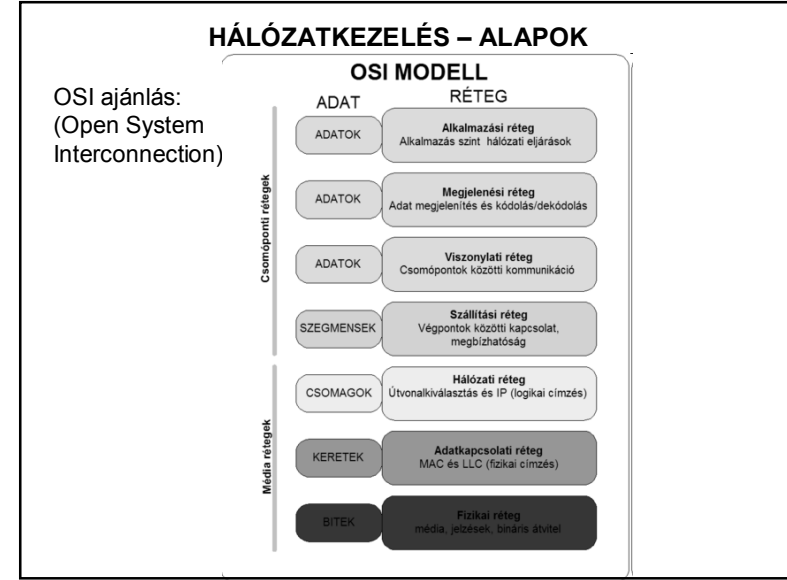
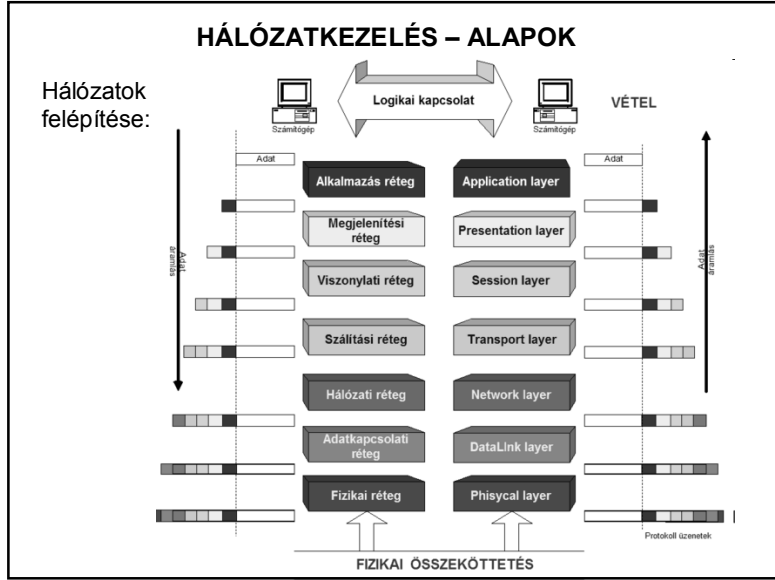
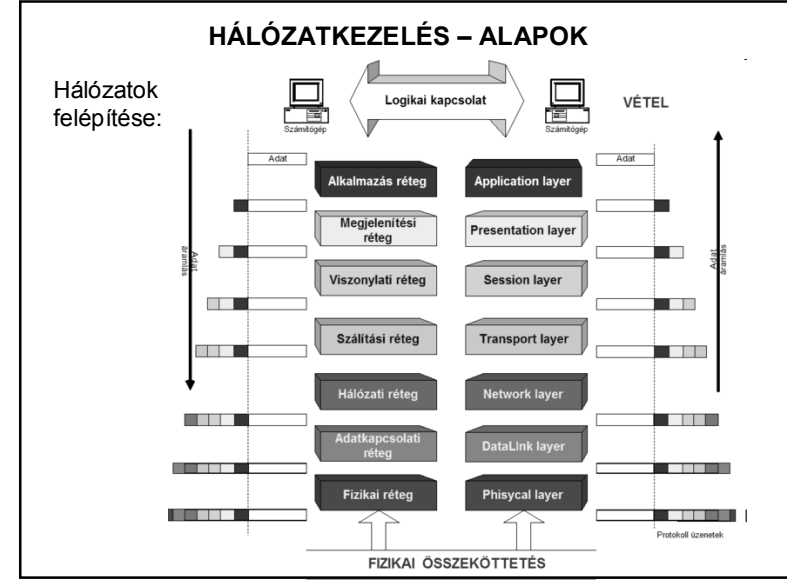


# Programozás III

HÁLÓZATI ALAPOK



## HÁLÓZATKEZELÉS – ALAPOK

Az adatátvitellel foglalkozó rétegek (1):

### A fizikai réteg (physical layer)

A **bitek** kommunikációs **csatornára való kibocsátásáért** felelős. Biztosítani kell, hogy az adó által küldött jeleket a vevő is azonosként értelmezze.

### Az adatkapcsolati réteg (data link layer)

Alapvető feladata a szomszédos gépek közötti **hibamentes** átvitel biztosítása. Az adatokat adatkeretekké (data frame) tördeli, továbbítja, a nyugtát fogadja, hibajavítást és forgalomszabályozást végez.

## HÁLÓZATKEZELÉS – ALAPOK

A logikai összeköttetéssel foglalkozó rétegek:

### A viszonyréteg (session layer)

Lehetővé teszi, hogy két számítógép **felhasználói kapcsolat**ot létesítsen egymással. Jellegzetes feladata a logikai kapcsolat felépítése és bontása, párbeszéd szervezése. Szinkronizációs feladatokat is ellát, ellenőrzési pontok beépítésével.

### A megjelenítési réteg (presentation layer)

Az egyetlen olyan réteg, amely megváltoztathatja az üzenet tartalmát. Tömörít, rejtjelez (adatvédelem és adatbiztonság miatt), **kódcserét** (pl.: ASCII - EBCDIC) végez el.

## HÁLÓZATKEZELÉS – ALAPOK

Az adatátvitellel foglalkozó rétegek (2):

### A hálózati réteg (network layer)

A kommunikációs alhálózatok működését vezérli, feladata az **útvonalválasztás** forrás és célállomás között. Ha az útvonalban eltérő hálózatok vannak, akkor fragmentálást, protokoll átalakítást is végez. Az utolsó olyan réteg, amely ismeri a hálózat topológiáját.

### A szállítási réteg (transport layer)

Feladata a **végpontok közötti hibamentes adatátvitel** biztosítása. Már nem ismeri a topológiát, csak a két végpontban van rá szükség. Feladata az összeköttetések felépítése, bontása, csomagok sorrendbe állítása.

## HÁLÓZATKEZELÉS – ALAPOK

A logikai összeköttetéssel foglalkozó rétegek (2):

### Az alkalmazási réteg (application layer)

Széles körben igényelt **szolgáltatásokat tartalmaz**. Pl.: fájlok gépek közötti másolása.

## HÁLÓZATKEZELÉS –PROTOKOLLOK

### A fizikai átvitel fölé épülő protokollok:

A csomópontok címzési módját, illetve a hálózaton továbbított adatcsomagok méretét és formátumát rögzítik.

### IP (Internet Protokoll):

- Nem biztosít megbízható átvitelt,
- az adatcsomagok késhetnek, sérülhetnek, duplikálódhatnak, elveszhetnek, de csak abban az esetben, ha az IP által használt hálózat hibázik.
- Az IP adatcsomagok formátuma viszonylag bonyolult, méretük maximum 64 kbyte lehet. Mindig egy ún. fejrész (header) kezdődnek, amely tartalmazza a küldő és a címzett csomópont IP-címét.

## HÁLÓZATKEZELÉS –PROTOKOLLOK

### TCP (Transport Control Protocol):

az IP protokollra épülve megbízható hálózati átvitelt garantál.

- A TCP – akár egy telefontársaság – garantálja, hogy a küldött adat helyes sorrendben érkezen meg a vevőhöz. Ha ez nem sikerül, hibajelzést küld vissza.
- Olyankor alkalmazzák, amikor a hálózaton átküldött és fogadott adatok sorrendje elengedhetetlen az alkalmazások helyes működéséhez.
- TCP-t használó legismertebb szolgáltatások:  
*Hyper Text Transfer Protokoll (HTTP),*  
fájlok átvitele távoli gépekre (*FTP: file transfer*),  
távoli gépekre történő belépés (*telnet*),  
*elektronikus mail (e-mail)* szolgáltatás alapját adó *SMTP* protokoll.

## HÁLÓZATKEZELÉS –PROTOKOLLOK



Fejrész (header)

Az IP protokollt megvalósító szoftver feladata az, hogy az IP-címet a helyi hálózatban használatos címmé alakítsa, és az adatcsomagot továbbküldje, elvégezve a használt alacsonyabb szintű protokollnak megfelelő változtatásokat az adatrészben.

Az IP protokollra számos más protokoll épül.

## HÁLÓZATKEZELÉS –PROTOKOLLOK

### UDP (User Datagram Protocol):

nem ad megbízható kapcsolatot egy hálózaton levő két számítógép között, mivel ez nem kapcsolat alapú, mint a TCP, inkább az adatok független csomagjait küldi az alkalmazások között. Ezeket a csomagokat adatcsomagoknak is hívjuk.

- A adatcsomagok küldése leginkább a levelek postán át való küldéséhez hasonlít: a szállítás nem garantált, a sorrend nem fontos, és mindegyik üzenet különbözik a másiktól.
- UDP-t használó szolgáltatások:  
távoli gépeken dolgozó felhasználók azonosítójának lekérdezése (*rwho*),  
fájlok hordozása (*TFTP*),  
*ping*.

## HÁLÓZATKEZELÉS –PROTOKOLLOK

A számítógép egy egyszerű fizikai kapcsolaton keresztül csatlakozik a hálózatra.  
Minden adat ezen a kapcsolaton át érkezik, függetlenül attól, hogy az adatok a számítógép melyik programjait használják.

Honnan tudja a számítógép, hogy melyik alkalmazásnak melyik adatot kell továbbítani?



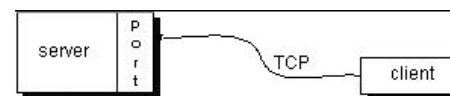
Port-ok használatával

A TCP és az UDP protokollok portokat használnak, hogy a bejövő adatokat a számítógép megfelelő programjai felé irányítsák.

## HÁLÓZATKEZELÉS –PORTOK

A kapcsolat alapú kommunikációnál (TCP)  
a szerver program leköt egy foglalatot egy jellemző port számára.  
Így a szerver megkap minden adatot, ami ezen a porton keresztül érkezik.

A kliens ezután a megadott porton keresztül kommunikálhat a szerverrel.



## HÁLÓZATKEZELÉS –PROTOKOLLOK

Az interneten küldött adat el van látva címzési információval: ez azonosítja

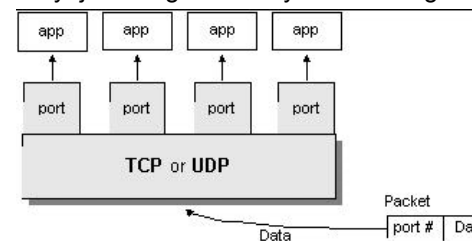
- a célszámítógépet és
- a portját.

A számítógép a 32 bites IP címmel van azonosítva, melyet arra használunk, hogy az adat a megfelelő számítógépre érkezzon meg.

A portot egy 16 bites számmal azonosítjuk, amit a TCP vagy UDP arra használ, hogy az adat a megfelelő programhoz jusson.

## HÁLÓZATKEZELÉS –PORTOK

Az adatcsomag alapú kommunikációnál (UDP)  
az adatcsomagok tartalmazzák a célállomás portszámát, és az  
UDP irányítja a megfelelő helyre a csomagot.



## HÁLÓZATKEZELÉS –PORTOK

A portok 16 bites számként vannak ábrázolva ⇒  
értékük 0 - 65535 közötti szám lehet

A 0 és 1023 közötti portok fent vannak tartva olyan ismert szolgáltatásoknak, mit például a HTTP vagy az FTP vagy más rendszerszolgáltatás. ⇒

A saját programjaink nem használhatják őket.

Egy internetes szolgáltatás elérési címe:

IP CÍM : port

Pl.: witch.pmmf.hu:2001

## HÁLÓZATKEZELÉS – JAVA

A Java programok a **java.net** csomag osztályain keresztül használhatják a TCP vagy UDP protokollokat.

TCP-n keresztül kommunikálnak az  
*URL*, *URLConnection*, *Socket* és *ServerSocket*  
osztályok.

Az UDP protokollt használják a  
*DatagramPacket*, *DatagramSocket* és *MulticastSocket*  
osztályok.