

Tokaj-Hegyalja  
Egyetem

# Hálózati Architektúrák és Protokollok

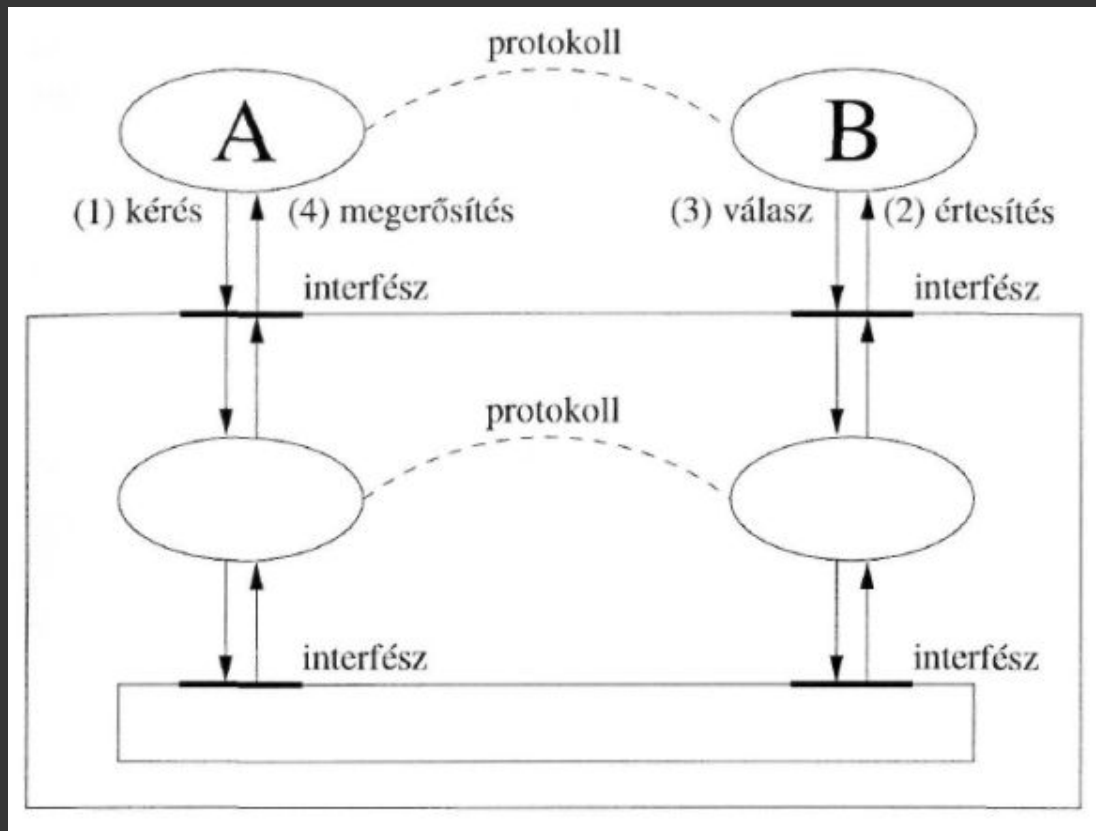
3. Hálózati rétegek, OSI / TCP IP modell

---

# Protokollhierarchiák

- Ahhoz, hogy két egymástól távoli számítógép kommunikálni tudjon egymással, számos feladatot kell megoldani.
- **A bonyolultság csökkentése érdekében a megoldandó feladatot rétegekre bontással strukturálhatjuk**
- A rétegek száma, elnevezése, tartalma és feladata más és más a különböző hálózatokban
- **Rétegek célja:**
  - bizonyos szolgáltatásokat nyújtson a felette elhelyezkedő rétegeknek,
  - miközben elrejtje előlük a szolgáltatások tényleges megvalósításának részleteit.
- **Rétegek értelmezése:**
  - minden réteg egy olyan virtuális gép, amely a felette levő rétegek számára szolgáltatásokat nyújt.
  - Egy adott réteg vizsgálatakor az alatta levő rétegeket fekete doboznak tekinthetjük:
    - nem foglalkozunk azzal, hogy milyen a belső működése, számunkra csak az érdekes, hogy hogyan viselkedik.

# Protokollhierarchiák



# Protokollhierarchiák

## Folyamat

- A rajz két oldalán található „A” és „B” entitások, vagy processzek egymással kommunikálni akarnak.
- Az „A” kérést intéz az alsóbb réteghez, amely számára fekete dobozként látszik, eltakarva a belső működését.
- A kérést ez a réteg valamilyen formában eljuttatja „B”-hez.
- „B” egy választ küld szintén az alsóbb réteget használva, mely „A” felé egy megerősítést továbbít.
- „A” kommunikált „B”-vel. A kommunikáció „nyelve” a **protokoll**, amelyet mindkét entitás ért.
  - **Fizikailag nem egymással „beszélnek”, hiszen nem „látják” egymást,**
  - mindketten az alsóbb szintet kérik meg a kommunikáció lebonyolítására.

# Protokollhierarchiák

- Az **entitás** (vagy más kifejezéssel: processz) az egyes rétegekben lévő aktív cselekvő. Az azonos rétegben lévő entitásokat társentitásoknak (peer entities) nevezzük.
- A **protokoll** a különböző gépeken futó n-edik szintű (= n-edik rétegben elhelyezkedő) entitások (vagy processzek) egymással való kommunikációja során használt szabályok és konvenciók összessége.
- A **szolgáltatás** elemi műveletek (primitívek) halmazával írható le, amelyek a szolgáltatást elérhetővé teszik a felhasználó vagy más entitások számára.

# Protokollhierarchiák

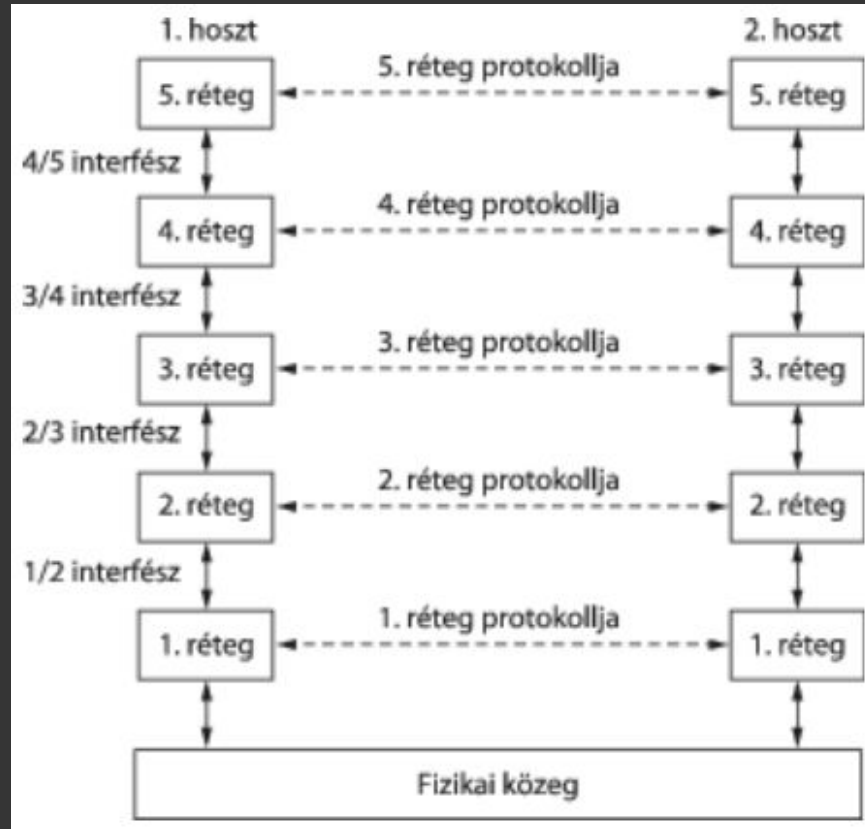
## Interfész:

- Az interfész azt definiálja, hogy az alacsonyabban levő réteg milyen elemi műveleteket és szolgáltatásokat nyújt a magasabban levő réteg számára.
- **A hálózattervezők eldöntik:**
  - hány réteget tartalmazzon egy hálózat,
  - mi legyen az egyes rétegek feladata
- A legfontosabb szempont az, hogy a rétegek közötti interfész minél világosabb legyen.
- **Szükséges:**
  - minden réteg jól definiált feladatokkal rendelkezzen.
  - a rétegek közötti információcserét minimalizálva legyenek
- Lehetővé teszi a réteg tényleges megvalósítását lecseréljük egy teljesen újra

# Protokollhierarchiák

- A rétegek és protokollok halmazát **hálózati architektúrának (network architecture)** nevezzük.
- Az architektúra specifikációjának elegendő információt tartalmaz:
  - hogy az implementálást végző szakember minden réteghez meg tudja írni a programot,
  - illetve meg tudja építeni a hardvert úgy, hogy az helyesen alkalmazza a megfelelő protokollt.
- Az implementáció részletei és az interfészek specifikációja nem része az architektúrának
  - mivel ezek a gép belsejében rejtve maradnak, tehát kívülről nem láthatók.

# Protokollhierarchiák



**Rétegek tervezési kérdései...**

# Rétegek tervezési kérdései

- Megbízhatóság kérdésköre:

- a hálózat annak ellenére is helyesen működjön, hogy önmagukban megbízhatatlan alkotóelemekből épül fel
  - egy hálózaton keresztül utazó csomag bitjei:
    - A bitek sérülve érkehetnek meg: oka lehet véletlenszerű elektromos zaj, kiszámíthatatlan vezeték nélküli jelek, illetve hardver- vagy szoftverhibák
- A fogadott adatok hibáinak észrevételére, azaz a **hibajelzésre (error detection)** az egyik lehetséges módszer a kódolás használata.
- A hibásan vett adatokat újra lehet küldeni, mígnem egyszer csak helyesen érkeznek meg.
- Hathatósabb kódolás révén **hibajavítás (error correction)** is lehetséges
  - a helyes üzenet helyreállítható a ténylegesen fogadott, esetlegesen hibás bitekből.
  - Mindkét eljárás redundáns információ hozzáadásával működik

# Rétegek tervezési kérdései

- Routing:

- másik megbízhatósági kérdés a működő útvonalak megtalálása a hálózatban.
- Gyakran több lehetséges útvonal is létezik egy forrásállomás és egy célállomás között,
  - ráadásul egy nagy hálózatban lehetnek meghibásodott kapcsolatok vagy útválasztók.
  - Pl. Németországban épp nem működik a hálózat. Ha Londonból Rómába Németországon keresztül küldenénk csomagokat, azok nem érnének célba,
    - de ehelyett elküldhetnénk azokat Londonból Rómába Párizson keresztül is.
  - A hálózatnak automatikusan kell meghoznia ezeket a döntéseket.
  - Ezt a témakört **útválasztásnak (routing)** nevezzük.

# Rétegek tervezési kérdései

- Protokoll:

- Mint második tervezési problémakör a hálózat fejlődésével foglalkozik
- Idővel a hálózatok nagyobbra nőttek, és újfajta elemeket kell a meglévő hálózathoz csatlakoztatni.
- A strukturálási mód, amely a teljes probléma felosztásával és a megvalósítás részleteinek elrejtésével segíti a változást:
  - ez a **protokollok rétegezése (protocol layering)**.
- Mivel a hálózatokban általában sok számítógép van összekötve, minden rétegben kell lennie egy olyan mechanizmusnak, amely egy üzenet küldőjét és vevőjét azonosítja.
  - Ezt a mechanizmust **címzésnek (addressing)** nevezik az alsóbb, illetőleg **névkezelésnek (naming)** a felsőbb rétegek esetében.

# Rétegek tervezési kérdései

- A növekedés egyik velejárója az, hogy a különböző hálózati technikák gyakran más-más korlátokkal rendelkeznek
- Pl. nem minden kommunikációs csatorna tartja meg a rajtuk elküldött üzenetek eredeti sorrendjét,
  - ezért olyan megoldásokat kellett bevezetni, melyek sorszámozzák az üzeneteket.
- Egy másik példa a hálózatok által továbbítható üzenetek méretkorlátai közötti eltérés.
  - Ez olyan eljárásokat igényel, melyek szétbontják, továbbítják, majd újra összerakják az üzeneteket.
  - A téma összefoglaló neve **internetworking**, azaz a hálózatok összekapcsolásával foglalkozó terület.

# Rétegek tervezési kérdései

- Amikor a hálózatok nagyon nagyra nőnek, új problémák kerülnek elő.
- A városokban kialakulhatnak forgalmi dugók, elfogyhatnak a telefonszámok.
- Nem sok embernek okoz ez problémát a saját körzetében, de városi léptékben gondolkodva már nagy gondot okozhat.
- Azokat a kialakításokat, melyek akkor is jól működnek, amikor egy hálózat nagyra növekszik, **skálázhatónak (scalable)** nevezzük.

# Rétegek tervezési kérdései

- **Erőforrások kiosztás:**

- A hálózatok az alattuk lévő erőforrások felhasználása révén szolgálják a hosztokat.
  - Ilyen erőforrás például az átviteli vonalak kapacitása.
- Olyan mechanizmusokra van szükségük, melyekkel úgy osztják fel az erőforrásaikat, hogy **az egyik hoszt ne nagyon zavarja a másikat**.
- Sok konstrukcióban a sáv szélesség kiosztása dinamikusan, a hosztok rövid távú szükségletei szerint történik
  - ahelyett, hogy minden hoszt a teljes sáv szélességnek egy rögzített hányadát kapná, melyet vagy kihasznál, vagy nem.
  - Ez a működési mód: statisztikai multiplexelés
    - az elosztás az igényekre vonatkozó statisztikák alapján történik.
    - Ez a megoldás az alsóbb rétegekben alkalmazható egyetlen kapcsolatra,
    - vagy a felsőbb rétegekben a teljes hálózatra, vagy akár a hálózatot használó alkalmazásokra is.

# Rétegek tervezési kérdései

- Erőforrások kiosztás:

- Minden rétegben felmerülő erőforrás-kiosztási probléma:
  - **hogyan akadályozzuk meg azt, hogy a gyorsabban adó gépek elárasszák adatokkal a lassabban vevőket.**
  - Ezek a módszerek gyakran a vevő és az adó közötti visszacsatoláson alapulnak.
  - Ez a témakör: **forgalomszabályozás**
- Máskor pedig:
  - túl sok számítógép kíván túl sok forgalmat átküldeni, és a hálózat nem képes mindent kézbesíteni.
  - A hálózat ilyen túlterhelődését **torlódás**nak (congestion) hívjuk
  - Az egyik követhető stratégia:
    - minden számítógép csökkentse az átviteli igényét, amikor torlódást tapasztal.
    - Ez a módszer is használható minden rétegben

# Rétegek tervezési kérdései

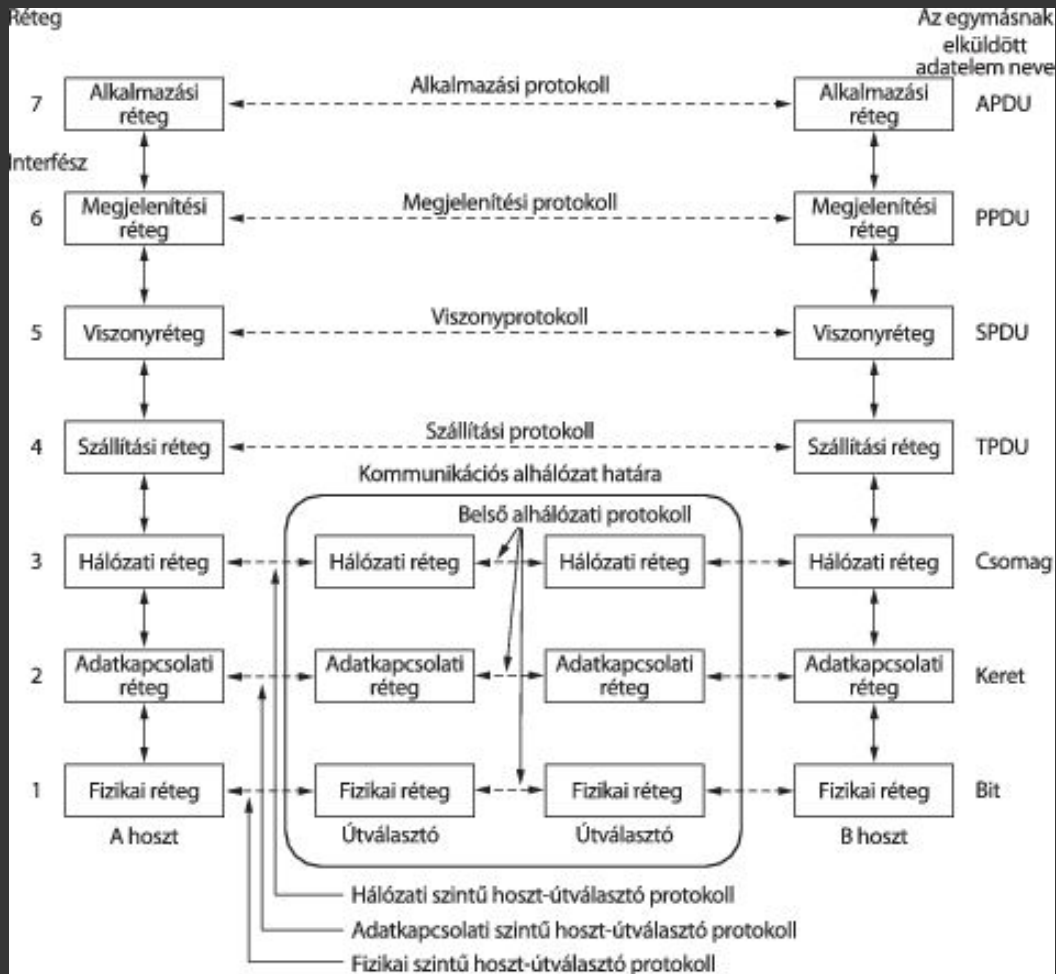
- Hálózat biztonsága:

- Az egyik fenyegetés a **kommunikáció lehallgatása**.
- A **titkosságot** (confidentiality) nyújtó technikák védenek ez ellen a veszély ellen, több rétegben is használjuk ezeket.
- A **hitelesítési (authentication)** eljárások akadályozzák meg, hogy valaki egy másik szereplőt személyesítsen meg.
- Más mechanizmusok a **sértetlenséget** (integrity) védik az üzenetek titokban történő módosítása ellen

**OSI modell...**

# OSI hivatkozási modell

- Az **OSI modell** a Nemzetközi Szabványügyi Szervezet (ISO – International Organization for Standardization) ajánlása.
- Ezt a modellt hivatalosan OSI (Open System Interconnection = nyílt rendszerek összekapcsolása) hivatkozási vagy referenciamodellnek nevezik
- Egy olyan ajánlást definiál, amelynek megfelelő rendszerek egymással elvileg képesek együttműködni.
- **Nyílt rendszerek összekapcsolásával foglalkozik.**
- A nyílt rendszerek olyan rendszerek, amelyek képesek más rendszerekkel való kommunikációra.



# OSI hivatkozási modell

- **Az OSI-modellnek hét rétege van**
- A hét rétegre történő felosztás elvei a következők voltak:
  - A rétegek különböző absztrakciós szinteket képviseljenek
  - Minden réteg jól definiált feladatot hajtson végre
  - A rétegek feladatának definiálásakor a nemzetközileg szabványosított protokollokat kell figyelembe venni.
  - A rétegek határait úgy kell meghatározni, hogy a rétegek közötti információcsere minimális legyen.
  - A rétegek számának elég nagyoknak kell lenni ahhoz, hogy eltérő feladatok ne kerüljenek szükségtelenül ugyanabba a rétegbe,
    - viszont elég kicsinek kell lennie ahhoz, hogy az architektúra ne váljon kezelhetetlenné

# OSI hivatkozási modell

- **Az OSI-modell nem hálózati architektúra**, mivel nem specifikálja az egyes rétegek által használt szolgáltatásokat és a protokollokat
- Csak annyit mond, hogy mit kell tenniük a rétegeknek
- Az ISO az egyes rétegekhez szabványokat is kidolgozott, azonban magának a hivatkozási modellnek ezek nem részei
- Viszont valamennyit közzétették, mint különálló nemzetközi szabványt
- A modellt (részleteiben) széles körben használják, annak ellenére, hogy a kapcsolódó protokollok már rég feledésbe merültek

# OSI hivatkozási modell

## Fizikai réteg:

- A fizikai réteg (physical layer) feladata az, hogy továbbítsa a biteket a kommunikációs csatornán.
- A rétegnek biztosítania kell azt, hogy az egyik oldalon elküldött 1-es bit a másik oldalon is 1-esként érkezen meg, ne pedig 0-ként.
- Ez a réteg tipikusan olyan kérdésekkel foglalkozik:
  - mekkora feszültséget kell használni a logikai 1, és mekkorát a logikai 0 reprezentálásához,
  - mennyi ideig tart egy bit továbbítása,
  - az átvitel megvalósítható-e egyszerre mindkét irányban,
  - miként jön létre az összeköttetés, hogyan bomlik le, ha már nincs szükség rá,

# OSI hivatkozási modell

- hány érintkezője van a hálózati csatlakozóknak,
- mire lehet használni az egyes érintkezőket stb.
- A tervezési szempontok:
  - főleg az interfész mechanikai, elektromos és eljárási kérdéseire, valamint a fizikai réteg alatt elhelyezkedő fizikai átviteli közegre vonatkoznak.
- **Az átvitel adataegysége a bit vagy szimbólum**

# OSI hivatkozási modell

## Az adatkapcsolati réteg: data link layer

- fő feladata az, hogy a fizikai átviteli rendszer szerény adottságait egy olyan vonallá alakítsa, amely a hálózati réteg számára felderítetlen átviteli hibáktól mentesnek látszik.
- Ezt a tényleges hibák elfedésével valósítja meg, hogy a hálózati réteg már ne lássa azokat.
- Ezt a feladatot úgy oldja meg, hogy az átviendő adatokat küldő fél oldalán **adatkeret**ekbe (data frame) tördeli és ezeket sorrendben továbbítja:
  - általában néhány száz vagy néhány ezer bájt
  - Ha a szolgáltatás megbízható, a fogadó fél egy **nyugtázókerettel** (acknowledgement frame) nyugtázza minden egyes keret helyes vételét.

# OSI hivatkozási modell

## Az adatkapcsolati réteg:

- **Felmerül az a kérdés:** hogyan lehet megelőzni azt, hogy egy gyors adó annyi csomaggal árásszon el egy lassú vevőt, amennyit az már nem képes fogadni.
- Valamilyen forgalomszabályozó eszközre van szükség ahhoz, hogy az adót tájékoztatni lehessen arról, ha a vevő készen áll további adatok fogadására.
- Az adatszóró hálózatok adatkapcsolati rétegében még egy dolgot szabályozni kell:
  - az osztott csatornához való hozzáférést.
  - Az adatkapcsolati réteg egy alrétege foglalkozik ezzel a feladattal, amelyet **közeghozzáférés-vezérlő alréteg**nek (medium access control sublayer) neveznek.
- A kommunikáció adataegysége tehát a keret.

# OSI hivatkozási modell

## Hálózati réteg:

- A hálózati réteg (network layer) az alhálózat működését irányítja.
- A legfontosabb kérdés itt az, hogy milyen útvonalon kell a csomagokat a forrásállomástól a célállomásig eljuttatni.
- Az útvonalak meghatározása:
  - történhet statikus táblázatok felhasználásával, amelyeket „behuzaloznak” a hálózatba, és csak nagyon ritkán változtatnak.
  - Az útvonalat minden egyes párbeszéd (például terminálviszony) előtt külön is meghatározhatjuk.
  - Végül az útvonal kiválasztása lehet kifejezetten dinamikus:
    - ilyenkor minden csomag számára a hálózat aktuális terhelésének ismeretében egyenként kerül kijelölésre az útvonal.

# OSI hivatkozási modell

## Hálózati réteg:

- Ha egyszerre túl sok csomag van jelen az alhálózatban, akkor azok akadályozhatják egymást, és ekkor torlódások alakulhatnak ki.
- Az ilyen torlódások kezelése is a hálózati réteg feladatai közé tartozik, a felsőbb rétegekkel együttműködve,
  - melyek a hálózati terhelésüket alakítják a helyzetnek megfelelően.
- Még általánosabban: a nyújtott szolgáltatásminőség (késleltetés, átviteli idő, sebesség-ingadozás stb.) szintén a hálózati réteg hatáskörébe tartozik.

# OSI hivatkozási modell

## Hálózati réteg:

- Ha viszont a csomagnak az egyik hálózatból át kell mennie egy másikba ahhoz, hogy elérje a címzett állomást, akkor még több probléma jelentkezik:
  - az első hálózatban használt címezési mód más, mint a második hálózatban;
  - a második hálózat egyáltalán nem fogadja a csomagot, mert az túl hosszú;
  - a két hálózat protokollja különbözik és így tovább.
- A hálózati rétegnek a feladata az, hogy legyőzze ezeket az akadályokat, és lehetővé tegye az egymástól eltérő hálózatok összekapcsolását.
- **Az adatszóró hálózatokban az útvonalválasztás viszonylag egyszerű feladat**
  - így ezekben a hálózatokban a hálózati réteg gyakran elég vékony, sőt van, amikor nem is létezik.

# OSI hivatkozási modell

## A szállítási réteg:

- A szállítási réteg (transport layer) legfontosabb feladatai:
  - adatokat fogadjon a viszonyrétegtől,
  - ha szükséges feldarabolja azokat kisebb egységekre,
  - továbbítsa ezeket a hálózati rétegnek,
  - biztosítsa azt, hogy minden kis egység hibátlanul megérkezzen a másik oldalra.
- Mindezt oly módon, hogy a felsőbb rétegek számára rejtve maradjanak a hardvertechnikában jelentkező változások
- A szállítási rétegben dől el az is:
  - hogy milyen típusú szolgáltatásokat nyújt a viszonyrétegnek
  - és ezzel tulajdonképpen az is, hogy milyen szolgáltatások állnak a hálózat felhasználóinak rendelkezésére

# OSI hivatkozási modell

## A szállítási réteg:

- A szállítási réteg egy igazi végpontok közötti réteg, a forráshoztól egészen a célhoz tart.
- Másképpen megfogalmazva: a forrásgépen futó egyik program beszélget egy hasonló programmal a célgépen,
  - felhasználva az üzenetek fejléceit és a vezérlőüzeneteket.
- Az alacsonyabb szintű rétegek protokolljait a gépek a közvetlen szomszédjukkal való kommunikációhoz használják,
- Nem pedig a tényleges küldő és a fogadó kommunikál velük, hiszen ezeket több útválasztó is elválaszthatja egymástól.

# OSI hivatkozási modell

## A viszonyréteg:

- A viszonyréteg (session layer) teszi lehetővé, hogy két gép egy viszonyt (session) vagy munkamenetet hozzon létre egymás között.
- A viszonyok sokféle szolgáltatást valósítanak meg:
  - pl a párbeszéd irányítás (dialog control) – az adás jogának kiosztása és nyomon követése,
  - a vezérlésközelítés (token management) – annak megakadályozására szolgál, hogy ketten egyszerre próbálják ugyanazt a kritikus műveletet végrehajtani,
  - és a szinkronizáció (synchronization) – ellenőrzési pontokat iktat a hosszú adásokba, hogy egy hibát követő helyreállítás után az ellenőrzési ponttól lehessen folytatni az adást).

# OSI hivatkozási modell

## A megjelenítési réteg:

- A megjelenítési réteg (**presentation layer**) nem a bitek mozgatásával foglalkozik, hanem az átvitt információ szintaktikájával és szemantikájával.
- Annak érdekében, hogy a különböző adatábrázolást használó gépek kommunikálni tudjanak, a párbeszéd során használt adatszerkezeteket és a „vezetékeken” használandó szabványos kódolást absztrakt módon kell definiálni.
- A megjelenítési réteg ezekkel az absztrakt adatszerkezetekkel foglalkozik, és lehetővé teszi a magasabb szintű adatszerkezetek (például banki ügyféladatlap) definiálását, valamint a gépek közötti átvitelét.

# OSI hivatkozási modell

## Az alkalmazási réteg:

- Az alkalmazási réteg (**application layer**) olyan protokollok változatos sokaságát tartalmazza, amelyekre a felhasználóknak gyakran szüksége van.
- Egy széleskörűen használt alkalmazási protokoll a HTTP (HyperText Transfer Protocol – hipertext-átviteli protokoll),
  - a világháló működésének alapja.
- Amikor egy böngésző meg akar szerezni egy weblapot, a HTTP segítségével küldi el a kért lap nevét a szervernek.
  - A szerver erre visszaküldi neki a lapot.
- További alkalmazási protokollok léteznek állományok átvitelére, e-levelezésre és a hálózati hírcsoportok eléréséhez.

**TCP/IP modell...**

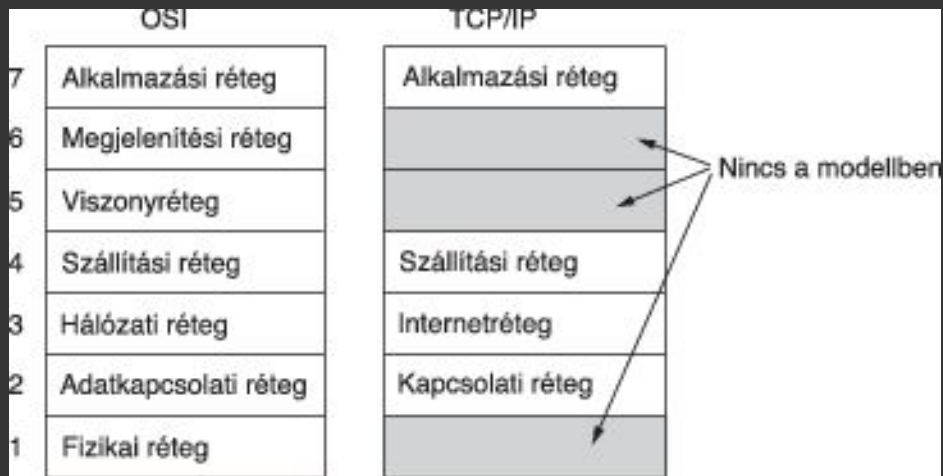
# TCP/IP hivatkozási modell

- Az **INTERNET** hivatkozási modellje. Korábban ARPANET
- Az **ARPANET** az amerikai védelmi minisztérium által támogatott kísérleti hálózat volt.
- Alkalmanként több száz egyetemi és kormányzati számítógépet kötött össze bérelt telefonvonalak segítségével.
- Később műholdas és rádiós hálózatokat is hozzákapcsoltak:
  - az akkori protokollok csak nehezen tudtak együttműködni,
  - egy új hivatkozási modell vált szükségessé.
- Ezért már a kezdetektől fogva az volt a legfőbb tervezési szempont:
  - lehetővé tegyék tetszőlegesen sok hálózat zökkenőmentes összekapcsolását.
- Később ez az architektúra **TCP/IP hivatkozási modell** néven vált ismertté

Transmission Control Protocol - átviteli vezérlő eljárás)

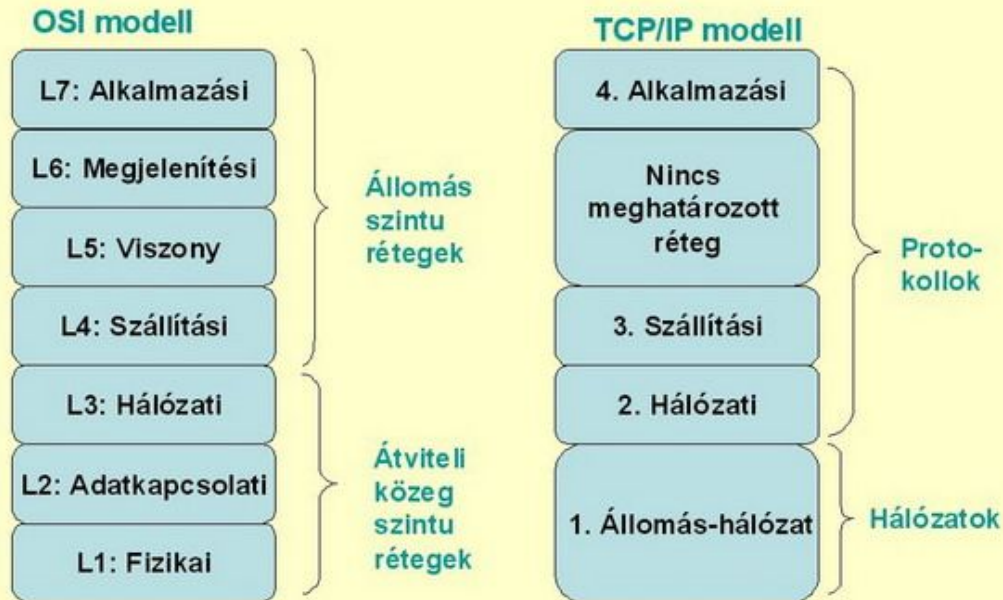
IP (Internet Protocol - internet eljárás)

# TCP/IP hivatkozási modell



# TCP/IP hivatkozási modell

## Az OSI modell és a TCP/IP modell összehasonlítása



# TCP/IP hivatkozási modell



## TCP/IP Model vs OSI Model

# TCP/IP hivatkozási modell

- Kapcsolati réteg:

- Az elvárások olyan csomagkapcsolt hálózathoz vezetnek:
  - amely egy összeköttetés nélküli rétegen alapul,
  - és különböző hálózatok között is működőképes.
- A modell legalsó rétege (**link layer**)
- Leírja: milyen képességekkel kell rendelkeznie az olyan átviteli elemeknek, hogy megfeleljenek az összeköttetés nélküli internetréteg igényeinek
  - Pl. soros vonalak vagy a klasszikus Ethernet, hogy megfeleljenek ennek az összeköttetés nélküli internetréteg igényeinek.
- Nem is valódi réteg, hanem egy csatlakozási felület a hosztok és az átviteli összeköttetések között.

# TCP/IP hivatkozási modell

- **Internetréteg:**

- Az internetréteg (**internet layer**) az összekötő kapocs, amely az egész architektúrát összefogja.
- Hozzávetőlegesen az OSI hálózati rétegnek feleltethető meg.
- **Feladata:**
  - lehetővé tegye a hosztok számára, hogy bármely hálózatba csomagokat tudjanak küldeni,
  - illetve a csomagok egymástól függetlenül célba jussanak (akár más hálózatokba is).
    - Az sem gond, ha a csomagok nem az elküldés sorrendjében érkeznek meg,
    - ugyanis, ha erre van szükség, akkor a magasabb rétegek visszarendezik őket a megfelelő sorrendbe.

# TCP/IP hivatkozási modell

- Internetréteg:

- Az internetréteg meghatároz egy hivatalos csomagformátumot, illetve egy protokollt
  - **Internetprotokoll** (Internet Protocol, **IP**)
- És egy ezt kísérő másik protokollt,
  - az internetes vezérlőüzenet protokollt (**Internet Control Message Protocol, ICMP**),
  - az előbbi működését segíti.
- Az internetréteg feladata:
  - kézbesítse az IP-csomagokat a címzetteknek.
  - A **csomagok útvonalának meghatározása** az egyik legfontosabb feladat,
  - a másik a **torlódások elkerülése** (bár az IP nem bizonyult túlságosan hatékonnak ez utóbbiban).

# TCP/IP hivatkozási modell

- Szállítási réteg:

- az internetréteg fölötti réteget általában szállítási rétegnek nevezik.
- Feladata: lehetővé tegye a küldő és címzett hosztokban található társentitások közötti párbeszédet
- Két különböző szállítási protokoll:
  - **Átvitelvezérlő protokoll** (Transmission Control Protocol, TCP), amely egy megbízható összeköttetés-alapú protokoll.
  - **Datagram protokoll** (User Datagram Protocol, UDP)

# TCP/IP hivatkozási modell

- Szállítási réteg:
  - TCP protokoll:
    - megbízható összeköttetés-alapú protokoll.
    - **Feladata az, hogy hibamentes bájtos átvitelt biztosítson bármely két gép között az interneten.**
    - A beérkező bájtos adatfolyamot diszkrét méretű üzenetekre osztja, majd azokat egyesével továbbítja az internetrétegnek.
    - A címzett host TCP-folyamata összegyűjti a beérkezett üzeneteket,
      - egyetlen kimeneti adatfolyamként továbbítja azokat.
    - A TCP forgalomszabályozást is végez
      - azért, hogy egy gyors forráshozt csak annyi üzenetet küldjön egy lassabb címzett hosztnak, amennyit az fogadni képes.

# TCP/IP hivatkozási modell

- Szállítási réteg:

- UDP protokoll:

- egy nem megbízható, összeköttetés nélküli protokoll.
- Jelentősége akkor van, amikor nem szükséges sem az üzenetek TCP-féle sorba rendezése, sem a forgalom szabályozás.
- **Hol hasznos?**
  - egyelővetű, kliens-szerver típusú kérés-válasz alkalmazásokban terjedt el, ahol a gyors válasz sokkal fontosabb, mint a pontos.
  - Pl. beszéd- vagy videoátvitel.

# TCP/IP hivatkozási modell



## TCP/IP Layers

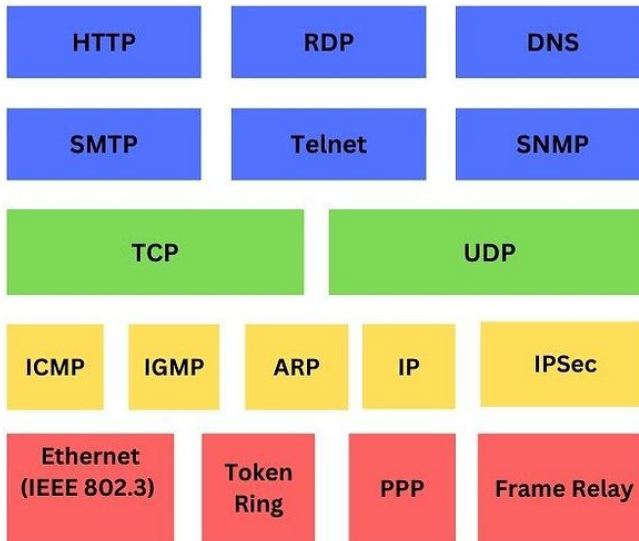
Application Layer

Transport Layer

Internet Layer

Network Access Layer

## Protocols (Examples)



# TCP/IP hivatkozási modell

- Alkalmazási réteg:

- A TCP/IP-modellben nincs viszony- és megjelenítési réteg.
  - nem kerültek bele a modellbe, mert nem mutatkozott irántuk igény.
- Helyette az alkalmazások tartalmazzák azokat a viszony és megjelenítési funkciókat, amelyekre szükségük van.
  - Az OSI-moddellel kapcsolatos tapasztalok is bizonyítják ezt a nézetet: a legtöbb alkalmazás nemigen használja ki e két réteget.
- A szállítási réteg fölött az alkalmazási réteg (application layer) található.
- Tartalmazza az összes magasabb szintű protokollt:
  - Pl. virtuális terminál (**TELNET**),
  - a fájltranszfer (**FTP**) és az elektronikus levelezés (**SMTP**) protokolljai.
  - (Domain Name System, **DNS**), ami a hosztnevek és hálózati címek megfeleltetésére szolgál,
  - a **HTTP**, a világháló oldalainak letöltésére szolgáló protokoll,
  - az **RTP**, a valós idejű média (hang és mozgókép) továbbításának protokollja.

**Köszönöm a figyelmet!**