

PROTOKOLLARCHITEKTÚRÁK

Mérnök-informatikus szak, BSc, 3. félév
2017/18 2.félév

Dr. Simon Vilmos
docens

BME Hálózati Rendszerek és Szolgáltatások Tanszék
svilmos@hit.bme.hu

2017.szeptember 27.

Bevezetés 1.

- „Protokoll”?
 - Más területeken: **orvosi protokollok, diplomáciai protokollok**

- A kommunikációs hálózatoknál is szükség van **viselkedési szabályok**, azaz **protokollok**
 - lerögzítésére
 - betartására

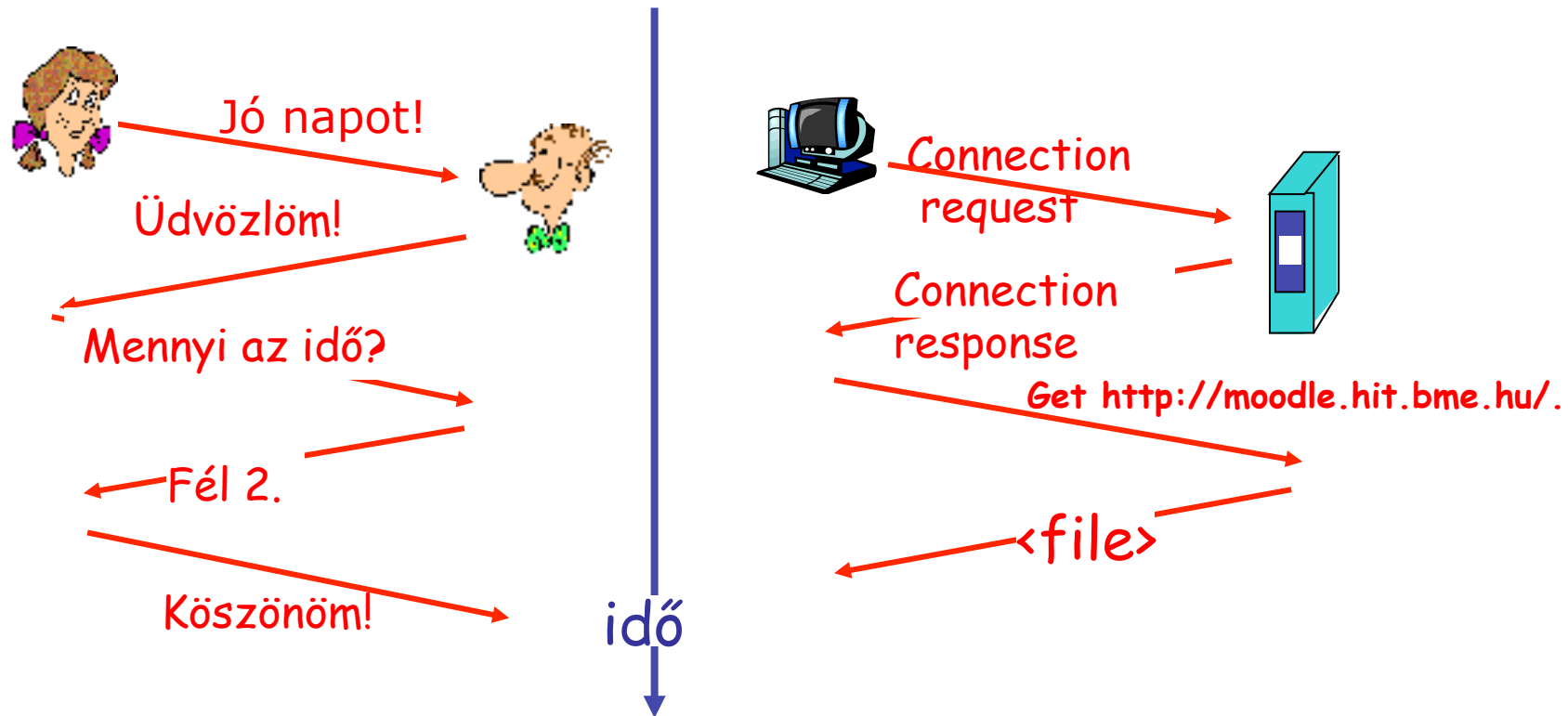
- Definíció:

A protokoll két vagy több kommunikáló egység közötti üzenetcsere **formátumait** és az **üzenetek sorrendjét** határozza meg, valamint az üzenetek vételéhez kapcsolódó és egyéb események által kiváltott **tevékenységeket**.

Bevezetés 2.

- Röviden:
 - Szintaxis (*syntax*) (formátum): adategységek formai leírása
 - Szemantika (*semantics*) (jelentés): az adatelemek értelmezése
 - Viselkedés (*behavior*): mit kell tenni valamely üzenet vételekor, küldésekor
- Helyesírás: protokoll és *protocol*
- **Módszer -> algoritmus -> protokoll** (l. pl. majd a routingnál: *linkállapot-módszer – Dijkstra-alg. – OSPF protokoll*)

Protokollok az emberi és a gépek közötti kommunikációban



Miért „protokollok”?

- Miért nem elég egyetlen protokoll?
 - Történelmi okai
 - **Részfeladatokra tördelés**
 - tevékenységek részekre bontása
 - az egyes részekre önálló szabályok

- Miért részekben, miért nem egyben?
 - egyszerűbb a tennivalók elvégzése
 - részenként lehet továbbfejleszteni

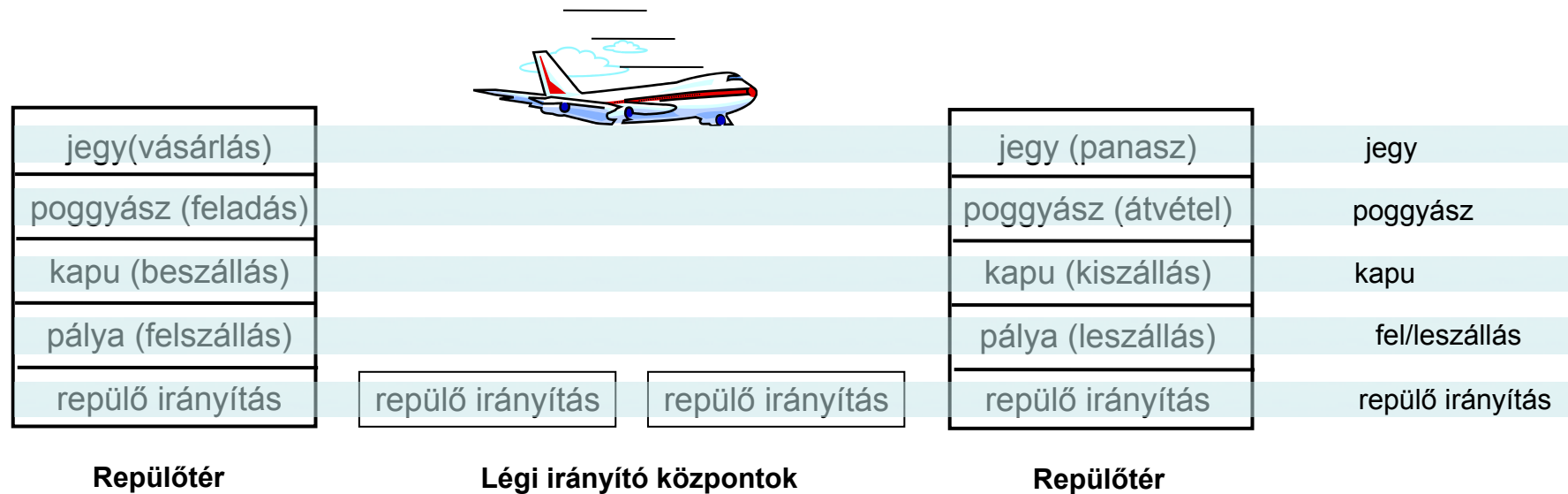
- Hogyan bontsuk részekre?
 - hány rész legyen?
 - hogyan viszonyuljanak egymáshoz?
 - vannak-e „természetes” határok?

- Önálló, jellegzetes, gyakori feladatok (pl):
 - illeszkedés a jelátvivő közeghez
 - hardware-közeli feladatok megoldása, problémák kezelése
 - összeköttetéseken megbízható adattovábbítás
 - végpontok közötti információtovábbítás
 - kívánt végpontok azonosítása
 - kedvezőtlen forgalmi állapot kezelése

- Mindezekkel kapcsolatban az előforduló hibák kezelése is

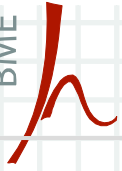
- Részek – funkcionalitások – „**rétegek**”

Analógia: légi közlekedés rétegezve



Rétegek: minden réteg egy szolgáltatást valósít meg

- a saját rétegen belüli tevékenységével
- támaszkodva az alatta lévő réteg által nyújtott szolgáltatásra

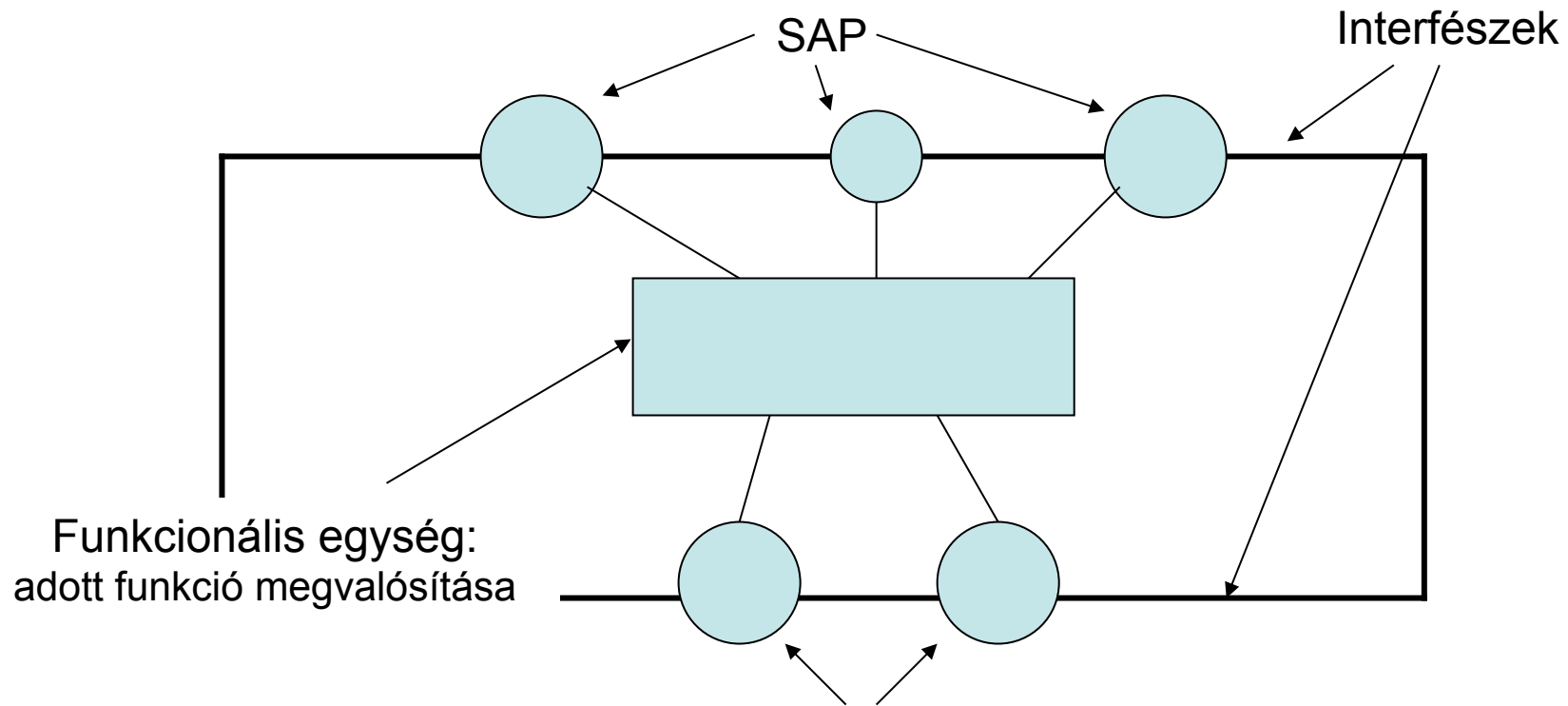


A rétegzett protokollarchitektúrák előnyei és hátrányai

- Előnyök:
 - Nagyon összetett feladat **kezelhető részekre** bontható
 - Az egyes részek megvalósítása **független más részekről**, így könnyen módosítható
 - Feljebb lévő feladatoknak **közös kiszolgálást** nyújthat egy lejjebb lévő rész

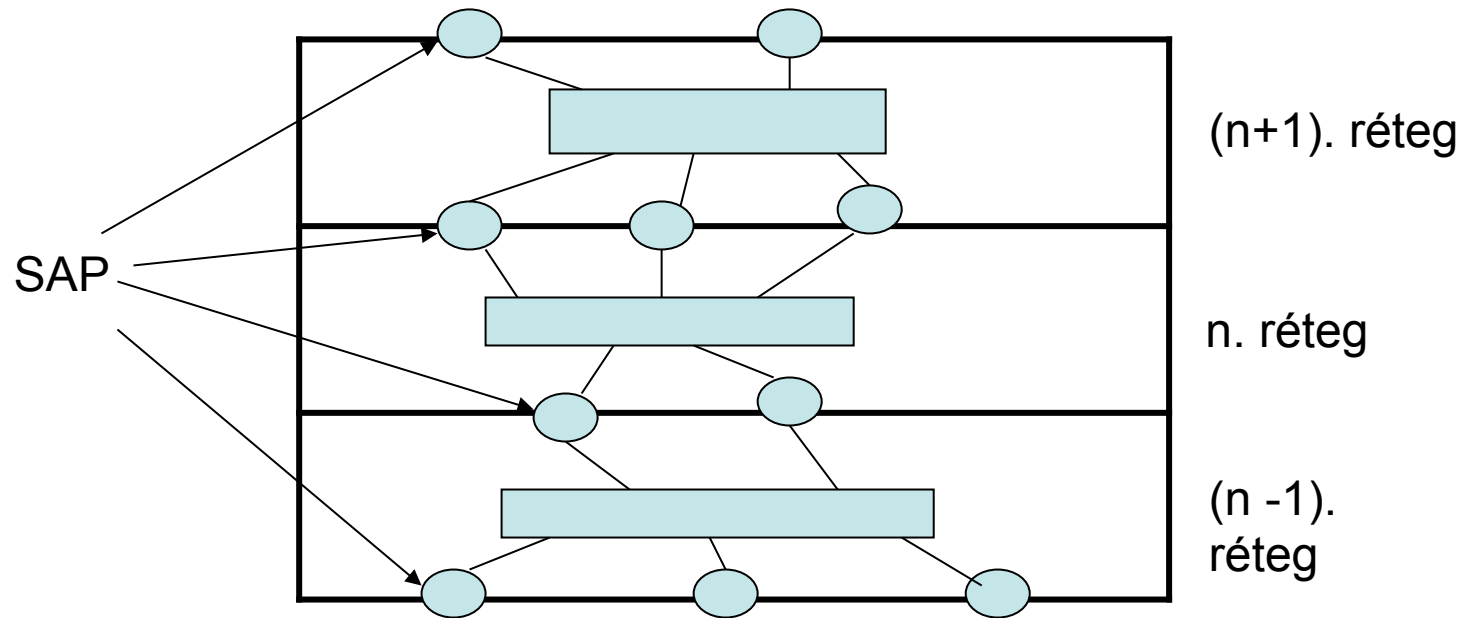
- Hátrány:
 - Adott réteg működéséhez szükséges információk egy másik rétegben → megsérül a rétegstruktúra
 - **Feladatok duplikálása** elkerülhetetlen
 - hatékonyság rovására megy (pl. hibavédelem több rétegben is)

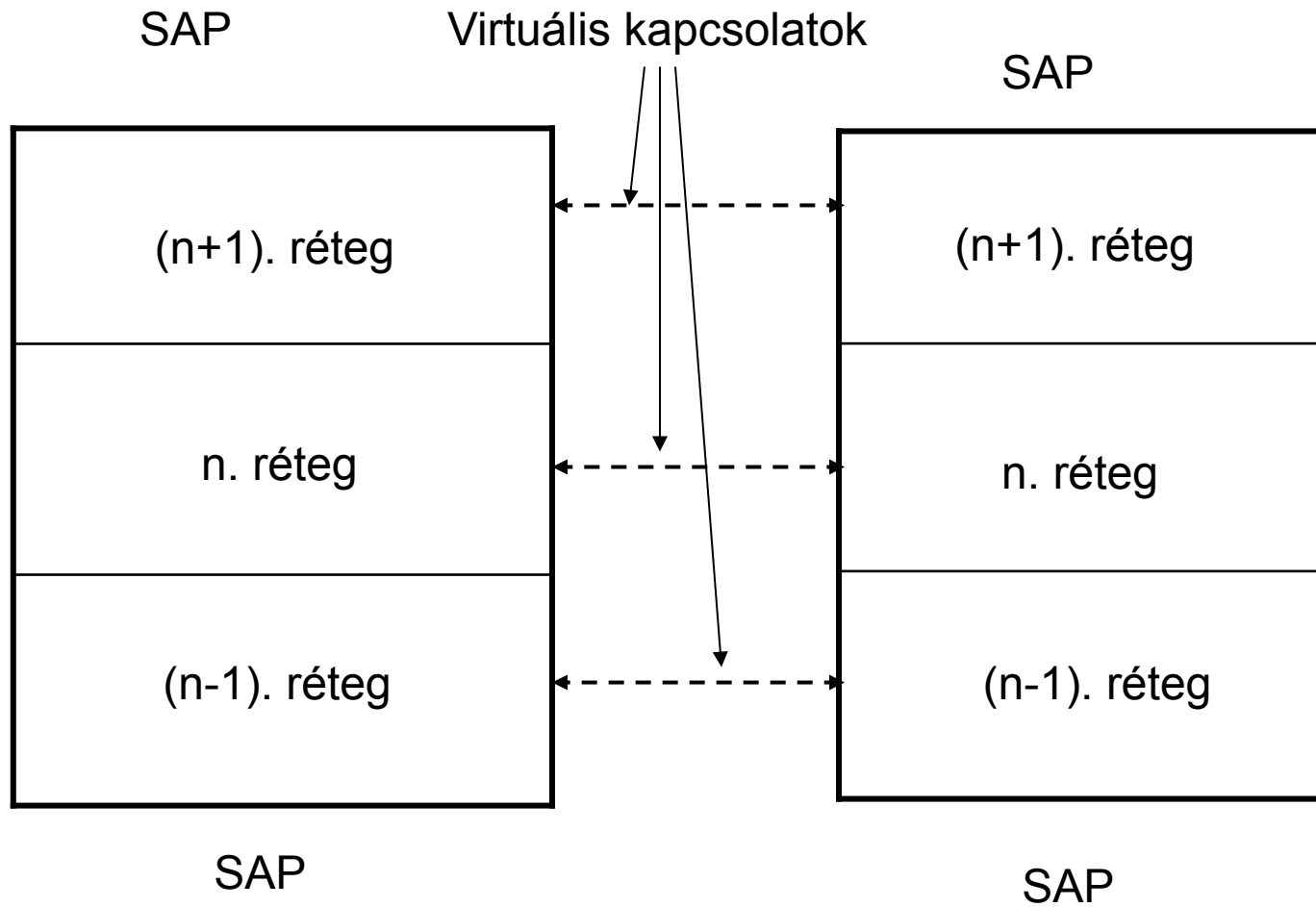
A réteg elve



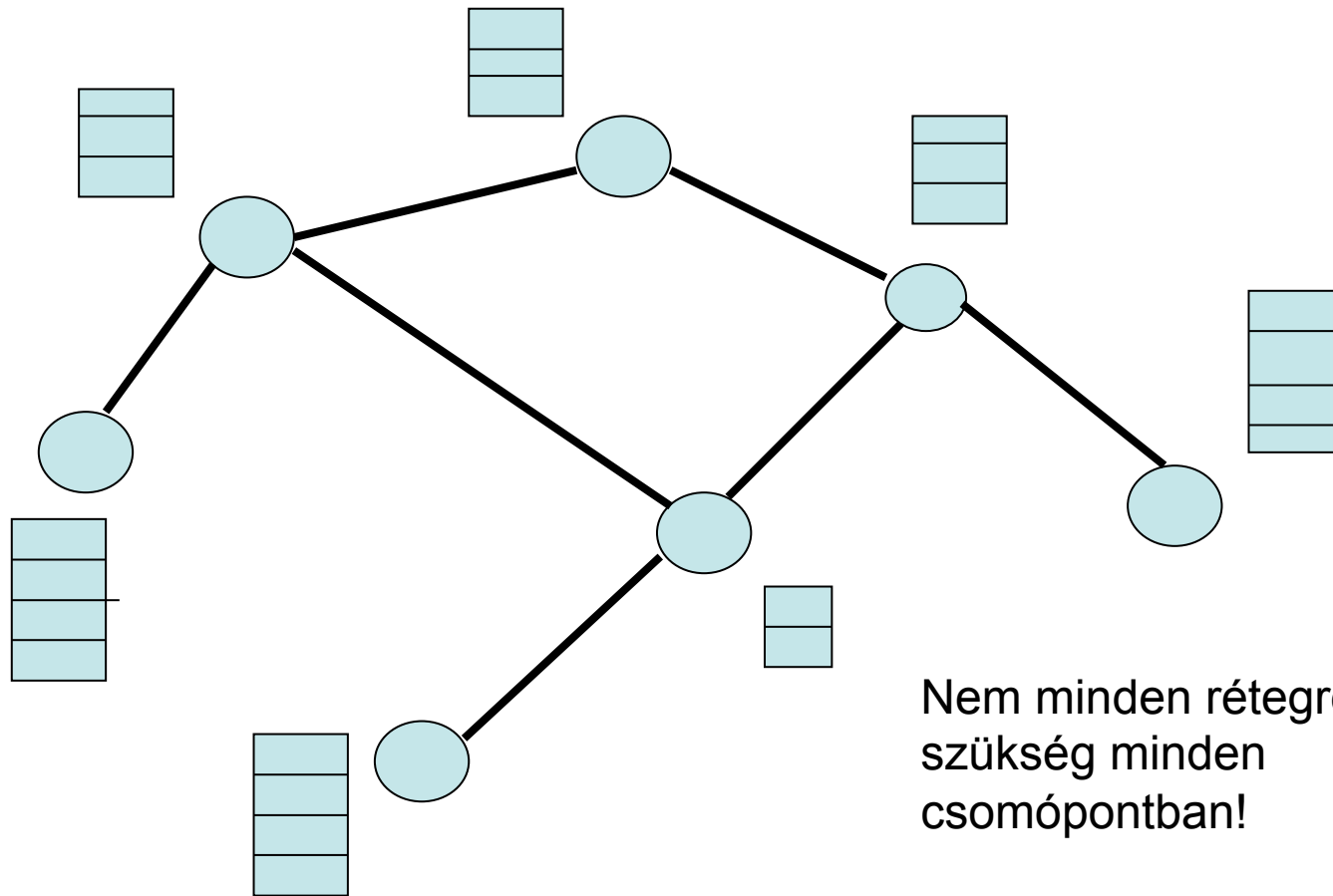
- Interfész-elv
- A megvalósítás elrejtése

SAP (Service Access Point) –
szolgáltatás-elérési pont



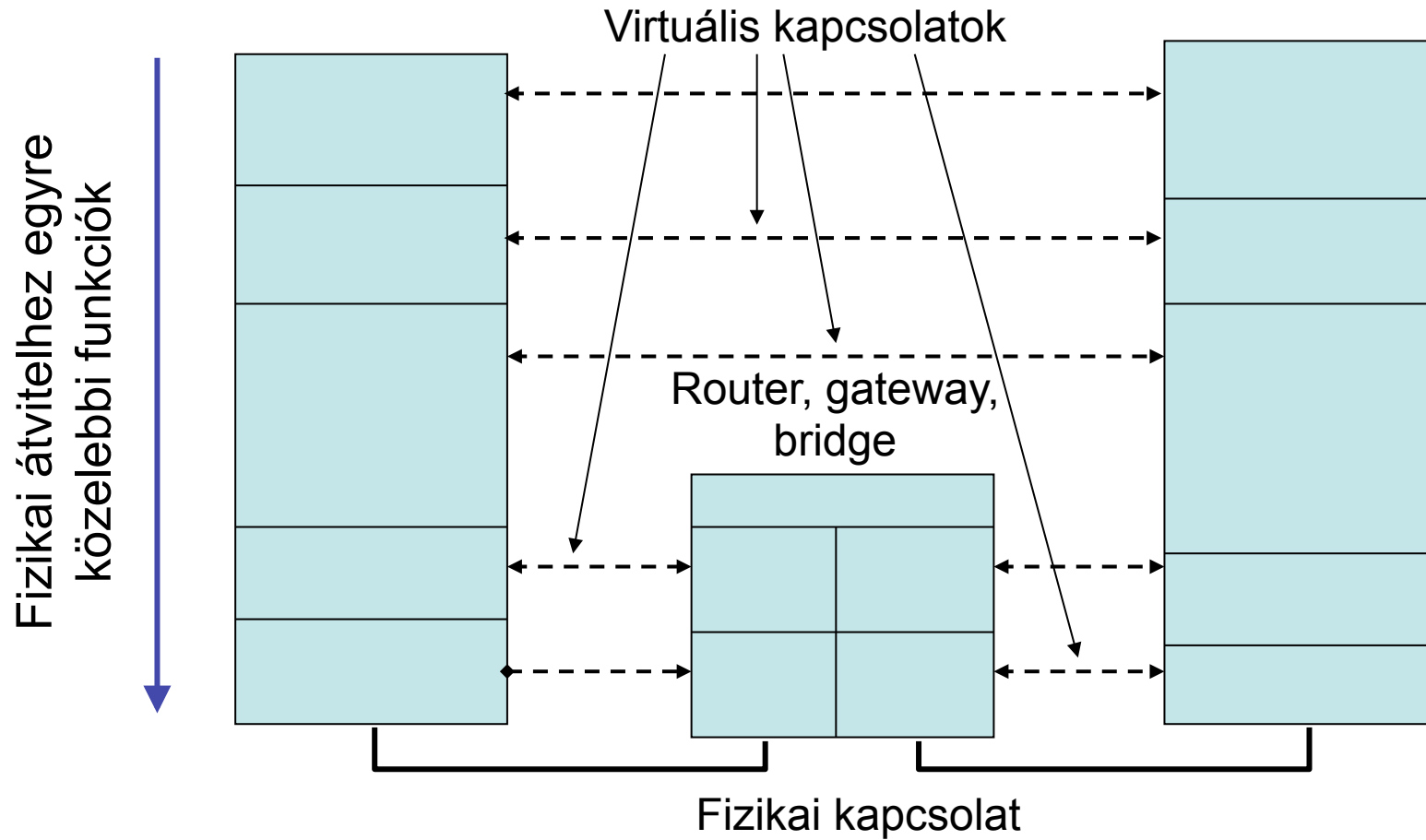


Protokollrétegek a hálózati csomópontokban

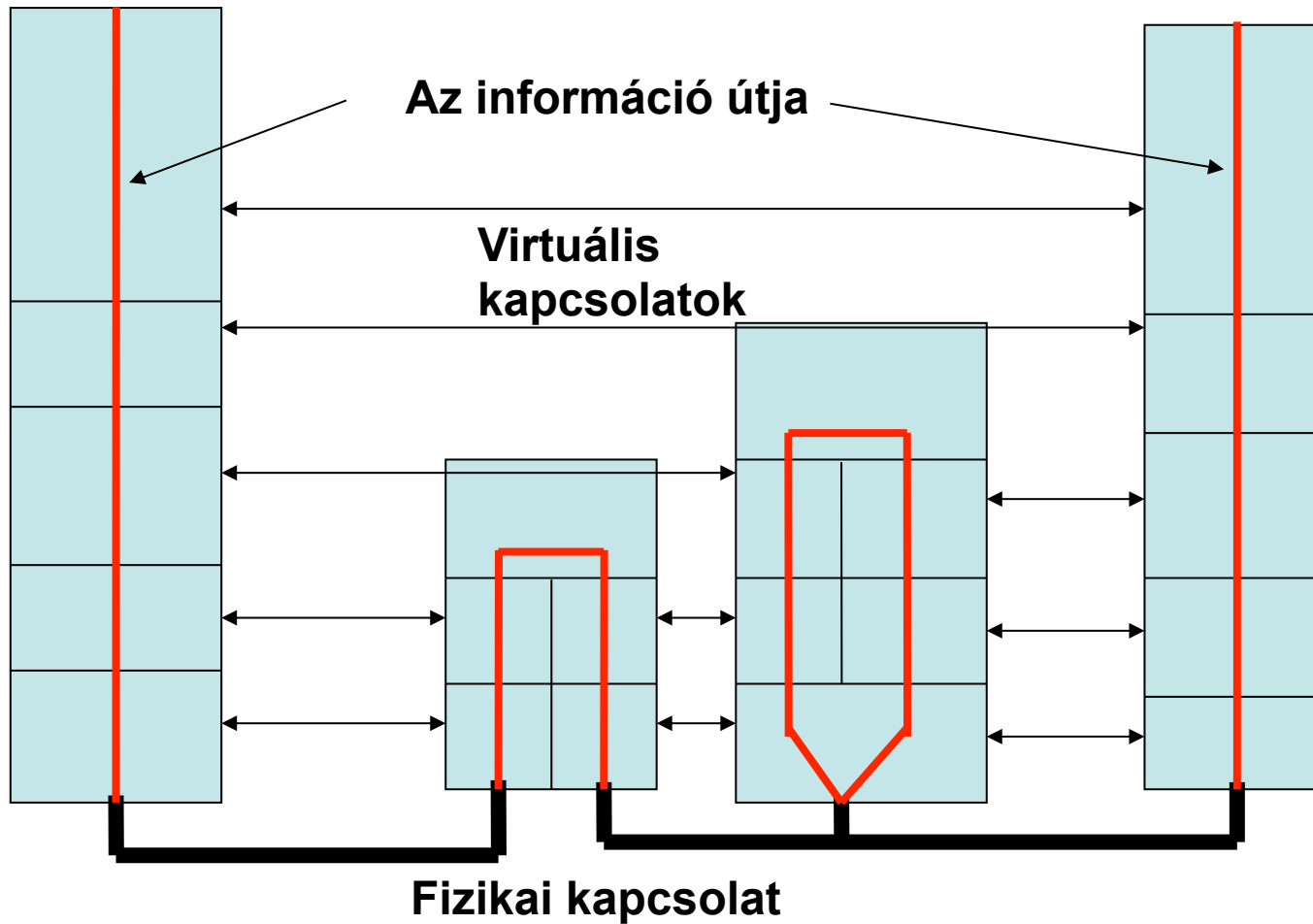


Nem minden rétegre van szükség minden csomópontban!

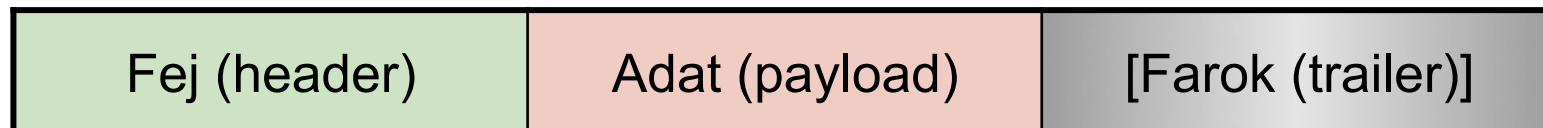
Protokollrétegek (1)



Protokollrétegek (2)

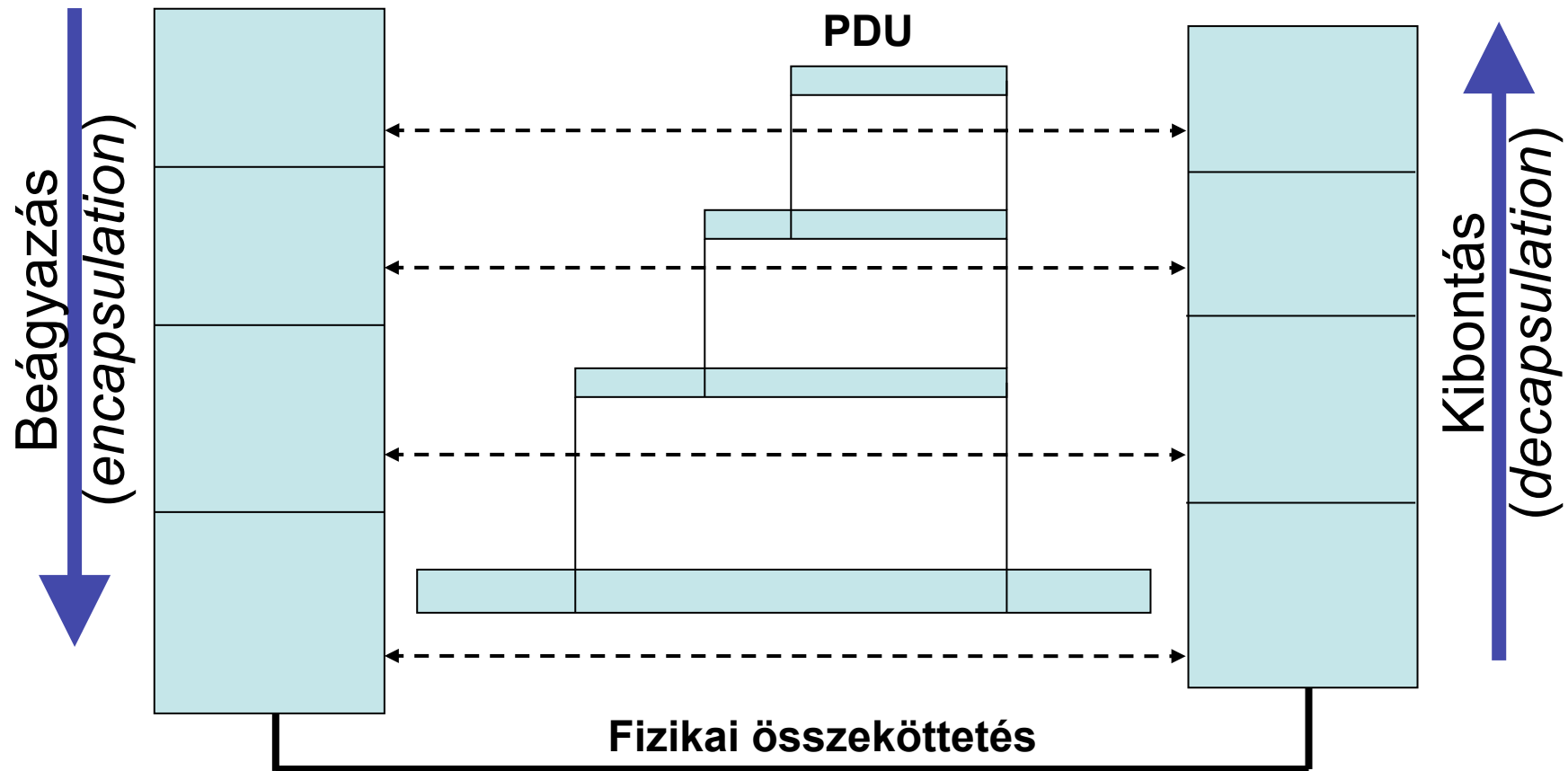


- Ezek kerülnek továbbításra a szomszédos rétegek között
- Általános felépítése, tartalma



- A fejrészben és farokrészben található kiegészítő információk:
 - Címek, sorszámok, hibavédelem, egyéb opcionális paraméterek

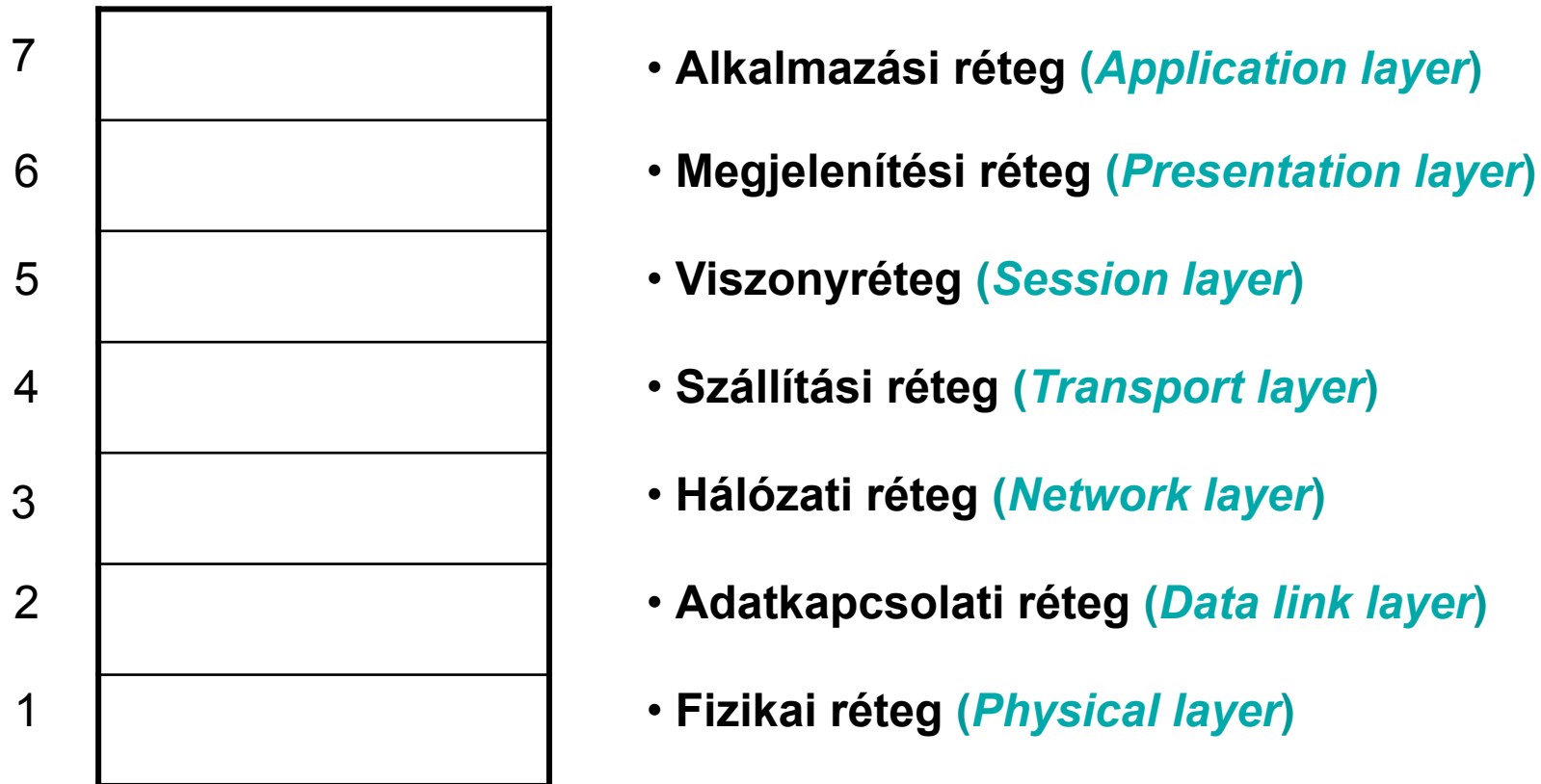
„Borítékolás” - beágyazás (encapsulation)



ISO OSI referenciamodell

ISO - International Organization for Standardization

OSI - Open Systems Interconnection

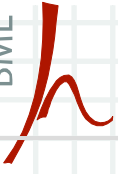


- Szigorúan véve:
 - az adott réteg protokoll-adategysége
 - vannak/voltak „szabványos” elnevezések
 - 2. réteg: keret (*frame*)
 - 3. réteg: csomag (*packet*)
 - 4. réteg: szegmens (*segment*)
 - Feljebb (főként alk. réteg): üzenet (*message*)

- Az egyszerűség kedvéért gyakran: mindent „csomag”-nak nevezünk

Fizikai réteg (KomHálók2)

- A fizikai közeg specifikációja
 - pl. érintkezők kiosztása, használatos feszültség szintek, kábel specifikációk, csatlakozók mérete, típusa stb.
- Bitek, bitcsoportok továbbítása a fizikai csatornán
 - **vonali kódolás**: szimbólumreprezentáció
 - **moduláció**: vivő viszi át az információt
- Számos Ethernet szabvány, pl. külön protokoll a csavart érpárra, külön az üvegszálra



Adatkapcsolati réteg (KomHálók2)

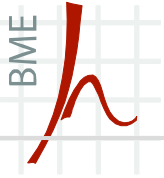
- Bitsorozatok „keretezése”
 - Csomagok összeállítása
- Fizikai címek kezelése (MAC címek)
- Közeghozzáférés vezérlése (Media Access Control)
- Forgalomszabályozás itt is lehet (manapság ritkán)
- Tipikusan itt működik: **bridge, switch**

- Mi innentől tárgyaljuk a KomHáló1 tárgy keretében
- Egyedi linkek logikai összeköttetése **végpontok közötti csatornává** (pl. IP protokoll)
 - Akár több hálózaton át
- **Logikai címezés**: hálózati eszközök közötti kapcsolásokhoz
- **Útvonalkeresés** a hálózaton belül
- Tipikusan itt működnek: **útvonalválasztók (router-ek)**

- Adatkapcsolati: **fizikai címzés**
 - MAC címek: gyártó fixen
- Hálózati: **logikai címzés**
 - hálózattervező határozza meg az értékeket
- Adatkapcsolati réteg egy **hálózaton belül** teremt kapcsolatot, hálózati réteg **több hálózaton át!**

- Végpontok közötti **megbízható** kommunikáció
- A hálózati réteg „nyers” csomagtovábbítása felett:
 - **Hibamentes** összeköttetés létrehozása:
 - Hibás csomagok ismétlése
 - Duplikált csomagok eldobása
 - Csomagsorrend helyreállítása
 - **Forgalomszabályozás**

- **Posta:** kiküldi és rendszerezi a leveleket, csomagokat
- Megbízható szolgáltatást nyújt: nem veszhetnek el levelek következmények nélkül
- Viszont: **csak a borítékát** látja a levélnek, a tartalmát nem



Viszonyréteg

- **Kapcsolat** irányának kezelése
 - Duplexitás kezelése
- Összeköttetés kezelése
 - **Kapcsolat** felépítése, lebontása
- Manapság: általában **a szállítási vagy alkalmazási rétegben** implementálják

Megjelenítési réteg

- Míg az előző rétegek a fejben lévő meta-adatokat kezelik, ez a réteg a **felhasználói adatokat**
- Adatábrázolás
 - az átvitt felhasználói információ szintaktikai ellenőrzése
 - „Syntax réteg”
 - szükség szerinti konvertálása
 - operációs rendszer miatti különbségek
- Adattömörítés
- Adattitkosítás

- Általában **az alkalmazási rétegben van megvalósítva**

- A végpontokon futó **alkalmazási programok**
- Felhasználják az alsóbb rétegek szolgáltatásait, ők maguk csak a felhasználót szolgálják ki

- Fő feladatai:
 - kommunikációs partnerek azonosítása
 - hálózati erőforrások elérhetősége
 - szinkronizáció a felhasználók között
 - formátum, biztonsági egyeztetés

- Implementációi: HTTP, FTP, SMTP

Az OSI referenciamodell szerepe, jelentősége

- Architektúraként **nem valósult meg** teljesen soha
- Ma már a legtöbb OSI-protokoll sincs használatban

- **Referenciamodellként** továbbra is **hasznos**
 - a különböző létező architektúrák megértéséhez, bemutatásához
 - mintegy „mérőléc”-ként használva

- Rétegek és viszonyítás az OSI modellhez



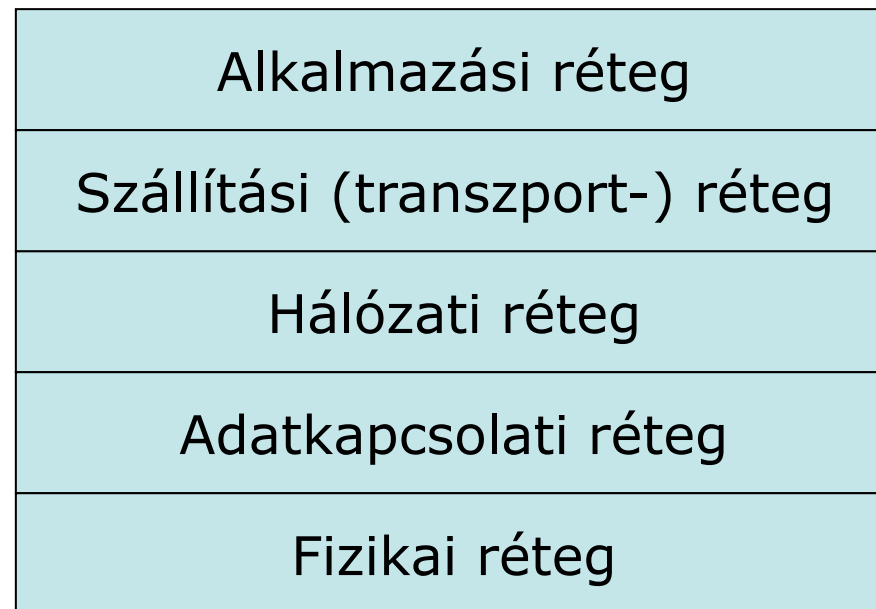
- **Interfész réteg: kommunikáció egy hálózati szegmensen belül**
 - Ethernet
- **Internet réteg: független hálózatokat köt össze**
 - Címzés és routing
 - IP
- **Transzport réteg: processz-processz kommunikáció**
 - Hibellenőrzés, forgalomszabályozás, portkezelés
 - TCP, UDP
- **Alkalmazási réteg: interfész a felhasználóhoz**
 - FTP, SMTP, HTTP

- RFC 3439 kimondja: „A rétegezés károsnak tekinthető!”
- Az IETF által koordinált protokoll-architektúra
 - „We reject: kings, presidents and voting. We believe in: rough consensus and running code.”
- Már az OSI modell előtt használták, Internet világát tükrözi
 - Összeköttetésmentes csomagkommunikáció
- A protokolloknak nem kell megfelelni szigorúan rétegeknek, hogy el legyenek fogadva szabványként
- Az **operációs rendszerek többsége** mind TCP/IP-t használ

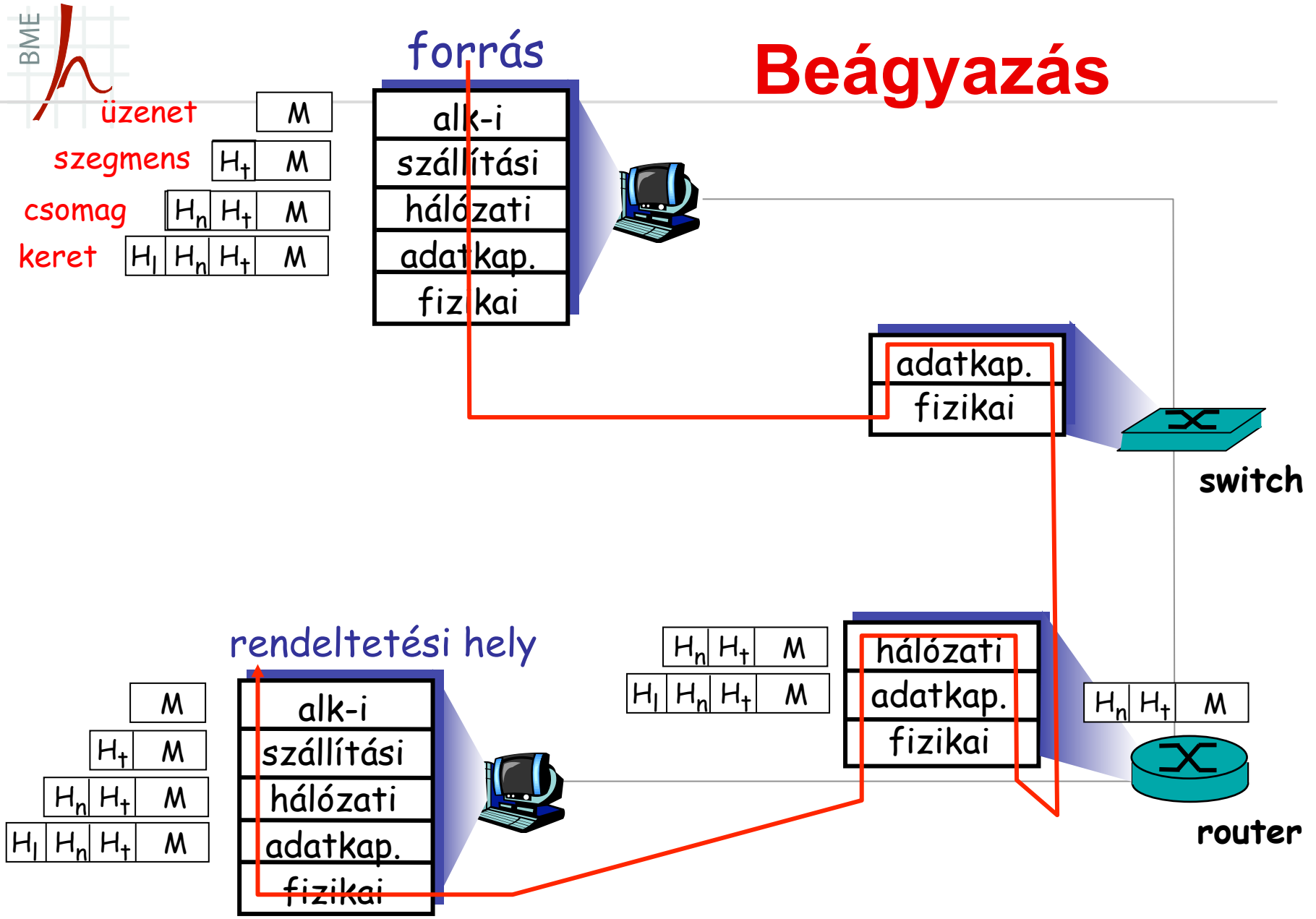
A gyakorlatban jól használható kombinált referenciamodell

[Tanenbaum: Számítógép-hálózatok],
[Kurose-Ross: Computer Networking]

5 réteg, a felső három a TCP/IP-nek, az alsó 2 az OSI-nak felel meg



Beágyazás



Kérdések?

KÖSZÖNÖM A FIGYELMET!



Dr. Simon Vilmos
docens

BME Hálózati Rendszerek és Szolgáltatások Tanszék
svilmos@hit.bme.hu