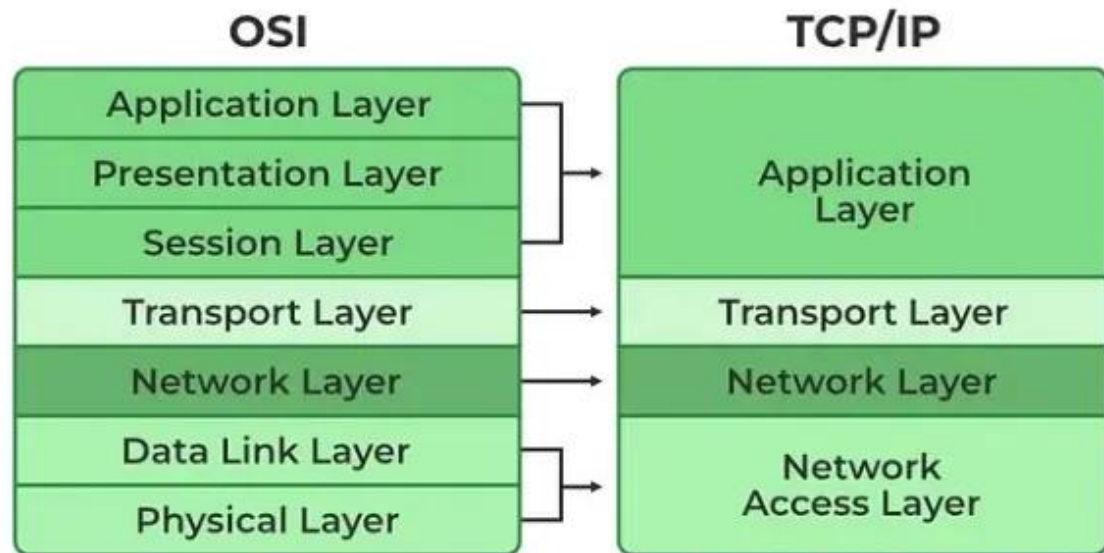
Kliens–szerver modell

- A legtöbb Internetes alkalmazás kliens-szerver modellben működik
- Kliens alkalmazás kéréseket küld
- Szerver alkalmazás válaszol
- Állapotmentes kommunikáció
- Állapottároló kommunikáció (munkamenet tárolás a szerveren)
- Szerver portokon várja a kéréseket (listening)
- Legismertebb portok: 80 port (HTTP), 443 port (HTTPS), 53 port (DNS), 25 port (SMTP)

TCP/IP modell áttekintése



- 4 rétegből áll (szemben az OSI model 7 rétegével):
 - **Alkalmazási** réteg (pl. HTTP, DNS, SMTP)
 - **Szállítási** réteg (TCP, UDP)
 - **Hálózati** réteg (IP)
 - Hálózati **hozzáférési** (fizikai) réteg (pl. Ethernet)
- Minden réteg egy-egy konkrét feladatkört lát el, és ezek egymásra épülnek.
- A TCP/IP modell nagyon praktikus, mert kifejezetten a modern internetes kommunikációhoz készült.



IP protokoll alapjai

- **Csomag alapú** továbbítás
- Best effort alapú (**legjobb szándék**) csomagszállítás a forrástól a célig
- **Összeköttetés mentes**: nem épít ki kapcsolatot a végpontok között.
- **Megbízhatatlan**: Nem garantálja a csomag kézbesítését, sem a sorrendet.
- A megbízhatóságot a felette lévő protokollnak (TCP) kell biztosítani
- IPv4 (32 bites) címek, IPv6 (128 bites) címek
- **Útválasztás** (routing): Az útválasztók továbbítják a csomagokat az útválasztó táblák segítségével

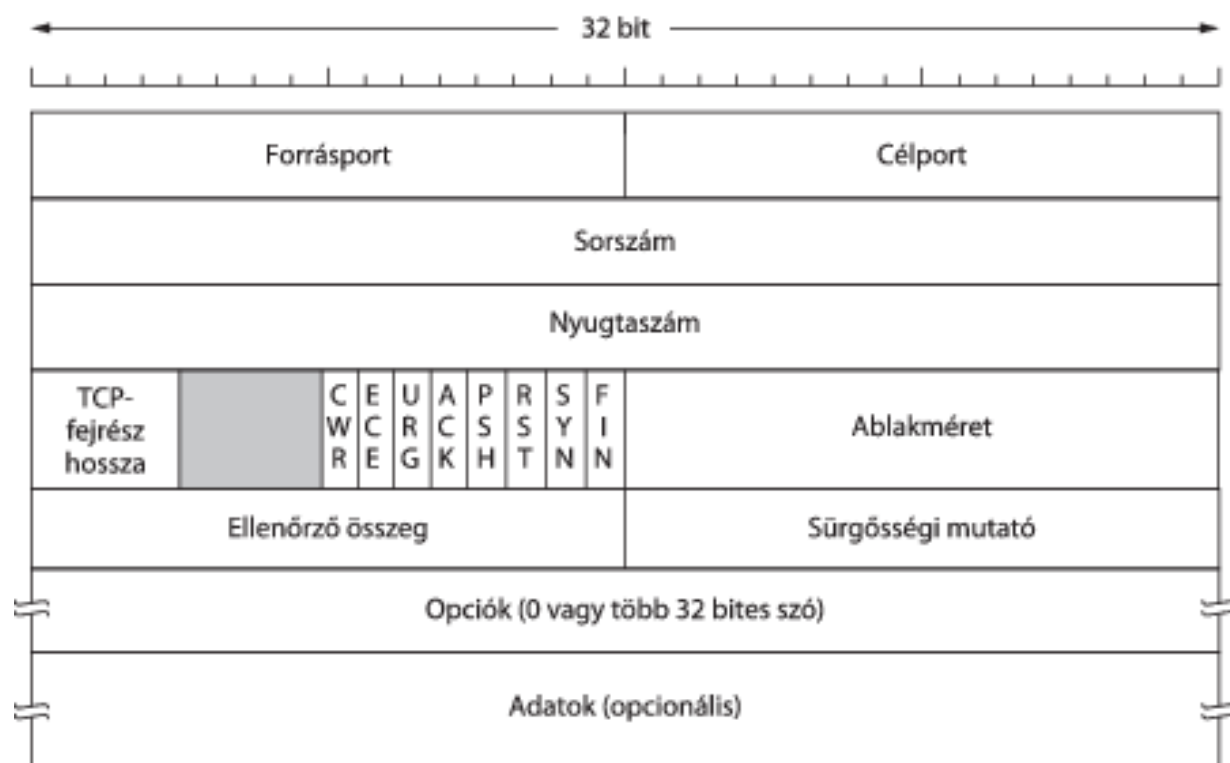
IP fejléc felépítése

32 bit				
Verzió	HL	TOS	Csomag hossza	
Fragmentation azonosító			Flag	Fragment offset
TTL		Protokoll	CRC	
Feladó címe				
Célpont címe				
Opciók (változó hosszúságúak)				

TCP protokoll alapjai

- Transmission Control Protocol – átvitelvezérlő protokoll
- **Megbízható**, kapcsolat-orientált
- **Háromlépcsős kézfogás** (3-way handshake)
 - SYN
 - SYN-ACK
 - ACK
- Az adatokat TCP szegmensekben viszik át
- **Sorrend, újraküldés**, csomagsorszámozás, nyugták
- Folyamvezérlés és torlódásvezérlés
- A legtöbb alkalmazási protokoll TCP-t használ a megbízhatóság miatt.

TCP fejléc felépítése



UDP protokoll alapja

- **Megbízhatatlan**, kapcsolatmentes
- Nem garantál semmit
- **Nincs kézfogás**
- Nincs sorszámozás, újraküldés, nyugtázás sem
- Nagyon gyors
- Használata: VoIP (RTP), videó, gaming, DNS

UDP fejléc felépítése

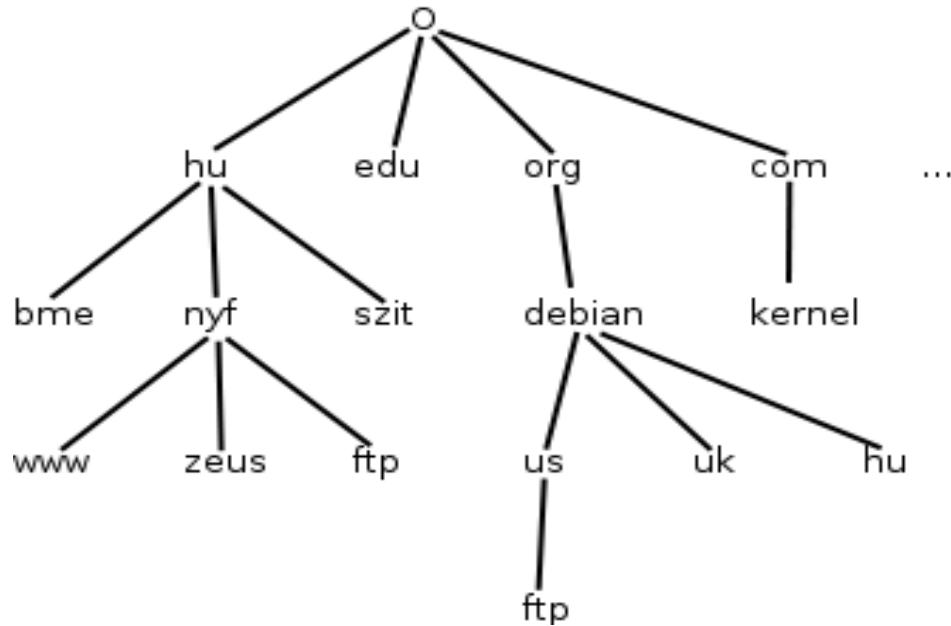
FORRÁSPORT	RENDELTEZÉSI PORT
HOSSZ	ELLENÖRZŐÖSSZEG
ADAT	

UDP csomagformátum

Portok szerepe

- Port: Az alkalmazás végpontja (Socket azonosító)
- A szerver szolgáltatások a portokon figyelnek
- Tipikus portok: 80 (HTTP), 443 (HTTPS), 22 (SSH), 53 (DNS)
- Kliensek is portot használnak saját végpontjukhoz
- A kliensek portjai véletlenszerűen kerülnek kiosztásra

DNS: Domain Name System



- Név → IP cím feloldás
- A DNS az „Internet telefonkönyve”
- Hierarchikus szerkezet
 - Root servers
 - TLD (Top Level Domains): .com, .hu, .org
 - second level domains
 - subdomains
- Cache-elés

Mi az URL / URI / URN ?

URI (Uniform Resource Identifier): az erőforrás univerzális azonosítója (Lehet URL vagy URN)

- **URL** (Uniform Resource Identifier): olyan URI, amely megadja, *hol* és *hogyan* érjük el az erőforrást
- **URN** (Uniform Resource Name): olyan URI, ami csak névként azonosít, nem ad elérési utasítást
- URL felépítése: *scheme://host:port/path?query#fragment*
- Példák:
 - <https://www.example.com/path/to/page.php?search=alma>
 - <ftp://ftp.example.org/pub/file.zip>
 - <mailto:info@example.com>
 - <tel:+36123456789>
 - urn:isbn:9780306406157

Hogyan tölt be egy weboldal? (high-level)

- DNS lekérdezés
- TCP kapcsolat
- HTTP kérés
- HTTP válasz
- HTML → CSS → JS betöltése
- DOM felépítése, Renderelés

HTTP nagyon röviden

- **Kliens–szerver** alapú
- Kérések és válaszok
- **Állapotmentes**: minden kérés önálló
- Módszerek: GET, POST, PUT, stb

Megjegyzés: A következő órákban erről részletesebben beszélünk majd!

HTTPS és TLS nagyon röviden

- Titkosítás
- Digitális tanúsítványok
- TLS handshake
- Miért fontos?

Megjegyzés: A következő órákban erről részletesebben beszélünk majd!

Összefoglalás

- Internet = sok hálózat kapcsolata (decentralizált)
- TCP/IP modell rétegei
- IP, TCP, UDP alapjai
- DNS, URL, HTTP működése
- Hogyan tölt be egy weboldal?