

*Petrik Lajos Vegyipari, Környezetvédelmi, Informatikai Kéttannyelvű  
Szakközépiskola*

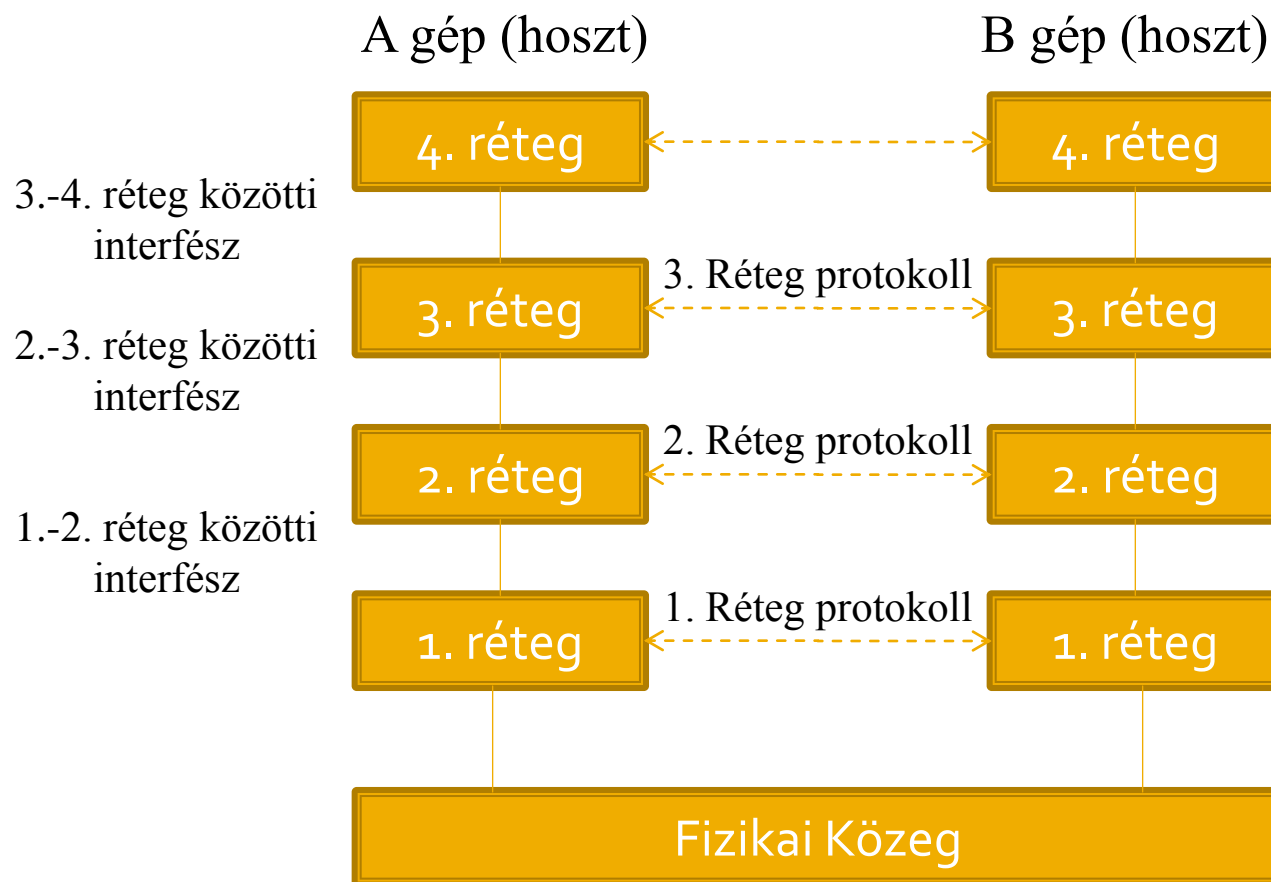
# Hálózatok Elmélet – 13 S

*Györgyi Tamás*

# Hálózati architektúrák

- A számítógép-hálózatok tervezését strukturális módszerrel végzik, azaz a hálózat egyes részeit réteg-ekbe (layer) vagy más néven szint-ekbe (level) szervezik, melyek mindegyike az előzőre épül.

# Általános réteg modell



# Réteg

- Hálózati kapcsolatnál az egyik gép k.-adik rétege a másik gép ugyanilyen szintű rétegével kommunikál.
- Ezt olyan módon teszi, hogy minden egyes réteg az alatta lévő elhelyezkedő rétegnek vezérlőinformációkat és adatokat ad át egészen a legalsó rétegig, ami már a kapcsolatot megvalósító fizikai közegehez kapcsolódik.

# Protokoll

- A kommunikációnál használt szabályok és megállapodások összességét protokollnak (*protocol*) nevezzük.
- A szomszédos rétegek között egy réteginterfész húzódik, amely az alsóbb réteg által a felsőnek nyújtott elemi műveleteket és szolgáltatásokat határozza meg.

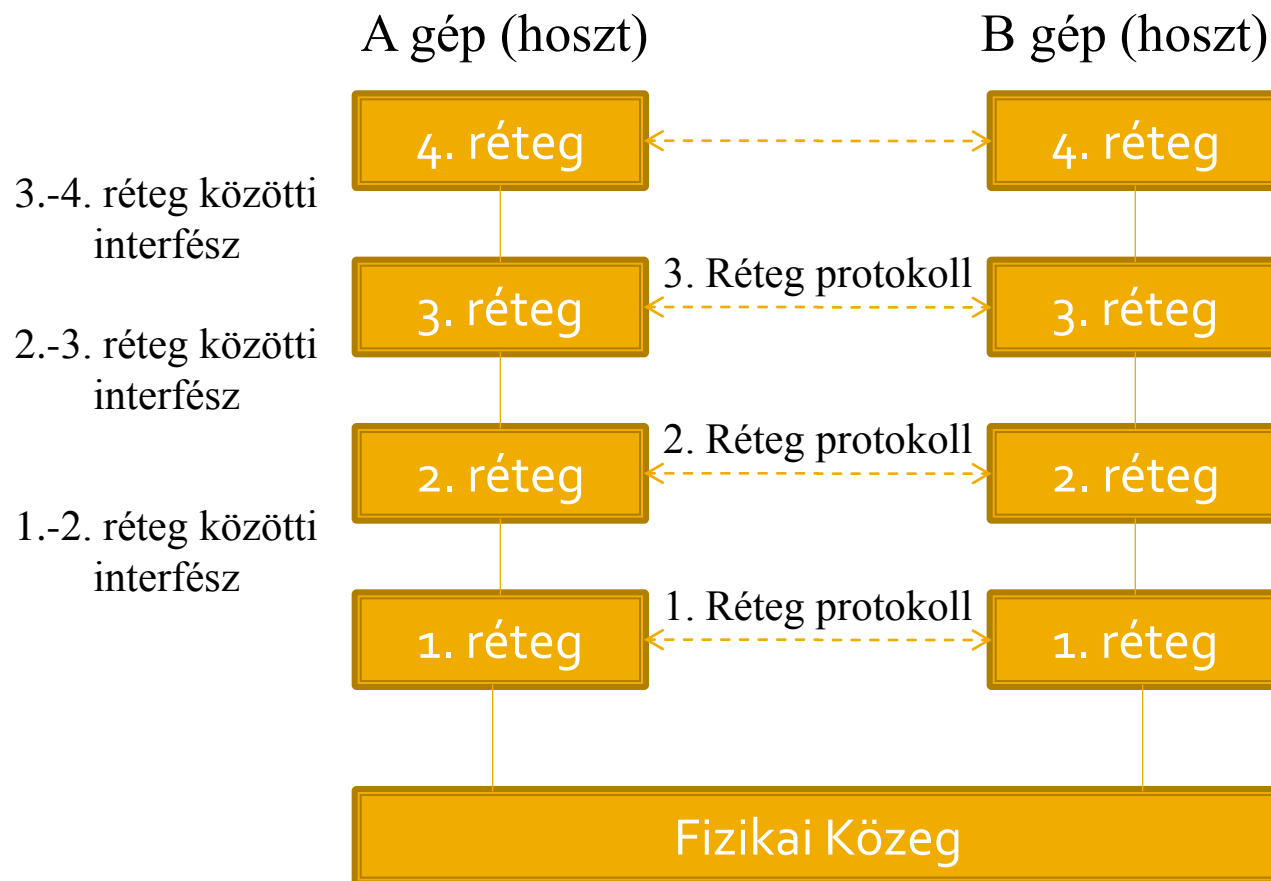
# Interfész

- A legfontosabb, hogy ez az interfész minden réteg között tiszta legyen olyan értelemben, hogy az egyes rétegek egyértelműen definiált funkcióhalmazból álljanak.
- Pl.
  - `Send(in: x,y out: z){ z=3*x-2*y; }`
  - `Send(in: x,y out: z){ z=x+x+x-y-y; }`
  - Valamilyen algoritmus számol, de a külső rész amit látnak az ugyanaz.

# Interfész használatának előnye

- Ez egyszerűvé teszi az adott réteg különböző megoldásainak a cseréjét, hiszen a megoldások az előbbiek alapján ugyanazt a szolgáltatást nyújtják a felettük levő rétegnek.
- Ez által segítve a nyílt rendszerek kialakítását.

# Általános réteg modell



# Hálózati architektúra

- A rétegek és protokollok halmazát nevezzük **hálózati architektúra**-nak.
- Az architektúra kialakításakor meg kell tervezni az egyes rétegeket a bizonyos elvek alapján.

# Tervezési Alapelvek

1. Minden rétegnek rendelkeznie kell a kapcsolat felépítését illetve annak lebontását biztosító eljárással.
2. Döntést kell hozni az adatátvitel szabályairól:
  1. az átvitel egyirányú (szimplex)
  2. váltakozóan két irányú (fél duplex)
  3. egyszerre két irányú (duplex)

# Szimplex

- Szimplex átvitel esetén a csatornán áramló információ csak egy irányú lehet, mindig van adó és van vevő a rendszerben, ezek szerepet nem cserélnek.
- Példa: (nem tudunk visszabeszélni...)
  - Rádió
  - TV adás

# Fél Duplex

- Fél duplex átvitel esetén a csatornán az információáramlás már kétirányú, felváltva történik, úgy hogy egyszerre mindig csak az egyik irány foglalja a csatornát.
- Pl.
  - Ilyen átvitel valósul meg nagyon sok rádiós kapcsolatban (pl. CB rádió)

# Duplex

- Duplex átvitel esetén egyidejű két irányban történő átvitel valósul meg.
- Pl.
  - emberi beszélgetés,
  - telefont

# Tervezési Alapelvek

3. Milyen legyen a rendszerben a hibavédelem, hibajelzés.
4. Hogyan oldható meg a gyors adók-lassú vevők együttműködése (ez a folyamat vezérlés = **flow control**).
5. Ha bizonyos okok miatt az üzenetek hossza korlátozott, és ezért a küldés előtt szét kell darabolni, felmerül a kérdés, hogy hogyan biztosítható a helyes összerakásuk.

# Tervezési Alapelvek

6. Az előbbi esetben biztosított-e az üzenetek sorrendjének a helyessége?
7. Nagyon sokszor ugyanazon a fizikai csatornán több párbeszéd zajlik. (Ez jobb vonalkihasználást eredményez. Hogyan kell ezt összekeveredés mentesen megoldani?)
8. Ha a cél és a forrás között több útvonal lehetséges, fontos a valamilyen szempontból optimális útvonal kiválasztása.

# Tervezés

- Ilyen és ehhez hasonló kérdésekre kell választ adni a tervezés során.
- Ezekre a kérdésekre nincs együttesen optimális válasz, ezért van az a keletkezett megoldások sokszínűek. Mindegyik mellett lesz érv és ellenérv is.

# Hálózatszabványosítás

- A hálózatok kialakításában alapvető szerepet játszik a szabványosítás.
- A szabványok központi szerepet játszanak a fejlődésben, ez teszi a rendszereket nyíltakká, egységeit cserélhetővé.
- Minden új dolog kialakulását megelőzi a kutatás, az ehhez kapcsolódó írásos és szóbeli információcserék (cikkek, konferenciák).

# Szabványosítás

- Felmerül a kérdés, hogy mikor célszerű az új dolgokkal kapcsolatos információhalmazt a szabványok által meghatározott útra terelni?!

# Szabványosítás Korán

- Ha ez a kutatási szakaszban következik be, ez azt jelenti, hogy az esetleg a még nem alapos ismeretek miatt a szabvány nem lesz megfelelő, mivel az új, későbbi kutatási eredményeket már nem lehet beilleszteni, kedvező megoldásokat kell elhagyni.

# Szabványosítás Későn

- Ha viszont túl későn következik be a szabványosítás, akkor a gyakorlatban már számos egymástól eltérő megoldás kerül megvalósításra, ami az ellenérdekek miatt nehezzé teszi az egységességet igénylő szabványosítást.

# Szabványok

- Ezért a gyakorlatban a szabványok két családja létezik:
  - a de-jure szabványok, amelyeket bizottságok deklarálnak, és hivatalos dokumentumokban rögzítenek
  - a de-facto szabványok, amelyek elterjedését már egy-egy konkrét megoldás széleskörű használata biztosítja.

# Hálózatok Szabványosítása

- A számítógép-hálózatok esetében sem volt másképp.
- Megjelenésükkor néhány vezető cég termékeivel de-facto szabványokat teremtett, és a későbbi ezeket figyelembe vevő de-jure szabványosítási törekvések kompromisszumos megoldásokat eredményeztek.

# OSI Modell

- A számítógép-hálózatok rétegezett struktúrájú modell segítségével írhatók le.
- A Nemzetközi Szabványügyi Szervezet, az ISO (*International Standard Organization*) kidolgozott egy olyan modell-ajánlást (*nem szabványt!*) amelyet ma már minden hálózati rendszer tervezésekor követnek.

# OSI Modell

- Az **OSI** az **Open System Interconnect** - nyílt rendszerek összekapcsolása kifejezés angol eredetijéből alkotott betűszó.
- *Nyílt rendszereknek olyan rendszereket hívjuk amelyek nyitottak a más rendszerekkel való kommunikációra.*
- Az OSI modell hét rétegből áll.

# OSI modell kialakulásának elvei

- Minden réteg feladata jól definiált legyen, és ez a nemzetközileg elfogadott szabványok figyelembe vételével történjen.
- A rétegek közötti információcsere minimalizálásával kell a rétegek határait megállapítani.
- Elegendő számú réteget kell definiálni, hogy a különböző feladatok ne kerüljenek feleslegesen egy rétegbe.

# OSI Modell 7 rétege

- A modell alsó három rétege a hálózattól függ.
- A felső négy réteg mindig alkalmazásfüggő, és mindig az alkalmazást futtató hosztokban történik a megvalósításuk (implementálásuk).
- A legalsó szinttől felfelé haladva a rétegek következnek.

# 1. Fizikai réteg (*physial layer*)

- Valójában ezen a rétegen zajlik a tényleges fizikai kommunikáció.
- Biteket juttat a kommunikációs csatornára, olyan módon, hogy az adó oldali bitet a vevő is helyesen értelmezze ( a 0-át 0-nak, az 1-et, 1-nek).

# 1. Fizikai réteg (*physial layer*)

- A fizikai közeg, és az információ tényleges megjelenési formája igen változó lehet:
  - elektromos vezeték esetén, a rajta lévő feszültség értéke, vagy a feszültség változásának iránya.
- Információhordozó és közeg más és más lehet még
  - fénykábel, rádióhullám, stb.
- Itt kell azt is meghatározni, hogy mennyi legyen egy bit átvitelének időtartama

# 1. Fizikai réteg (*physial layer*)

- Egy vagy kétirányú kapcsolat.
  - A kétirányú kapcsolat egyszerre történhet-e?
- Hogyan épüljön fel egy kapcsolat és hogyan szűnjön meg.
- Milyen legyen az alkalmazott csatlakozó fizikai, mechanikai kialakítása?

## 2. Adatkapcsolati réteg (*data link layer*)

- Feladata adatok megbízható továbbítása az adó és fogadó között. Működése:
  - az átviendő adatokat **adatkeretekké (data frame)** (bitcsoportba kódolt forma) tördeli
  - ellátja kiegészítő cím, ellenőrző és egyéb információval
  - ezeket sorrendhelyesen továbbítja
  - majd a vevő által visszaküldött **nyugtakereteket (acknowledgement frame)** véve ezeket feldolgozza.

## 2. Adatkapcsolati réteg (*data link layer*)

- Hogyan jelezzük a keretek kezdetét és a végét?
- Mi történjék akkor ha egy keret elveszik?
- Mi történjék akkor ha a nyugtakeret vész el?
  - Ilyenkor, ha az adó újra adja, kettőzött keretek jelennek meg a rendszerben.
- Mi legyen akkor, ha az adó adási sebessége jelentősen nagyobb, mint a vevőké?

## 2. Adatkapcsolati réteg (*data link layer*)

- Ha a csatorna kétirányú adatátvitelre használt, felmerülhet problémaként, hogy mennyire legyen szimmetrikus a két különböző irányban történő adatátvitel, és ezt milyen megoldással lehet biztosítani azt, hogy az egyik irányú átvitel ne kerüljön túlsúlyba.

### 3. Hálózati réteg (*network layer*)

- Lényegében a kommunikációs alhálózatok működését vezérli.
- Nagyobb hálózatok esetén a keretek vevőtől a célba juttatása elvileg több útvonalon is lehetséges, feladat a bizonyos szempontból optimális útvonalnak a kiválasztása.
  - Ez a tevékenység az útvonalválasztás (routing), és több megoldása lehetséges.

# Routing

- A rendszer kialakításakor alakítjuk ki az útvonalakat.
- A kommunikáció kezdetén döntünk arról, hogy a teljes üzenet csomagjai milyen útvonalon jussanak el a rendeltetési helyükre.
- Csomagonként változó, a hálózat vonalainak terhelését figyelembe vevő alternatív útvonalválasztás lehetséges.

### 3. Hálózati réteg (*network layer*)

- Itt kell megoldani a túl sok csomag hálózatban való tartózkodása okozta torlódást
- Különböző (heterogén) hálózatok összekapcsolását is a feladata közé tartozik.

## 4. Szállítási réteg (*transport layer*)

- Feladata a hosztok közötti átvitel megvalósítása.
- A kapott adatokat szükség esetén kisebb darabokra vágja, átadja a hálózati rétegnek.
- Fontos része a címzések kezelése.
- Forrás-cél összeköttetések létrehozása a névadási mechanizmus felhasználásával.

## 5. Együttműködési réteg (*session layer*) (*Viszony réteg*)

- A különböző gépek felhasználói viszonyt létesítenek egymással, például bejelentkezés egy távoli operációs rendszerbe, állománytovábbítás két gép között.
- Átvitt adatfolyamokba szinkronizációs ellenőrzési pontok beiktatása.
  - Ez azt biztosítja, hogy hosszú átvitt adatfolyam átvitele alatt bekövetkező hiba esetén elegendő az utolsó ellenőrzési ponttól ismételni az elvesztett adatokat.

## 6. Megjelenítési réteg (*presentation layer*)

- Feladata az adatok egységes kezelése.
- A legtöbb alkalmazói program nem egy csupán egy bitfolyamot, hanem neveket, dátumokat, szövegeket küld.
- Ezeket általában adatstruktúrákban ábrázolják.
- A kódolás sem minden esetben egységes, pl. a karakterek kódolására az ASCII mellett az EBCDIC kód is használt.

## 6. Megjelenítési réteg (*presentation layer*)

- Más lehet egy több bájtos kód esetén az egyes bájtok sorrendje.
- Ezért egységes, absztrakt adatstruktúrákat kell kialakítani, amelyek kezelését a megjelenítési réteg végzi.
- További, e réteg által kezelt vonatkozások:
  - az adattömörítés
  - átvitt adatok titkosítása

## 7. Alkalmazási réteg (*application layer*)

- Ez kapcsolódik legszorosabban a felhasználóhoz, itt kell a hálózati felhasználói kapcsolatok megoldásait megvalósítani.
- Számos termináltípust használnak a hálózati kapcsolatokban, amelyek természetesen kisebb-nagyobb mértékben egymástól eltérnek, ezért egy hálózati virtuális terminált definiálnak, és a programokat úgy írják meg, hogy ezt tudja kezelni.

## 7. Alkalmazási réteg (*application layer*)

- A különböző típusú terminálok kezelését ezek után egy kis programrészlet végzi.
- Másik tipikus, e réteg által megvalósítandó feladat
  - a fájlok átvitelekor az eltérő névkonvenciók kezelése,
  - az elektronikus levelezés,
  - és mindazon feladat, amit internet szolgáltatásként ismerünk.