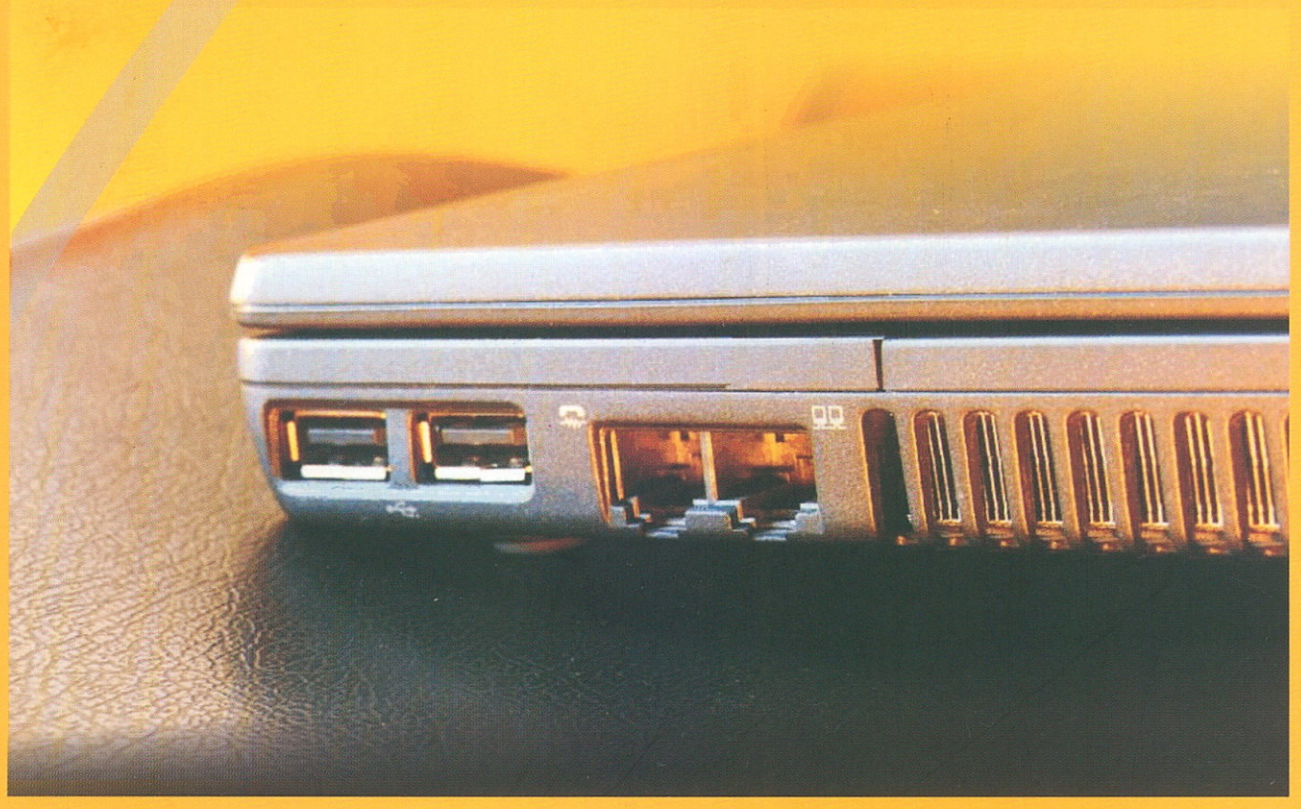
Kónya László

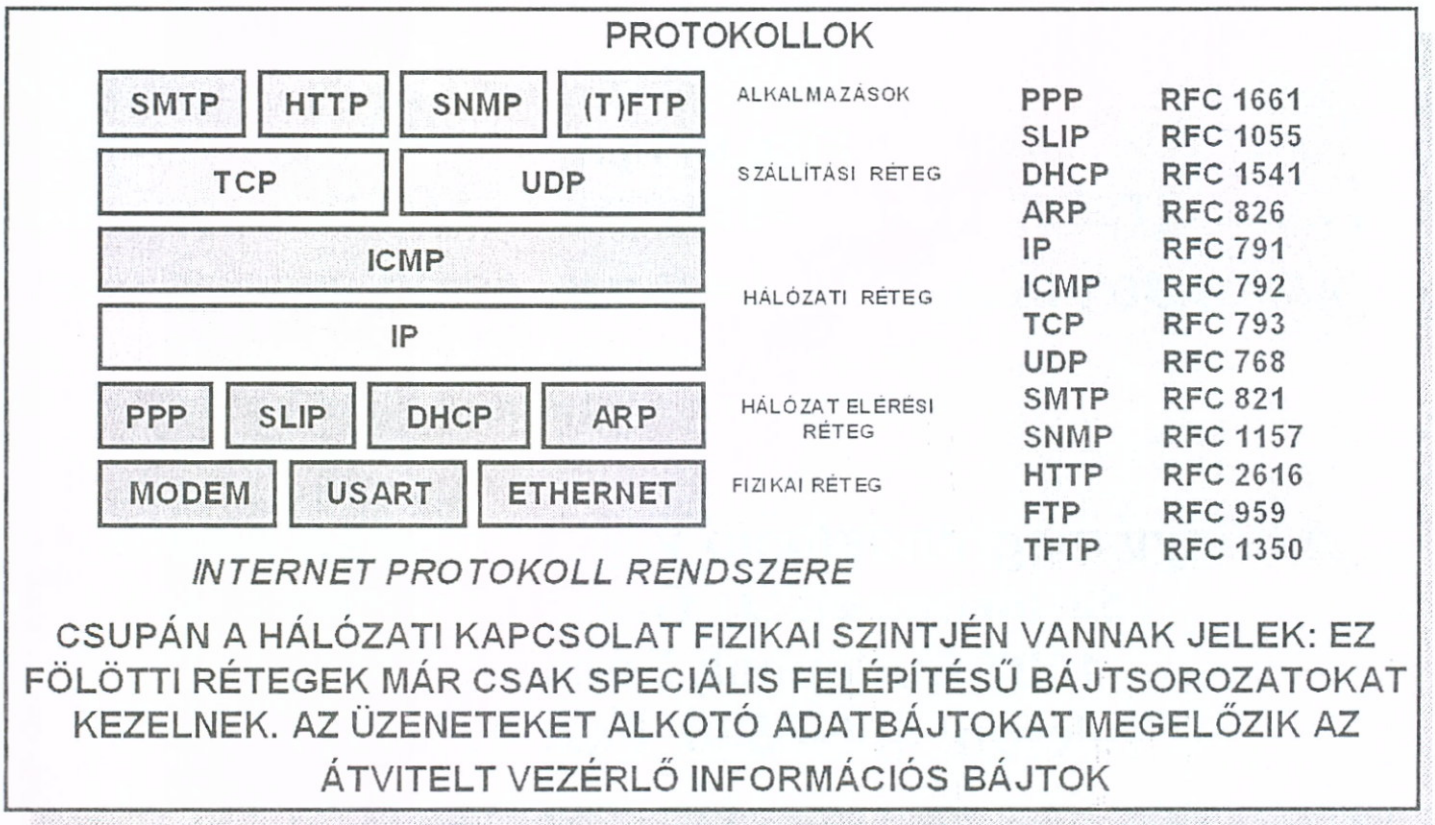
# SZÁMÍTÓGÉP-HÁLÓZATOK



INOK

DR. KÓNYA LÁSZLÓ

# Számítógép-hálózatok




Nyitott rendszerű képzés
– távoktatás – oktatási segédlete
Felsőoktatási tankönyv

**INOK KIADÓ**
Budapest, 2006

*Lektorálta:*

***FÜZI ATTILA***

***A könyv megrendelhető vagy megvásárolható:***
**LSI INFORMATIKAI OKTATÓKÖZPONT**
**1037 Budapest, Bécsi út 324.**
**Telefon: 436-6520 Fax: 436-6521**

**ISBN 963 9625 16 7**

Kiadó: **INOK KFT**
Kiadásért felel az INOK Kft. ügyvezetője
Témafelelős: **Flier István**

LIGATURA KFT – NASZÁLY PRINT KFT

# TARTALOMJEGYZÉK

*Csak azt látjuk és értjük meg, amit látni és érteni akarunk.*

# BEVEZETÉS

*Ami működik, az csodálatos...*

Tisztelt Olvasó!

A könyv első kiadása 1997-ben íródott. Azóta a könyv már kisebb módosításokkal többször átdolgozásra került, és eljött az idő, hogy a számítógépes hálózatok jelenlegi fejlettségét figyelembe véve, a témát újragondolva, megszülessen a könyv jelenlegi változata. A könyv alapvető célja nem változott: a Gábor Dénes Műszaki Informatikai Főiskolán oktatott **Számítógép-hálózatok** c. tárgy tanulásához nyújt segítséget.

Jól tudjuk, hogy ilyen témával foglalkozó könyvek, a tématerület aktualitása miatt magyar nyelven is hozzáférhetők, de azok témaválasztása néha sokkal átfogóbb, illetve speciálisabb, hogy azt kötött óraszámban, és az oktatás korlátait figyelembe véve, oktatni lehessen.

Ez a könyv, amely elsődlegesen oktatási célokra íródott, tartalmának kialakításánál négy szempont játszott szerepet:

- Didaktikailag jól felépített legyen,
- önálló tanulásra is alkalmas legyen,
- az átfogó elméleti ismeretek tárgyalása után konkrét, a gyakorlatban is jól használható témákkal foglalkozzon.
- A témák tárgyalásánál vegye figyelembe a kommunikáció és a számítástechnika konvergenciáját (egymáshoz való közeledését).

Mivel a könyvben szereplő ismeretek már sok helyen és jól megfogalmazva megjelentek, ezért nem volt cél az eredetiség: sokszor használtunk fel ábrákat és magyarázatokat az adott témák tárgyalásánál. Ilyenkor utaltunk az eredeti irodalomra is.

A könyv alapos átdolgozását - lényegében újraírását - az is indokolja, hogy az emberi kommunikáció elektronikus eszközökkel való megvalósítása: mobiltelefonok, internet ma már az emberek társadalmi létét is meghatározza. Hasonlóan az autókhoz - amelyeket majdnem mindenki tud vezetni - a kommunikáció említett formáit ismerni, használni, és érteni kell.

## A könyv célja

A számítógép alapú kommunikáció alapjainak, felépítésének, működésének a megismerése, azok hatékony és eredményes alkalmazása érdekében. Ez azt jelenti, hogy az olvasónak a könyv elolvasását és megértését követően, áttekintő képpel kell rendelkeznie:

- Általában a számítógépes kommunikációról,
- az alkalmazott, a gépek teljesítményétől függő megoldási módokról,
- az egyes kommunikációs részegységek működésének alapjairól.

## A könyv felépítése

Röviden összefoglaljuk a könyv felépítését, ami didaktikai szempontból nem szerencsés, hiszen a következőkben számos olyan fogalmat fogunk használni, amit itt nem magyarázunk, és lényegüket a könyv későbbi fejezeteinek tanulmányozásával tudjuk megérteni.

A könyv két részből áll, amely összesen tíz fejezetre tagozódik. Az első rész (1-6. fejezet) a **Hálózati elméleti ismeretek.** A tárgyalásmód az OSI modell rétegeit követi.

A második rész**: Hálózatok a gyakorlatban**. Ebben a részben (7-10. fejezet) az első rész ismereteit felhasználó gyakorlati, napjainkban aktuális ismeretanyagot adunk közre.

Az **1. Fejezet** a számítógép-hálózatok elméleti alapjait foglalja össze. Bemutatja a hálózatok kialakításának a célját, a rétegszemléletet, és az ehhez kapcsolódó alapismereteket. A fejezet legfontosabb része annak az ún. OSI-modellnek a bemutatása, amely mindenfajta kommunikáció tárgyalásához alapot nyújt.

A **2. Fejezet** a hálózatok fizikai jellemzőit mutatja be, foglalkozik a vonalmegosztási kérdésekkel, a vezetékes (réz, üvegszál) alapú illetve vezeték nélküli átvitel jellemzőivel, különféle megoldásaival.

A **3. Fejezet a**z analóg és digitális átvitel megoldásait taglalja: az alapsávi analóg átvitel a vezetékes telefonhálózathoz kapcsolódik; a széles sávú analóg jelátvitelt használja a kábel tv, az ADSL. A továbbiakban foglalkozunk a digitális jelek kódolásával, majd a karakterábrázolást mutatjuk be. Ez után következik az adatkeretek átvitelénél használt protokollok bemutatása, mivel két eszköz közötti adatcsere megvalósításához adatkapcsolati protokollok kidolgozása szükséges. A következőkben, a hálózaton közös csatornán kommunikáló eszközök adási-jog megszerzési stratégiáit, a közegelérési módszereket mutatjuk be. Az IEEE802 szabványok ismertetésén keresztül a módszer gyakorlatban használt megoldásai is bemutatásra kerülnek.

A **4. Fejezet** az adatkeretek átviteli megoldásait mutatja be: soros adatátviteli szabványok, ISDN, ATM, USB, CAN busz, illetve az ADSL, és kábelmodem megoldásokat. A mobiltelefon hálózatokat is bemutatjuk röviden.

Az információcsere több csomópontot tartalmazó hálózatban, az információ útjának kijelölését igényli. Ezt a feladatot hálózati réteg látja el, a kapcsolódó ismeretek a hálózati réteget bemutató **5. Fejezet**-ben találhatók.

Mikor az információ eljut a távoli csomópontba, még számos feladatot kell megoldani, amely feladat a felsőbb rétegekre hárul. **6. Fejezet**-ben a szállítási, együttműködési, megjelenítési és az alkalmazási réteg feladatait mutatjuk be.

A második részben a környezetünkben megjelenő hálózati megoldásokkal ismerkedünk meg.

Az Internet kisebb kiterjedésű számítógépes hálózatokból (LAN) álló globális számítógépes rendszer, adatátviteli protokollja a TCP/IP. Mivel az Internet hatalmas léptekkel fejlődik, ezért egy külön fejezetet (**7. Fejezet**) szenteltünk az Internet hálózattal kapcsolatos legfontosabb ismeretek összefoglalására. Itt természetesen már felhasználjuk az előbbiekben szereplő tudásanyagot.

A számítógépes hálózatok egyik legdinamikusabban fejlődő területe a kisebb helyi hálózatok, azaz a lokális hálózatok. A legtöbb felhasználó a hálózatokkal ilyen formában találkozik, a csoportos munka, a kommunikáció, és az elosztott információkezelés hatékony eszközeként. Ilyen hálózatokban – mivel egymással az interneten keresztük összekapcsolódnak – célszerű az internetes technológiákat felhasználni, amit **intranet**-nek nevezünk. A **8. Fejezet** a helyi hálózatokat, a LAN-okat és az intranetet mutatja be. Érdekes, és itt bemutatandó terület a virtuális magánhálózatok alkalmazása, amely az internet infrastruktúráját, és kommunikációs közegét felhasználva, lehetővé teszi egymástól távol lévő pontok összeköttetését.

A **9. Fejezet**-ben külön jelenik meg a napjainkban a hálózatok fejlődésével együtt járó problémakör: a hálózatok biztonsági kérdései. Mivel az emberek közösen használják a hálózati szolgáltatásokat, ezért meg kell teremteni, a hálózat erőforrásainak és felhasználóinak egymástól elszigetelt felhasználást. A biztonsági kérdések ismerete és a megoldások alkalmazása ma már mindenki számára kötelező.

A **10. Fejezet** a hálózatok üzemeltetésével kapcsolatos, és sok esetben a hálózat használóit is érintő hálózatfigyelés és felügyelet alapjait mutatja be.

A könyv végén található Irodalomjegyzék — a teljességre törekvés igénye nélkül — sorol fel néhány, a témakörrel kapcsolatos könyvet.

Természetes a hálózatok ilyen tárgyalása, felosztása egy kicsit önkényes, hiszen a hálózatokat lehetséges kiterjedtség szerint is tárgyalni.

- A legnagyobbak a nagykiterjedésű világméretű hálózatok. Ezeket **WAN**-oknak (Wide Area Network) nevezzük és azért fontosak, mert a jelenleg robbanásszerűen terjedő Internet hálózat használatához nyújtanak alapokat.
- Ez alatt helyezkednek el a városi, nagyobb területre kiterjedő **MAN**-ok (Metropolitan Area Network).
- A harmadik szint a helyi hálózatok. (**LAN** — Local Area Network) Ezek általában egy intézményhez kapcsolódnak, segítve az intézmény szervezettségét, az intézmény hatékony működéséhez szükséges intézményen belüli kommunikációt.
- A negyedik szint a termelés- és folyamatirányításban egyre nagyobb szerepet játszó mikroszámítógép alapú eszközök kapcsolatát lehetővé tevő kommunikációs hálózatok.

## *Hogyan kell a tantárgyat tanulni?*

A téma tanulásához a "Bevezetés a számítástechnikába" és az "Elektrotechnikai és elektronikai alapismeretek" valamint a "Mikroszámítógépek" című tárgyak előzetes hallgatása szükséges.

A tananyag elsajátítását a minden fejezet végén megtalálható kérdésekre adott válaszokkal ellenőrizhetjük. Természetesen az elsajátításban sokat segít az előadásokon történő részvétel, vagy a készült videofelvétel meghallgatása, mert az ott elhangzottak áttekintést adnak a tananyag összefüggéseiről, segítséget nyújtanak a bonyolultabb részek megértéséhez.

A szerző honlapja (*http://www.aut.bmf.hu/konya*), valamint az évfolyamnak kiadott CD is számos jól tanulható anyagot tartalmaz. Az elhangzottak otthoni feldolgozása (az előadási jegyzet átolvasása, átgondolása, a lényegi részek kiválasztása, az összefüggések feltárása, rendszerezése, — valamint kiegészítése a tankönyvek, vagy más kapcsolódó szakirodalom, folyóirat alapján — biztosítja a felkészülést a következő tananyagrész megértéséhez, és a számonkéréshez

## *Megváltozott az információhoz való viszonyunk*

Régebben egy- egy információhoz való jutás igen nehézkes volt: a világ más részein született eredményeket csak hosszú idő után, sokszor hosszas utánjárással lehetett megszerezni, vagy esetleg könyvtárakat és azokban lévő katalógusokat kellett bújni. Jelenleg a helyzet más: az Interneten hatalmas mennyiségű naprakész információ áll rendelkezésre.

Mikor a könyv anyagához forrásokat gyűjtöttem, elképesztett az a bőség, ami például a számítógépes hálózatokkal kapcsolatban az Interneten megtalálható. Például egy számítógépes rendszer üzemeltetéséhez is minden információ megtalálható, letölthető és felhasználható. Csak egy kicsi baj van: a dokumentumokat el kell olvasni, és meg kell érteni, ami sok erőfeszítést és önálló tanulást igényel. **Vagyis olyan korban élünk, ahol már nem az információ megszerzése, hanem annak az elsajátítása okozhat gondot.**

A nagy mennyiségű információhoz való jutás lehetősége – az információbőség – azonban komoly problémákat is felvet. Nevezetesen azt, hogyan tudjuk a rengetegből kiválasztani a lényegeset, a fontosat, milyen elveket érvényesítsünk a kiválasztásban. Fontos az információközlő felelőssége: egyszerre szeretne mindent közölni az adott tématerületről, de jó lenne, ha Ő maga — mint a legavatottabb szakértő — korlátozná, szűrné az átadásra kerülő információt,

Alapgondolatként egy olyan, már régen jól ismert, hasznos elvet kívánunk felidézni, amely egy információközlési módszernek is tekinthető: a **lényegkiemelést**. Ez azt jelenti, hogy bármilyen téma bemutatásakor nagyon egyszerűen, lényegre törően, akár bizonyos gondolati pongyolaság árán, próbáljuk a lényeget kiemelni, utalva az egyszerűsítés korlátaira is.

Tudjuk, hogy a túlzott egyszerűsítés nagyon veszélyes, felületességre késztet, de pszichológiailag azt az előnyt adja, hogy az adott téma elsajátítója önbizalmat kap, és hitet arra, hogy a részleteket is képes megérteni, illetve megtanulni.

## *A könyv belső formátuma*

Az oldalakon használtuk a lényegkiemelést segítő, és a szövegek monotonítását megtörő szövegformázási eszközöket: aláhúzást, dőlt és vastagított szövegrészek alkalmazását. A könyvben sokszor találunk egy kisebb betűnagyságot. Ez általában magyarázó szöveg, vagy példa. Lapszéleken két jelölést helyeztünk el:

? a kérdéseket, ! nagyon fontos részeket jelöli.

A könyvben szereplő fogalmaknál, ahol lehetett, ott már a megszokott magyar elnevezéseket használtuk.

Kérem a könyv olvasóit, a könyvből tanuló hallgatókat, hogy a könyvvel kapcsolatos észrevételeiket, az esetleges hibákat részemre eljuttatni szíveskedjenek.

Budapest, 2005. szeptember

*a szerző*

E-mail: *konya.laszlo@kvk.bmf.hu*

http://www.aut.bmf.hu/konya

# I. HÁLÓZATI ELMÉLETI ISMERETEK (1-6 FEJEZET)

# 1. HÁLÓZATOK CÉLJA, ALKALMAZÁSA, ALAPFOGALMAK

*Először jön az elmélet, aztán a gyakorlat. (Leonardo da Vinci)*

A számítógépek megjelenésekor mindegyik egymástól elkülönülve, önállóan dolgozott. Még a személyi számítógép, a „personal computer" nevében is hordozza az elkülönültségre utaló „személyi" jelzőt. A fejlődéssel azonban megjelent az igény a számítógépek összekapcsolására.

**Számítógép-hálózatok alatt az egymással kapcsolatban lévő, önálló számítógépek rendszerét értjük.** A meghatározás nagyon egyszerű, de mint sok más dolog, ez is bonyolultságot rejt magában.

## Milyen előnyökkel jár a gépek hálózatba kapcsolása?

- Lehetővé teszi a berendezések, perifériák, programok, adatok **közös használat**át, azaz a külön-külön meglévő erőforrások megosztását. Ez azt jelenti, hogy ezek az erőforrások felhasználók fizikai helyétől függetlenül bárki - ténylegesen a megfelelő jogosultságokkal rendelkezők - számára elérhetők.
- A rendszerben lévő **eszközök teljesítményének egyenletesebb megosztására** is lehetőséget biztosít ez a megoldás.
- A kialakított rendszer **nagyobb megbízhatóságú működés**t eredményezhet. Például, egy nyomtató meghibásodása nem jelenti azonnal a nyomtatási lehetőségek megszűnését, mivel szerepét a rendszerben lévő másik nyomtató is átveheti. A fontosabb programok, adatok a rendszer több számítógépének lemezegységén is tárolódhatnak, és az egyik példány megsemmisülésével sem történik helyrehozhatatlan károsodás.
- A fenti előnyök anyagi oldalról tekintve részben **költségmegtakarítás**sal járnak. Az eszközöket (pl. nyomtatókat, háttértárakat) kevesebb példányban kell beszerezni (de a hálózati interfészt minden számítógépbe meg kell vásárolni...).
- A hálózatok további célja a **skálázhatóság**, ami azt jelenti, hogy a teljesítmény növelése fokozatosan úgy lehetséges, hogy újabb processzorokat adunk a rendszerhez. (pl.: Fürtözési technika).
- **Hozzáférés távoli információkhoz**. Lehetővé válik adatbázisok elérése, a benne lévő adatok felhasználása, sőt az adatbázis sok pontról történő bővítése. Erre példa lehet egy multinacionális vállalat rendelési rendszere. Olyan programok is futtathatók ilyen módon, amelyek erőforrásigénye nagyobb, mint ami egy gépen rendelkezésre áll.

- A jelenlegi egyik legizgalmasabb kibővítés az, amikor a **hálózati rendszert kommunikációs közegként használjuk**.
- **Személyek közötti kommunikáció.** Ez azt jelenti, hogy a rendszer használói egymásnak üzeneteket, leveleket vagy egyéb információt tudnak küldeni. Jelenleg a számítástechnika fejlődése ebbe az irányba mutat. A hálózati kapcsolatok egyre bővülő lehetősége azt is lehetővé teszi, hogy olyan számítógépeket készítsünk, amely a futtatandó programjait, adatait nem saját maga tárolja, hanem a hálózat valamelyik kiszolgáló gépén van elhelyezve. Ez a megoldás nagymértékben csökkentheti egy számítógépben elhelyezett egységek számát, és ezért nagyon olcsó.
- **Interaktív szórakoztatás.** Mivel a számítógépek és a hálózatok az otthonokban is megjelentek már lehetővé válik a hálózati videózás, amely a hálózaton eljuttatott videoinformációt tartalmazó adatok segítségével valósul meg.

### *Milyen hátrányokkal jár a gépek hálózatba kapcsolása?*

- A hálózatban lévő információk és felhasználók védelmét, egymástól való elszigetelését meg kell oldani.
- A központi program- és adattárolás, a felhasználók bizonyos kiszolgáltatottságát eredményezi.
- A hálózat kritikus pontjain történő meghibásodás az egész hálózat működését lehetetlenné teheti.
- Új típusú bűnözési módok jelennek meg: adatlopások, rendszerek bénítása, adatok illegális felhasználása, számítógépes vírusok használata, stb.

## *1.1 Hálózati struktúrák*

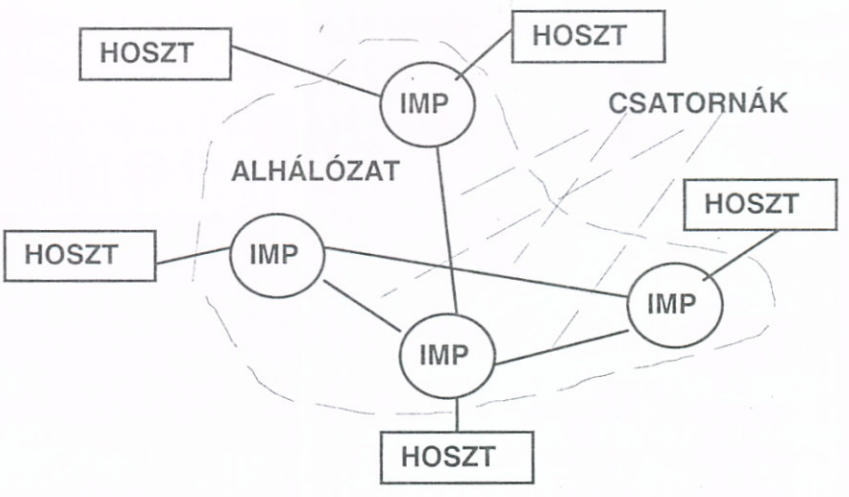
A következő részben összefoglaljuk a számítógépes hálózatokkal kapcsolatos legfontosabb alapfogalmakat: a hálózat építőelemeit, a kapcsolódásuk módjait.

### *1.1.1 Fontos alapfogalmak*

**Azokat a számítógépeket, amelyeket egy számítógépes hálózattá összekötünk hosztoknak (host) nevezzük.** Ezt magyarul gazdagépnek is hívjuk, itt futnak a felhasználói programok, helyezkednek el az adatbázisok.

Az elnevezés onnan származik, hogy a hálózatokat UNIX alapú miniszámítógépeknél használtak először. Itt több felhasználó terminálon (képernyő+billenyűzet) keresztül kapcsolódott a „gazdához".

A gépeket kommunikációs alhálózatok kötik össze,




*1-1. ábra Kommunikációs alhálózatok*

amelyek feladata a hosztok közötti_kommunikáció megvalósítása, azaz üzenetek továbbítása.

Általában ezek az alhálózatok két jól szétválasztható részből állnak:

- Az információ fogadását, küldését, szétosztását megvalósító **kapcsolóelemekből,**
- az átvitelt biztosító **csatornákból** (itt áramlik az információ).

A kapcsolóelemek elterjedt neve **IMP (Interface Message Processor)** azaz **interfész üzenet feldolgozó.** Az IMP-ek lehetnek a hoszt részei (pl. hálózati kártya és a programja), de sokszor valójában speciális számítógépek, amelyek a csatornák kapcsolását végzik, a bemenetükre jutó adatot valamelyik meghatározott kimenetre kapcsolják. További szokásos nevük: belső hálózati kapcsoló pontok (internal network switching node).

A kommunikációban résztvevők közötti információcsere alapvetően két módon lehetséges:

- Az egyik megoldásnál a csatorna kizárólag a két kommunikáló fél között kerül kialakításra. Ezt **pont-pont** típusú összekötetésnek hívjuk.
- A másik esetben a kommunikáló feleknek csak egy közös csatorna áll rendelkezésre: ezt kell megfelelő szabályok szerint a felek között megosztani. Ezek az **üzenetszóró** (broadcasting) alhálózatok.

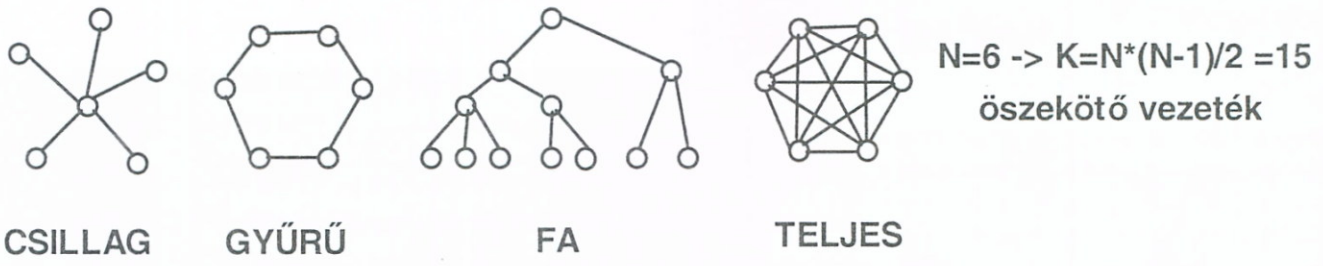
Melyik a jobb megoldás? Erre a kérdésre a válasz nem egyértelmű. Ezért a következőkben részletesebben megvizsgáljuk mindkét lehetőséget.

### 1.1.2 Pont-pont összeköttetés

Ebben az esetben a két kommunikációs végpontot, például kábel köti össze, és az üzenetek (vagy annak a szétdarabolt részei, a **csomagok (=packet))** ezen a kábelen keresztül haladnak. Abban az esetben, ha a csomagok rövidek és azonos méretűek gyakran **celláknak** (cells) nevezik őket.

Két ember közötti négyszemközti vagy telefonon keresztül történő beszélgetés is lényegében pont-pont alapú összeköttetés szerint történik.

Amikor egy vevő megkapja a csomagot és annak nem Ő a címzettje, akkor azt továbbadja egy következő pont-pont összeköttetésen keresztül. Ezért az ilyen típusú hálózatokat más néven szokták két pont közötti (**point-to-point**), vagy **tárol és továbbít** (store-and-forward), vagy **csomagkapcsolt** hálózatoknak nevezni.




*1-2. ábra Pont-pont topológiák*

Ha a kapcsolatokat úgy alakítjuk ki, hogy a felek csak egymással kommunikálnak, akkor nem szükséges a másik fél azonosítása (nem kell címzés!) Az ilyen kialakításnak lényeges előnye az, hogy a két pont közötti kapcsolatból adódóan **a kommunikációs problémákat elsődlegesen a pontok közötti csatorna hordozza,** és hibák behatárolásánál is előnyös ez a kialakítás.

**Hátrányának** lehet felróni**,** hogy több pontot tartalmazó hálózatban a pontok közötti kommunikáció csak a közvetlen összeköttetések kialakításával lehetséges.

Általában igaz, hogy N pontot tartalmazó hálózatban, ahhoz, hogy minden állomás minden állomással közvetlenül tudjon kommunikálni N*(N-1)/2 darab pont-pont összeköttetést kell kialakítani. (például 6 pont összekötéséhez 6*5/2=15 pont-pont összeköttetés szükséges.

Több pont-pont kapcsolatú végpont összeköttetése különféle módokon valósítható meg. Az *1-2. ábrán* néhány lehetséges elrendezést mutatunk be.

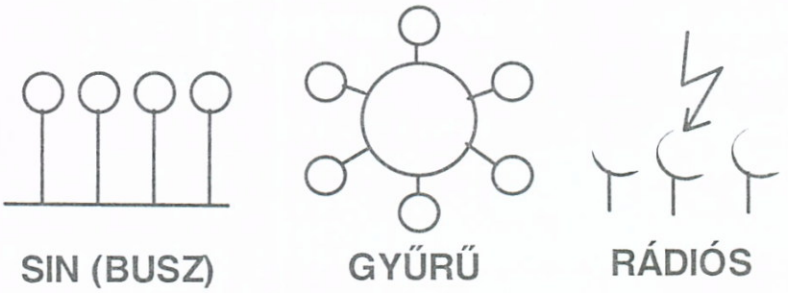
## 1.1.3 Üzenetszórásos csatornahasználat

Ilyen típusú hálózatoknál a kommunikációban résztvevők számára ténylegesen egy kommunikációs csatorna van, és ezen az egy csatornán osztozik az összes hálózatba kapcsolt számítógép. A küldött üzeneteket a hálózat minden állomása veszi, (ami nehezíti az adatvédelmet) és azt, hogy az üzenet kinek szól, az üzenetben elhelyezett egyedi — gépet címző — címinformáció hordozza. A közös csatornahasználat már felveti azt a problémát, hogy ki, és mikor használhatja a csatornát, (**egy küldi - mindenki veszi elv**), ezért a **felek azonosításához címzés szükséges**. A forráscím (vagyis a küldő címe) minden esetben ún. **egyedi** (unicast, vagy single-node) cím. A célcímet pedig három csoportba tartozó címzési móddal jelölhetjük ki:

- egyedi (unicast)
- csoport (multicast)
- üzenetszórásos (broadcast).

A csatornán küldött üzeneteket minden gép először olyan mértékben dolgozza fel, hogy a címmező értelmezésével eldönthesse hogy a csomag neki szól-e.

Ezek után az üzenet feldolgozását csak az az állomás folytatja, amelynek címe megegyezett az üzenetben lévő címmel. Ez a kialakítás az egyedi gépcímzés (**unicasting**) mellett csoportcímzés (**multicasting**) használatára is lehetőséget biztosít, amely segítségével több gépnek (csoportnak) szóló üzenetet csak egy példányban kell elküldeni. Ha a hálózat minden résztvevőjének üzenetet küldünk, akkor beszélünk **broadcast**-ról.




*1-3. ábra Üzenetszórásos topológiák*

A gyakorlatban a címek bináris számok. Bizonyos címkombinációkat (pl. a cím egy részét, fenntartjuk **csoportcím**-nek. Ha ezt a címet küldjük ki, akkor a csoporthoz tartozó hosztok mindegyike a magáénak tekinti a hozzá kapcsolt üzenetet.

**0Például,** ha 8 bites címzést használunk, és a legfelső 3 bitet tartjuk fenn a csoportok címzésére, akkor a 3 bit segítségével 8 csoportot képezhetünk, és minden csoportban, a maradék öt bit alapján 32 csoport-tag lehetséges. Célszerű egy hárombites csoportcímet - pl. az 111 kombinációt - olyan üzenetek - csupán egy példányban történő - elküldésére felhasználni, ami minden csoportnak szól. Vagyis az 111 csoportcímű üzenet minden csoportnak szól, de igy akkor csak hét csoport lehetséges. Hasonló módon, a csoporton belüli kommunikáció rövidítésére szolgálhat az 11111 csoporton belüli címzés, amely a csoport minden tagjának szól. (De csak 31 tagja lehet a csoportnak). Akkor vajon kinek szól az 111 11111 című üzenet? (keniknednim...)

Mindenkinek szóló címzésre célszerű egy speciális címet fenntartani: a gyakorlatban ez a minden címpozíción 1-eseket tartalmazó cím.

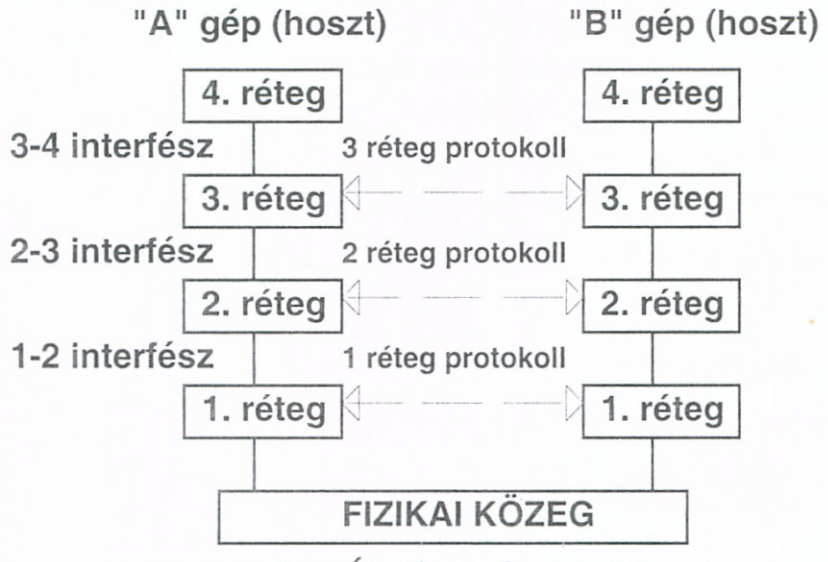
Az ilyen módon működő hálózatok esetén a jellegzetes topológiák a *1-3. ábrán* jelennek meg. Látható, hogy itt is szerepel a gyűrű topológia, de itt a gyűrű a közös üzenetszóró közegként szerepel. Természetesen ehhez a címeket úgy kell kialakítani, hogy a cím, az egyedi gépcímek mellett, a csoportcímzésre vonatkozólag is tartalmazzon egy a csoportra utaló bitsorozatot.

Az adatszórásos hálózatoknál a csatorna használata nem olyan egyszerűen kezelhető, mint pont-pont összeköttetés esetén. Ugyanis elképzelhető, hogy egyszerre egynél több állomás akar adni a csatornán, versenyhelyzetet alakítva ki. Ki kell találni olyan ún. **közeghozzáférési eljárás**-t, amely ezt a versenyhelyzetet feloldja. Ezt a feloldást hívják **arbitráció** – nak. (A közeghozzáféréssel kapcsolatos kérdésekre egy későbbi fejezetben még visszatérünk.)

Fontos két hasonló kifejezés megkülönböztetése: a **topográfia** kifejezés arra utal, hogy a hálózat fizikailag (és pl. a térképen) hogyan helyezkedik el, míg a **topológia** az összekapcsolás felépítését (struktúráját) jelenti.

## 1.2 Hálózati architektúrák

A mai modern számítógép-hálózatok tervezését úgy végzik, hogy a hálózat egyes, egymásra illeszkedő részeit hierarchikusan (egymásra épülően) **réteg**-ekbe (**layer**) vagy más néven **szint**-ekbe (**level**) szervezik, amelyik mindegyike az előzőre épül. Ez **a rétegszemlélet lehetővé teszi az összetett, bonyolult rendszerek egyszerűbb leírását**, a rétegen belüli tervezési megoldások elszigetelt kezelését. A kommunikáció megvalósítása bonyolult feladat, nehéz teljes egészében tárgyalni, ezért célszerű egymásra épülő, részekre bontani. Lényegében ezek a rétegek. A **réteg a kommunikáció egy jól definiált feladatát végrehajtó** megoldás, ami kapcsolatban van, egy szintén jól definiált **réteg interfészen**




*1-4. ábra Általános rétegmodell*

keresztül, az alatta és felette elhelyezkedő tétegekkel, azoknak szolgálatatást nyújt, illetve fogad el. A kommunikációs ellenállomás hasonló rétegével logikai kapcsolatban van, és a vele való kapcsolat pontos formáját a rétegprotokoll írja le. (*1-4 ábrán* mutatjuk be példaként, egy négyrétegű modell felépítését.)

Gondoljuk csak meg: középkorban, amikor két szomszédos ország királya üzenetet kívánt váltani, ezt mai szemmel vizsgálva a rétegszemlélet alapján tette meg. Az egyik király az üzenetét elmondta az udvarmesterének, azt megfelelő szóvirágokkal kiegészítve - esetleg más nyelven - lediktálta az írnoknak, aki ezt szép levélben megírta. A levelet odadták a futárnak, és az valamilyen közlekedési eszközzel vitte el a levelet a szomszéd királyi udvarba, odaadva azt az írnoknak, aki továbbította a levelet a másik király udvarmesterének. Az a királynak felolvasta, estleg lefordította az üzenetet. Nagyon fontos hogy mindenki csak a saját feladatát végezve működött, és a másik udvar megfelelő szintjével került csak kapcsolatba.

**Fontossága miatt, ezt foglaljuk mégegyszer össze:** Hálózati kapcsolatnál az egyik gép k.-adik rétege a másik gép ugyanazon szintű rétegével kommunikál, ezt **rétegprotokoll**-nak nevezik. A réteg az alatta és a feletti réteggel a **réteginterfész**-en keresztül kapcsolódik. Ezt olyan módon teszi, hogy minden egyes réteg az alatta elhelyezkedő rétegnek vezérlőinformációkat és adatokat ad át egészen a legalsó rétegig, ami már a kapcsolatot megvalósító fizikai közeghez kapcsolódik.

**Egy adott kapcsolatnál (kommunikációnál) használt szabályok és megállapodások összességét protokollnak (protocol) nevezzük.**

A fizikai közeg közvetítésével jut el az információ az egyik hoszttól a másik hoszthoz.

A legfontosabb az, hogy ez a réteginterfész minden réteg között tiszta legyen - olyan értelemben - hogy az egyes rétegek működését egyértelműen definiált funkciók írják le. Ez egyszerűvé teszi az adott réteg különböző megoldásainak a cseréjét, hiszen a megoldások az előbbiek alapján ugyanazt a szolgáltatást nyújtják a felettük és alattuk lévő rétegnek, segítve a nyílt rendszerek kialakítását.

**A rétegek és rétegprotokollok halmazát együttesen nevezzük hálózati architektúrának.**

Az architektúra kialakításakor meg kell tervezni az egyes rétegeket a következő elvek alapján:

- Minden rétegnek rendelkeznie kell a kapcsolat felépítését, illetve annak lebontását biztosító eljárással.
- Milyen irányú legyen a kommunikáció a két réteg között:
    - **Szimplex átvitelnél** a csatornán áramló információ csak egy irányú lehet, mindig van adó és van vevő a rendszerben, ezek szerepet nem cserélnek. Ilyen kommunikáció a szokásos rádió vagy TV adás (nem tudunk visszabeszélni...)
    - **Fél duplex** átvitelnél a csatornán az információáramlás már kétirányú, felváltva történik, úgy hogy egyszerre mindig csak az egyik irány foglalja a csatornát. Ilyen átvitel valósul meg nagyon sok rádiós kapcsolatban (pl. CB rádió)
    - **Duplex átvitel** esetén egyidejű két irányban történő átvitel valósul meg, hasonlóan az emberi beszélgetéshez, és technikai példaként a telefont említhetjük meg.
- Milyen legyen a rendszerben a hibavédelem, hibajelzés.

- Hogyan oldható meg a gyors adók, lassú vevők együttműködése (ez a folyamat - vezérlés, angolul **flow control**).
- Ha bizonyos okok miatt az üzenetek hossza korlátozott, akkor a küldés előtt szét kell darabolni csomagokra. Felmerül a kérdés, hogyan biztosítható a helyes összerakásuk.
- Csomagokra bontott üzenetek esetében biztosított-e az üzenetek sorrendjének a helyessége?
- Nagyon sokszor ugyanazon a fizikai vonalon, több csatornán zajlik a párbeszéd. Ez jobb vonalkihasználást eredményez. Hogyan kell ezt összekeveredés-mentesen megoldani?
- Ha a cél és a forrás között több útvonal lehetséges, fontos a valamilyen szempontból optimális útvonal kiválasztása.

Ilyen és ehhez hasonló kérdésekre kell választ adni a tervezés során, és talán már most, kezd világossá válni, hogy a kérdésekre nincs együttesen optimális válasz, ami a későbbiekben leírt megoldások sokszínűségét igazolja.

## 1.3 Hálózatszabványosítás

Minden új dolog kialakulását megelőzi a kutatás, az ehhez kapcsolódó írásos és szóbeli információcserék (cikkek, konferenciák), majd az új dolgot gyártó rendszerek kialakítása. Már az eddigiek alapján is nyilvánvaló, hogy a hálózatok kialakításában (de ez igaz minden műszaki tudományra) alapvető szerepet játszik a szabványosítás.

A szabványok központi szerepet játszanak a fejlődésben, ez teszi a rendszereket nyílttakká, egységeit cserélhetővé. Felmerül a kérdés, hogy mikor célszerű az új dolgokkal kapcsolatos információhalmazt a szabványok által meghatározott útra terelni?

- Ha ez a kutatási szakaszban következik be, ez azt jelenti, hogy esetleg a még nem alapos ismeretek miatt a szabvány nem lesz megfelelő, mivel az új későbbi kutatási eredményeket már nem lehet beilleszteni, és ezért esetleg kedvező megoldásokat kell elhagyni.
- Ha viszont túl későn következik be a szabványosítás, akkor a gyakorlatban már számos egymástól eltérő megoldás kerül megvalósításra, ami az ellenérdekek miatt nehézzé teszi az egységességet igénylő szabványosítást.

Sajnos a műszaki fejlődés számos esete bizonyítja az állításunkat még napjainkban is. Ezért a gyakorlatban a szabványok két családja létezik:

- **de-jure szabványok**, amelyeket nemzetközi szakmai bizottságok deklarálnak, és hivatalos dokumentumokban rögzítenek, és
- **de-facto szabványok**, amelyek elterjedését már egy-egy konkrét megoldás széleskörű használata biztosítja. Példa ez utóbbira a nyomtatók Centronics interfésze, vagy az IBM-PC-ben alkalmazott számos megoldás. Már napjaink, illetve a közelmúlt története a nagysebességű modemek illetve a DVD R+ és R- szabványok

Természetesen számos esetben a **de facto** szabványokat célszerű utólagosan **de-jure** szabványokká alakítani.

A szabványosítás története 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, azaz adott műszaki problémára több elterjedt megoldást tettek szabványossá.

A hálózatokban történő adatátvitel szabványosítását a régebben a **CCITT (Comité Consultatif International Télégraphique et Téléphonique)**, ma már az **ITU (International Telecommunications Union)** nemzetközi szervezet végzi. A szabvány rendelkezik a jelvonalakról azok kialakításáról és funkcióiról.

A telefonvonalakon történő átvitelt a V sorozatú szabványok írják le. Az X sorozatú szabványok rögzítik a digitális hálózatokon történő adatátvitel módját. A szabványoknál esetleg megjelenő „bis" tag utal arra, hogy a szabványt úgy fejlesztették tovább, hogy régebbi berendezések használata is lehetővé váljék. Egy kis információ:

| | | |
|---|---|---|
| E sorozat: | Telefon szabványok | |
| F sorozat: | Távíró szabványok | |
| G sorozat: | Digitális hálózatok (PCM) | |
| I sorozat: | ISDN | |
| Q sorozat: | Digitális elérésű jelzési rendszer | |
| S sorozat: | Távíró terminálok | |
| T sorozat: | Teletex, fax | |
| U sorozat: | Távíró kapcsolás-telex | |
| V sorozat: | Telefonvonalakon történő adat kommunikáció | |
| X sorozat: | Adatátvitel | |
| OSI modell | | (X.200-X.299) |
| Internetworking (Hálózatok közötti kommunikáció) | | (X.300-X.330) |
| Üzenetkezelő rendszerek | | (X.400-X.430) |
| Katalógus szolgálatok | | (X.500) |

## 1.4 OSI modell

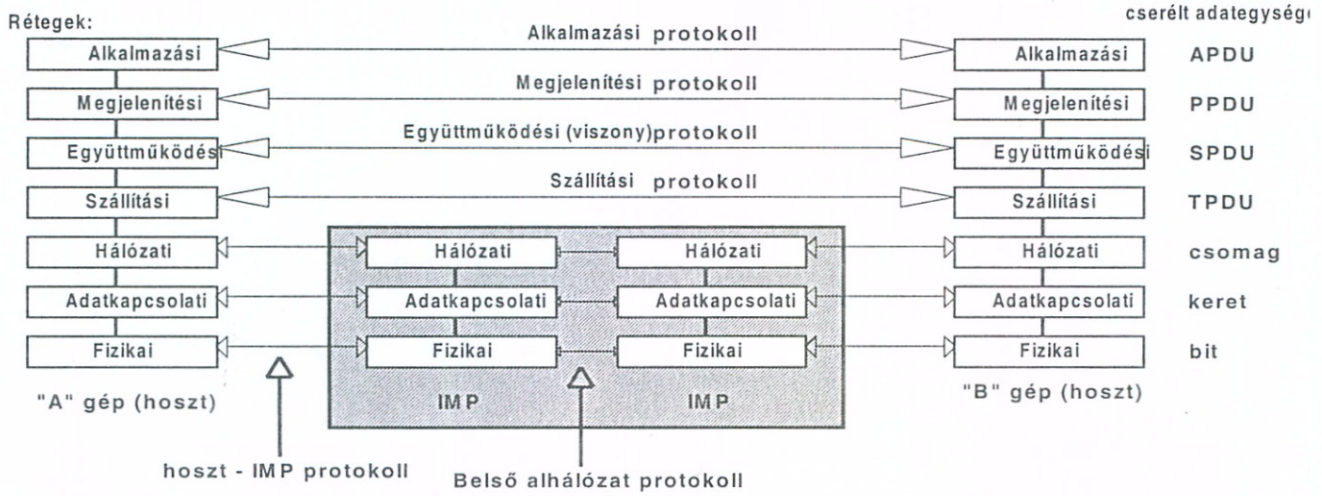
Az előzőkben leírtak alapján már látható, hogy a számítógép-hálózatok rétegezett struktúrájú hierarchikus modell segítségével írhatók le. A **Nemzetközi Szabványügyi Szervezet**, az ISO (International Standard Organization) kidolgozott egy olyan modell-ajánlást (**nem szabványt!**), amelyet ma már minden hálózati, illetve kommunikációs rendszer tervezésekor követnek.

Ezt a megoldást **OSI-modell**-nek hívják. Az OSI az Open System Interconnect — nyílt rendszerek összekapcsolása kifejezés angol eredetijéből alkotott betűszó. Nyílt rendszereknek az olyan rendszereket hívjuk, amelyek nyitottak a más rendszerekkel való kommunikációra. Az, hogy a modell éppen hét rétegből áll, ez a létrehozó bizottságban kialalakított kompromisszum eredménye.

A hétrétegű OSI modell kialakításánál a következő elveket vették figyelembe:

- 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.

Az *1-5 ábrán* megjelenített 7 rétegű OSI modell, legalsó, fizikai rétegei vannak egymással tényleges kapcsolatban. A modell alsó három rétege a hálózattól függ, míg a felső négy réteg mindig alkalmazásfüggő, és mindig az alkalmazást futtató hosztokban valósítják meg (implementálják). A kommunikációban résztvevő felek közötti üzenet-




***1-5. ábra Az OSI modell***

váltást hálózaton átviendő adategységek segítségével végzik. Ezeket – mivel az A és B azonos szintű társréteg között kapcsolatot hordozzák, **protokoll adategységeknek** (Protocol Data Unit (=PDU) nevezik.

A következőkben röviden összefoglaljuk az egy-egy réteg által ellátott feladatokat, a legalsó szinttől felfelé haladva, zárójelben megnevezve a rétegek angol elnevezését is.

**Fizikai réteg (physical layer):**

**Ebben a rétegben zajlik a tényleges fizikai kommunikáció**. A fizikai réteg 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). A fizikai közeg, és az információ tényleges megjelenési formája igen változó lehet: pl. elektromos vezeték esetén, a rajta lévő feszültség értéke, vagy a feszültség változásának iránya. Információt hordozó közeg lehet még: fénykábel, rádióhullám, stb.

Itt kell azt is meghatározni, hogy mennyi legyen egy bit átvitelének időtartama, egy vagy kétirányú kapcsolat kerüljön kialakításra. A kétirányú kapcsolat egyszerre történhet-e? Hogyan épüljön fel egy kapcsolat, és hogyan szűnjön meg? Milyen legyen az alkalmazott csatlakozó fizikai, mechanikai kialakítása?

**Adatkapcsolati réteg (data link layer):**

**Feladata adatok megbízható továbbítása az adó és fogadó között.** (üzenetszórásos, vagy pont-pont kapcsolat kialakításával). Ez általában úgy történik, hogy az átviendő adatokat (amelyek általában bitcsoportba kódolt formában — pl. bájtokban jelennek meg) adatkeretekké (data frame) alakítja (tördeli), ellátja kiegészítő cím, egyéb és ellenőrző információval, ezeket sorrendhelyesen továbbítja, majd a vevő által visszaküldött, az átvitelt igazoló nyugtakereteket (acknowledgement frame) véve ezeket feldolgozza.

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

Az első pillanatban egyszerűnek és teljesnek tekinthető megoldást a gyakorlatban számos kialakuló esemény kezelésével is ki kell egészíteni:

- Hogyan jelezzük a keretek kezdetét és a végét?
- Mi történjék, ha egy keret elveszik?
- Mi történjék, ha a nyugtakeret vész el? (Ilyenkor, ha az adó egy idő múlva újra adja, kettőzött keretek jelennek meg a rendszerben.)
- Mi a teendő, ha az adó keretadási sebessége jelentősen nagyobb, mint a vevőké?

**Hálózati réteg (network layer):**

**A kommunikációs alhálózatok működését vezérli.** Mit is jelent ez? Nagyobb hálózatok esetén a széttördelt üzenetrészeknek (ezeket szokták csomagoknak nevezni) a vevőtől a célba juttatása elvileg több útvonalon is lehetséges, a feladat bizonyos szempontból optimális útvonal kiválasztása. Ez a tevékenység az útvonalválasztás (routing), és több megoldása lehetséges. A hálózaton átküldött információ útvonalának meghatározása kiterjedt hálózatok esetén alapvető jelentőségű.

- 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.

Itt kell megoldani a túl sok csomag hálózatban tartózkodása miatti torlódást, valamint különböző (heterogén) hálózatok összekapcsolását.

**Szállítási réteg (transport layer):**

Feladata a **hosztok közötti átvitel megvalósítása. Valójában nem a gépek, hanem a bennük futó, akár több folyamat (program) kommunikál más gépeken futó programokkal.** A felsőbb rétegek felől érkező üzeneteket szükség esetén kisebb darabokra vágja, és úgy adja át a hálózati rétegnek. Fontos feladata a hoszton belüli, esetleg egyszerre működő kommunikáció címzéseinek a kezelése. Például, ha nagyobb hálózati sebességet akarunk elérni, akkor több hálózati kapcsolatot is ki lehet párhuzamosan alakítani. Ezeket egymástól meg kell különböztetni, címezni kell. Fontos tudnunk, hogy a kommunikáló hosztok sokszor távol vannak egymástól, az üzenetváltásaik több csomópontot érintve történnek. A szállítási réteg feladata annak megvalósítása, hogy erről a két hoszt „mit sem tudva" úgy kommunikáljon egymással, mintha pont-pont összeköttetés lenne közöttük.

**Együttműködési réteg (session layer):**

Más néven: viszony réteg. **A hosztok között adatátvitel megoldása még nem elegendő a folyamatok kommunikációjához, a folyamatoknak is illeszkedniük kell egymáshoz. Ez a felhasználói viszony.** Például: bejelentkezés egy távoli

operációs rendszerbe, állománytovábbítás két hoszt között. Feladata még az 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.

Gondoljuk meg, különben mi történne, ha egy 10 Mbájtos fájl átvitele megszakadna. Az együttműködési réteg szolgáltatása nélkül a megszakadt adatfolyam átvitelét elölről kellen kezdeni, és a megszakadásig átvitt rész elveszne. (A Getright programmal történő adatátvitel is ezt használja ki.)

**Megjelenítési réteg (presentation layer):**

**Feladata az adatok egységes kezelése**. A következő kérdésekkel foglakozik:

**Szám és adatábrázolás – kódolás – adattömörítés – titkosítás.**

A legtöbb alkalmazói program nem 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. 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, illetve az átvitt adatok titkosítása. **Kétségtelen, hogy az utolsónak említett titkosítási kérdéseknek a jelentősége igen megnövekedett a hálózati kommunikáció társadalmivá válásával.**

**Alkalmazási réteg (application layer):**

Mivel ez kapcsolódik legszorosabban a felhasználóhoz, itt kell a **hálózati felhasználói kapcsolatok megoldásait megvalósítani.** Mivel számos terminháltípust 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. A különböző típusú terminálok kezelését ezek után egy kis — a valódi és e hálózati absztrakt terminál közötti megfeleltetését végző — 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 is ismerünk (pl. böngészés).

***Szolgálatok a rétegek között***

Minden rétegben vannak aktív, működő elemek úgynevezett **funkcionális elem**-ek (más néven: **entitás**-ok), **amelyek a rétegtől várt funkciókat valósítják meg.** Kommunikáló berendezésekben, több, egymástól jól elkülöníthető egységet tételezünk fel, amelyek mindegyike különböző feladatot valósít meg. Ez lehet egy program, vagy egy hardver elem (pl. egy be-kimeneti áramkör).

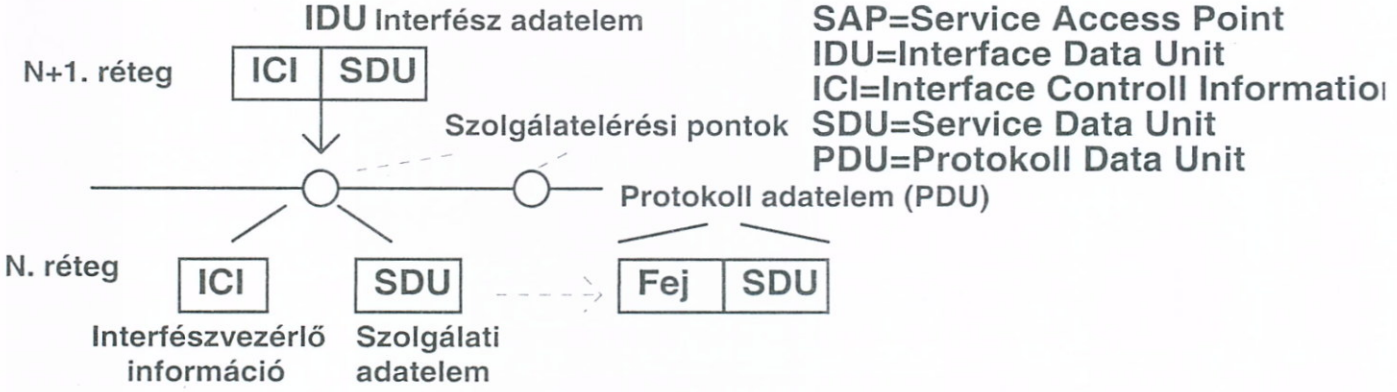
Az egymás alatti vagy fölötti rétegek közötti kommunikáció **szolgálatok** segítségével valósul meg.

A szolgálatok elérése nem tetszőlegesen, hanem pontosan meghatározott módon, a rétegeknél definiált ki/bemeneti szolgálat elérési pontokon **SAP**-okon(=Service Access Point) keresztül valósul meg. Ezek mindig két szomszédos réteg között találhatók.

Például egy telefonrendszerben a SAP a telefon fali csatlakozója, és a SAP címe az a telefonszám, amelyen keresztül a csatlakozóba dugott telefon hívható.

Általánosan fogalmazva az N+1 rétegbeli entitás (funkcionális elem) kapcsolati adatelemet (IDU-t) küld a SAP-on keresztül az N rétegben lévő entitásnak *(1-6. ábra)*. Egy adatcsomag esetén, (amely címet és adatot hordoz) ICI a csomag címe és SDU maga az adat.

Az IDU két részből, a vezérlőinformációból (ICI) és az adatelemből (SDU) áll. Az ICI csak az interfész megfelelő működéséhez szükséges, a tényleges információt az SDU hordozza. Elképzelhető, hogy az adatelemet az N.-edik rétegbeli entitás még szétda-




***1-6. ábra Kapcsolat a rétegek között***

rabolja, és független protokoll adatelemként küldi tovább.

A szállítási, viszony és alkalmazási protokoll adategységekre (Protocol Data Unit=PDU-kra) rendre TPDU, (T=Transport), SPDU (S=Session), és APDU (A=Application) néven hivatkoznak.

A kommunikációt biztosító szolgálatoknak alapvetően két különböző típusa lehetséges: az **összeköttetés-alapú** és az **összeköttetés-mentes szolgálat.**

Az összeköttetés alapú szolgálatok esetén közvetlen kapcsolat van a felek között, míg az összeköttetés-mentes szolgálatban a kapcsolat közegen átviendő adatcsomagok segítségével valósul meg.

### Összeköttetés-alapú szolgálat

A lényegét a telefonrendszer segítségével érthetjük meg. Ha valakivel beszélni akarunk, akkor felemeljük a kagylót, a tárcsázás segítségével a telefonközponton keresztül kapcsolatot létesítünk (azaz felépítjük az összeköttetést), információt cserélünk (azaz használjuk), majd a beszélgetés végeztével letesszük a kagylót (vagyis bontjuk a kapcsolatot).

Tehát a folyamat**: a kapcsolat felépítése, használata, majd bontása,** és az információ átvitel sorrendjét szigorúan az adó határozza meg. Ez azt jelenti, hogy amilyen sorrendben küldjük az információt, a vevő pontosan ilyen sorrendben kapja meg. Az összeköttetés kialakítása időt vesz igénybe, így sok esetben csak akkor célszerű alkalmazni, ha nagyobb mennyiségű információt akarunk átvinni.

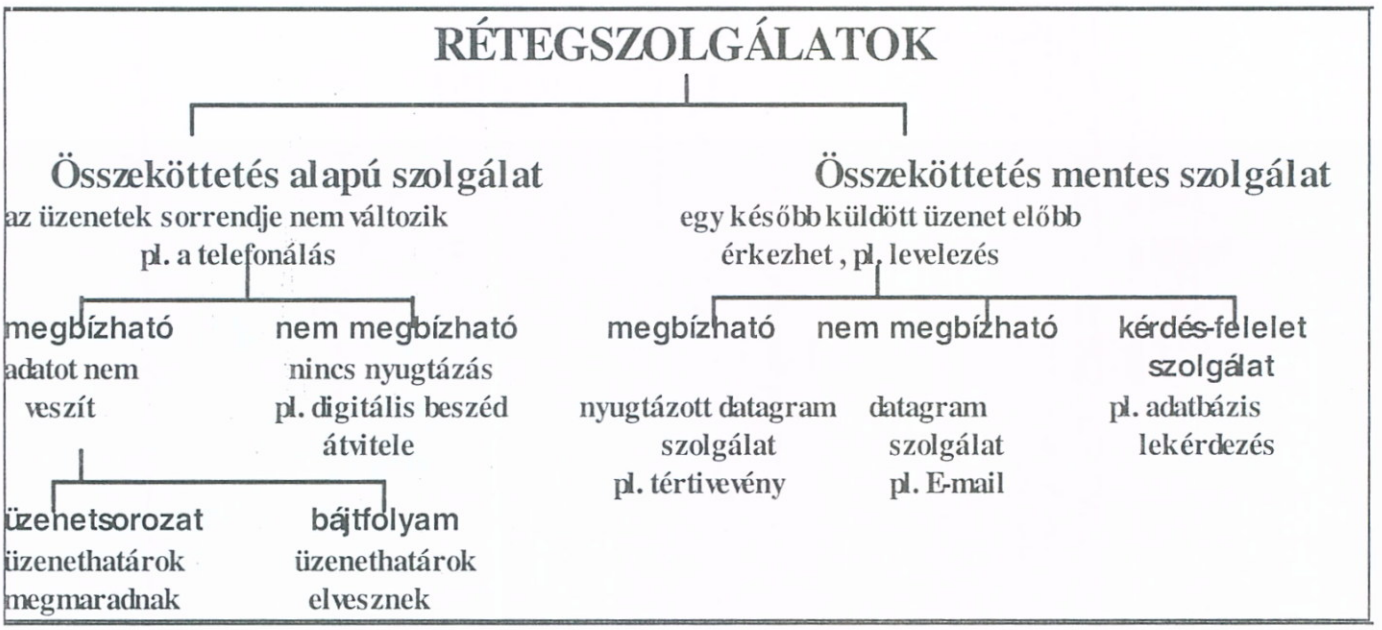
### Összeköttetés-mentes szolgálat

Az információ ilyenkor az adó és a vevő között a vevő címét is tartalmazó információrészek (csomagok) segítségével kerül átvitelre, a levélkézbesítő rendszer működéséhez hasonlító módon. Ilyenkor elképzelhető, hogy a részekre bontott információt a

vevő nem az adó által küldött sorrendben kapja meg, felmerül a csomagok helyes sorrendben történő összerakásának a szükségessége is.

Melyik a jobb megoldás? — kérdezhetnénk. Mindkét megoldást annak megbízhatóságával minősíthetjük, ami azt jelenti, hogy az átvitel során nem vesztünk adatot. A megbízhatóság megvalósításának az a szokásos módja, hogy a vevő az információvétel tényét visszajelzi a küldőnek, azaz nyugtázza (nyugtát küld). Mivel a nyugtázás esetén az adónak meg kell várnia a küldött információ nyugtáját, ezért ez lassítja a kommunikációt.

A gyakorlatban a megbízhatatlan (azaz nem nyugtázott) összeköttetés-mentes szolgálatot datagram szolgálatnak (datagram service) nevezik. A megbízható összeköttetésmentes szolgálat neve: **nyugtázott datagram szolgálat** (acknowledged datagram service).




*1-7. ábra A rétegszolgálatok osztályozása*

Természetesen összeköttetés alapú szolgálatok esetén is megkülönböztethetünk megbízható és a nyugtázást nélkülöző megbízhatatlan szolgálatokat.

Az előbbiek illusztrálására, a rétegszolgálatokat az *1-7. ábrán* foglaltuk össze.

Egy szolgálatot bizonyos alapműveletek (primitívek) segítségével írhatunk le. Ezekkel definiáljuk, hogy egy szolgálat milyen tevékenységet végez el, és milyen jelzést ad tovább egy másik primitívnek. Az OSI modellben a primitívek mindössze négy osztálya lehetséges. Megerősített szolgálatnál ezt mind hasznájuk, míg megerősítetlen szolgálatnál csak a KÉRÉS és BEJELENTÉS primitíveket.

| Primitív | Mit csinál |
|---|---|
| Kérés | Valamilyen tevékenység végrehajtásának kérése |
| Bejelentés | Információ adása eseményről |
| Válasz | Egy eseményre való válaszadás |
| Megerősítés | A kérést kérő informálása |

A köztük lévő kapcsolatokat a *1-8. ábrán* foglaltuk össze.

Ha a kapcsolat létrehozását a **CONNECT,** az adatátvitelt a **DATA** és a lebontást a **DISCONNECT** szavakkal jelöljük, akkor egy összeköttetés-alapú szolgálat nyolc szolgálatprimitívből áll [1]:

CONNECT.kérés - Hívó összeköttetés létesítését kéri (telefont tárcsáz)

CONNECT.bejelentés - Hívó jelez a hívott félnek (kicseng)

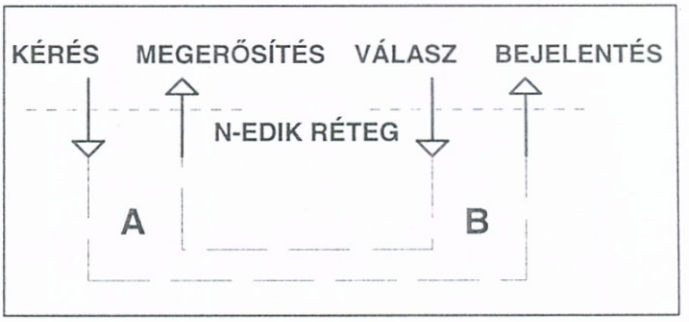
CONNECT.válasz - A hívott fél válasza a hívásra (elfogadja-elutasítja) (felveszi a telefont)

CONNECT.megerősítés - Közli a hívóval, hogy a kérését elfogadta-e (halljuk, hogy a csöngetés végetért)

DATA.kérés - Hívott az adat küldését kéri (valamit kérdezünk)

adatátvitel----------

DATA.bejelentés - Hívott az adat érkezését jelzi a hívónak (valamit válaszol)




*1-8. ábra Összefüggés a primitívek között*

DISCONNECT.kérés - Hívó összeköttetés bontását kéri (letesszük a telefont)

DISCONNECT.bejelentés - Hívott jelez a hívónak hogy elfogadta (leteszi a telefont)

A példában a CONNECT egy megerősített szolgálat (nyugtázott), míg a DISCONNECT megerősítetlen szolgálat (nincs nyugtázva külön a kérés).

## Ellenőrző kérdések: 1. Fejezet

?

1. Sorolja fel milyen előnyökkel, illetve hátrányokkal jár a számítógépek hálózatba kapcsolása!
2. Hogyan épül fel, és mlyen részekből áll egy számítógépes hálózat?
3. Magyarázza el a következő fogalmakat: hoszt, IMP, csatorna!
4. Az összeköttetés kialakítása alapján hogyan csoportosíthatjuk az alhálózatokat?
5. Ismertesse a pont-pont kialakítás jellemzőit, és rajzolja fel a megoldási lehetőségeit!
6. Ismertesse az üzenetszórásos kialakítás jellemzőit, és rajzolja fel a megoldási lehetőségeit! Mi az a közeghozzáférés?
7. Mi az a csoportcímzés?
8. Mi a topográfia és a topológia közötti különbség?
9. Mik azok a hálózati rétegek? Mi a rétegprotokoll?
10. Határozza meg a protokoll fogalmát!
11. Mi a hálózati architektúra?
12. Milyen elveket kell követni a rétegek megtervezésekor?
13. Adja meg a szimplex, félduplex és duplex átvitel meghatározását!
14. Mi a flow controll (áramlás szabályozás)?
15. Miért fontos a hálózatok szabványosítása? Mik azok a de-jure és a de-facto szabványok? Melyek a szabványosítás nemzetközi szervezetei?
16. Fogalmazza meg az OSI-modell filozófiáját! Miért modell, és nem szabvány?
17. Milyen rétegekből épül fel az OSI modell?
18. Mi a fizikai réteg feladata?
19. Mi az adatkapcsolati réteg szerepe?
20. Mit biztosít a hálózati réteg?
21. Milyen célokat valósít meg a szállítási réteg?

22. Mire használják az együttműködési rétegeket?
23. Milyen feladatokat lát el a megjelenítési réteg?
24. Ismertesse az alkalmazási réteg szerepét az OSI modellben!
25. Mi az entitás?
26. Határozza meg a rétegszolgálat fogalmát! Mi az a SAP? Hol helyezkedik el?
27. Milyen részekből áll egy kapcsolati adatelem?
28. Ismertesse az összeköttetés alapú szolgálat lényegét! Mikor célszerű alkalmazni?
29. Ismertesse az összeköttetés-mentes szolgálat lényegét! Mikor célszerű alkalmazni?
30. Adjon példát az összeköttetés alapú és az összeköttetés-mentes szolgálatokra!
31. Mit jelent egy szolgálat megbízhatósága?
32. Mik azok a szolgálati primitívek? Milyen primitív osztályokat definiáltak az OSI modellben?

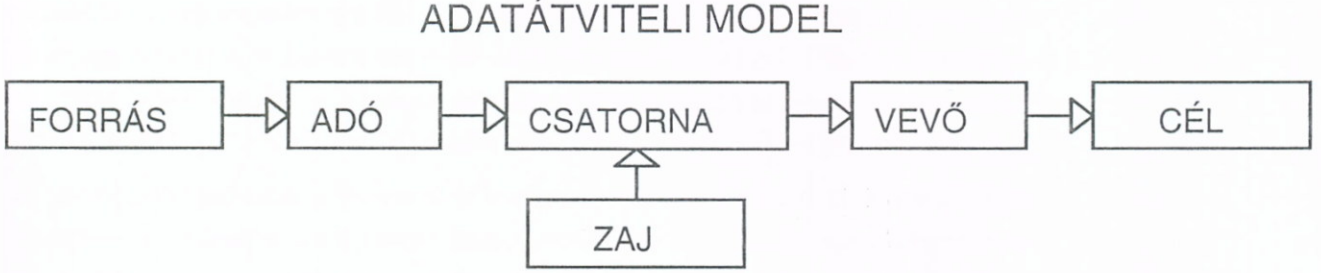
# 2. FIZIKAI ÁTVITELI JELLEMZŐK ÉS MÓDSZEREK

*A kellően fejlett technika már megkülönböztethetetlen a mágiától. (Murphy)*

A fejezetben a fizikai átviteli csatorna jellemzőivel foglakozunk. Bemutatjuk a számítógépes hálózatoknál leggyakrabban alkalmazott átviteli közegeket: a rézvezetőt, a fényt használó megoldásokat (üvegszál), és a rádiós kapcsolatot.

## 2.1 Általános elméleti alapok

Az adatátviteli csatornán történő információátvitel során, az adó megváltoztatja a csatorna fizikai közegének valamilyen tulajdonságát. Ez a közegen továbbterjed, és a




***2-1. ábra Az adatátvitel modellje***

vevő ezt a fizikai közegváltozást érzékeli. Például vezetékek esetén a folyó áram változhat, vagy a feszültség, vagy ha elektromágneses hullámot használunk, akkor a hullám amplitudója, frekvenciája, vagy kezdeti fázisszöge.

Azért kell ilyen általánosan fogalmaznunk, mert például a börtönben rabok úgy kommunikálnak egymással, hogy ott a kommunikációs közeg a vízcsővezeték hálózat, és ez vezeti a kopogtatást, azaz a hangot, vagy gondoljunk az indiánok által használt füstjelzésre.

Sajnos az átvitel során fellépnek a tovaterjedést befolyásoló tényleges fizikai korlátok: A közeg fizikai jellemzői változtathatóságának mértéke, a változtatás lehetséges sebessége, a tovaterjedés során fellépő jelgyengülés.

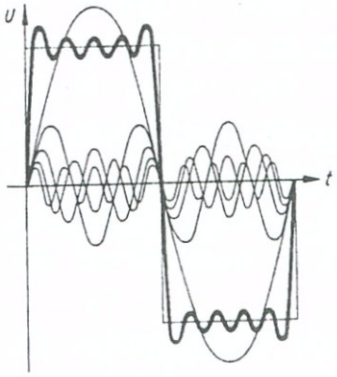
### 2.1.1 Csillapítás, sávkorlátozás, és zaj hatása a jelek átvitelére.

A témához kapcsódó részletes ismereteket a [16] irodalomból sajátíthatjuk el, itt csupán a teljesség a didaktika (mindig a megismert dolgokkal magyarázzuk az újat) szem előtt tartása miatt tekintjük át.

Amikor jelet viszünk át egy fizikai csatornán, három akadállyal kell szembenéznünk: a **csillapítás**-sal (a jel terjedéskor gyengül), **sávkorlátozás**-sal (nem lehet tetszőleges sebességű jelváltozást átvinni), és a **zajok**-kal (a jel tényleges értékét külső hatások megváltoztatják).

A jelek átviteléhez — amely lényegében az átviteli közeg valamelyik jellemzőjének megváltoztatásához (más szóval modulálása) — mindig energia kell. Ennek nagysága

**EMLÉKEZTETŐ:**




a jel összetevőitől függ. Ha jelet a vizsgálat érdekében szinuszhullámok összegének tekintjük (Fourier analízis), akkor a jel Fourier sorának egyes tagjai (az összetevők) az átvitelkor különböző mértékben csillapodnak, amelynek eredménye a kimenő jelalak torzulása.

Mivel a színuszhullámok matematikája nagyon jól kidolgozott, ezért a jeleket a könyebb vizsgálhatóság érdekében úgy tekintjük, mint periódikus szinuszhullámok összegét. Amint az ábrán is látható, egy négyszögjel is felfogható, mint színuszhullámok összege.

Mivel a komponensek közül a jelről a legtöbb információt az első ún. alapharmonikus hordozza, ezért, ahogy növeljük az adatátviteli sebességet, annál nagyobb lesz az alapharmonikus frekvenciája.

Általában 0 és egy **fc** úgynevezett vágási frekvencia között az összetevők lényegében csillapodás nélkül terjednek, míg az e frekvencia feletti összetevők erősen csillapodnak. Az fc a közegtől függ, és például telefonhálózatoknál értéke a beépített szűrők miatt kb. 3000 Hz -nél van.

Ha az alapharmonikus megközelíti a vágási frekvenciát, ennek jelentős csillapodása a jel teljes eltorzulását okozza. Összefoglalva: ***a közeg sávszélessége korlátozza az adatátviteli sebességet.*** Kicsit szerencsétlen módon a digitális hálózatok esetén is ezt az elnevezést használják: a csatornán másodpercenként átvitt bitek számát értik alatta, és bit/sec-ad adják meg az értékét.

A csatornán átküldött jel fontos jellemzője még a jel (S=Signal), és a jelenlévő zaj (N=Noise) teljesítményének aránya, más néven a **jel-zaj viszony (signal to noise ratio)** : **S/N.** A gyakorlatban általában helyette a **10*lg(S/N)** számot adjuk meg (lg-10-es alapú logaritmus), és mértékegysége a dB (decibel). Például 10 dB-es jel-zaj viszony esetén a jel és a zaj aránya 10, 30 dB esetén 1000.

Elméleti számítások alapján megmondható, hogy egy sávkorlátozott zajos csatornán mekkora lehet a maximális adatátviteli sebesség:

***Maximális adatátviteli sebesség= H*log (1+S/N)***

***(itt: H-sávszélesség (Hz) - log - 2-es alapú logaritmus.)***

Például 3000 Hz-es sávszélességű telefonvonalon tipikus 30 dB jel-zaj viszony esetén a sebesség nem lehet több, mint 30 000 bit/sec. Ez az érték a jel mintavételi gyakoriságától, állapotainak számától nem függ!

Valójában a sávszélesség analóg rendszerek esetén használt fogalom: egy adott analóg jel maximális és minimális frekvenciájának a különbségét értjük alatta.

Például az emberi beszéd alsó frekvenciája 300Hz, a felső frekvenciája 3300 Hz, így a sávszélessége: 3300-300=3 kHz

Az átvitelt jellemezhetjük a felhasznált jel értékében 1 másodperc alatt bekövetkezett jelváltozások számával is, amit **jelzési sebesség**nek, vagy közismert néven **baud**-nak nevezünk.

***1 baud = $\log_2 P$ [bit/s]*** *ahol* ***P*** *a kódolásban használt jelszintek száma.*

Például olyan átvitelnél, ahol ezt kétállapotú jelekkel valósítjuk meg, ott a baud és a bit/sec azonos számér-

téket ad, de ha a jelet négy szint felhasználásával visszük át (pl. 0V, 1/3V, 2/3V 1V jelszinteket használva), ott a baud számértéke már csak fele a bit/sec-ban megadott valós adatátviteli sebességnek. (Mert minden jelállapot 2 bitet kódol).

Ezért mindig gondosan, ne egymás szinonimájaként használjuk a baud és bit/s mértékegységeket!

## 2.2 Vonalak megosztása

A következőkben leírtak megértéséhez gondolatban mindig meg kell különböztetnünk a **csatornákat,** amelyeken az információcsere történik, és a felhasznált, tényleges, fizikailag létező, összeköttetéseket biztosító, **vonalakat.**

A csatornák, amelyeken az üzenetek áramlanak, igen jelentős költséggel megépített és üzemeltetett összeköttetéseken (vezeték, rádióhullám) keresztül valósulnak meg. Ezért nem célszerű, ha egy kommunikációs csatorna számára kisajátítunk egy vonalat, mert nagyon sok esetben, a kommunikáció jellegéből fakadóan, nincs folyamatos információcsere rajta, azaz a legtöbb kapcsolatban a vonalhasználat időszakosan jelentkezik. (Gonoljunk csak a telefonálás közbeni beszédszünetekre.) Mivel az ADÓ és VEVŐ oldal számára csak a végeredmény, az információ a fontos, ezért több csatorna is kialakítható egy vonalon, amelynek kialakítására több lehetőség van.

- Az egyik megoldás az, mikor a fizikai közeget osztjuk meg több csatorna között. Ezt, vagyis az **adott vonal felosztását csatornákra több adó, illetve vevő között multiplexelés-nek nevezzük.** A multiplexelés során egy adatvonalat előre meghatározott, rögzített módszer szerint osztunk fel elemi adatcsatornákra. Minden bemenő elemi csatornához egy kimenő csatorna is tartozik. Ezek a későbbiekben bemutatott **frekvenciaosztásos** és az **időosztásos multiplexelési módszerek**, illetve ezek kombinációja.
- A másik lehetőség a vonalak maximális kihasználására, az átviendő információ kisebb adagokra bontása. A vonalon egymás után történik ezek átvitele, majd a darabokból az összerakásuk. Ez az ADÓ és a VEVŐ számára folyamatos összeköttetés látszatát kelti. Ezek az **üzenet** és **csomagkapcsolási** módszerek.
- A harmadik lehetőségként az adatcsatornákat nem egy ADÓ-hoz és egy VEVŐ-höz rendeljük, hanem a kommunikáció szükséglete szerint kapják meg a felek. Ennél a **vonalkapcsolás**-nak hívott módszernél a kapcsolat a kommunikáció részeként jön létre, és a kommunikáció befejezésekor szűnik meg.

A következőkben az említett vonalmegosztási módszereket fogjuk bemutatni.

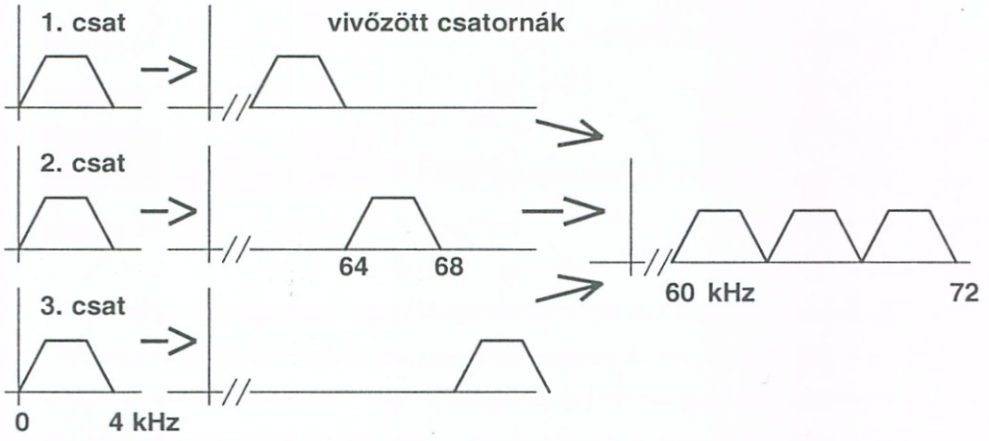
### 2.2.1 Multiplexelés frekvenciaosztással

Frekvenciaosztásos multiplexelés (FDM — Frequency-Division Multiplexing) üzemmódban régebben a távbeszélő hálózatok vivőfrekvenciás rendszereinek szélessávú fővonalait használták. Ma a kábeltelevíziózás, a mobil telefónia, illetve az ADSL technika elképzelhetetlen lenne felhasználása nélkül. Egy széles frekvenciasávban, időben egyszerre haladnak a különböző vivőfrekvenciákra ültetett jelek.

**A módszer alapelve azon a tényen alapul, hogy szinuszos hullámok összegéből bármelyik összetevő egy megfelelő szűrővel leválasztható.**

Az ADÓ oldalon a csatornák jeleit egy-egy vivőfrekvenciára ültetik (a vivőfrekvenciának az amplitudóját a jelekkel modulálják), ezeket összegzik, az összegzett jelet átviszik a VEVŐ oldalra, és ott ezeket szűrőkkel választják szét. A jelösszegzés más néven a multiplexelés, a szűrőkkel történő szétválasztás a demultiplexelés.

A sávszélességet általában az ekvivalens 4 kHz-es beszédcsatornák számával adjuk meg.

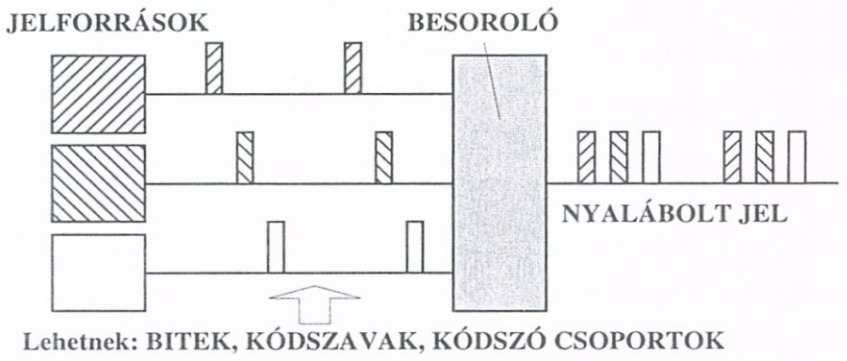



*2-2. ábra Klasszikus telefon frekvenciaosztásos multiplexelés*

Az egyes elemi vivőfrekvencia-tartományok között elválasztó frekvenciarésre van szükség, mivel a különböző jeleket szétválasztó szűrők meredeksége véges. A frekvenciarések jelentősen csökkentik a fővonal sávszélességének kihasználhatóságát. Ráadásul az éppen nem dolgozó berendezésekhez rendelt frekvenciasávok is kihasználatlanok. Az összefoglalóból az is nyilvánvaló, hogy ez a módszer nem igazán alkalmas számítógépek közötti információátvitelre, a csatornák emberi beszédre alapozott sávszélessége miatt. Igazán fontos a módszer akkor volt, amikor a földrészeket tenger alatti kábelekkel kötötték össze. Itt nagy volt annak a jelentősége, hogy egy kábelen minél több beszélgetést vigyünk át egyidőben.

A frekvenciaosztás előnye, hogy a vonalak tetszőleges helyen megcsapolhatók, az egyes alcsatornák egymástól földrajzilag eltolva kezdődhetnek és végződhetnek, a csoportba fogott jeleknek nem szükséges kis körzetben elhelyezkedő adatállomásokhoz tartozni. Természetesen a multiplexelt vonal minden egyes megcsapolásánál külön demultiplexer szükséges.

A módszert a telefontechnikában kiszorították az időosztást alkalmazó eljárások, de a módszert jelenleg is alkalmazzák a mobil telefon, a kábel TV hálózatokban, vagy az ADSL átvitelnél. (ld. később.)




*2-3. ábra Időosztásos multiplexelés*

## 2.2.2 Multiplexelés szinkron időosztással

Digitális átvitelnél az időmultiplex (STDM — Synchronous Time-Division Multiplexing) berendezések a nagyobb sávszélességű adatvonalat időben osztják fel több, elemi adatcsatornára.

**Minden elemi, digitális**

**jeleket szállító adatcsatorna egy-egy időszeletet kap**. A fővonal két végén elhelyezkedő vonali multiplexerek előre meghatározott időben, periodikusan, egymással szinkronban működve összekapcsolják egy-egy rövid időre — néha egyetlen bit, legtöbbször egyetlen karakter vagy bájt, esetleg néhány bájt átviteli idejére — az összetartozó be-, illetve kifutó csatornákat.

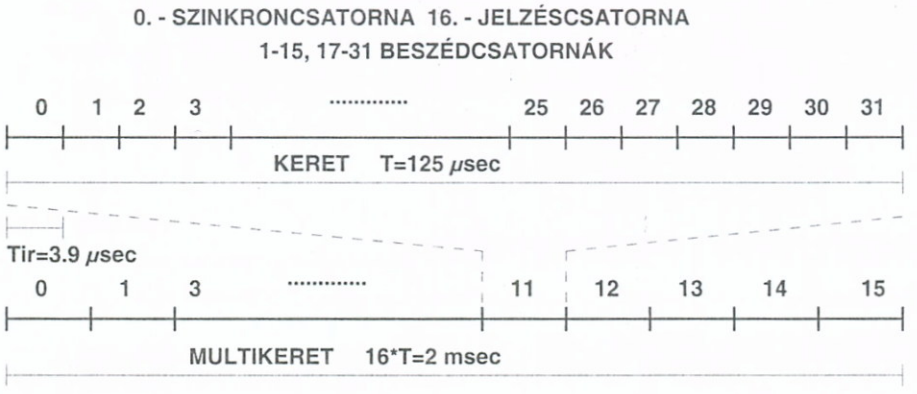
Bármilyen típusú is az átvitel és bármekkora a multiplexelt információ-egység, szükség van arra, hogy a vonal két végén elhelyezett multiplexerek szinkronizmusát biztosító vezérlő jeleket is elhelyezzük az információ-egységek között. Ezek a szinkronjelek csökkentik a fővonal kihasználhatóságát. A frekvenciaosztással és időosztással működő multiplexerek egyaránt akkor felelnek meg jól rendeltetésüknek, ha jelenlétük nem befolyásolja az adatkapcsolat szintű vezérlést. (Az adatkapcsolati réteg számára egy fizikai rétegnek tekinthetők.)

Például a telefontechnikában ma már általánosan használt **PCM (Pulse Code Modulation)** - impulzus kód moduláció - esetén az eljárás a következő:

A mintavételezés (mivel a telefon sávszélessége 300...3400 Hz, és a Nyquist elv alapján, a maximális frekvencia legalább kétszeresével kell mintavételezni) szokásos értéke fv=8000Hz, illetve a periódusidő T = 125 $\mu$sec. A mintavétel 8 bites felbontással történik, azaz 256 lépcsőből áll és logaritmikus léptéket használnak. Ennek az az oka, hogy az emberi fül is ilyen: tízszeres hangnyomást hallunk kétszer erősebbnek. A lényeg: **8000 db 8 bites mintát kell átvinnünk másodpercenként.**

A PCM átvitelben mivel minden impulzushoz n = 8 bit tartozik, az átviteli sebesség 8*8000 = 64 kbit/s. Vagyis, amíg a beszéd analóg átviteléhez 300-3300 Hz-=3kHz-es sávszélesség elegendő, ugyanezen beszéd digitális át-vitele 64 kbit/s-os adatátviteli sebességet igényel, igaz, csak kétállapotú je-leket használva.

Multiplexelés esetén a CCITT szabvány szerint az ún. primer csoport N = 32 csatornával. Az átviteli sebesség: N*n* $f_{minta}$ = 32*8*8000 = 2048 kbit/s.




*2-4. ábra PCM csatornák kiosztása*

Egy csatornára jutó időrés Tir = T/32 = 3.9 $\mu$sec, és mivel 8 bitet tartalmaz, egy bit időtartama Tir/8 = 488 nsec. Az egység, amelyen belül minden csatorna átvitelre kerül, a multikeret.

A digitális multiplexelésnek ezt a szintjét Európában E1-el jelölik és 32 darab 64 kbit/s-os csatornából áll, és összesen 2048 kbit/s adatátviteli sebességű. Az USA-ban ennek megfelel a T1-el jelölt kialakítás, amely 24 darab 64 kbit/s-os csatornából áll, és ez 1544 kbit/s adatátviteli sebességű.

Egy másik példa: audio CD-knél a HIFI sztereo hangminőség 22.1 kHz-es sávszéles-

séget igényel, ezért a mintavétel 44.2 kHz-el történik sztereo csatornánként, a jobb hangminőség miatt 16 bites felbontással. Ezért az átviteli sebesség: 2*44 200 minta/sec*16 bit/minta=141.4 kbit/sec. Ezt szokták egyszeres CD sebességnek hívni. (Audio DVD lemezeknél: a mintavétel 192 kHz, 24 bites felbontás és 5+1 csatorna kerül rögzítésre.)

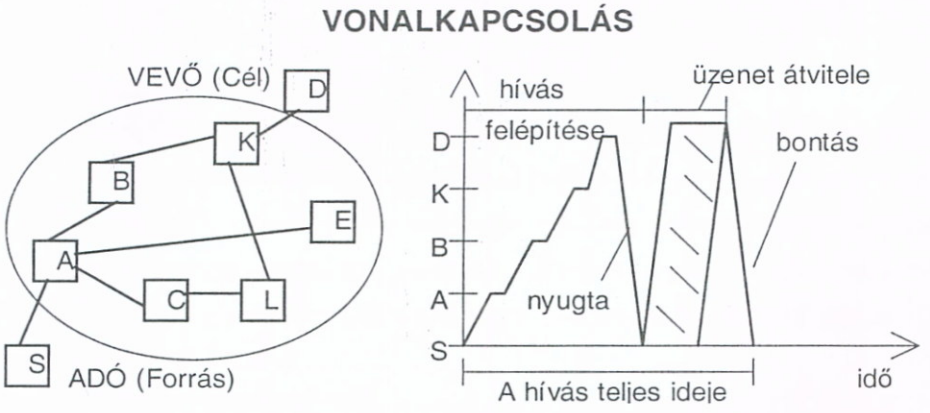
Vegyük észre, hogy az adatátviteli sebességnek nagyon magas a felső korlátja. Lényegében a mindenkori műszaki színvonal határozza meg azt, hogy milyen rövid idő alatt tudunk egy bitet kibocsátani, és milyen rövid időtartamú bitet tudunk helyesen detektálni.

## 2.2.3 Vonalkapcsolás

Az ADÓ és a VEVŐ közti összeköttetés megteremtésére ki kell alakítani azt az útvonalat, amelyeknek részei kapcsolóközpontokon keresztül vannak összekötve.

Első lépésben fizikai kapcsolat létesül az ADÓ és VEVŐ között, ami az összeköttetés idejére áll fenn. Az összeköttetésen keresztül megvalósul az adatátvitel, majd annak befejeztével a kapcsolat lebomlik. A vonalkapcsolás esetén a rendelkezésre álló teljes sávszélességet csak akkor tudjuk kihasználni, ha az információáramlás folytonos.

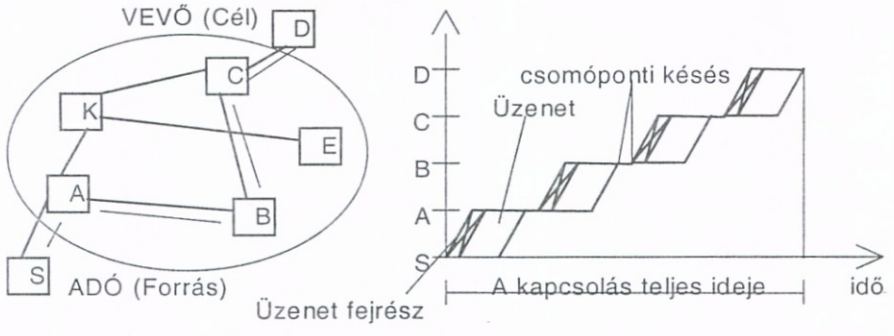
Az ábrából az is látható, hogy elvileg különböző útvonalakon is létrejöhet a kapcsolat a két végpont között, ami a kapcsolat megbízhatóságát növeli.




*2-5. ábra Vonalkapcsolás elve*

A folyamatot a távbeszélő technikában hívásnak nevezik. Fontos tény, hogy az információátvitelt meg kell előznie a híváskérés hatására létrejövő összeköttetés felépítésének. Előnye az, hogy ténylegesen fizikai összeköttetést hozunk létre. Ezek után a két állomás úgy képes kommunikálni, mintha pont-pont összeköttetés valósult volna meg közöttük.

Ilyenkor az adatok késleltetését már csak az elektromágneses jel terjedési ideje határozza meg, amely kb. 6 msec 1000 km-enként. Hátránya a kapcsolat létrehozásához szükséges sokszor jelentős időtartam, és az, hogy ilyenkor a csatorna kisajátítja a vonalat. Ha a csatorna nem teljes kapacitással üzemel (telefonnál: hosszú csend), akkor ez a vonal kihasználtságát rontja.

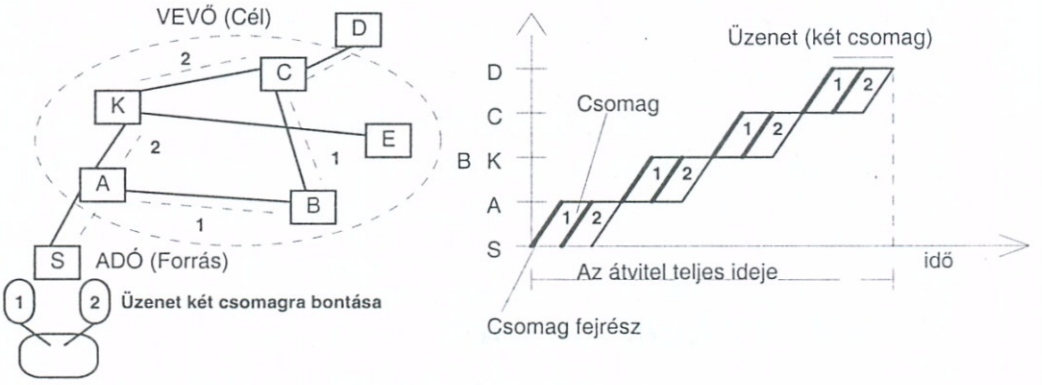



*2-6. ábra Üzenetkapcsolás elve*

## 2.2.4 Üzenet és csomagkapcsolás

Ilyenkor nincs előre kiépített út az ADÓ és a VEVŐ között. Az ADÓ az elküldendő adatblokkját elküldi az első IMP-nek, az pedig továbbküldi a következőnek, egészen a VEVŐ hoszt-hoz kapcsolódó IMP-ig. Az ilyen hálózatok a tárol és továbbít (store and forward) hálózatok. **Az üzenetkapcsolás esetén nincs az adatblokk méretére korlátozás,** ami nagy tárolókapacitású fogadó, és továbbító IMP-ket igényel a teljes átviteli szakasz csomópontjain.

Másik hátránya az, hogy egy nagy üzenet akár percekre lefoglalhatja a közreműködő IMP-ket és a köztük lévő átviteli csatornát.




*2-7. ábra Csomagkapcsolás elve*

Ezért gyakrabban használatos (számítógépes hálózatoknál csaknem kizárólagosan használt) az a módszer, mikor az **átviendő adatblokk méretét korlátozzuk, és csomagokká bontjuk.** *(2.7 ábra)* Az S pontban az ADÓ feldarabolja az üzenetet azaonos méretű csomagokra, és úgy továbbítja. D pontban a VEVŐ összeállítja a csomagokból az eredeti üzenetet.

A csomagkapcsoló hálózatok hatékonyan alkalmazhatók interaktív forgalom (ember-gép kapcsolat) kezelésére is, mivel biztosítják, hogy bármelyik felhasználó csupán néhány ezredmásodpercre sajátíthat ki egy vonalat.

A csomagkapcsolás nagyon hatékonyan képes a vonalak kihasználására, mivel adott két pont között összeköttetést több irányból érkező és továbbhaladó csomag is használja. Másrészről fennáll annak a veszélye, hogy a bemenő adatforgalom csomagjai úgy elárasztanak egy IMP-t, hogy korlátozott tárolókapacitása miatt csomagokat veszít. **Míg vonalkapcsolás esetén az üzenet lényegében egyben kerül átvitelre, csomagkapcsoláskor a csomagok sorrendje megváltozhat, és a sorrendhelyes összerakásukról is gondoskodni kell.**

# 2.3 Vezetékes átviteli közegek

A számítógép-hálózatok vonatkozásában az összekötő átviteli közeg természetétől függően megkülönböztetünk **fizikailag összekötött (bounded)** és **nem összekötött (unbounded)** kapcsolatokat. Az előbbihez tartoznak az elektromos jelvezetékek, az optikai kábel, míg az utóbbira jó példa a rádióhullám, (mikrohullámú) illetve az infravörös vagy lézeres összeköttetés. Mindegyiknek van előnye és hátránya. Kétségtelen, hogy a már kialakított telefonvonalak megléte miatt jelenleg a legszélesebb körben a rézvezetékeket használó huzalpárok terjedtek el.

A jelenlegi, a hálózatokat fokozottabban használó világban a fentieket mind mérlegelni kell, és ha már egy meglévő infrastruktúrát kell hálózati kapcsolatokkal kiegészíteni, sokszor csak a nem fizikailag összekötött megoldások jöhetnek szóba, hiszen egy for-

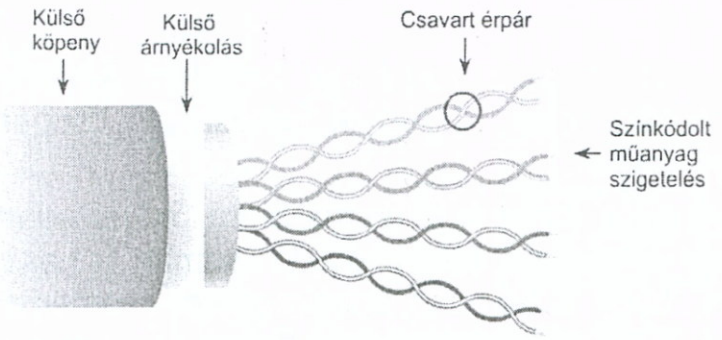
galmas főút két oldalának összekötése — ha nincsenek kábelalagutak — kábelekkel szinte lehetetlen. Mégis számos előnye van a vezetékes hálózati kapcsolatoknak: megbízható, nagyobb sebességet lehet elérni, mint a vezeték nélküli összeköttetéseknél, a lehallgatás elleni védelem is jobban megoldható.

Ha valaki figyeli a hálózatok fejlődését, egyre nagyobb jelentősége van a vezetéket nem használó megoldásoknak. A fizikailag nem összekötött rendszerek mozgékonyak, könnyen áthelyezhetők, a hosszú kábelcsatornák helyett elég egy-két antennaoszlopot kialakítani, de mivel a jel a széles környezetben terjed, az adatbiztonságra fokozottan kell ügyelni a lehallgatás könnyebb kivitelezhetősége miatt.

Bár vezetékes összeköttetésnél független vezetékekből kialakított párhuzamos huzalpárok használata is elképzelhető, de igen rossz csillapítási és zajfelvevő tulajdonságai miatt ezt a gyakorlatban csak kisebb távolságokra használják. (pl. telefonvezetékek)

## 2.3.1 Réz alapú kábelek

Gyorsabb jelváltozásoknál az ilyen vezetékpár antennaként jeleket sugároz a környezetébe. A probléma megoldására a gyakorlatban két kialakítást használnak: a csavart érpárt, illetve az árnyékolt (koax) kábeles megoldást.




*2-8. ábra STP kábel*

### *Csavart érpár (UTP, STP)*

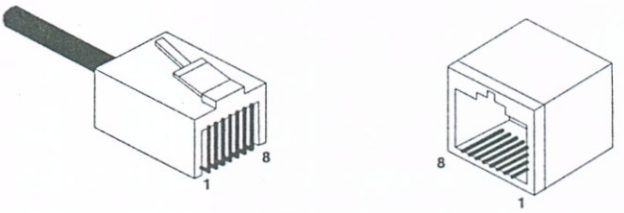
A csavart, vagy más néven sodrott érpár (Unshielded Twisted Pair = UTP) két szigetelt, egymásra spirálisan felcsavart rézvezeték. Ha ezt a sodrott érpárat kívülről egy árnyékoló fémszövet burokkal is körbevesszük, akkor árnyékolt sodrott érpárról (Shielded Twisted Pair = STP) beszélünk. *(2-8. ábra)* A csavarás a két ér egymásra hatását küszöböli ki, jelkisugárzás nem lép fel.

Pontosan a sodrás biztosítja azt, hogy a szomszédos vezetékpárok jelei ne hassanak egymásra (ne legyen interferencia). Az épületekben lévő telefonhálózatoknál is csavart érpárokat használnak. A felhasználásuk számítógép-hálózatoknál is ebből a tényből indult ki: ezek a vezetékek már rendelkezésre állnak, nem kell új vezetékeket kihúzni a munkahelyekhez.

Általában több csavart érpárt fognak össze közös védőburkolatban. Ma már akár 1000 Mbit/s adatátviteli sebességet is lehet ilyen típusú vezetékezéssel biztosítani. (Terjed az 1Gbit/sec sebességet biztosító Ethernet technológia!)

Alkalmasak mind analóg mind digitális jelátvitelre is, áruk viszonylag alacsony. Az UTP kábelek minősége a telefonvonalakra használtaktól a nagysebességű adatátviteli kábelekig változik.

**Általában egy kábel négy csavart érpárt tartalmaz közös védőburkolatban.** Minden érpár eltérő számú csavarást tartalmaz méterenként, a köztük lévő áthallás csökkentése miatt. Szabványos osztályozásuk a táblázatban van összefoglalva, elnevezésük: CAT1...CAT7. Az adó és vevő adatjelek a sodrott érpárú szegmensen polarizáltak, az egyik érpár vezetője a pozitív (+) a másik a negatív (-) jelölésű. A kategóriák közötti egyetlen lényeges különbség a csavarás sűrűsége. Minél sűrűbb a csavarás, annál nagyobb az adatátviteli sebesség (és a méterenkénti ár...). Az UTP kábeleknél általában az RJ-45 típusjelű telefoncsatlakozóhoz hasonlító kivitelt használják a csatlakoztatásra.




| *Csatlakozási Pont* | *Hozzárendelés* |
|---|---|
| *1* | *Kimenő adat +* |
| *2* | *Kimenő adat -* |
| *3* | *Bejövő adat +* |
| *6* | *Bejövő adat -* |
| *4,5,7,8* | *Más célokra fenntartva* |

***2-9. ábra Csavart érpár bekötése RJ-45-ös csatlakozóban***

Egyik érkettős adásra, a másik vételre szolgál, akár egy időben (duplex átvitel). Az érpárban a vezetékeket sodorni kell a teljes szegmenshosszon a jelátviteli tulajdonságainak javítása érdekében. A megengedett legnagyobb szegmenshossz 100 méter, ami lehet rövidebb vagy hosszabb a kábel minőségétől függően. Nagyobb adatátviteli sebességeknél a más célokra fenntartott érpárakat is használják az átvitel során. A következő táblázatban a sodrott érpár technikát foglaltuk össze:

**A sodort érpárú vezetékek osztályozása, ISO 11801, illetve TIA 568A, 568B szabványban rögzített átviteli sebesség és üzemi frekvencia értékekkel:**

| | |
|---|---|
| **CAT 1** | **Hang átvitel telefon** |
| **CAT 2** | **4 Mbit/s – Local Talk** |
| **TIA 568A** | |
| **CAT 3 (16 MHz)** | **10 Mbit/s – Ethernet (10BaseT)** |
| **CAT 4 (20 MHz)** | **20 Mbit/s – Token Ring (16 Mbit/s)** |
| **CAT 5 (100 MHz)** | **100 Mbit/s – Fast Ethernet** |
| **TIA 568A – TSB95** | |
| **Category 5 (defines new tests)** | |
| **TIA 568B (2001)** | |
| **CAT 3** | |
| **CAT 5E (100 MHz)** | **1 Gbit/s – Gigabit Ethernet** |
| **CAT 6 (250 MHz)** | **ATM hálózatok** |
| **CAT 7** | **ATM, és egyéb speciális hálózatok** |

| |
|---|
| **ISO 11801:1995 or "1st Edition"** |
| Class C (16 MHz) |
| Class D (100 MHz) |
| **ISO 11801 : 2000 or "1.2 Edition"** |
| Class C |
| Class D ".1.2" (100 MHZ) |
| **ISO 11801 : "2nd Edition" (Pending 2002?)** |
| Class C (16 MHz) |
| Class D (100 MHz) |
| Class E (250 MHz) |
| Class F (600MHz) |

### *Koaxiális kábelek*

A másik vezeték kialakítási megoldás a koaxiális kábelek használata. Felépítése a *2-10. ábrán* látható. Széles körben két fajtáját alkalmazzák:

Az egyik az **alapsávú koaxiális kábel**, amelyet digitális jelátvitelre alkalmaznak, a másik a **szélessávú koaxiális kábel**, amelyet pedig analóg átvitelre használnak.

- **Alapsávi átvitelről** akkor beszélünk, amikor magát a fizikai jelváltozást visszük át a kábelen, pl. a telefonbeszélgetésnél a mikrofonba beszélve, a mikrofonáram változást, vagy videojel átvitelét TV-s rendszereknél.
- **Széles sávú átvitel** esetén az információt hordozó jellel moduláljuk egy állandó frekvenciájú vivőjel amplitudóját, és ezt a jelet visszük át. A frekvenciaosztásos multiplexelés elvét alkalmazva, így egy vezetéken több ilyen modulált jel vihető át egy időben (pl. kábelTV rendszerek)

Az alapsáv elnevezés még abból az időből származik, amikor telefonbeszélgetésekre alkalmazták a kábeleket, és itt a sávszélesség az érthető emberi hangnak megfelelő kb. 0-4 kHz volt. A televíziós rendszerek megjelenésével a több tv csatorna video és hangjelének az átviteléhez jelentősen nagyobb sávszélesség kellett, ezeket a szélessávú kábelekkel oldották meg.

A koaxiális kábeleknek három igen lényeges jellemzője van: **a hullámellenállása** (Z0), **a hosszegységre eső késleltetési ideje** és a **hosszegységre eső csillapítása**.

- A leggyakrabban az 50 Ω és 75 Ω **hullámellenállású** kábelt használnak: az 50 Ω-ost alapsávú, a 75 Ω-ost szélessávú és kábelTV hálózatokban. Ez utóbbival azonban alapsávúként is találkozhatunk, főként akkor, ha a hálózat alapsávúként és szélessávúként egyaránt működhet.
- A **késleltetési idő** a kábel szigetelésének permittivitásától (dielektromos állandójától) függ. A nagy késleltetési idő a hálózatok működése szempontjából hátrányos, ezért csökkentésére törekednek. Igyekeznek minél kisebb permittivitású szigetelőanyagot alkalmazni, de ezen túl, ez még az anyag szerkezetének lyukacsossá tételével tovább csökkenthető.
- A **kábel okozta veszteség** az ohmos komponensekből, a dielektrikumban keletkező, és a elektromágneses sugárzás okozta veszteségekből tevődik össze. A frekvencia növekedésével a bőr (skin) hatás is jelentkezik. A tömör központi huzallal készülő kábel késleltetése és csillapítása kisebb, mint a több összesodrott fémszálat alkalmazóé (ha egyébként minden más változatlan). A tömör huzalú kábel viszont merevebb, mint a sodrott változat.

### Alapsávú koaxiális kábelek

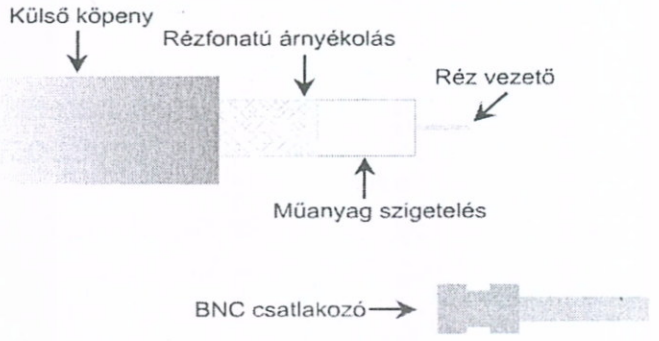
Az alapsávú koaxiális kábeleket leggyakrabban helyi számítógép-hálózatok kialakítására alkalmazzák. Az alapsávú koaxiális kábelek jellemző maximális adatátviteli sebessége 100 Mbit /sec 1 km-es szakaszon. Az átvitel sávszélessége nagymértékben függ a távolságtól. Tehát kisebb távolságon nagyobb sebesség is elérhető.

A digitális átviteltechnikában **vékony koaxiális kábeleket** Arcnet és Ethernet hálózatok kialakításánál alkalmaznak. Csatlakozásra BNC dugókat és aljzatokat használnak. Mivel a csatlakozások mindig a kábelezés

legkritikusabb pontjai, célszerűbb, a biztonságosabb kötést biztosító sajtolt (krimpelt) csatlakozók használata, a csavaros vagy forrasztott BNC csatlakozókkal szemben.

A **vastag koaxiális kábeleket** is az Ethernet hálózatok kialakításánál alkalmazzák. A vastag kábel előnye, hogy lényegesen kisebb a csillapítása, mint a vékony változatnak, ezért nagyobb távolságok hidalhatók át vele.

Mivel a kábel vastagságánál fogva merev, ezért nehezen szerelhető. Csatlakozások kialakítása is speciális: úgynevezett **vámpír csatlakozókat** alkalmaznak. Ez a kábelre kívülről rásajtolt csatlakozó, amely a rásajtoláskor úgy szúrja át a kábel szigetelését, hogy a külső árnyékolással és a belső vezetékkel is önálló elektromos érintkezést biztosít.




*2-10. ábra Koaxiális kábel felépítése*

## Széles sávú koaxiális kábelek

A másik fajta koaxiális kábelrendszer, a kábeltelevíziózás szabványos kábelein keresztül, analóg átvitelt teszi lehetővé. Mivel ezek a szélessávú hálózatok a szabványos kábeltelevíziós technikát használják, ezért az analóg jelátvitelnek megfelelően — amely sokkal kevésbé kritikus, mint a digitális — a kábelek közel 100 km-es távolságig 300 MHz-es (időnként 450 MHz-es) jelek átvitelére alkalmasak. Digitális jelek analóg hálózaton keresztül átviteléhez minden interfésznek tartalmaznia kell egy konvertert, amely a kimenő digitális jeleket analóg jelekké, és a bemenő analóg jeleket digitális jelekké alakítja. Egy 300 MHz-es kábel tipikusan 150 Mbit/s-os adatátvitelt tesz lehetővé. Mivel ez egy csatorna számára túlzottan nagy sávszélesség, ezért a szélessávú rendszereket általában több csatornára osztják.

Az egyes csatornák egymástól függetlenül képesek pl. analóg televíziójel, csúcsminőségű hangátviteli jel, vagy digitális jelfolyam átvitelére is.

Az alapsávú és a szélessávú technika közötti egyik legfontosabb különbség az, hogy a szélessávú rendszerekben analóg erősítőkre van szükség. Egykábeles rendszerben egyetlen kábelen két különböző frekvenciatartomány van az adó (adó-sáv) és a vevő (vevősáv) részére.

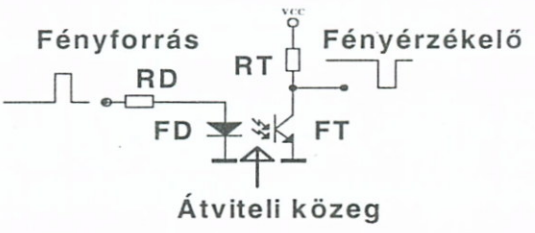
Ezek az erősítők a jelet csak az egyik irányba tudják továbbítani, ezért csak szimplex adatátvitelt képesek megvalósítani. A probléma megoldására kétféle szélessávú rendszert fejlesztettek ki: a kétkábeles és az egykábeles rendszert.

A kétkábeles rendszerben két azonos kábel fut egymás mellett. A két kábelen ellentétes irányú az adatforgalom.

A szélessávú rendszerek nagy előnye, hogy egyazon kábelen egyidejűleg egymástól függetlenül többféle kommunikációt valósíthatunk meg, hátránya azonban a telepítés és az üzemeltetés bonyolultsága és a jelentős költségek.

## 2.3.2 Üvegszálas kábel

A jelenlegi legkorszerűbb vezetékes adatátviteli módszer, az üvegszál technológia alkalmazása. Az információ fényimpulzusok formájában terjed egy fényvezető közegben, praktikusan egy




*2-11. ábra Optikai adatátvitel alapelve*

üvegszálon. Az átvitel három elem segítségével valósul meg: **fényforrás — átviteli közeg — fényérzékelő.**

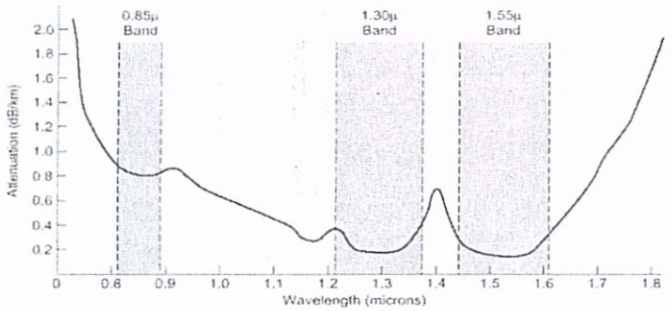
A fényforrás egy LED-dióda, vagy lézerdióda. Ezek a fényjeleket a rajtuk átfolyó áram hatására generálják. A fényérzékelő egy fotótranzisztor vagy fotodióda, amelyek vezetési képessége a rájuk eső fény hatására megváltozik.

Az átviteli közeg egyik oldalára fényforrást kapcsolva a közeg másik oldalán elhelyezett fényérzékelő a fényforrás jeleinek megfelelően változtatja a vezetőképességét.

Az elektronikában használt optikai kapu működése jól illusztrálja a működési elvet: A fotodiódára az RD ellenálláson keresztül kapcsolt pozitív feszültség a LED-diódát nyitja, az átfolyó áram hatására fényt bocsát ki.

Az átviteli közegen (ami esetünkben egy átlátszó műanyag) a fény átjutva az FT tranzisztort kinyitja és a felső pontjának feszültsége közel nulla lesz. Az, hogy ez a módszer nagyobb távolságokon is működjön, átviteli közegként vékony üvegszálat kell alkalmazni,és minimálisra kell csökkenteni a fényveszteségeket.

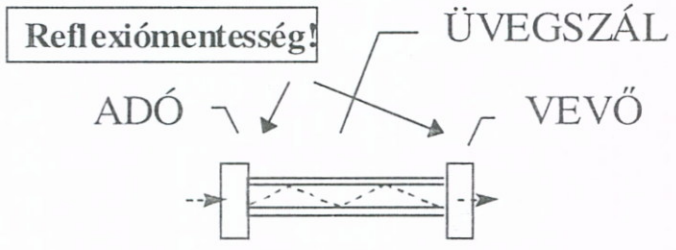
Fényveszteség három részből áll: a két közeg határán bekövetkező **visszaverődés (reflexió)**, a közegben **létrejövő csillapítás** (amely a hullámhossz függvényében minimumokat mutat (*2-12. ábra*)), és a közegek **határfelületén átlépő fénysugarak.** Az első hatás a határfelületek gondos összeillesztésével minimálisra csökkenhető. Döntő jelentőségű az a tény, hogy a csillapítás nem az üveg alapvető tulajdonsága, hanem ezt az üvegben lévő szennyeződések okozzák. A csillapítás megfelelő anyagválasztással minimalizálható. A csillapítás, vagyis a fényerősség csökkenése a használt fény hullámhosszától is függ. Az üvegszálas kábeleknek két alapvető típusa van:




***2-12. ábra Csillapítás-hullámhossz függvény***

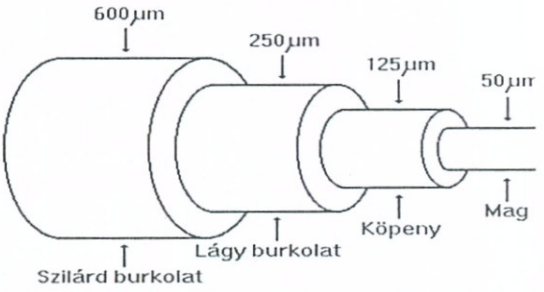
- többmódusú üvegszál (multimode fiber)
- egymódusú üvegszál (monomode fiber)

A közeg határfelületén való átlépés megakadályozására a megoldás az optikában jól ismert teljes visszaverődés jelensége. Ha a közeg határfelületére érkező fénysugár beesési szöge elér egy kritikus értéket, akkor a fénysugár már nem lép ki a levegőbe, hanem visszaverődik az üvegbe. **A** *2-13. ábrán* mutatjuk be az elmondottakat. Az üvegszálban az adóból kibocsátott számos fénysugár fog ide-oda verődni. Az ilyen optikai szálakat többmódusú üvegszálnak (multimode fiber) nevezik.




***2-13. ábra Teljes visszaverődés az üvegszálban***

Ha azonban a szál átmérőjét közel a fény hullámhosszára csökkentjük, akkor a fénysugár már verődés nélkül terjed. Ez az **egymódusú üvegszál** (single (mono) mode fiber). ADÓ-ként ilyenkor lézerdiódát kell alkalmazni, de sokkal hatékonyabb, nagyobb távolságú összeköttetés alakítható ki segítségével.




***2-14. ábra: Optikai kábel egy szálának a felépítése***

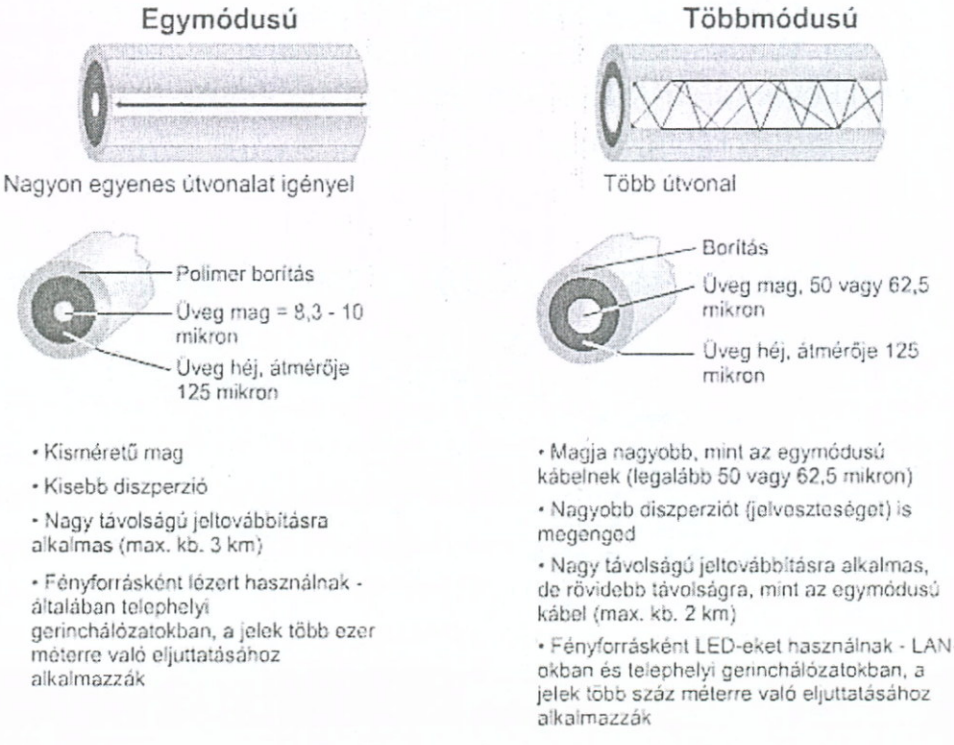
Érdekes és gyakorlati feladat az üvegszálak összetoldása, amelyre három módszer van:

- Csatlakozók használata (10-20%-os veszteség).
- A megtisztított két üvegszálvéget egy olyan vékony csőben tolják össze, amelyben az üvegszállal megegyező törésmutatójú folyadék van. Ilyenkor a fény a folyadékon is áthaladva, keveset csillapodik csak (10%).
- A két kábel összeforrasztása (üvegolvasztással). Igen kicsi veszteség. Ha nem sikerül a forrasztás jól, azaz a veszteség nagy, akkor a forrasztó berendezés automatikusan újra eltöri az összforrasztott szálat.

Tudni kell, hogy az üvegszállal végzett munka nagyon veszélyes, testbe kerülve, (pl. szem) ez igen kisméretű idegen anyag.

Jelenleg a nagytávolságú összeköttetésben általában 0.2-2 db/km csillapítású fényvezető szálakat használnak, amelyek legfeljebb 20-100 km távolság, közbenső regenerálás nélküli áthidalását teszik lehetővé.

Gondoskodni kell arról, hogy az optikai szálat csak minimális fizikai terhelés érje, minden nagyobb és hosszabb ideig tartó terhelést más szerkezeti elem vegyen át, mely védelmet és terhelésátvitelt a kábel konstrukciónak kell biztosítania. (Nem lehet csak úgy „húzni", vagy kis sugárban meghajlítani, mert az üvegszálak megszakadhatnak.)




***2-15 ábra Egymódusú és többmódusú üvegszálak***

A hagyományos, rézvezetékeket tartalmazó kábel és a fénykábel konstrukciós kö-vetelményei között az alapvető különbség az, hogy míg a rézvezetéknél nagy, 15%-os nyújtás is megengedhető, addig a kvarcüveg esetében az 1%-os nyújtás is idő előtti öregedéshez, mikrorepedésekhez, esetleg törésekhez vezethet, ezért elsődleges követelmény a fénykábel szálainak a tehermentesítése. Az üvegszálak alkalmazásánál kritikus kérdés a jelek be és kicsatolása, amire kétféle illesztés, a passzív és az aktív használatos.

- **Passzív illesztő** két, az üvegszálra kapcsolódó csatlakozóból áll. Az egyik csatlakozón egy LED dióda, a másik csatlakozón egy fotódióda van. Az illesztő teljesen passzív, segítségével jeleket tudunk a fénykábelből kicsatolni, illetve jeleket tudunk a kábelbe bejuttatni. Az illesztés természetesen fényveszteséggel (és így csillapítással) jár, ezért adott távolságon csak adott számú illesztő használható.
- **Aktív illesztő** jelismétlőként vagy más néven jelregenerálóként is működik, azaz a beeső fényjelet villamos jellé alakítja, majd az ADÓ részén ezt LED dióda segítségével felerősítve továbbsugározza. Mivel a regenerálás folyamán, a kábelen haladó fényjel villamos jelként is megjelenik, ezért ez közvetlenül elektromos jelillesztésre is felhasználható.

Ahogy az eddigiek szerint is nyilvánvaló, az üvegszálon adott hullámhosszú fényt használva csak egyirányú adatátvitel képzelhető el. Gyűrű kialakítású topológiánál az állomások illesztővel csatlakoznak a hálózatra, így egy vonalon is képesek venni (jel az illesztőbe bejön) és adni (illesztőn továbbadni).

Kétirányú pont-pont átvitel esetén már két üvegszálas kapcsolat szükséges: egyik irány az adásra, másik a vételre. Ez szerencsére a legtöbb esetben nem igényli újabb kábel lefektetését, mivel egy kábel több független üvegszálat tartalmaz.

Az optikai szálon a nagyobb sávszélesség elérése érdekében több technikát alkalmaznak: WDM (Wavelenght Division Multiplexing – hullámhossz multiplexálás) az a módszer, amikor több, egymástól független optikai jelfolyamot vihetünk át ugyanazon az optikai szálon különöző hullámhosszúságú fényimpulzusok segítségével. DWDM (Dense Wavelenght Division Multiplexing - nagysűrűssége hullámhossz integrálás) technikával, akár 40 különböző frekvencián továbbítjuk az információt.

A következő táblázatban a vezetékes átviteli közegek jellemzőit foglaltuk össze:

## 2.4 Vezeték nélküli átviteli közegek

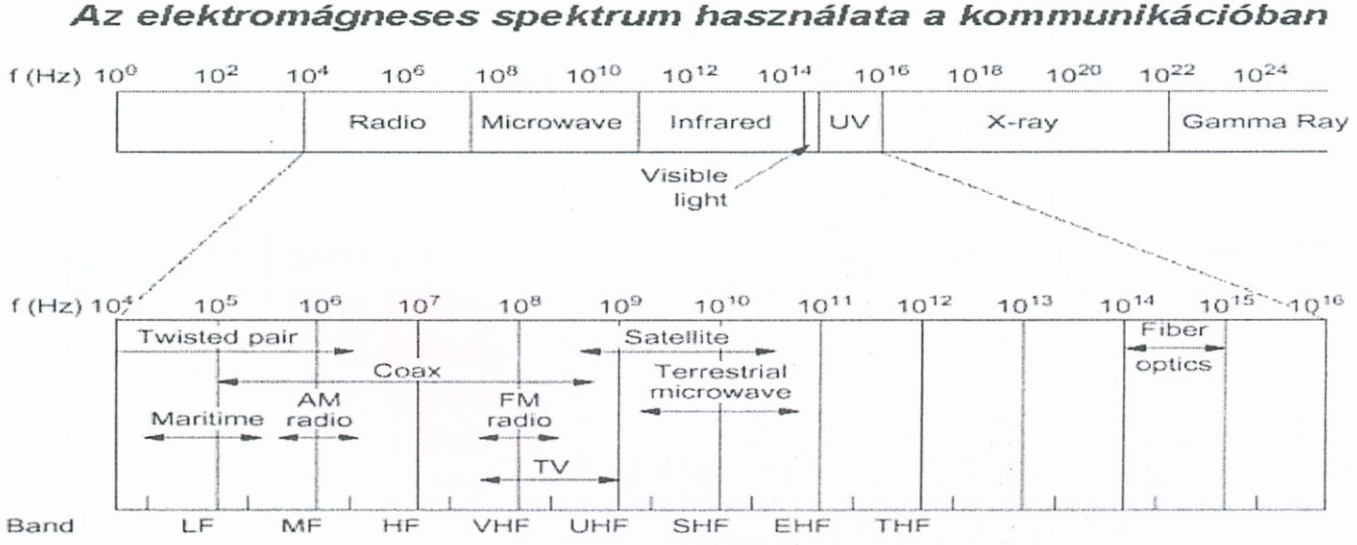
VEZETÉKES ÁTVITELI KÖZEGEK

| Átviteli közegek | Maximális elméleti sávszélesség | Maximális fizikai távolság |
|---|---|---|
| 50 ohmos koaxiális kábel<br>Ethernet 10Base-2, ThinNet | 10-100 Mbps | 200 m |
| 75 ohmos koaxiális kábel<br>Ethernet 10Base-5, ThickNet | 10-100 Mbps | 500 m |
| 5-ös kategóriájú, árnyékolatlan, sodort érpár (UTP, STP, S-UTP)<br>Ethernet 10Base-T, 100Base-TX | 10 Mbps | 100 - 500 m |
| 5-ös kategóriájú, árnyékolatlan, csavart érpár (UTP, STP, S-UTP)<br>Ethernet 100Base-TX | 100 Mbps | 100 m |
| 5E, 6, 7 –es kategóriájú szabványos, vagy még nem szabványosított réz kábel | 1000 Mbps | 100 m |
| Többmódosú (62,5/125 um)<br>Optikai szál 100Base-FX | 100 Mbps | 2000 m |
| Egymódosú (10 um mag)<br>Optikai szál 1000Base-LX | 1000 Mbps | 3000 m |
| Kutatási fázisban lévő más technológiák | 2400 Mbps | 40 Km |
| Vezeték nélkül | 2 – 56 Mbps | 100 m |

Hálózat kiépítésekor gyakran adódik olyan helyzet, amikor vezetékes összeköttetés kialakítása lehetetlen. Utcákat kellene feltörni, ott árkokat ásni, és ha mindez mondjuk egy forgalmas, sűrűn beépített terület? Ilyenkor a vezeték nélküli átviteli megoldások közül kell választani, amelyek fény (infravörös, lézer) vagy rádióhullám alapúak lehetnek.

A *2-16. ábrán* foglaltuk össze, hogy a különféle kommunikációs módszerek az elektromágneses sugárzás melyik tartományát használják.

Az ábrán látható, hogy mind a rádióhullám, mind a fény elektromágneses hullám, de a nagymértékben eltérő frekvenciájuk miatt, a tulajdonságaik nagyon eltérnek. Az is észrevehető, hogy a látható fény (=visible light), milyen keskeny sávot foglal el az elektromágneses spektrumban.




***2-16. ábra Elektromágneses spektrum***

## 2.4.1 Infravörös, lézer átvitel

A lézer és infravörös fényt alkalmazó ADÓ-VEVŐ párok könnyen telepíthetők háztetőkre, a kommunikáció teljesen digitális, a nagyobb távolság áthidalását lehetővé tévő energiakoncentrálás miatt rendkívül jól irányított, amely szinte teljesen védetté teszi az illetéktelen lehallgatás, illetve külső zavarás ellen.

Sajnos a láthatósági feltételek miatt az eső, a köd, a légköri szennyeződések zavarként jelentkeznek. A számítógépes rendszerekben az információátvitel ilyen módja fokozatosan terjed, **IrDA** néven szabványos megoldása is létezik.

## 2.4.2 Rádióhullám

Nagyobb távolságok áthidalására gyakran használják a mikrohullámú átvitelt. A frekvenciatartomány 2-40 GHz között lehet. A kiemelkedő antennatornyokon (a láthatóság itt is feltétel!) elhelyezkedő parabola adó és vevőantennák egymásnak rádióhullám nyalábokat küldenek és akár száz kilométert is átfoghatnak. A jelismétlést itt reléző állomásokkal oldják meg, azaz a vett jelet egy más frekvencián a következő, reléző állomásnak továbbítják.

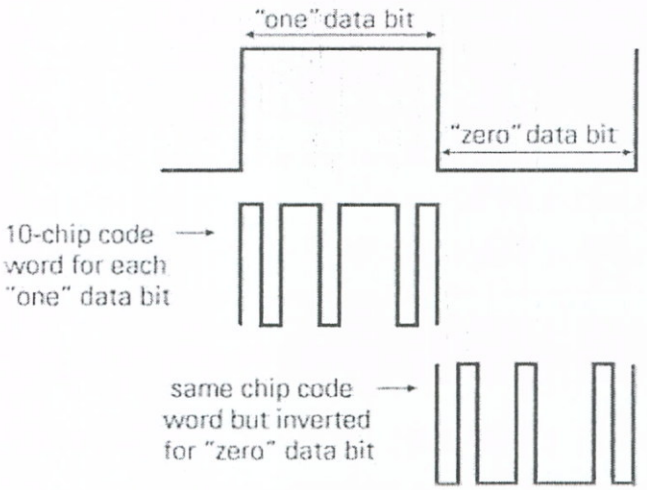
Problémaként jelentkeznek a viharok, villámlás, egyéb légköri jelenségek. Átgondolást igényel a frekvenciasávok kiosztása, és ez hatósági feladat. A rádiós számítógépes hálózatokat WLAN-nak (=Wireless Local Area Network: vezeték nélküli helyi hálózat) nevezik.

### *Vezeték nélküli hálózati szabvány: Az IEEE 802.11 szabvány alapjai*

A rádióhullámokat használó vezeték nélküli hálózatok elterjedésében az igazi áttörést a vezeték nélküli hálózatok szabványosítása hozta. A lokális hálózatok szabványosításánál nagy szerepet játszó szervezet, az IEEE saját rendszert kidolgozva, létrehozta a 802.11-es szabványcsoportot.

A szabvány célja egy közös működési mód definiálása volt. Azt, hogy egy megoldás eleget tesz-e a szabványnak egy szövetség,a WECA (Wireless Ethernet Compatibility Alliance – Vezeték nélküli Ethernet kompatibilitási szövetség) teszteli, és az eredmény alapján kapják meg a kompatibilis termékek a **Wi-Fi ( Wireless Fidelity )** jelzést.

A szabvány az OSI modell fizikai (PHY) és közeg-hozzáférési (MAC) alrétegét definiálja. Meghatározták a kommunikációhoz használatos modulációs eljárásokat és jelszinteket. Háromféle modulációs eljárást szabványosítottak.

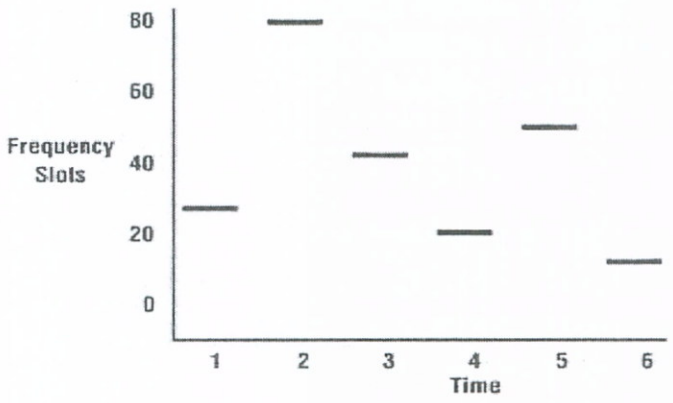



*2-17. ábra DSSS technológia*

- ***DSSS*** – Direct Sequence Spread Spectrum – közvetlen szekvenciális szórt spektrum
- ***FHSS*** – Frekvency Hop Spread Spectrum – frekvenciaugrásos szórt spektrum
- ***OFDM*** – Ortogonal Frekvency Division Multiplex – ortogonális frekvenciaosztásos multiplexelés

**A DSSS**

A DSSS technika az átviendő adatokat nagyobb frekvenciájú digitális kóddal kombinálja. Minden egyes adatbitet olyan mintába ágyaz, amit csak az adó- és a kívánt vevőállomás ismer.




*2-18. ábra FHSS technológia*

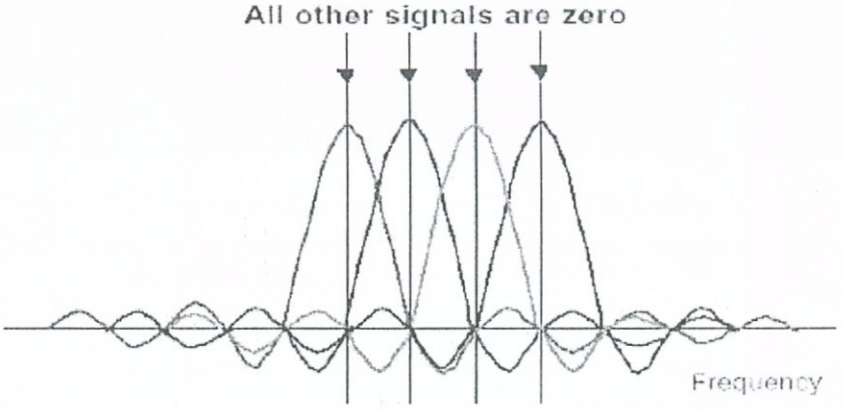
Ezt a bitmintázatot „forgácskódnak" (chipping code) nevezik. Ez a kód magas és alacsony jelek véletlen sorozata, amelyek az épp aktuális bitet jelentik. Hogy az ellentétes állapotú bitet is kódolják, ezt a "forgácskódot" invertálják. Ez a frekvenciamodulálás, amennyiben az átvitelt jól szinkronizálták, magában foglalja saját hibajavítását is, így ez a módszer jobban elviseli az interferenciát, illetve a véletlen bitsorozat „beépített" hibajelzőként is működik.

### FHSS

A frekvenciaugrásos szórt spektrumú jel egy a kommunikáló felek által előre ismert algoritmus szerint modulálja frekvenciákat. Az átvitel csatornáról csatornára ugrál. A csatornák a kommunikációra használt frekvenciatartomány keskenyebb sávokra való felosztásából keletkeznek. Ez a módszer egyrészt a többi hálózattal való interferenciát küszöböli ki, másrész a lehallgatás lehetőségét csökkenti.

### OFDM

Olyan eljárás, amely nagyszámú, ortogonális frekvenciát alkalmaz. Ortogonálisnak nevezzük a különböző frekvenciájú jelekből álló csoportot, ahol a nem azonos frekvenciájú tagok szorzatának átlaga nulla. A jelet frekvencia összetevőkkel kódolva, és a frekvencia-összetevőket modulálva viszi át a kívánt információkat a küldőtől a vevőhöz. Az OFDM az adatszimbólumokat a sok frekvencia-összetevő segítségével párhuzamosan viszi át. Az adatszimbólumok továbbítása az egyes különálló frekvencia-összetevők segítségével, keskenysávú átvitellel történik, vagyis a szélessávú OFDM jel nagyszámú keskenysávú jelösszetevőre bontható.




*2-19. ábra OFDM modulált jel spektruma*

Az IEEE 802.11 szabvány szerint, a működési frekvenciának az úgynevezett ISM (Industrial, Science, Medical – ipari, tudományos, orvosi) sávot használják, mivel ezen világszerte az előírt adóteljesítmény (100mW) betartásával, engedély és bejelentés nélkül lehet forgalmazni. Ez a tartomány a 2.4 – 2.485 GHz frekvenciasávot jelenti. A csatornakiosztás viszont országonként változó. Európában 13 csatorna van kijelölve. A frekvencia sávok kiosztásánál mindig arra kell vigyázni, hogy az egyes frekvenciasávok ne zavarhassák egymást. Ennek a sávnak a hátránya, hogy nagyon zavart, ha csak azt nézzük, hogy például a mikrohullámú sütő is ezt használja. (A víz gerjesztése alapján melegít). Ebből a példából látszik, hogy a víz nagy csillapítást okoz ilyen frekvenciákon. A csillapítás jelentősen megnő, ha az eszközeink között víztartalmú dolgok (fák, folyók..) vannak. Az átvitelt az időjárás is nagymértékben befolyásolhatja, pl ködben, esőben, havazáskor, az egyébként kifogástalan átvitel megszakadhat.

A Wi-Fi hálózati eszközök kommunikációjához rálátás szükséges a másik félre. Kismértékű árnyékolás mellett még működőképes a hálózat néhány 100 méterre, de falakon, domborzati árnyékoláson túl már nagymértékben romlik az átvitel. Megfelelő erősítéssel, és irányított antennákkal azonban a technológia 30 km feletti távolságot is képes áthidalni.

### *Műholdas átvitel*

**A műholdakon lévő transzponderek a felküldött mikrohullámú jeleket egy másik frekvencián felerősítve visszasugározzák.** Hogy a földön lévő műholdra sugárzó, illetve a műhold adását vevő, antennákat ne kelljen mozgatni, az Egyenlítő

fölött kb. 36.000 km magasságban keringő műholdakat használnak, amelyek sebessége megegyezik a Földnek a forgási sebességével. Ezek a geostacionárius pályára állított műholdak a Földről állónak látszanak. A mai technológia mellett 90 geostacionárius műhold helyezhető el ezen a pályán (~4 fokonként). A frekvenciatartományok a távközlési műholdaknál: 3,7...4,4 GHz a lefelé, 5,925...6,425 GHz a felfelé irányuló nyaláb számára.

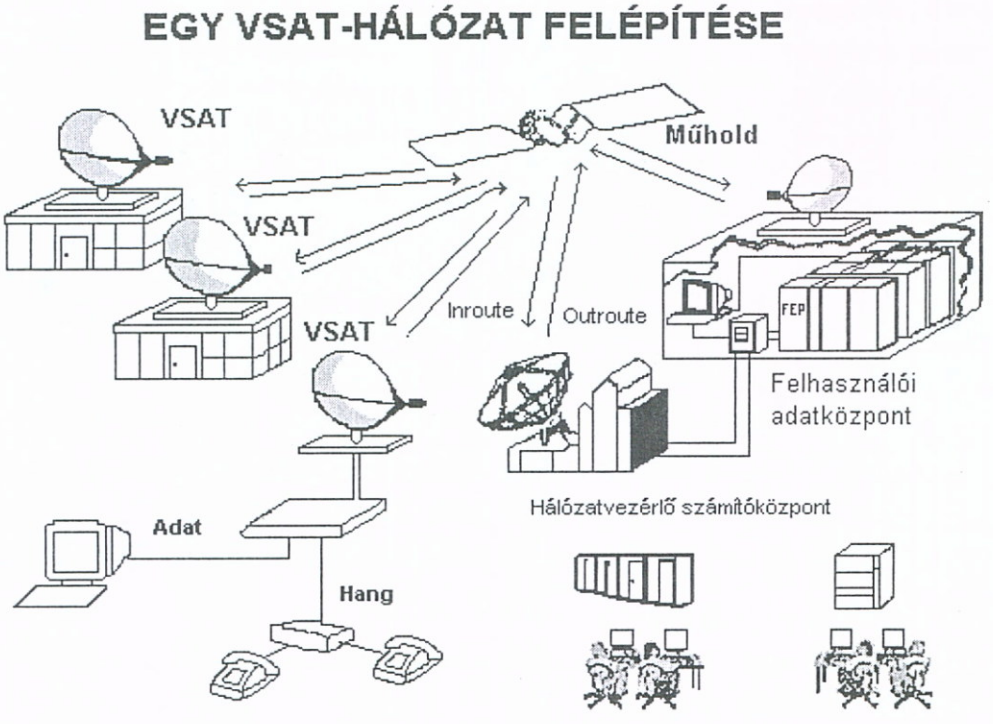
Tudnunk kell, hogy a műholdas átvitel késleltetése a földi mikrohullámú illetve a vezetékes rendszerekhez képest jelentős a nagy távolság miatt: 250-300 msec. A műholdak tipikus sávszélessége 500 MHz (12 db 36 MHz-es transzponder, egy transzponderen 50 MB/s-os adatforgalom, vagy 800 db 64 kbit/s-os hangcsatorna.

A frekvenciatartományok kiosztása a transzponderek között lehet statikus: azaz a frekvenciák fixen ki vannak osztva a transzponderek között, de ma inkább azt a módszert használják, hogy először az egyik transzponder majd utána a következő kap egy-egy frekvenciaszeletet. (Osztott idejű multiplexelés).

A visszasugárzott hullámnyaláb mérete is befolyásolható: nagy kiterjedésű hullámnyalábot leginkább a TV-s műsorszórás igényel, de ma már lehetséges kis kiterjedésű (néhány km átmérőjű) pontnyalábok (spot beam) használata is. Ez utóbbi távközlési rendszereknél előnyös, a lehallgathatóságot csökkenti.

### VSAT rendszerek

A VSAT (Very Small Aperture Terminal) mozaikszó magyarul nagyon kicsi nyílásszögű antennájú vevőberendezést jelent. Így neveznek minden olyan műholdra néző antennát, amelynek átmérője 2,4 méter alatt van. Míg az első kisméretű terminálokat adatszórási célra használták, a 80-as évek elejétől a **kétirányú** VSAT-ok kezdtek elterjedni.




*2-20. ábra: VSAT hálózat*

Napjainkban a VSAT terminálok több generációja él. Ezek sebességben, méretben térnek el egymástól. Kezdetben a VSAT technika a 6/4 GHz–es ún C sávot, a mai rendszerek általában a 14/12 GHz-es ún. Ku sávot, a jövő rendszerei pedig a 30/20 GHz-es ún. Ka sávot fogják használni. (Az első szám mindig a felfelé, a második

szám pedig lefelé irányt jelenti.). A műholdas hálózatok két fő alkotórészre oszthatók: a műholdas űrszegmens és a földi szegmens.

A VSAT-hálózatokban a felhasználók **VSAT terminálok** segítségével kerülnek összeköttetésbe a központi földi állomással, amit az angol terminológia alapján "hub"-nak hívnak.

Magát az összeköttetést egy geostacionárius távközlési műhold biztosítja. A tipikus VSAT-hálózat lényegében csillaghálózat, amelynek a középpontjában a hub helyezkedik el.

A VSAT-terminál több részből áll. A kültéri egység az antennából, a tápfejből, a mikrohullámú erősítőből és a kis zajú keverőből áll. A beltéri egység foglalja magában a modemet, az alapsávi jelfeldolgozó egységeket és a különféle adatátviteli protokollokkal kommunikáló mikroszámítógépes rendszert. Kiemelten kell foglalkozni a távbeszélő rendszerekkel, mivel sok felhasználó hálózati elérése csak ezeken keresztül valósulhat meg.

A hálózat központja a hub állomás, amellyel a VSAT-terminálok együttműködnek. Akár több száz VSAT-terminál kiszolgálását végezhetil el, antennamérete 5-9 méter.

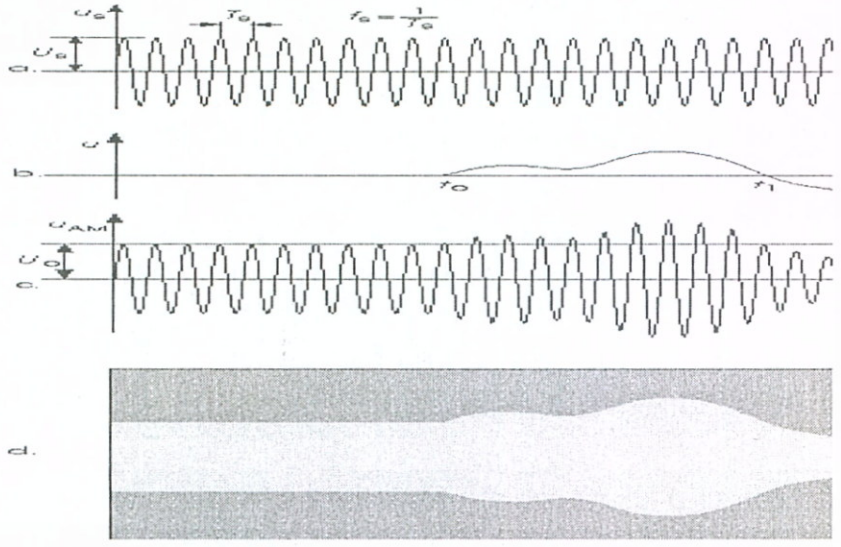
## *Ellenőrző kérdések: 2. fejezet*

?

1. Mit jelentenek jelátvitelnél a csillapítás, és a sávkorlátozás fogalmak? Mi a sávszélesség és az adatátviteli sebesség? Ez utóbbinak mitől függ az értéke?
2. Magyarázza meg a baud és bit/s mértékegységek közötti különbséget!
3. Mi a jel-zaj viszony?
4. Mi a vonal és a csatorna közötti különbség? Milyen vonalmegosztási módszereket ismer? Hogyan történhet több csatorna átvitele egy vonalon? Mi a multiplexelés?
5. Ismertesse a frekvenciaosztásos multiplexelés módszerét! Mi az alapelve?
6. Ismertesse a szinkron időosztásos multiplexelés módszerét! Mi az a PCM?
7. Ismertesse a vonalkapcsolás elvét!
8. Mutassa be az üzenet és csomagkapcsolást! Mi köztük az alapvető különbség?
9. Melyek a fizikailag összekötött és össze nem kötött kapcsolatok jellemzői, előnyei, hátrányai?
10. Ismertesse a csavart érpáras összeköttetés jellemzőit! Milyen kategóriái vannak?
11. Ismertesse a koaxiális kábelt használó összeköttetés jellemzőit! Hogyan jellemezné az alapsávú és szélessávú átvitelt?
12. Hogyan működik az optikai adatátvitel? Mitől függ a fényveszteség? Ismertesse az üvegszálas (optikai) kábelt használó összeköttetés jellemzőit!
13. Mit jelentenek a egymódusú, (monomódusú) illetve többmódusú (multimódusú) fogalmak?
14. Milyen optikai kábelillesztő egységeket ismer? Jellemezze ezeket!
15. Jelemezze a mikrohullámot használó rádiós átvitelt! Mi az a szórt spektrumú sugárzás? Mi a WIFI?
16. Mi az transzponder és mi a geostacionárius pálya? Milyenek a késleltetések egy műholdas rendszerben?
17. Foglalja össze a VSAT rendszerek jellemzőit!

# 3. ANALÓG ÉS DIGITÁLIS JELEK ÁTVITELE

*Ne fukarkodj az őszinte elismeréssel!*



*3-1. ábra Amplitudó moduláció*

Az analóg átvitel a folyamatosan (nem diszkrét módon) változó jelek átvitelén alapszik. Ilyen jeleknél egy folytonosan változó fizikai jellemző hordozza az információt. Például az állandó frekvenciájú vivőhullám amplitudóját változtatjuk az átviendő jel feszültségének megfelelően. Ezt hívjuk amplitudó modulációnak. *(3-1. ábra)*

A múltat teljes egészében az analóg átvitel jellemezte. A berendezések, az átviteli módszerek mindegyike analóg volt, gondoljunk a telefonra, a rádióra és a televízióra. A kialakított kommunikációs infrastruktúra is döntően analóg. Jelenleg már egyre szélesebb körben terjednek a digitális átviteli rendszerek, de a még meglévő és továbbfejlődő analóg rendszerek alapismerete is fontos.

## 3.1 Alapsávi, analóg jelátvitel: vezetékes telefonok, modemek

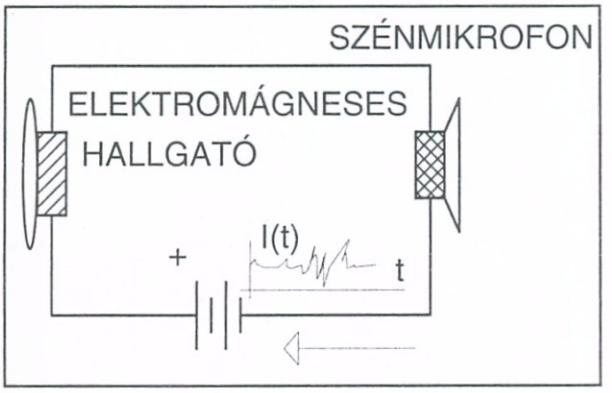
A nyilvános távbeszélő hálózatokat az emberi hang többé-kevésbé felismerhető módon történő átvitelére tervezték. Számítógépek kommunikációjára való felhasználásuk, csak speciális módszerekkel lehetséges, de mégis szükséges, mert egyszerűen ez áll rendelkezésre. Számítógépeket összekötő adatátviteli kábeleken az adatátviteli sebesség minimum Mbit/s-os nagyságrendű, igen kis hibaarány mellett. Telefonvonalon keresztül hagyományos módszerekkel, ez mindössze 10 kbit/s nagyságrendű, jelentősen nagyobb hibaaránnyal, amit vonal és a kötések öregedése folyamatosan növel. Ma egy átlagos felhasználó, ha lehetséges, két hálózatra csatlakozik: a kábeltévé hálózatra koax kábelen keresztül, illetve a telefonhálózatra egy rézhuzalpáron keresztül. Meg kellett találni azt a megoldást, amely a kapcsolódásokat felhasználva biztosítja a nagysebességű hálózati kapcsolatot: ezek a kábelmodem és az A(DSL) rendszerek.

A gyakorlatban még használható a villamos energia továbbítására használt erősáramú hálózat, amit kisebb sebességen már most is használnak információ átvitelre: az erősáramú hálózaton kialkított körvezérléssel lehet villamos fogyasztókat távolról ki- illetve bekapcsolni: az éjszakai/nappali tarifa bevezetését ez a módszer tette lehetővé.

### 3.1.1 Telefon működése

A szénmikrofon ellenállása (amely egy membránnal lezárt szénpor réteget tartalmaz) a rábeszélt hang hatására változik. A hanghullámok a membránba ütközve mozgatják

azt, és a szénszemcsék változó mértékben összepréselődnek. Ezért a körben folyó áram a han erőssége és frekvenciája által meghatározott mértékben változik. Ez a változó áram átfolyva egy elektromágnes tekercsén, annak vasanyagú membrán fegyverzetét az átfolyó áram által meghatározott erővel vonzza. Az ilyen módon mozgatott rezgő membrán hallható hangot fog kibocsátani.




*3-2. ábra A telefon működésének vázlata*

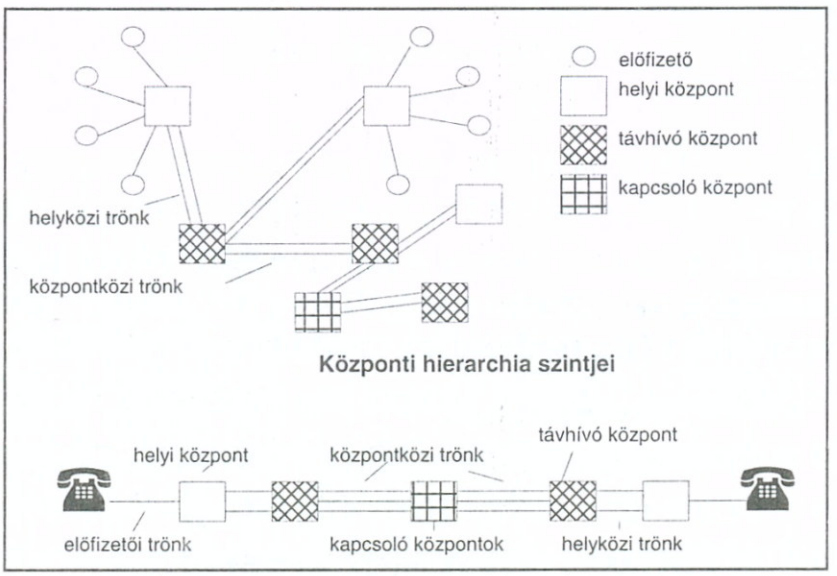
Ez a megoldás csak egyirányú (szimplex) átvitelt biztosít, ezért az áramkört fordított irányban duplázni kell. A beszélgetés kezdeményezését váltakozó áramot használó csengető áramkör hozzáadásával lehet jelezni. Ilyen módon két huzallal összekötve két távbeszélő állomás már képes egymással teljes duplex módon kapcsolatba lépni.

Több állomás esetén az egymással való beszélgetés telefonközpont közbeiktatásával lehetséges. Ilyenkor a beszélgetés célját szolgáló vezetéken a központba egy vezérlő információt (jelzést) is el kell juttatni: a hívott állomás számát. A telefonközpont a szám vétele után létrehozza az összeköttetést a hívott állomással.

A kézi kapcsolású központok korában az előfizető, a vele kapcsolatban lévő telefonközpontban lévő telefonos kisasszonyt hívta fel, megmondta neki a hívni kívánt számot, és a kisasszony hozta létre az összeköttetést.

Minden előfizető egy vezetékpárral (két vezeték) a hozzá közeli **helyi központ**-hoz kapcsolódik. Ezeket **előfizetői hurkoknak (local loop)** nevezik.

- Ha két — azonos helyi központhoz kapcsolódó — előfizető hívja egymást, akkor a központon keresztül, az összeköttetés, a beszélgetés idejére létrejön.
- Ha nem azonos helyi központhoz tartoznak az előfizetők, akkor a kapcsolat kialakításában a távhívó központok játszanak fontos szerepet.




*3-3. ábra Telefonhálózat felépítése*

A helyi központok több vezetékpárral (nevük: helyközi trönk) kapcsolódnak a távhívó központhoz Ezeken keresztül a helyi központok közötti információcsere történik.

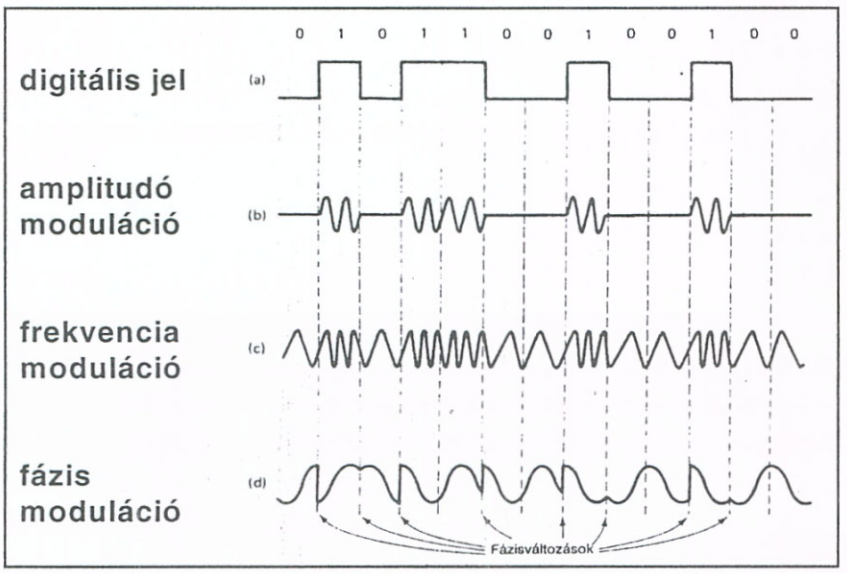
Ha a távhívó központ sem közös, akkor az összeköttetés kialakítása a kapcsolóközpont-hierarchiának a következő, magasabb szintjén történik.

Legáltalánosabban a lakásokban levő telefonvonalakat használják, ezek két vezetékes **kapcsolt vonalak** (**dial-up-line**). Ez azt jelenti, hogy az előfizető csupán az összeköttetés idejére kapcsolódik a telefonközponton keresztül a hálózathoz. A **bérelt vo-**

**nalak** dedikáltak, (fix összerendelésűek), amelyeket a telefontársaság a központban állandó jelleggel összeköt, vagyis tárcsázás nélkül is mindig van összeköttetés.

A telefoncsatorna alkalmas mind hang, mind adat átvitelére és a frekvenciája 300 és 3400 Hz között van, a sávszélessége tehát 3100 Hz.

Szokták még **PSTN**-ek is hívni a hagyományos távbeszélő hálózatokat. **PSTN** (=Public Switched Telefon Network): nyilvános kapcsolt telefonhálózatok, analóg (hang-) átviteli távbeszélőrendszer.




*3-4. ábra Modulációs módszerek*

## 3.1.2 Modemek

Az előfizetők a helyi központhoz egy fémes vezető párral csatlakoznak, és elvileg ezeken akár 1-2 Mbit/s-os adatátviteli sebesség is lehetséges, azonban a telefonbeszélgetések váltakozó áramú jeleit szűrők segítségével 300 Hz – 3.3 kHz között tartják. Ezért az információt ebbe a hangfrekvenciás tartományba eső szinuszos hullám valamelyik jellemzőjéhez (és annak diszkrét változásához) célszerű hozzárendelni. *(3-4. ábra)*

Ezt, a telefonhálózaton átvitt színuszos hanghullám modulálásával érhetjük el. Szinuszos jel esetén annak amplitudóját, frekvenciáját, illetve fázisát modulálhatjuk.

**Azt az eszközt, amely a bemenetére adott bináris jel vezérlésével a modulációt elvégzi (modulálja), illetve a modulált analóg jelből a bináris jelet visszaállítja (demodulálja) modem-nek (modulátor - demodulátor) nevezzük.**

Modemek működését funkcionális protokollok határozzák meg. Ezek:

- modulációs protokoll (milyen modulációs módszert használ)
- hibajavító protokoll (error correcting)
- adattömörítő protokoll (compression)

### *Modulációs protokollok*

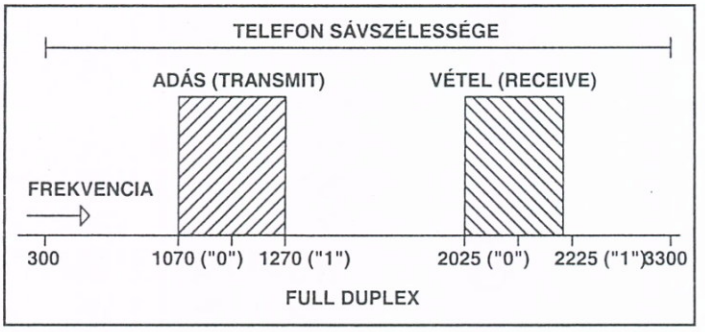
Szinuszos jel esetén annak amplitudóját, frekvenciáját, illetve fázisát lehet megváltoztatni (modulálni).

#### Amplitudó moduláció

Erre jó példa a morse jelek használata. Egy adott amplitudóju jel (a hang), illetve annak hiánya (csend) hordozza az információt. Jól működő, de lassú átvitelt biztosító megoldás. Természetesen több, diszkrét analóg jelértékkel modulálva, a vele elérhető sávszélesség megnövekszik.

## Frekvencia moduláció

Ezt használták először a modemeknél, jó zajtűrése és a biteket hordozó frekvenciák szűrőkkel való könnyű szétválaszthatósága miatt. Szokták a módszert FSK-nak (Frequency Shift Keying) is hívni. Mivel a telefonösszeköttetések duplex rendszerűek, ezért a szabványos adási és vételi, 0 és 1 értékű bitekhez tartozó frekvencia kiosztás a 30. *ábrán* látható.

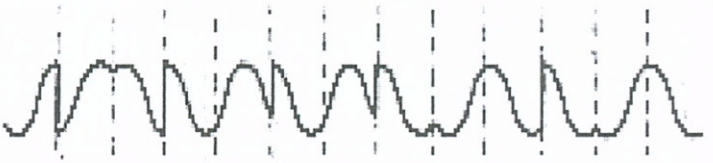



*3-5. ábra Teljes duplex FSK adatátvitel*

Az adatátviteli sebességet a használt alacsony frekvencia erősen korlátozza, mivel például a legkisebb, 1070 Hz-es frekvencián a minimális, egy teljes szinusz-hullám átvitele:~1 msec, ami 1 kbit/s átviteli sebességet jelent (1000 színuszhullám másodpercenként).

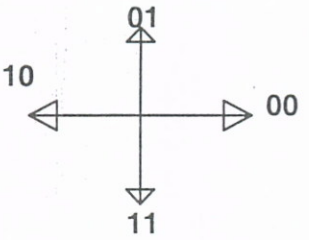
## Fázismoduláció és fázis-amplitudó moduláció (PAM)

Ez a módszer nyújt lehetőséget, a telefonvonalon történő, nagyobb adatátviteli sebesség elérésére. Ilyenkor a modem folyamatosan küldi a szinuszhullámot. Ez viszi a fázis- és amplitudó információt. Ezért szokásos ezt a jelet vivőhullámnak (carrier) is hívni. A vivő frekvenciája a telefoncsatorna sávszélességének közepén van: kb 2 kHz.



*3-6. ábra Fázis moduláció*

Az adónak 0 - 90 - 180 és 270 fokos fázisszög - kezdettel kell szinusz hullámot elküldenie, ami két bit kódolását, és így, egyszerre való elküldését biztosítja. Szokásos a vektorokkal történő jelölés, amivel az egyes fáziszögek jól illusztálhatók. Ha például 2400 Hz-es szinuszhullámot használunk, akkor másodpercenként 2400 darab szinusz hullámot küldünk át, amelynek négy különböző fázishelyzete lehetséges. Ezekhez két bit információt rendelve, az adatátviteli sebesség 4800 bit/s lesz!

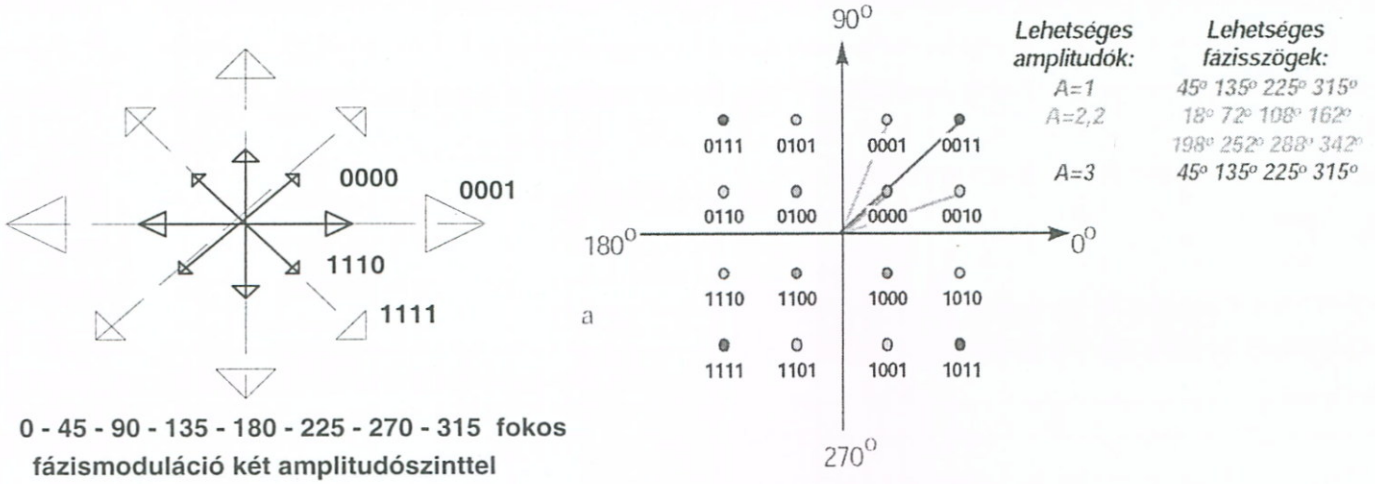



**0 - 90 - 180 - 270 fokos fázismodulációs**

*3-7. ábra Négy fázisszögű fázismodulációs jelvektor*

Az adatátviteli sebesség további növelése úgy lehetséges, ha növeljük a fázisszögek számát, illetve a hordozó szinusz hullámot különböző amplitudókkal küldjük. Például 8 fázisszög és 2 amplitudó esetén már egy jel 4 bitet képes kódolni! Ezt a megoldást szokták **QAM16**-nak, vagyis **Kvadratúra amplitudó moduláció**-nak is hívni. A *3-8. ábra* jobb oldalán a gyakorlati megoldás látható, ahol a megbízhatóbb felismerés miatt három amplitudót használnak.

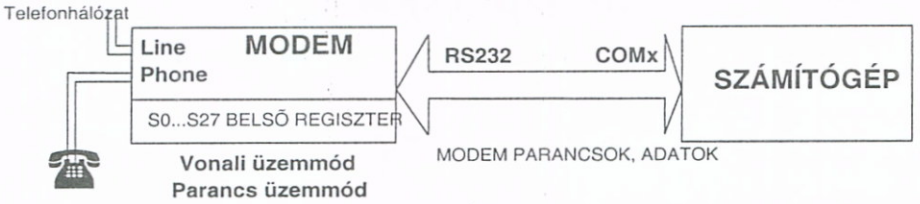
Belátható, hogy a sikeres adatátvitel az amplitudók és a fáziszögek korrekt detektálását igénylik a vevő oldalán. Ezt a korszerű modemekben az analóg áramköri technikák kifinomult alkalmazásával lehet csak megoldani, és a modemek minőségét alapvetően meghatározza.




*3-8. ábra Nyolc fázisszögű, két amplitudószintű illetve a QAM16 fázis-moduláció vektorábrája*

## A modemek felépítése

A modemek önállóan működő számítógépes perifériák, és az adatátvitel megvalósításához ezeket a számítógépnek kell felprogramozni, parancsokkal vezérelni, és állapotát (státuszát) ellenőrizni. Két modem csak akkor tud kapcsolatba lépni, ha van közös modulációs protokolljuk. Ha több sebességű is van, akkor a leggyorsabb közöst választják ki (kezdeti "összefütyülés" alatt, ill. később is).




*3-9. ábra Modem bekötése*

Az összekapcsolás a később ismertetett szabványos soros vonalon keresztül valósul meg. A nagysebességű modemek esetén fel kell gyorsítani a soros portot, hogy a modem ne kényszerüljön várakozásra és az **adatátvitelt vezérlő flow control**-t kell alkalmazni.A flow control rendszerint szoftver- vagy hardver-vezérlést (handshake) alkalmaz.

- A **szoftver handshake** (vagy inband flow control) speciális karaktereket a Ctrl-S-t (amit XON-nak is hívunk), és a Ctrl-Q-t (neve: XOFF) helyez az adatfolyamba az adatátvitel vezérlésére, az adatvesztés megelőzése céljából. Így például a modem egy XOFF-ot fog küldeni a számítógépnek, ha az adatokat túl gyorsan küldi. A modem XON-t fog küldeni, ha már kész a többi adat fogadására.
- A **hardver handshake** (vagy out-of-band flow control) villamos jelet használ a számítógép és a modem közötti kábel egy vezetékén. Az RS-232 modem illesztő szabvány az RTS-CTS jelpárt használja az adatfolyam vezérlésére (ld. később!). A hardver handshake megoldás előnyös, mert nem keverednek össze a vezérlő jelek az aktuális adatjelekkel, de külön vezetékeket igényel.

A számítógép a szöveges formájú parancsokat soros vonalon keresztül adja ki a modemnek, a modem parancs üzemmódjában értelmezi azokat, és szintén szöveges, általában "OK" üzenettel válaszol. A parancsok csak azért szövegesek, hogy mi, emberek könnyen tudjuk olvasni.

Minden parancs az AT karaktersorzattal kezdődik, és ezt követi (betüköz nélkül!) a parancs további része. Csupán az AT utána Enter begépelésére a modem OK üzenettel jelzi a kapcsolat meglétét. A legfontosabb parancsok ismertetése egy sorban:

### *Modem parancsok*

| | |
|---|---|
| AT | Parancs prefix |
| A/ | Ismételd az utolsó parancsot (pl. Ismételt tárcsázás) |
| Bn | n=0 vagy 1. Protokoll kiválasztása (BELL/CCITT) |
| D | Tárcsázási parancs |
| P | Pulse mód |
| T | Tone mód |
| , | Szünet tárcsázás közben |
| ; | Tácsázási parancs végén a modemet parancs üzemmódben tartja. |
| R | Fordított kapcsolat, a hívást kezdeményező modem üzemmódba kerül |
| W | A modem tárcsázás közben tárcsahangra vár |
| Hn | Vonali relé H0 esetén a modem lelép a vonalról (on hook, v. hung up) H1 esetén rálép. |
| In | Gyártási kód és memória ellenőrzés |
| F4 | Fax üzemmódra váltás |
| Ln | Hangerő szabályozás. N=0....3 |
| Mn | hangszóró ki-be kapcsolása |
| O | Vonali üzemmód |
| Qn | Eredménykód küldés engedélyezés/tiltás |
| Sn? | Regiszter (n=0...27) tartalmának lekérdezése. |
| Sn=X | X érték írása a regiszterbe |
| Vn | Eredménykód formátum |
| Xn | Eredménykód részletes kiírásának engedélyezése |
| Y | A hosszú szünet: kapcsolat megszakítása. |
| Zn | Reset parancs |
| +++ | Kilépő parancs vonali üzemmódból parancs üzemmódba. |

Például:

**AT DP 1754568;**

**OK**

A fenti példa arra utasítja a modemet, hogy a pulzus módot használva, hívja fel a 175-4568-as telefonszámot, aminek teljesítését a visszaküldött OK üzenettel jelzi.

### *Modemregiszterek*

A legtöbb modemben 28 regiszter van (jelölésük: S0-S27), amelyek a modem működési paramétereit határozzák meg. Ezek szerepe lehet az, hogy időzítőként vagy számlálóként működnek, vagy az, hogy a tartalmuk határoz meg bizonyos jellemzőket (bitminta). Egyes jellemzők értékei nem törlődő memóriában (NVRAM) tárolhatók és a későbbiekben újra bekapcsoláskor ezek jelentik az alapbeállítást. A regiszterek tartalmának módosítása és kiolvasása két modemvezérlő paranccsal lehetséges: (n = 0...27 és X = 0...255)

**Módosítás: ATSn=X** **Kiolvasás: ATSn? - kiírja az Sn regiszter értékét decimálisan**

### *Modemek hibakezelése, adattömörítés*

A modemek által használt telefonvonalak, nem tesznek lehetővé fizikailag megbízható

átvitelt. Ezért meg kellett találni azokat az átviteli hardver és szoftver megoldásokat, ami ezt mégis megbízhatóvá teszi. Adatátviteli sebességnövekedést biztosít az adattömörítés alkalmazása.

## Szabványos hibajavító protokollok

- MNP (The Microcom Networking Protocol) Az amerikai hadsereg megbízására a kifejlesztője a MICROCOM cég. Az MNP egy különleges hibajavító és adattömörítő eljárás, amely zajos vonalakon is biztosítja a hibátlan adatátvitelt. Az OSI modell hálózati rétegének része, azaz szabványos adatkapcsolatot biztosít a különböző eszközök között. Lehet szoftveres és hardveres megoldású. MNP1-MNP10 kategóriái vannak.
- CCITT V.42 (LAP-M, Link Access Procedure for Modems), Hibajavító protokollal ellátott modemek hibamentes adatátvitel biztosítanak. A vonali hibák változatlanul jelen vannak, de a két modem - az adatok esetleges újraküldésével – kiszűri azokat.

## Szabványos adattömörítő protokollok

- MNP-5 (szükséges, hogy a modem MNP-4-et tudja), max. tömörítés: 2:1.
- CCITT V.42bis (szükséges, hogy a modem V.42-t tudja), max. tömörítés: 4:1

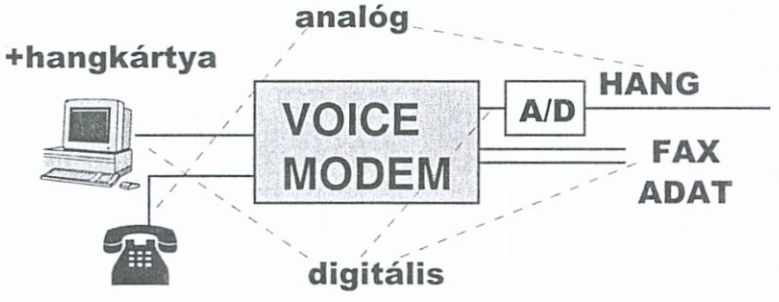
A 2:1 és 4:1 arány csak elméleti, rendkívüli mértékben függ az átvitt adattól. Eleve tömörített fájl esetén az átviteli sebesség még rosszabb is lehet, mint tömörítetlenül. (Normál text kb. 50%)

## *Fájlátviteli protokollok*

Mivel nagyon sokszor a modemeket adatátviteli célokra használják ezért kifejlesztettek számítógépes adatátvitelre alkalmas modem protokollokat is.Elterjedt szabványos fájl átviteli protokollok:

| | |
|---|---|
| Xmodem | 128bájtos csomagok átvitele CRC ellenőrzéssel. |
| Xmodem-1K | mint az Xmodem, de 1024 bájtos csomagokkal |
| Ymodem | mint az Xmodem-1K de többszörös batch fájl átvitellel |
| Ymodem-g | mint az Ymodem de hibajavító protokollal (hibajavító modem kell hozzá) |
| Zmodem | mint az Ymodem de megszakadásból is fel tud épülni |
| Kermit | régi, egyszerű, jólismert |

## *Modemek fejlődése*



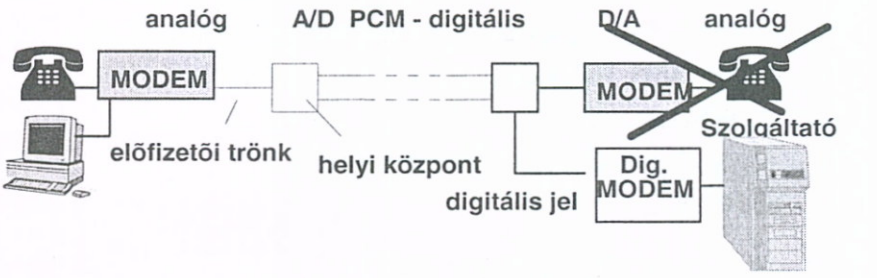

*3-10. ábra Voice-modem*

A modemek fejlődése az előbbiekben kifejtett okok miatt töretlen. Folyamatosan jelennek meg az egyre nagyobb sebességű modemek (a könyv írása idején 56 kbit/s a sebesség), és a modemek segítségével újabb szolgáltatásokat is megvalósítanak.

Az eredeti, telefonvonalon terjedő hanginformáció is digitalizálható. Az előbbiek alapján nyilvánvaló, hogy az analóg telefonvonalon vagy modemes adatátvitel, vagy beszédátvitel folyik, ezek egymást kizárják.

A **voice-modem**-ek képesek a hangot is felismerni, átalakítóval digitálissá, illetve a digitális jelet analóggá átalakítani, és így a számítógépen tárolni.

A jelenlegi **V92 szabványú modemek** már képesek 56 kbit/sec sebességü adatfolyam küldésére is. **A sebesség növekedést a jel/zaj viszony növekedését okozó analóg-digitál-analóg átalakítás egyikének az elhagyásával érték el.** Hagyományos modemeknél a számítógép digitális jelét a modem analóggá alakítva viszi át az előfizetői vonalon a központba. Mivel a legtöbb központ digitális, üvegszálas átvitelt használ a központközi kapcsolataiban, ezért ezt az analóg fázis- és amplitudó- modulált jelet digitális PCM jellé alakítják. A célközpontban megtörténik a küldött jel analóggá alakítása, majd a vezetéken történő átvitele, ahol ismét digitálissá alakítják. Ez az utolsóként említett átalakítás hagyható el: a vételi oldalon olyan úgynevezett **digitális modemet** alkalmazunk, amely képes a digitális PCM jelből előállítani az adatot.




*3-11. ábra 56 kbit/sec-os modem működési elve*

Két megjegyzés:

- Egy ilyen modem sebessége csak akkor használható ki, ha a fogadó oldalon digitális modem van.
- A leírtakból nyilvánvaló, hogy két 56 kbit/sec-os modem egymással nem tud ilyen sebességen kommunikálni, csak az elméleti 33 kbit/sec sebességen.

### *Modem szabványok*

A modemek fejlődését jól mutatják a szabványok, a következőkben összefoglaljuk a legfontosabbakat.

| | |
|---|---|
| V21 | 300bit/s teljes duplex MODEM szabvány, megfelel a BELL 103-nak. |
| V22 | 1200bit/s teljes duplex MODEM szabvány, megfelel a BELL 212-nek. |
| V22bis | 2400bit/s teljes duplex MODEM szabvány. |
| V23 | 600/1200bit/s felduplex MODEM szabvány, amely rendelkezik egy 75bit/s-os ellenirányú ún. felügyeleti csatornával. |
| V24 | A DTE - DCE közti interfész legelterjedtebb fajtája megfelel az RS232C-nek. |
| V32 | Max. 9600bit/s sebességű, duplex szinkron MODEM szabvány. Kombinált többszintű fázis- és amplitudó-modulációt használ, speciális bitcsoport kódolással, amely lehet 16 szintű ún. nem redundáns kódolás, illetve 32 szintű redundáns ún. trellis kód. |
| V42 | Hibajavító eljárás, amely az átvitel során keletkezett hibákat felismeri és részben a redundáns kódolás segítségével javítja, illetve szükség esetén ismétlést kér. Két változatban működhet: egyrészt a saját LAMP eljárása szerint, másrészt az MNP4-es eljárás szerint a kompatibilitás biztosítása céljából. |
| V42bis | Adattömöritő szabvány, max. 4-szeres tömöritést lehetővé. Csak a V42 szabványú hibajavító protokollt alkalmazó MODEM-ekkel használható. Az előzetesen (pl. PKZIP-el) végrehajtott tömörítéstől függetlenül működik. |
| V92, X92 | Ez jelöli az 56 kbit/sec-os adatátviteli sebességet biztosító szabványt |

A *3-12. ábrán* látható felsorolás tartalmazza mindazokat a jellemzőket, amelyek egy modem vizsgálatakor, vagy vásárlásakor figyelembe kell venni.

### *Szoft modemek*

Ma már a digitális eszközök sebessége lehetővé teszi, hogy a modemeknél alkalmazott modulációs, demodulációs technikát, nem különállő hardverrel, hanem a számítógépen háttérben futó programmal valósítsuk meg. Ebben az esetben az illesztő áramkör jóval egyszerűbb, és olcsóbb, és az eszközhöz adott programot kell a számítógépen installálni a modem működésének megvalósítására. Az eszközöket **szoft modem**eknek nevezik.

**MODEMTULAJDONSAGOK**

- AUTOMATIKUS TÁRCSÁZÁS IMPULZUSOS (PULSE) VAGY HANGEFFEKTUSOS (TONE) ÜZEMMÓDBAN
- TELJES DUPLEX MŰKÖDÉS
- HAYES-KOMPATIBILITÁS
- AUTOMATIKUS VÁLASZOLÁSI KÉPESSÉG
- KEZDEMÉNYEZŐ/VÁLASZOLÓ ÜZEMMÓD (ANSWER/ORIGINATING)
- AUTOMATIKUS SEBESSÉGVÁLASZTÁS
- HÍVÁSFIGYELÉS: TÁRCSÁZÓ ÉS FOGLALTSÁGI JELEK ÉRZÉKELÉSE
- HANGERŐ SZABÁLYOZÓS HANGSZÓRÓ
- HIBAÉSZLELÉS ÉS –JAVÍTÁS
- ADATTÖMÖRÍTÉS
- AZ ELŐLAP JELZŐFÉNYEI
- TELEFONSZÁMOK TÁROLÁSA
- HÍRKÖZLÉSI FŐFELÜGYELET ENGEDÉLYE
- MODEM ÁTVITELI SEBESSÉGE
- COMX, IRQX VÁLASZTHATÓSÁGA
- FLASH EPROM
- HANGSZÓRÓ+MIKROFON -> ADAT+HANG KÉPESSÉG
- MELLÉKELT SZOFTVER

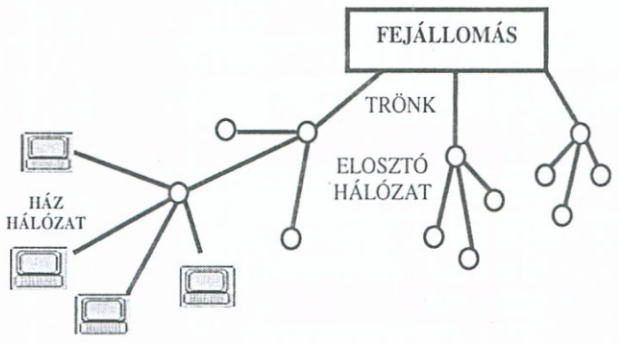
*3-12. ábra Modemtulajdonságok összefoglalása*

### FAX-ok

Írott anyagok képek átvitelére szintén a modem technológiát felhasználó faxok (facsimile = hasonmás) szolgálnak.

A FAX egy lapolvasót (szkennert) egy nyomtatót és egy modemet tartalmaz. Két szín, (fekete=0, fehér=1 alkalmazása esetén a letapogatott jelek egy bináris jelsorozatot alkotnak, ami egy modem segítségével telefonvonalon átvihető.

## *3.2 Széles sávú analóg jelátvitel: Kábel-TV (=KTV)*




*3-13. ábra KTV hálózat felépítése*

A kábeltelevíziós hálózatok – hasonlóan a telefonhálózatokhoz – széles körben elterjedtek, és a tv adások átvitele mellett, számítógépes hálózatok kialakítására is alkalmasak: egy kábelen halad a televízióadás és a hálózati forgalom.

A kábeltelevíziós hálózatok hierarchikus felépítésűek. A legfelső szintet alkotják a nagy távolságot áthidaló gerinchálózatok (szokásos elnevezésük: trönk), kis csillapítású koaxiális kábeleket és számos erősítőt tartalmaznak. Ezek továbbítják a fejállomások műholdról, vagy antennákról kapott jeleit a vonal- vagy elosztóhálózatnak nevezett második szintnek. Ezek 300-1200m hosszúságúak, már az erősítőkön kívül megjelennek a passzív elosztást végző csomóponti kialakítások is. Ezt a házhálózati szint követi. Építőelemei a házerősítő és a közvetlenül az előfizetőig menő kábel. Bár számos topológia terjedt el, a legelterjedtebb, mikor a törzs és a vonalhálózat fa struktúrájú, a házhálózatok pedig csillag kialakításúak.

A kábeltelevíziózás átviteli közege a már megismert 75 Ω-os koaxiális kábel. Természetesen a hierarchia szintjein más és más mechanikai és elektromos tulajdonságú

kábeleket alkalmaznak. A modern kábelhálózaton analóg (folyamatos értékekkel rendelkező) és digitális (diszkrét állapotokból álló) jelek egyaránt átvitelre kerülnek, pedig jellemzőik lényegesen eltérnek. Azért, hogy a kábelen számos tv-adás jelét át tudják vinni, az átvitel, a már megismert **frekvenciaosztásos multiplexeléssel** történik. Az adások analóg video- és hangjeleinek átvitelénél a már megismert kétfajta modulációs módszert használják:

- **Amplitudó moduláció**: a vivőjel amplitudóját változtatják, a tényleges információt hordozó jel függvényében
- **Frekvencia moduláció**: a vivőjel frekvenciája változik, a tényleges információt hordozó jel függvényében

Digitális jelek átvitelekor a telefonmodemeknél már bemutatott fázisszög modulációt használják, zajos környezetben csak a jel fáziszögét változtatják, egyébként a nagyobb átviteli sebesség elérés érdekében kombinált fázis-amplitudómodulációt alkalmaznak (QAM).

**Frekvenciakiosztás**

A kábelen a frekvenciaosztásos multiplexelés frekvenciatartománya kb. 5MHz-800MHz. Itt helyezik el az egyes adások video és hangjeleivel modulált, egymást 6...8MHz-es lépésekben követő vivőfrekvenciákat. Mivel a mai korszerű interaktív televíziózás, illetve az internet megjelenése kétirányú kommunikációt igényel, ezért szükséges az egy kábelen történő kétirányú átvitel miatt a rendelkezésre álló frekvenciasáv irányok szerinti „kettéhasítására". Ez több módon elvégezhető de jellemzően az 5MHz...100 MHz-es tartomány a visszafelé mutató irány, és e felett helyezkedik el a műsorokat is tartalmazó előre mutató irány. A két irányhoz rendelt frekvenciasáv között van egy 30...100MHz nem használt frekvenciatartomány, a kölcsönös zavarások elkerülése érdekében. (Részletesen: a www.hif.hu weboldalon.)

## 3.3 Digitális átvitel kétállapotú jelekkel

A távközlés területén nagyon sokáig az analóg átvitel volt az uralkodó. A jeleket valamelyik fizikai jellemzőjük (pl. feszültségük) időben folytonos változtatásával vitték át. A digitális elektronika és a számítógépek gyors fejlődése során, a telefonközpontok közötti nagysebességű trönkökön folyamatosan a digitális átvitelre tértek át (azaz folyamatos jelek helyett 0-kból és 1-ekből álló sorozatok haladnak a vonalakon).

**Analóg átvitel esetén a jel értéke folytonos, a nagyságával jellemzhető. Digitális jelek esetében a jelnek több diszkrét állapota lehetséges. Bináris jelekről akkor beszélünk, ha jeleknek csak két állapota van, amit szimbolikusan 0-val és 1-el szoktak jelölni.**

A bináris átvitel több fontos szempontból jobb az analóg átvitelnél. Először is nagyon kicsi a hibaaránya. Analóg áramkörök esetén erősítőket használnak a vonalon fellépő csillapítások kompenzálására, azaz a jel regenerálására. Mivel a szükségképpen két irányban elhelyezett erősítők paraméterei folyamatosan változnak (öregedés, külső hőmérséklet, stb.) ezért ez soha nem lehet tökéletes. Mivel a hiba halmozódik, ezért a sok erősítőn átmenő jelek várhatóan komolyan torzulnak.

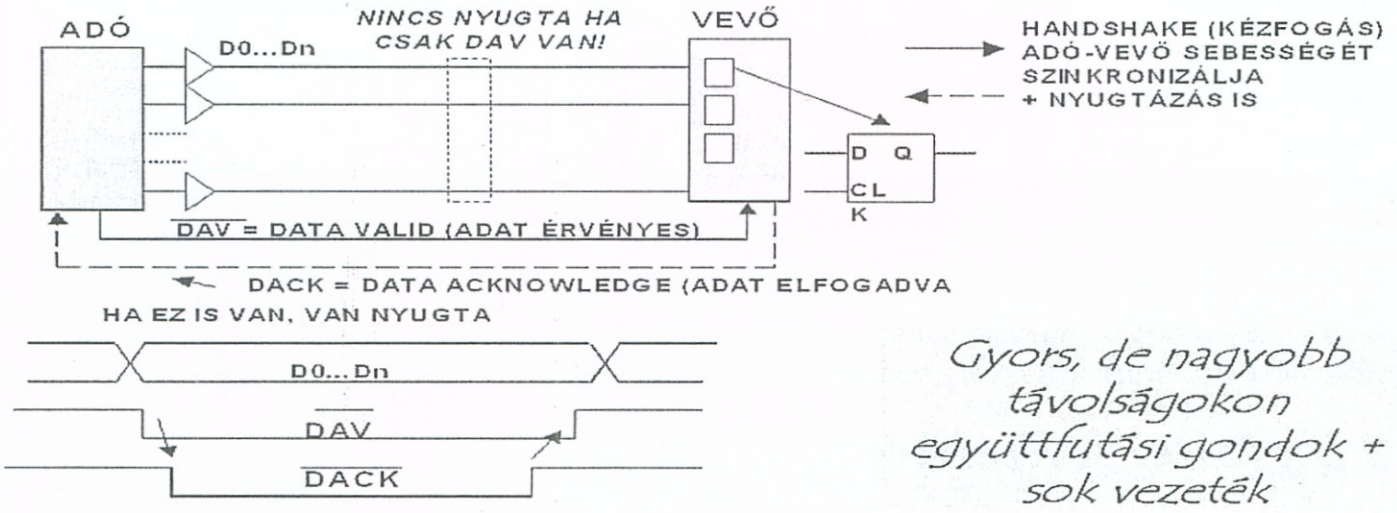
Ezzel szemben a digitális jelek tökéletesen helyreállíthatók, hiszen két lehetséges értékük van, az 1 és a 0. A digitális jelek helyreállításakor nem lép fel halmozódó hiba. A jel regenerálásához is egy összehanlításon alapuló döntést kell hozni. A digitális átvitel egy másik előnye az, hogy egyetlen eszköz hatékonyabb kihasználásával, különböző típusú adatok (hang, zene, adat, kép pl. televíziós kép vagy videotelefon stb.) kevert

átvitelét teszi lehetővé. Ez a különféle típusú adatok bináris alakra kódolása segítségével valósul meg.

A jelek analóg-digitális átalakítása, majd a csatornán történő átvitel utáni visszaalakítása során is fellépnek hibák (felbontás, mintavételi idők, stb). A digitális adatok tömörítésekor is fellépnek veszteségek. A minőségi javulás az átviteli közegre korlátozódik!

## 3.3.1 Párhuzamos és soros adatátvitel

Az átvinni kívánt információt mindig bitekből álló csoportok kódolják – egy bit csak egy igen-nem típusú információt tud hordozni. Ez sok esetben elegendő, mert egy kapcsoló vagy lámpa állapotát, egy bittel le tudjuk írni. Mérési eredmények, vagy szö-

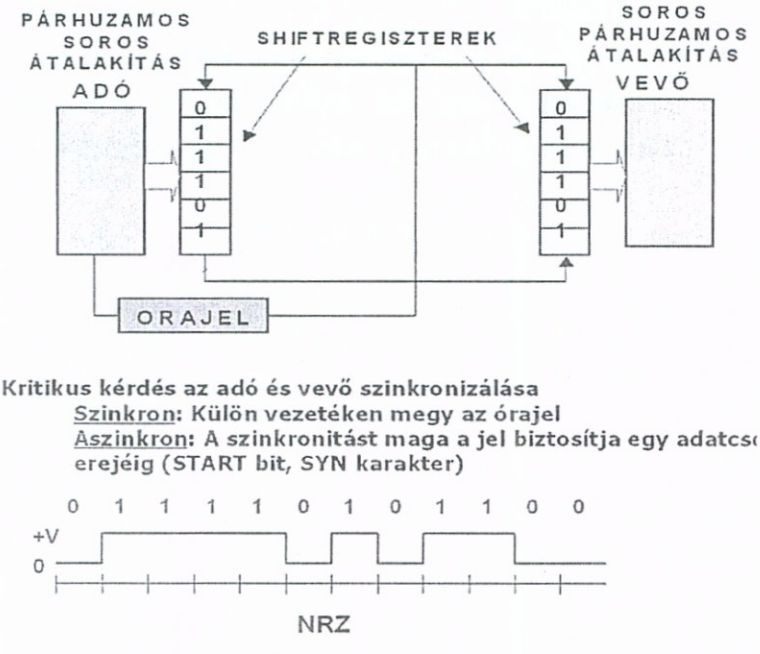



*3-14. ábra Párhuzamos adatátvitel*

vegek átviteléhez már bitcsoportokkal kell kódolni az információt, és úgy küldeni. Bitcsoportok átvitelekor, több csatornát felhasználva, a bitek átvitele történhet párhuzamosan, egyidőben. Ez a **párhuzamos átvitel.**

A másik lehetőség a bitcsoportokat alkotó bitek átvitele bitenként, egy csatornát felhasználva. Ez a **soros átvitel.**

Melyik a jobb? – kérdezhetnénk. A válasz – mint számos más kérdésre – esetünkben sem egyértelmű. Kétségtelen, hogy az adatátviteli sebesség, a párhuzamos átvitelnél nagyobb. Azonban a jelek egymásra való hatása (áthallás), – valamint az egyes vezetékek környezet felé mutatott különböző szórt kapacítása miatt – az




*3-15. ábra Soros adatátvitel*

egyszerre induló bitek a végpontba nem egyszerre érkeznek, és vételkor a leglassabb bitet kell megvárni. A vezetékek száma is hátrányos az összehasonlításban.

Soros adatátvitel esetén az ilyen együttfutási problémákkal nem kell foglalkozni, és a vezetékek száma is jelentősen lecsökken. A digitális áramkörök sebességének a növekedése a párhuzamos átvitel sebességelőnyét kiegyenlíti.

Ha megfigyeljük, az a tendencia, hogy áttérnek a párhuzamos átvitelről a soros átvitelre: A nyomtatók párhuzamos portját a soros átvitelű USB, míg a merevlemezek illesztésénél használt párhuzamos (ATA) interfészt, a soros SATA (serial ATA) váltja fel. A processzorok belső áramköri működése során a sebesség miatt a párhuzamos síneket használják. Itt ugyanis a mérettől és távolságoktól függő szórt kapacítás hatása jelentéktelen, a méretek miatt.

A továbbiakban – az előzőekben leírtakból következően - elsősorban a soros adatátvitellel foglalkozunk. A digitális kommunikáció kezdetein eleinte szövegátvitelt valósítottak meg, ezért az átvitt információ egysége egy olyan bitcsoport volt, amely a szöveg egy karakterét kódolta. A bitcsoportok átvitelét használó módszert szokták **karakterorientált átviteli eljárás**-nak nevezni. A hálózatok elterjedésével a szöveges jellegű információk mellett más jellegű információk átvitele is szükségessé vált, sokszor eltérő szóhosszúságú és adatábrázolású számítógépek között. Ezért a bitcsoportos átvitel helyett a tetszőleges bitszámú üzenetátvitel került előtérbe, ezek a **bitorientált eljárás**-ok.

Karakterorientált átviteli eljárásnál az átvitt információ egysége a karakter. A hálózati szabványokban, leírásokban a bájt kifejezés helyett az **oktet (octet)** fogalmát használják, ami egy 8 bites csoportot jelöl.

**Ne feledjük, az átvitel során mindig biteket viszünk át, legfeljebb az adó és a vevő egységes bitcsoportként kezeli!**

**Hogyan tudjuk biztosítani az adó és a vevőoldal szinkronizmusát?**

(Szinkronizmus: amikor az adó sorrendben küldi a biteket, a vevő ezt helyesen veszi). Az átvitel során, mivel sorban, bitenként történik, valahogy biztosítani kell az adó és a vevő szinkronizmusát, azaz azt, hogy pl. az ötödiknek elküldött bitet, a vevő szintén az ötödik bitnek érzékelje. Elméletileg két megoldás kínálkozik:

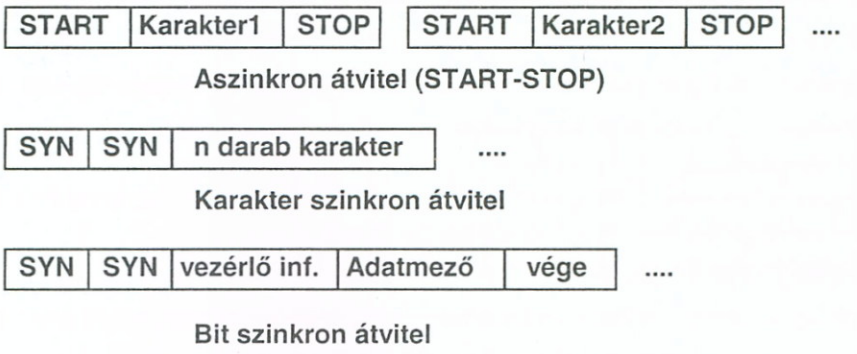
- Az egyik esetben az adó, a bitekkel együtt küld egy érvényesítő órajelet, amely jelzi, hogy a bit értékét mikor kell leolvasni.
- Másik módszer alkalmazásakor az ADÓ és a VEVŐ szinkronizmusát úgy biztosítjuk, hogy mindkét egységben lévő belső órát szinkronizáljuk egymáshoz, egy szinkronjel segítségével. A következő szinkronizálásig az ADÓ és VEVŐ belső órája szinkronban van egymással, és a küldött biteket a VEVŐ helyesen fogja értelmezni. (A két óra eltérése hibahatáron belül van a következő szinkronizálásig.)

Gyakran, a külön órajelet használó átvitelt szokták szinkron átvitelnek nevezni. Sajnos megtévesztő módon, az adatfolyamban elhelyezett, szinkronizációs jeleket használó módszernek is van szinkron és aszinkron változata. Az ismételt szinkronizáció időtartamától függően beszélünk **szinkron**, és a kicsit megtévesztő, **aszinkron** átvitelről.

**Szinkron átviteli módszer**-nél az egyes bitek jellemző időpontjai (kezdetük, közepük és a végük) egy meghatározott alapidőtartam egész számú többszörösére helyezkednek el egymástól. Ez azt jelenti, hogy egy üzenet bitjei szigorú rendben követik

egymást, és a szinkronizmus akár több száz bit vagy karakter átvitele alatt fennáll.

Az ADÓ és VEVŐ szinkronizálását egy speciális bitcsoport (SYN vagy más terminológiában FLAG) érzékelése biztosítja. A vevő ezt érzékelve, már helyesen tudja a következő biteket vagy bitcsoportokat (karaktereket) értelmezni. A 8 bites SYN bitcsoport (karakter) hat egymást követő 1 értékű bitet tartalmaz: **01111110**. Ha az adó olyan bitfolyamot küld, amiben 5-nél több 1-es van egymás után, akkor beszúr az öt küldött 1-es után egy 0 értékű bitet, a VEVŐ, ha így átalakított bitsorozatot vesz, az öt 1-es után kiveszi a töltelék 0-át. Ha mégis hat darab 1-es jön egymás után, akkor a szinronizálást jelző SYN karakter érkezett. Természetesen a módszer sok egymás utáni nullákból álló bitsorozatok esetén is használható, de ekkor a 0-ák és 1-ek értelemszerűen szerepet cserélnek. Ezt hívják **bitbeszúrás**nak.




*3-16. ábra Szinkron és aszinkron átviteli módszerek*

A televízió technikában is alkalmazott ez a módszer: a soron belüli képpontok helyes megjelenítését a sorszinkron jellel (START jel!) szinkronizált sor-oszcillátor egy soron belül, közel állandó frekvenciája biztosítja.

**Aszinkron karakterorintált eljárások** legrégibb módszere a START-STOP átvitel. Ennél a szinkronizmus az adó és a vevő között csak egy-egy karakter átvitelének idejére korlátozódik. A leírtakat *a 3-16. ábrán* foglaltuk össze.

## 3.3.2 Digitális jelek vonali kódolása

A vonali kódolás a digitális jelátvitelben az a művelet, mely során a továbbítandó bitsorozathoz egy olyan jelsorozatot - vonali szimbólumsorozatot - rendelünk, mely az **átviteli út**on (a csatornán) a legkisebb torzítással halad át. Sebesség definíciók a digitális jelátvitelben:

- **bitsebesség**: az időegység alatt továbbított információ mennyisége [bit/s]
- **jelzési sebesség**: az időegység alatt továbbított vonali szimbólumok száma [Baud]

A fizikai vonalon való átvitelnél a bitek ábrázolására több lehetőség is van, amely közül a legegyszerűbb az, mikor minden bitet, értékétől függően két feszültségszinttel ábrázoljuk. Szokásos az „1" állapotot MARK-nak, a 0-át SPACE-nek is nevezni. Az alábbiakban a különféle, a gyakorlatban használt lehetőségeket tekintjük át, a következők figyelembe vételével:

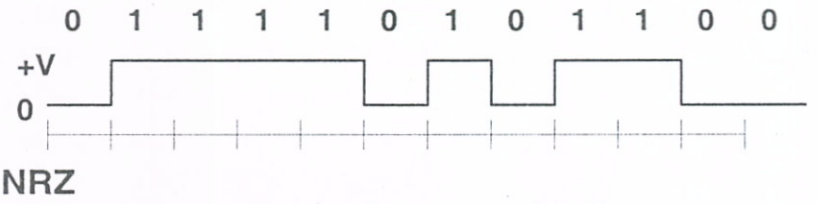
- A használt kódolás kis sávszélességű (kevés váltást tartalmaz), akkor felhasználásával több információ is átvihető egy adott kommunikációs csatornán.
- Kicsi legyen a jelek egyenfeszültség összetevője, mivel a magas DC szintű jelek jobban gyengülnek, így az átviteli távolság csökken.
- Legyen elég váltás a jelállapotokban, hogy az ADÓ és VEVŐ közötti szinkronizáció

ezen váltások segítségével, minden külön eszköz, külön vonal nélkül legyen megvalósítható.

- A jelek ne legyenek polarizáltak, így kétvezetékes átvitelnél közömbös lehet a bekötés.

### *NRZ — Non Return to Zero — Nullára vissza nem térő.*

Azaz mindig az a feszültség van a vonalon, amit az ábrázolt bit határoz meg. Ez a leginkább gyakori, "természetes" jelforma.



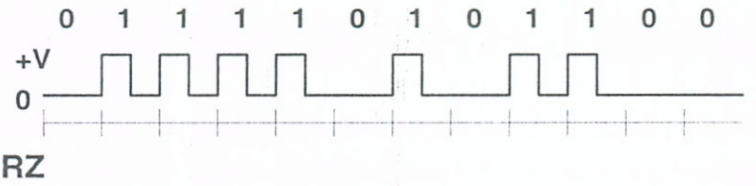

*3-17. ábra NRZ kódolás*

Ha egy bit 1-es, akkor a feszültség teljes bit idő alatt magas (H) szintű, ha 0-ás, akkor alacsony (L) szintű. Két, vagy több egymás utáni 1-es bit esetén a feszültség megszakítás nélkül H-ban marad a megfelelő ideig, az egyesek között nem tér vissza 0-ra. Nem túl jó megoldás, mert: magas egyenfeszültség összetevője van (V/2), nagy sávszélességet igényel közel 0 Hz-től (ha csak csupa 1-est vagy csupa 0-át tartalmaz a sorozat) az adatátviteli sebesség feléig (ha sorozat: 10101010...). Polarizált jel. Hosszú 0-s sorozatok esetén a szinkronizáció biztosítása itt is probléma, de a bitbeszúrási módszerrel ez megoldható.

### *RZ — Return to Zero — Nullára visszatérő.*

A nulla a "nyugalmi állapot", 1 bitnél a bitidő első felében a +V, a második felében a jel visszatér a 0-ra. Az NRZ kódoláshoz képest vannak előnyei: egyenfeszültség összetevője csak V/4, ha az adat csupa 1-est tartalmaz, akkor is vannak jelváltások (szinkronizáció).

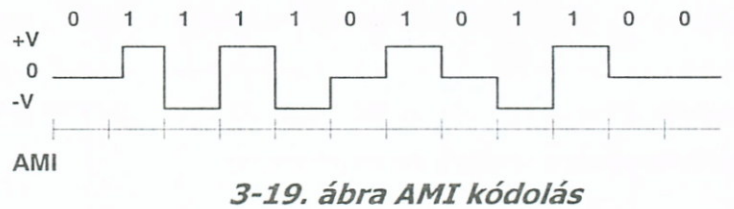
A legrosszabb a sávszélesség igénye: ez maga az adatátviteli sebesség (ha az adatfolyam csupa 1-est tartalmaz). Bárkiben felmerülhet, hogy mi a helyzet a sok nullát tartalmazó sorozat esetében, hiszen ekkor sincsenek jelváltások, azaz a szinkronizáció problémás. Ilyen esetben a már bemutatott bitbeszúrásos módszert alkalmazzák.



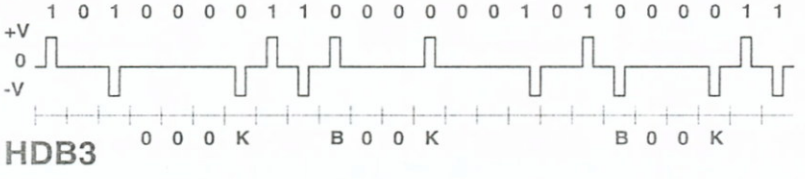

*3-18. ábra RZ kódolás*

### *AMI — Alternate Mark Inversion — váltakozó 1 invertálás*

A módszer nagyon hasonló az RZ módszerhez, de nullára szimmetrikus tápfeszültséget használ, így az egyenfeszültségű összetevője nulla. Minden 1-es-hez rendelt polaritás az előző 1-eshez rendelt ellentettje, a nulla szint jelöli a 0-át. Természetesen hosszú 0-s sorozatok esetén a szinkronizáció itt is problémás, de a bitbeszúrási módszer itt is használható.



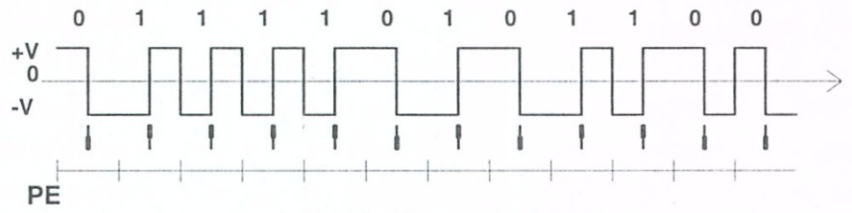

*3-19. ábra AMI kódolás*




*3-20. ábra HDB3 kódolás*

### *HDB3 — High Density Bipolar 3 — Nagy sűrűségű bipoláris 3*

Optikai kábeleknél használt. Mikor 4 egymás utáni „0" bit következik, az utolsót megváltoztatjuk 000K-ra, ahol K polaritása azonos az előző 1-eshez rendelt polaritással. A két egymás utáni azonos polaritásból a VEVŐ már tudja, hogy a második nem 1-et, hanem 0-át jelöl. Így már mindig van hosszabb nulla sorozatoknál is jelváltás, de a jelnek keletkezne egyen-feszültségű összetevője. Ezt is meg lehet oldani, ha a következő 0000 sorozat első B bitjét K bitjével azonos polaritásúnak választjuk. Mikor a VEVŐ egy B bitet vesz, azt hiszi, hogy az 1-hez tartozik, de mikor a K bitet is veszi, a B és K azonos polaritása miatt tudni fogja, hogy azok nullákat jelöltek.




*3-21. ábra PE kódolás*

### *PE — Phase Encode (Manchester) — Manchester kódolás*

Ennél jel-átmenet, ugrás jelképezi a biteket, de itt az ugrás irányának is jelentősége van: pl. 0-1 átmenet 1-es bitet, 1-0 átmenet 0-ás bitet jelöl.

Akkor, amikor több azonos bit követi egymást, akkor a jelnek a két bit között "félidőben" vissza kell térnie az eredeti szintre azért, hogy a következő bit idején ugyanolyan irányú átmenet következhessen. A sok előnyös tulajdonsága mellett az egyetlen hátránya a gyakori jelváltások miatti nagy sávszélessége, mivel az információt ennél a formánál a jelátmenetek hordozzák. Kiválóan alkalmas mágneses adatrögzítéshez is. Mivel minden bitnél van jelváltás, ezért a szinkronizálás nem okoz problémát.

## *3.3.3 Karakterek ábrázolása — az ASCII kódrendszer*

Mivel a karakter alapú átvitel még napjainkban is elterjedt módszer – számos internet protokoll használja, ezért ismerkedjünk meg a karkteres ábrázolás szabványos megoldásaival.

A számítógép, a külvilág, vagy egy másik számítógép közötti kapcsolat megvalósítása során az információk átvitele kódolva történik. A bitcsoportoknak — amelyek elvileg tetszőleges számú bitből állhatnak — **jelentést tulajdonítunk (kódolást végzünk**). Az információ átvitele során a bitcsoportokat továbbítjuk, és a vevő oldalon a **jelentésének megfelelően értelmezzük (dekódoljuk).**

Természetesen a bitcsoportokban lévő bitek száma, és a bitcsoportokhoz rendelt jelentés elvileg számtalan féle lehet, azonban a karakterek ábrázolásánál — a számítástechnika és informatika fejlődése során — csaknem kizárólag az ASCII kódrendszer vált egyeduralkodóvá.

Az **ASCII** rövidítés az American Standard Code for Information Interchange (=Amerikai szabványos kód az információ kölcsönös cseréjére) kifejezés rövidítése. ASCII karaktereknek nevezzük az ilyen módon kódolt bitcsoportokat

**Az ASCII karakterkészlet 128 hétbites, különböző kódot tartalmaz, amelyik mindegyike egy egyedi karaktert reprezentál.**

Természetesen felmerülhet a kérdés, hogy miért 7, és nem 8 bites kódot választottak, hiszen ekkor 256 különféle kód volna lehetséges (és ez a bájtos tárolási módhoz is illeszkedne). Az ASCII kód ANSI X3.4-1977-es szabványának függelékében szerepel az a megállapítás, hogy minimum 7 bit a legtöbb felhasználásban elegendő. Ez érthető is, mert ha az angol ABC-t tekintjük, annak 26 kis, 26 nagybetűje, az írásjelek (vessző, kérdőjel, stb.) valamint a 10 szám együttesen már 64 különféle karaktert jelent, aminek kódolásához már 6 bit szükséges, így a 7 bites kódhossz választás több, mint elegendő.

Az ANSI szabvány az ASCII karakterkészlet definiálásakor a kódokat két fő csoportba osztotta: **grafikus karakterek** és **vezérlő karakterek** csoportjába.

Grafikus karakterek esetén a megjeleníthető, látható, nyomtatható karaktereket értjük, míg a vezérlő karakterek, a megjelenítés vezérlésére, formájának kialakítására, valamint az információcsere vezérlésére szolgálnak.

A vezérlőkaraktereket három kategóriába soroljuk: — **információcsere vezérlők, — formátumot befolyásolók — információ_elkülönítők**. Az első 32 karakter, és az utolsó DEL karakter tartozik e kategóriákba.

Információcsere vezérlő karakterre példa a 04H kódú EOT karakter, amit annak a jelzésére használnak, hogy a karakterek átvitele befejeződött és azt jelzi, hogy nincs több átviendő karakter.

Formátum befolyásoló karakterekkel lehet a karaktersorozat megjelenési formáját befolyásolni. Például az LF Line Feed (Soremelés) karakter hatására a karakterek megjelenítése az adott pozícióban, de új sorban folytatódik. Pl. az A,B,C,D,LF,E,F karaktersorozat az

**ABCD**

**EF** formában jelenik meg.

Az információ elkülönítő karakterek az információ logikai értelemben való elkülönítésére szolgálnak. Ilyen módon lehetséges különböző hosszúságú karaktersorozatok — rekordok — átvitele. Ha például három különböző hosszúságú rekordot akarunk átvinni, akkor a rekordokat a Rekord Separator (RS) (kód: 1EH) karakterrel lehet egymástól elválasztani. A vezérlőkarakterek némelyike a fentiek egyikébe sem sorolható be, ezeket általános vezérlőkaraktereknek nevezzük.

### *A vezérlő karakterek jelentése*

Az ANSI szabvány minden ASCII karaktert részletesen meghatároz. A vezérlőkarakterek értelmezése és jelentése általánosan nem közismert, ezért a következőkben ezeket ismertetjük, így jobban megértjük egy adott készüléknél, berendezésnél való felhasználásuk célját és értelmét. Minden ASCII vezérlőkarakter speciális vezérlési feladat megvalósítására szolgál. Egy rövid ismertető:

**NUL (null)** : ez a karakter bárhol elhelyezhető az adatfolyamban, annak információ tartalmának megzavarása nélkül. Például lassú nyomtatóknál a kocsi-vissza (CR), soremelés (LF) karaktereket egy, vagy több NUL karakter követhet, aminek az a szerepe, hogy a mechanika képes legyen a parancsokat végrehajtani, azaz a nyomtatófejet a sor balszélére visszavinni.

**SOH (Start of Heading):** Adatátvitelnél a tényleges adatok átvitele blokkonként (karaktercsoportonként) történik. Az adatokra vonatkozó információkat (hány adat, milyen fajta, típusa, stb.) egy külön blokkban viszik át. Ennek a speciális blokknak a kezdetét jelöli a SOH karakter.

**STX (Start of Text):** Az előbbi speciális blokk az STX karakterrel fejeződik be, és egyben jelöli, hogy ezután adatblokkok (szöveg) átvitele következik.

**ETX (End of Text):** Az utolsó adatblokk befejezését jelöli (szövegvége).

**EOT (End of Transmission):** Ezt a karaktert szokás használni a teljes átvitel befejezésére. Maga az átvitel több speciális blokkból, és az azokat követő adatblokkokból állhat. Hogy jobban megértsük ezeknek a vezérlő karaktereknek a jelentését, tegyük fel, hogy egy terminálra (aminek címe mondjuk legyen 16-os), ki akarjuk vinni a "STOP" üzenetet. Az üzenet négy karakter hosszúságú. A szabványos átvitel szerinti karakter sorozat:

SOH,1,6,STX,S,T,O,P,ETX,4,EOT

Természetesen a konkrét megvalósításokban még más specifikus részek is lehetnek az üzenetben.

**ETB (End of Transmission Block):** ez a karakter használható egy-egy adatblokk átvitelekor a végső lezáró karakterként.

**ENQ (Enquiry):** Az adatátviteli rendszerekben, ha választ várunk egy távolabbi állomástól, ezt a karaktert küldjük ki (ki vagy?), hogy az beküldje az azonosítóját, és az állapotára (státuszára) vonatkozó információt.

**ACK (Acknowledge):** ezt a jelet a vevő küldi ki, azért, hogy választ kapjon a küldőtől.

**NAK (Negative Acknowledge):** a vevő küldi ki az adónak, ha valamilyen okból nem képes az adóval együttműködni, mert foglalt.

**BEL (Bell):** vezérlő karakter a figyelem felhívására, ha a vevő ezt veszi, általában hallható hangjelzést ad (csengő).

**BS (Backspace):** a formátumot befolyásolja, kiküldésével a vevőnél ugyanabban a sorban egy pozícióval való visszalépés (és esetleg az utolsó karakter törlése) hajtódik végre.

**HT (Horizontal Tabulation):** szintén formátum vezérlő, a vevő a jel hatására az aktuális karakter pozícióból a következő, előre meghatározott tabulátor pozícióba lép.

**LF (Line Feed):** soremelés az aktuális pozícióban, de a következő sorban folytatódik a nyomtatás. Sok esetben az ilyen karakter vétele, nem a fenti hatást, hanem új sor parancsot is jelent.

**VT (Vertical Tabulation):** formátumvezérlő, ugyanabban a pozícióban, de előre meghatározott sor átlépése után folytatódik a nyomtatás.

**FF (Form Feed):** formátumvezérlő, ugyanabban a pozícióban, de a szövegformátum (pl. lap) következő oldalának előre meghatározott sorában folytatódik a nyomtatás.

**CR (Carriage Return):** kocsi-vissza, ugyanazon sor első pozíciójába lép.

**SO (Shift Out):** az SI karakterrel együtt a grafikus karakterkészlet kiterjesztésére szolgál. Ekkor az SI karakter vételéig az ASCII grafikus karaktereknek más (pl. grafikai szimbólumok) jelentése van.

**SI (Shift In):** vétele után visszaáll az eredeti állapot és karakterértelmezés.

**DLE (Data Link Escape):** átvitelvezérlő karakter, ami az ezt követő korlátozott számú karakter jelentését megváltoztatja. Kizárólag további adatátviteli vezérlő funkciók biztosítására szolgál.

**DC1,DC2,DC3,DC4 (Device Controls):** vezérlőkarakterek Szokásos használata a DC1 és DC3 karaktereknek, a különböző átviteli

| DEC | HEX | KAR | DEC | HEX | KAR |
|---|---|---|---|---|---|
| 0 | 00 | NUL | 64 | 40 | @ |
| 1 | 01 | SOH | 65 | 41 | A |
| 2 | 02 | STX | 66 | 42 | B |
| 3 | 03 | ETX | 67 | 43 | C |
| 4 | 04 | EOT | 68 | 44 | D |
| 5 | 05 | ENQ | 69 | 45 | E |
| 6 | 06 | ACK | 70 | 46 | F |
| 7 | 07 | BEL | 71 | 47 | G |
| 8 | 08 | BS | 72 | 48 | H |
| 9 | 09 | HT | 73 | 49 | I |
| 10 | 0A | LF | 74 | 4A | J |
| 11 | 0B | VT | 75 | 4B | K |
| 12 | 0C | FF | 76 | 4C | L |
| 13 | 0D | CR | 77 | 4D | M |
| 14 | 0E | SO | 78 | 4E | N |
| 15 | 0F | SI | 79 | 4F | O |
| 16 | 10 | DLE | 80 | 50 | P |
| 17 | 11 | DC1 | 81 | 51 | Q |
| 18 | 12 | DC2 | 82 | 52 | R |
| 19 | 13 | DC3 | 83 | 53 | S |
| 20 | 14 | DC4 | 84 | 54 | T |
| 21 | 15 | NAK | 85 | 55 | U |
| 22 | 16 | SYN | 86 | 56 | V |
| 23 | 17 | ETB | 87 | 57 | W |
| 24 | 18 | CAN | 88 | 58 | X |
| 25 | 19 | EM | 89 | 59 | Y |
| 26 | 1A | SUB | 90 | 5A | Z |
| 27 | 1B | ESC | 91 | 5B | [ |
| 28 | 1C | FS | 92 | 5C |  |
| 29 | 1D | GS | 93 | 5D | ] |
| 30 | 1E | RS | 94 | 5E | ^ |
| 31 | 1F | US | 95 | 5F | - |
| 32 | 20 | SP | 96 | 60 | Ś |
| 33 | 21 | ! | 97 | 61 | a |
| 34 | 22 | " | 98 | 62 | b |
| 35 | 23 | # | 99 | 63 | c |
| 36 | 24 | $ | 100 | 64 | d |
| 37 | 25 | % | 101 | 65 | e |
| 38 | 26 | & | 102 | 66 | f |
| 39 | 27 | Ś | 103 | 67 | g |
| 40 | 28 | ( | 104 | 68 | h |
| 41 | 29 | ) | 105 | 69 | i |
| 42 | 2A | * | 106 | 6A | j |
| 43 | 2B | + | 107 | 6B | k |
| 44 | 2C | , | 108 | 6C | l |
| 45 | 2D | - | 109 | 6D | m |
| 46 | 2E | . | 110 | 6E | n |
| 47 | 2F | / | 111 | 6F | o |
| 48 | 30 | 0 | 112 | 70 | p |
| 49 | 31 | 1 | 113 | 71 | q |
| 50 | 32 | 2 | 114 | 72 | r |
| 51 | 33 | 3 | 115 | 73 | s |
| 52 | 34 | 4 | 116 | 74 | t |
| 53 | 35 | 5 | 117 | 75 | u |
| 54 | 36 | 6 | 118 | 76 | v |
| 55 | 37 | 7 | 119 | 77 | w |
| 56 | 38 | 8 | 120 | 78 | x |
| 57 | 39 | 9 | 121 | 79 | y |
| 58 | 3A | : | 122 | 7A | z |
| 59 | 3B | ; | 123 | 7B |  |
| 60 | 3C | < | 124 | 7C |  |
| 61 | 3D | = | 125 | 7D |  |
| 62 | 3E | > | 126 | 7E |  |
| 63 | 3F | ? | 127 | 7F | DEL |

sebességű adók és vevők közötti adatátvitel vezérlése az un. XON/XOFF protokoll szerint.

**SYN (Synchronous Idle):** átvitelvezérlő karaktert a soros szinkron adatátviteli rendszerekben használják. (Ez nem szinkron adatátvitel szinkron karaktere! – Az különben is 8 bites.)

**CAN (Cancel):** vétele azt jelzi a vevőnek hogy a küldött adatban hiba van, vagy az adatot törölni kell. Pontos jelentését adott esetben külön kell definiálni.

**EM (End of Medium):** vezérlőkarakter, ami az adatokat tartalmazó adathordozó fizikai végét, befejeződését jelzi.

**SUB (Substitute):** vezérlőkarakter, amit hibás, vagy érvénytelen karakter helyettesítésére használják.

**ESC (Escape):** vezérlőkarakter, a kódrendszer kiterjesztésére. A karakter maga egy jelölőkarakter, ami az utána következő véges számú bitalakzat speciális értelmezését jelzi.

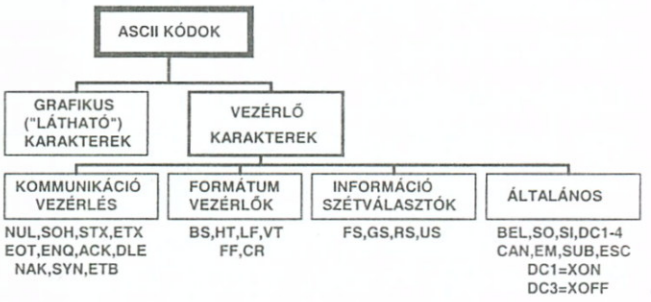
**FS,GS,RS,US (File-,Group-,Record-,Unit-Separator):** elválasztók, melyek fájl, csoport, rekord struktúrájú adatok elválasztására használhatók. FS a legmagasabb "rendű" elválasztó (azaz a struktúra legmagasabb szintjén álló egységek szétválasztására szolgál), míg US a legalacsonyabb.

**DEL (Delete):** karakter, ami az utolsónak bevitt karaktert helyettesíti, felülírja, gyakorlatilag törli. Mivel nem nyomtatható, és egyéb jelentése nincs, ezért adatátvitel során kitöltő (helyet és időt) karakternek használható.

| | | |
|---|---|---|
| NUL | NULL Character | Null (semmi) |
| SOH | Start of Heading | fejléc kezdete |
| STX | Start of Text | szöveg kezdete |
| ETX | End of Text | szöveg vége |
| EOT | End of Transmission | adás vége |
| ENQ | Enquiry | kérés |
| ACK | Acknowledge | elfogadás, nyugtázás |
| BEL | Bell | hangjelzés |
| BS | Backspace | visszaléptetés |
| HT | Horizontal Tabulation | vízszintes tabuláció |
| LF | Line Feed | soremelés |
| VT | Vertical Tabulation | függőleges tabuláció |
| FF | Form Feed | lapdobás |
| CR | Carriage Return | kocsi-vissza |
| SO | Shift Out | kódváltás |
| SI | Shift In | kód visszaváltás |
| DLE | Data Link Escape | adat átkapcsolás |
| DC1 | Device Control 1 | általános vez.jel (XON) |
| DC2 | Device Control 2 | általános vezérlőjel |
| DC3 | Device Control 3 | általán. vez.jel (XOFF ) |
| DC4 | Device Control 4 | általános vezérlőjel |
| NAK | Negative | negatív nyugtázás |
| SYN | Synchronous Idle | szinkronizáló jel |
| ETB | End of Transm. Block | egy blokk adás vége |
| CAN | Cancel | érvénytelenítés |
| EM | End of Medium | inform. hordozó vége |
| SUB | Substitute | helyettesítés |
| ESC | Escape | átkapcsolás |
| FS | File Separator | fájl elválasztó |
| GS | Group Separator | csoport elválasztó |
| RS | Record Separator | rekord elválasztó |
| US | Unit Separator | egység elválasztó |
| SP | Space | szóköz |
| DEL | Delete | törlés |

***Az ASCII vezérlőkarakterek***

Ahogy a felsorolásból is látható volt, az ASCII vezérlő karakterek alapvetően a karakter orientált átviteli eljárások kialakításának támogatását végzik. Ilyen szempontból az ASCII karakterkód készletet egy adatátviteli (kommunikációs) kódnak is nevezhetjük. Az ASCII kódrendszer kialakítása abban az időszakban történt meg, amikor még az adatmegjelenítő perifériák csaknem kizárólag mechanikus működésűek voltak (teletype, telex, géptávíró). Ezért a kódrendszerből hiányoznak azok a vezérlő kódok, amelyek a már csaknem kizárólagosan használt képernyő orientált rendszerekben használatosak: a kurzormozgató, képernyőtörlő, stb. funkciókhoz rendelt kódok. Mivel ezekről a szabvány nem rendelkezik, ezért bizonyos inkompatibilitás van az egyes megjelenítők között, hogy konkréten melyik vezérlőkaraktereket rendelték hozzá az adott funkcióhoz.




***Az ASCII kódok osztályozása***

A PC-k megjelenésekor az IBM által hozzáadott 1 bites kiterjesztéssel újabb 128 karakter használatát szabványosította, amely kódrendszer Latin1 néven ismert. Ez tartalmazza számos európai nyelv — pl. francia, német spanyol, stb. — speciális nemzeti karaktereit, valamint a görög ABC betűit, táblázat-rajzoló karaktereket is. Nyomtatóknál szokásos megoldás a nyomtató paramétereinek beállításához, felhasználó által definiált karakternek a nyomtató elektronikába való betöltéséhez, az ún. "escape szekvencia" használata.

### *Unicode*

A PC-k nemzetközi elterjedésével felmerült az igény hogy más nemzetek karaktereit is lehessen használni, ezért bevezették a kódlapokat: ez olyan 256 karakterből álló táblázat, amely az alsó felén az ASCII kódokat, míg a felső 128 karakter adott földrajzi terület népeinek speciális karakterei közül került ki. Magyarország számára a szláv nyelvekkel együtt a 852-es kódlap volt használható. Ez sajnos csak enyhítette, de nem oldotta meg a problémát, hiszen a kódlapok használata azonnal problémához vezetett kevert nyelvű szövegek esetében. Ezért célszerűbbnek látszott egy olyan kódrendszert kidolgozni — és természetesen nem 8 bites alapon — amely kódlap váltása nélkül is képes eltérő nyelvek speciális karaktereit helyesen ábrázolni.

1987-ben a XEROX cég egy új 16 bites kód fejlesztésébe kezdett. Az Unicode elnevezést az egyik fejlesztő javasolta, mivel: unique (egyedi), universal (univerzális) és uniform (egységes) karakterkódolást biztosít.

A kifejlesztett kezdeti specifikációkat tartalmazó Unicode 1.0 kódrendszer a világ összes írott ABC-jének kódolását tartalmazza. A fejlesztés során követett alapelvek a következők voltak:

**Teljesség** Az Unicode-t úgy tervezték, hogy a szövegek létrehozásához használt összes karaktert tartalmazza, ebbe még olyan "holt" nyelvek is tartoznak, mint pl. a szankszkrit.

**Egyszerűség és hatékonyság** Minden Unicode kód azonos, 16 bites hosszúságú és mindegyik egy tényleges létező karaktert reprezentál. Nincsenek sem vezérlő kódok sem vezérlő kódsorozatok. Ezek mind bonyolultabbá teszik a számítógépes szövegkezelést és feldolgozást.

**Egyértelműség** Minden kód egyértelműen egy karaktert jelent. Ezért egy karakter hibás olvasásakor csak egy a hibás és nincs előre mutató következménye.

**Pontosság** Minden kódolt karakter szabványos, a nyelvi szakértők által ismert és elfogadott.

| ASCII/8859-1 Text | |
|---|---|
| A | 0100 0001 |
| S | 0101 0011 |
| C | 0100 0011 |
| I | 0100 1001 |
| I | 0100 1001 |
| / | 0010 1111 |
| 8 | 0011 1000 |
| 8 | 0011 1000 |
| 5 | 0011 0101 |
| 9 | 0011 1001 |
| - | 0010 1101 |
| 1 | 0011 0001 |
| | 0010 0000 |
| t | 0111 0100 |
| e | 0110 0101 |
| x | 0111 1000 |
| t | 0111 0100 |

| Unicode Text | |
|---|---|
| A | 0000 0000 0100 0001 |
| S | 0000 0000 0101 0011 |
| C | 0000 0000 0100 0011 |
| I | 0000 0000 0100 1001 |
| I | 0000 0000 0100 1001 |
| | 0000 0000 0010 0000 |
| 天 | 0101 1001 0010 1001 |
| 地 | 0101 0111 0011 0000 |
| | 0000 0000 0010 0000 |
| س | 0000 0110 0011 0011 |
| ل | 0000 0110 0100 0100 |
| ا | 0000 0110 0011 0111 |
| ر | 0000 0110 0100 0101 |
| | 0000 0000 0010 0000 |
| α | 0000 0011 1011 0001 |
| κ | 0010 0010 0111 0000 |
| γ | 0000 0011 1011 0011 |

*3-22. ábra UNICODE példa*

Az Unicode tervezésénél is azért kompromisszumokat kellett kötni. Például a kompatibilitás érdekében az Unicode helyet foglal le néhány eredeti ASCII vezérlőkódnak, de ezt nem használja.

Az első 8192 helyet a szabványos alfabetikus karakterek számára foglalták le, helyet hagyva a később szabványba bekerülő régi írások ABC-inek. A következő 4096 kód tartalmazza az írásjeleket, matematikai, műszaki és piktogram szimbólumokat. Az ezt követő 4096 karaktert foglalták le a kínai, japán és koreai ABC-nek és írásjeleknek. A kód legnagyobb részét mintegy 27000 karaktert az egységesített Han karakterek részére foglalták le. Az egységesített Han karakterkészletet a GB 13000 Kínai Nemzeti Szabvány definiálja. Végül az utolsó előtti 5632 hely a felhasználók által használható és definiálható, az utolsó 495 kód az Unicode alá konvertálást segítő karakterek tartománya.

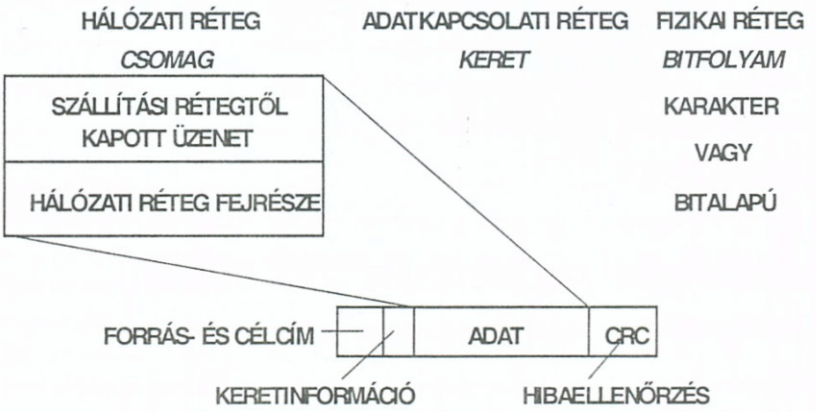
A rendszerszoftverek íróinak a kétbájtos kódkészlet miatt újra kell írniuk a programjaikat, de ez részben automatizálható, és nem érinti a teljes programot. Unicode egy de facto szabvány, az ISO ezt továbbfejlesztette, és ISO 10646 néven szabványosította. A szabvány már 32 bites kódokból felépített karakterkészleteket ír le. Ebben a szabványban a Unicode az 32 bites kódlap,aminek a felső 16 bitje nulla.

## 3.3.4 Adatkapcsolati protokollok

**Az adatkapcsolati protokollok feladata egy összeállított keret átvitele két csomópont között.** Az adatokat a hálózati rétegtől kapja az adatkapcsolati réteg, és az általa összeállított információcsomagokat, vagy más néven kereteket átadja a fizikai rétegnek, ami bitenként küldi át a fizikai közegen. Elsőként foglalkozzunk az üzenetszórásos adatátviteli közegek hozzáférési módszereivel.

### *Adatátviteli protokollok*

Mivel nincs olyan eljárás, amely folyamatos tetszőleges bitfolyamban a hibát képes jelezni, az átküldés hibátlanságát valahogy ellenőrizni kell, ezért **a bitfolyamot bitsorozat darabokra, vagy szokásos nevén keretekre kell szétdarabolni (tördelni).** Mindegyik keretet egy ellenőrző összeggel kell kiegészíteni. A keret megérkezése után ez az ellenőrző összeg a vételi oldalon a vett adatokból is kiszámításra kerül, és ha nem egyezik meg a küldő által számítottal, akkor a keretet a vevő eldobja, és a küldőnek ismételten el kell küldenie.




*3-23. ábra Információk a rétegekben*

A keretek átvitele két csomópont között első gondolatra egyszerűnek látszik, de ez csak látszat, hiszen az ADÓ és VEVŐ tulajdonságai, a keretek adatútját jelentő csatorna minősége, a váratlan eseményekre való felkészülés mind igényeket támaszt a protokollal szemben, ami ezért

változó bonyolultságú lehet. Külön gondot jelent a jól megkülönböztethető keret-kezdet és keret-vég határok jó megválasztása.

A bitfolyam átvitele bár a legtöbb esetben sorosan, egymás után bitenként történik, azonban sokszor célszerűbb a több egymást követő bitből álló bitcsoportos átvitelt alkalmazni. A bitcsoportok tetszőlegesek lehetnek, de legtöbbször az ASCII karakter kódolást alkalmazzák. Ilyenkor az átvitel során mindig egész számú karaktert viszünk át, azaz az átvitt információ alapegysége a karakter.

Ez a **karakterorientált átvitel** (szöveges információ átvitele esetén nem is rossz választás.). Amennyiben a bitfolyam bitjeit bitenként értelmezzük, **bitorientált átvitelről** beszélünk, ami általános információ átvitelekor rugalmasabb megoldás.

Elsőként a keretek képzéséről, majd a hibakezelésről írunk, majd a különféle keretátviteli protokollokat mutatjuk be.

### *Keretek képzése*

A bitfolyam (illetve a ezt alkotó bitcsoportok) keretekké tördelésére több módszer használatos, függően attól, hogy karakter- vagy bitorientált átvitelt használunk.

**Karakterszámláló módszer:** a keret fejlécében megadjuk a keretben lévő karakterek számát. Ez a VEVŐ oldalán meghatározhatóvá teszi a keret végét.

**Kezdő és végkarakterek alkalmazása karakterbeszúrással:** az előző módszernél a keret karaktereinek vételénél egy számlálót is folyamatosan kell egyesével csökkenteni (dekrementálni), amely kezdeti értékét is a keretből töltjük fel. Amikor a számláló értéke nulla, akkor értük el a keret végét. Jobb megoldás az ha egy speciális karaktersorozattal jelöljük a keret kezdetét és végét.

Szokásos megoldás a DLE STX karakterkettőssel jelezni a keret kezdetét és DLE ETX-el a keret végét. Ezek speciális, az ASCII kódtáblában megtalálható karakterek, és keret adatrészében lévő esetleges szövegekben nem fordulnak elő. Más a helyzet, ha karakteralapú módszerrel bináris adatokat (pl. egy programkódot) kívánunk átvinni. Ilyenkor, mivel bármilyen bináris bitcsoport előfordulhat, az adatmezőben megjelenhet a fenti két karakterkombináció, és ez hibás kerethatárt jelez. A megoldás: az ADÓ a keret összeállításakor az adatmezőben megjelenő minden DLE kód után, ***azonnal beszúr*** még egy DLE karaktert. A VEVŐ pedig, ha a DLE karakter vétele után ismét DLE következik, egyszerűen a második DLE-t eldobja.

A hálózati réteg által küldött üzenet:

`I T T E Z DLE V O L T`

Az ADÓ adatkapcsolati réteg keretképzése és karakter beszúrása:

`DLE STX I T T E Z DLE DLE V O L T DLE ETX`

Az VEVŐ adatkapcsolati rétege leválasztja a kettőzött beszúrt karaktert:

`DLE STX I T T E Z DLE V O L T DLE ETX`

A VEVŐ hálózati rétegének átadott üzenet: `I T T E Z DLE V O L T`

**Kezdő és végjelzők bitbeszúrással:** ezt a módszert a rugalmasabb bitorientált átvitelnél használják. Minden keret egy speciális (a gyakorlatban legtöbbször) 01111110 bitmintával kezdődik és végződik. Ha az ADÓ öt egymást követő 1-est tar-

talmazó mintát talál az adatmezőben, akkor egy 0 bitet szúr be utána. A VEVŐ a másik oldalon pedig ezt a beszúrt bitet az öt egymás utáni 1-es bit érzékelése után kiveszi a bitfolyamból.

A hálózati réteg által küldött üzenet:

`1111111110111`

Az ADÓ adatkapcsolati réteg keretképzése és bitbeszúrása:

`01111110 1111101111100111 01111110`

Az VEVŐ adatkapcsolati rétege leválasztja a beszúrt biteket:

`01111110 11111111110111 01111110`

A VEVŐ hálózati rétegének átadott üzenet:

`1111111110111`

Ha a 0 és 1 bitek kódolásán kívül még létezik nem használt kód, ez a keretképzésre felhasználható. Például, ha +U feszültség kódolja az 1, illetve –U feszültség a 0 állapotot, akkor a nulla feszültségszint jelölheti a keretek határát.

### *Hibakezelés*

Az adatátvitel és a kommunikáció fontos kérdése az átvitel során fellépő hibák kezelése. A vonalakon fellépő hibákat különböző fizikai jelenségek okozzák: termikus zaj, a vonalakat kapcsoló berendezések impulzus zaja, a légköri jelenségek (villámlás) okozta zajok. A rétegfelosztást figyelembe véve ezt az alsó három rétegben lehet megoldani, de igen jó minőségű vonalak esetén akár a felsőbb rétegekben is elvégezhető.

Az első hibakezelés a fizikai rétegben, a bitek és karakterek átvitelénél végezhető el. A

## HIBAKEZELÉS

**ÁTVITELI HIBA mindig lesz, akármilyen jó az átviteli rendszer (még akkor is, ha a jel-zaj viszony kicsi. Oka: termikus zaj, impulzus zaj.**

**Megoldás: hibakorlátozó kódolás. Alapelve: A kódszavakat redundanciával egészítjük ki, amely az átviendő bitek alapján képződik az ADÓ oldalon. A VEVŐ oldalon az eljárás megismétlésével és az eredmények összehasonlításával döntjük el, hogy van-e hiba.**

**HIBAKORLÁTOZÁS (ÉS HIBAKEZELÉS) FAJTÁI:**

- **HIBAJELZÉS: a redundancia alapján a hibát a vevő jelzi**
  - **Paritásvizsgálat (kódszóban lévő 1-esek száma a paritásbittel együtt páros (vagy páratlan) pl. 7 bites ASCII kód, a 8. bit a paritásbit)**
  - **Tömbparitás vizsgálat ( hossz- és keresztirányban) sokkal hatékonyabb mint az előző**
  - **CRC – Cyclic Redundancy Check Ciklikus Redundancia Ellenőrzés ld. később.**
- **HIBAJAVÍTÁS: a redundancia alapján a hibát a vevő ki is javítja, nagy redundanciát igényel, (pl. CD-nél 8-bitből 14 bit)**

*3-24 ábra Hibakezelés*

zajok időtartamából következően, lehetnek egyedi és csoportos bithibák. Gyakoribb esetekben a hibák fennállási ideje általában egy bit átviteli idejének a többszöröse,

ezért ezek a hibák csoportosan, hibacsomók formájában jelentkeznek. Ezek a **csoportos bithibák.** Mivel az adatátvitel blokkos (keretes) formában történik, ezért az eredmény az, hogy egy-egy blokk tönkremegy.

**Egyedi bithibák kezelésére a hibajavító (error correcting codes — ECC) és hibajelző kódok (error detecting codes) alkalmazása ad lehetőséget.** Mindkét esetben az adatblokkokat redundanciával küldik, hogy a vevő az esetleges hiba tényét felfedezhesse (hibajelzés), illetve megállapíthassa, hogy minek kellett volna jönnie (hibajavítás).

A redundáns kódok alkalmazásakor a bitcsoportot alkotó eredetileg **m** bites kódot **r** darab bittel egészítik ki, így a redundáns bitcsoport (aminek általános elnevezése: kódszó) n=m+r bitből fog állni.

## Hamming távolság

Két tetszőleges kódszót megadva, mindig megállapítható, hogy hány bitben különböznek egymástól: a két szó kizáró vagy (XOR) kapcsolata által adott eredményben az 1-esek száma adja a különbséget, és ezt szokták a két kódszó Hamming távolságának nevezni.

Pl.: az A karakter (41H) és a B karakter (42H) Hamming távolsága:

100 0001 .XOR. 100 0010 = 000 0011 azaz, kettő, mert két bitben különböznek. B és C karakter (43h) között: 100 0010 .XOR. 100 0011 = 000 0010 csak egy.

LEGYEN EGY 3 BITBŐL ÁLLÓ KÓD:

| | |
|---|---|
| X0 | 0 0 0 |
| X1 | 0 0 1 |
| X2 | 0 1 0 |
| X3 | 0 1 1 |
| X4 | 1 0 0 |
| X5 | 1 0 1 |
| X6 | 1 1 0 |
| X7 | 1 1 1 |

- Ha X1, X2, X4 és X7 kódokat használjuk fel adattovábbításra akkor H=2. Ekkor, ha 1 bit meghibásodik, akkor azt felismerjük.
- Ha minden kódszót felhasználunk, akkor a Hamming távolság H=1. Bármelyik bit meghibásodik az átvitel során, nem fedezzük fel.
- Ha csak az X0 és X7 kódokat használjuk fel adat továbbításra, akkor H=3. Ekkor, ha 1 bit meghibásodik, akkor azt felismerjük, és ki is tudjuk javítani!

**EGYSZERES BITHIBÁK ESETÉN:**

Nem redundáns kód használata: H=1.
Hibafelfedés feltétele: H=>2
Hibajavítás feltétele: H=>3

***3-25. ábra Példa Hamming kódra***

Egy kódszavakból álló kódrendszerben kiszámítjuk az összes kódszó pár egymás közötti Hamming távolságát, és ezek közül a legkisebb lesz a kódrendszerre jellemző Hamming távolság.

ASCII kódrendszerben a Hamming távolság: 1. A paritásbittel történő kiegészítés során olyan kódszavakat generálunk, amelyek hossza eggyel nagyobb, mint az eredeti kódszó. Ez természetesen redundáns kód, de a Hamming távolságuk 2, és ezért egyszeres bithibák kimutatására alkalmasak.

Ennek az a jelentősége, hogy **ha két kódszó k Hamming távolságú, akkor az egyik a másikba k darab egyedi hibával konvertálódhat át.**

Ha egy kódszavakból álló blokkhoz szavanként egyetlen paritásbitet adunk, akkor csoportos hiba esetén a hibajelzés valószínűsége csak 0.5 lesz (pl. a kódszóban lévő valamelyik két bit az ellentettjére változik, vagy egyszerre kettő azonos módon változik).

A hibajelzés valószínűsége olyan módon növelhető, hogy a blokkot n*k elemű mátrix-

nak tekintjük, ahol n a paritásbittel kiegészített kódszó hossza, és k a blokkban lévő kódszavak száma.

A paritásbitet oszloponként is kiszámítjuk, és a mátrix utolsó soraként azt is elküldjük. A vett blokkot a vevő mindkét paritás szerint ellenőrzi.

A módszer egyetlen n bites csoporthiba észlelésére képes, mert oszloponként 1 bit változik csak.

### CRC — Cyclic Redundancy Check

Csoportos bithibák esetén, mivel egy kódszóban több bit is megváltozhat, a paritásvizsgálat nem hatékony, hanem inkább egy másik hibafeleismerő módszert használnak, amely a kódszóban lévő bitek sorrendjére is tekintettel van. Ez a hibavédelmi eljárás úgy működik, hogy egy keretnyi adatot egy előre az ADÓ és a VEVŐ oldalon is ismert bitsorozattal „elosztunk", és a „maradékot" a keret részeként továbbítjuk. Az osztási elvből következik, hogy az osztandó bármelyik elemének a megváltoztatása az osztás eredményét befolyásolja. Az osztási művelet során mind az osztandó keret bitsorozatát, mind a közös ismert osztó bitsorozatot egy 1, és 0 együtthatókból álló n-edfokú polinomnak tekintjük. Az osztó bitsorozatból képzett polinom neve: **generátorpolinom**. Az így képzett polinomokkal polinomosztást végzünk.

A vevő oldalon ugyanezt az osztást végezzük el, és ha a keret részeként átküldött maradékkal egyezik az osztás maradéka, akkor hibátlannak fogadjuk el a keretet. Három szabványos generátorpolinom terjedt el:

| | |
|---|---|
| CRC-12 | 6 bites karakterek átvitelekor használt |
| CRC-16 | 8 bites karakterek átvitelekor használt |
| CRC-CCITT | 8 bites karakterek átvitelekor használt |

Az olyan 16 bites ellenőrző összeg, amit az utóbbi két módszer alapján állítunk elő,

**CRC – Cyclic Redundancy Check**

**Csoportos bithibák kezelése. CRC - Cyclic Redundancy Check.**

***A lényeg: az osztás eredményét több szomszédos számjegy határozza meg.***

**Egy k bitből álló keret egy r bitből álló redundanciával kiegészítve (k+r) bites blokk kerül átvitelre. A redundancia polinom osztással képződik. A keret bitjeit „elosztjuk" egy generátor polinommal. Pl. $G(x)= x^4+x+1$ ->10011**

**A VEVŐ a a (k+r) bitből álló sorozatot ugyanavval a G(x) genrátor polinommal osztja. Ha az osztás maradékmentes, akkor feltételezhető, hogy nincs hiba.**

1. $x^r M(x)$ (M(x)=keretpolinom) ALGORITMUS !
2. $x^r M(x)/G(x) = Q(x)+R(x)/G(x)$ R(x)= maradékpolinom
3. $T(x) = x^r M(x)+R(x)$ a „+" itt modulo 2 összeadás !!!
4. $T(x)/G(x) = [x^r M(x)+R(x)]/G(x) = Q(x) + R(x)/G(x) + R(x)/G(x)$

*3-26. ábra CRC*

ezért ezek a hibák csoportosan, hibacsomók formájában jelentkeznek. Ezek a **csoportos bithibák.** Mivel az adatátvitel blokkos (keretes) formában történik, ezért az eredmény az, hogy egy-egy blokk tönkremegy.

**Egyedi bithibák kezelésére a hibajavító (error correcting codes — ECC) és hibajelző kódok (error detecting codes) alkalmazása ad lehetőséget.** Mindkét esetben az adatblokkokat redundanciával küldik, hogy a vevő az esetleges hiba tényét felfedezhesse (hibajelzés), illetve megállapíthassa, hogy minek kellett volna jönnie (hibajavítás).

A redundáns kódok alkalmazásakor a bitcsoportot alkotó eredetileg **m** bites kódot **r** darab bittel egészítik ki, így a redundáns bitcsoport (aminek általános elnevezése: kódszó) n=m+r bitből fog állni.

## Hamming távolság

Két tetszőleges kódszót megadva, mindig megállapítható, hogy hány bitben különböznek egymástól: a két szó kizáró vagy (XOR) kapcsolata által adott eredményben az 1-esek száma adja a különbséget, és ezt szokták a két kódszó Hamming távolságának nevezni.

Pl.: az A karakter (41H) és a B karakter (42H) Hamming távolsága:

100 0001 .XOR. 100 0010 = 000 0011 azaz, kettő, mert két bitben különböznek. B és C karakter (43h) között: 100 0010 .XOR. 100 0011 = 000 0010 csak egy.

LEGYEN EGY 3 BITBŐL ÁLLÓ KÓD:

| | |
|---|---|
| X0 | 0 0 0 |
| X1 | 0 0 1 |
| X2 | 0 1 0 |
| X3 | 0 1 1 |
| X4 | 1 0 0 |
| X5 | 1 0 1 |
| X6 | 1 1 0 |
| X7 | 1 1 1 |

- Ha X1, X2, X4 és X7 kódokat használjuk fel adattovábbításra akkor H=2. Ekkor, ha 1 bit meghibásodik, akkor azt felismerjük.
- Ha minden kódszót felhasználunk, akkor a Hamming távolság H=1. Bármelyik bit meghibásodik az átvitel során, nem fedezzük fel.
- Ha csak az X0 és X7 kódokat használjuk fel adat továbbításra, akkor H=3. Ekkor, ha 1 bit meghibásodik, akkor azt felismerjük, és ki is tudjuk javítani!

**EGYSZERES BITHIBÁK ESETÉN:**

Nem redundáns kód használata: H=1.
Hibafelfedés feltétele: H=>2
Hibajavítás feltétele: H=>3

***3-25. ábra Példa Hamming kódra***

Egy kódszavakból álló kódrendszerben kiszámítjuk az összes kódszó pár egymás közötti Hamming távolságát, és ezek közül a legkisebb lesz a kódrendszerre jellemző Hamming távolság.

ASCII kódrendszerben a Hamming távolság: 1. A paritásbittel történő kiegészítés során olyan kódszavakat generálunk, amelyek hossza eggyel nagyobb, mint az eredeti kódszó. Ez természetesen redundáns kód, de a Hamming távolságuk 2, és ezért egyszeres bithibák kimutatására alkalmasak.

Ennek az a jelentősége, hogy **ha két kódszó k Hamming távolságú, akkor az egyik a másikba k darab egyedi hibával konvertálódhat át.**

Ha egy kódszavakból álló blokkhoz szavanként egyetlen paritásbitet adunk, akkor csoportos hiba esetén a hibajelzés valószínűsége csak 0.5 lesz (pl. a kódszóban lévő valamelyik két bit az ellentettjére változik, vagy egyszerre kettő azonos módon változik).

A hibajelzés valószínűsége olyan módon növelhető, hogy a blokkot n*k elemű mátrix-

nak tekintjük, ahol n a paritásbittel kiegészített kódszó hossza, és k a blokkban lévő kódszavak száma.

A paritásbitet oszloponként is kiszámítjuk, és a mátrix utolsó soraként azt is elküldjük. A vett blokkot a vevő mindkét paritás szerint ellenőrzi.

A módszer egyetlen n bites csoporthiba észlelésére képes, mert oszloponként 1 bit változik csak.

### CRC — Cyclic Redundancy Check

Csoportos bithibák esetén, mivel egy kódszóban több bit is megváltozhat, a paritásvizsgálat nem hatékony, hanem inkább egy másik hibafeleismerő módszert használnak, amely a kódszóban lévő bitek sorrendjére is tekintettel van. Ez a hibavédelmi eljárás úgy működik, hogy egy keretnyi adatot egy előre az ADÓ és a VEVŐ oldalon is ismert bitsorozattal „elosztunk", és a „maradékot" a keret részeként továbbítjuk. Az osztási elvből következik, hogy az osztandó bármelyik elemének a megváltoztatása az osztás eredményét befolyásolja. Az osztási művelet során mind az osztandó keret bitsorozatát, mind a közös ismert osztó bitsorozatot egy 1, és 0 együtthatókból álló n-edfokú polinomnak tekintjük. Az osztó bitsorozatból képzett polinom neve: **generátorpolinom**. Az így képzett polinomokkal polinomosztást végzünk.

A vevő oldalon ugyanezt az osztást végezzük el, és ha a keret részeként átküldött maradékkal egyezik az osztás maradéka, akkor hibátlannak fogadjuk el a keretet. Három szabványos generátorpolinom terjedt el:

| | |
|---|---|
| CRC-12 | 6 bites karakterek átvitelekor használt |
| CRC-16 | 8 bites karakterek átvitelekor használt |
| CRC-CCITT | 8 bites karakterek átvitelekor használt |

Az olyan 16 bites ellenőrző összeg, amit az utóbbi két módszer alapján állítunk elő,

**CRC – Cyclic Redundancy Check**

**Csoportos bithibák kezelése. CRC - Cyclic Redundancy Check.**

***A lényeg: az osztás eredményét több szomszédos számjegy határozza meg.***

**Egy k bitből álló keret egy r bitből álló redundanciával kiegészítve (k+r) bites blokk kerül átvitelre. A redundancia polinom osztással képződik. A keret bitjeit „elosztjuk" egy generátor polinommal. Pl. $G(x)= x^4+x+1$ ->10011**

**A VEVŐ a a (k+r) bitből álló sorozatot ugyanavval a G(x) genrátor polinommal osztja. Ha az osztás maradékmentes, akkor feltételezhető, hogy nincs hiba.**

1. $x^r M(x)$ (M(x)=keretpolinom) ALGORITMUS !
2. $x^r M(x)/G(x) = Q(x)+R(x)/G(x)$ R(x)= maradékpolinom
3. $T(x) = x^r M(x)+R(x)$ a „+" itt modulo 2 összeadás !!!
4. $T(x)/G(x) = [x^r M(x)+R(x)]/G(x) = Q(x) + R(x)/G(x) + R(x)/G(x)$

*3-26. ábra CRC*

észleli az összes egyes és kettős hibát, az összes páratlan hibás bitet tartalmazó hibát, az összes 16 bites, vagy ennél rövidebb csoporthibát, a 17 bites csoporthibák 99.997%-át, valamint a 18 bites és annál hosszabb csoporthibák 99.998%-át. [1]

**A lényeg: az osztás eredményét több szomszédos számjegy határozza meg.**

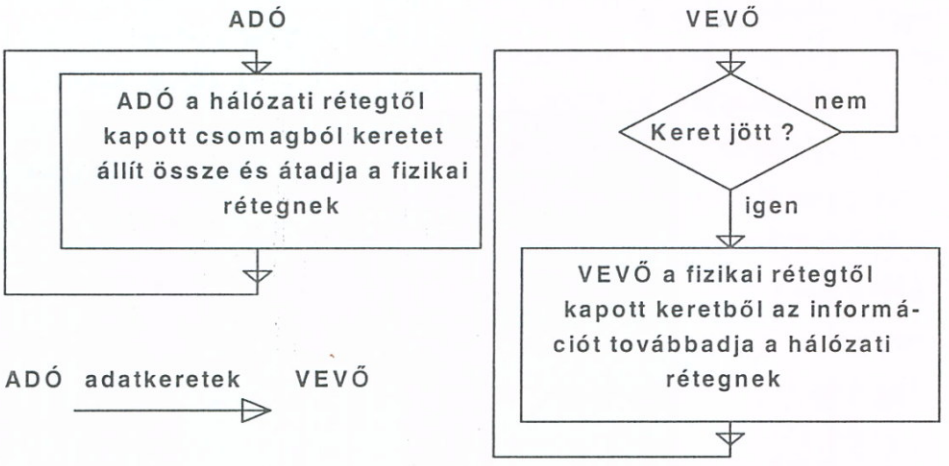
Hogyan tudunk meggyőződni nagy valószínűséggel, hogy egy adatátviteli csatornán átvitt szám nem hibásodott – e meg? Legyen a szám M. Osszuk el számot G (prím)számmal.

Ekkor általában M / G = Q + R , ahol R a maradék. Vigyük át az M számot és R maradékot. Ha a másik oldalon az M-R számot osztjuk G-vel (ez a másik oldalon is ismert, és azonos osztó), akkor: (M-R) / G = Q egész számot kapunk maradék nélkül! (Q értéke nem is érdekes...)

Pl: m=123456; g=129 m/g=957 r=3. Átmegy: 123456,3 test: 123453/129 maradéka: 0.

Persze, ha 123456 úgy változna, hogy 3-at kivonva az eredmény 129 többszöröse lenne, akkor elfogadnánk...

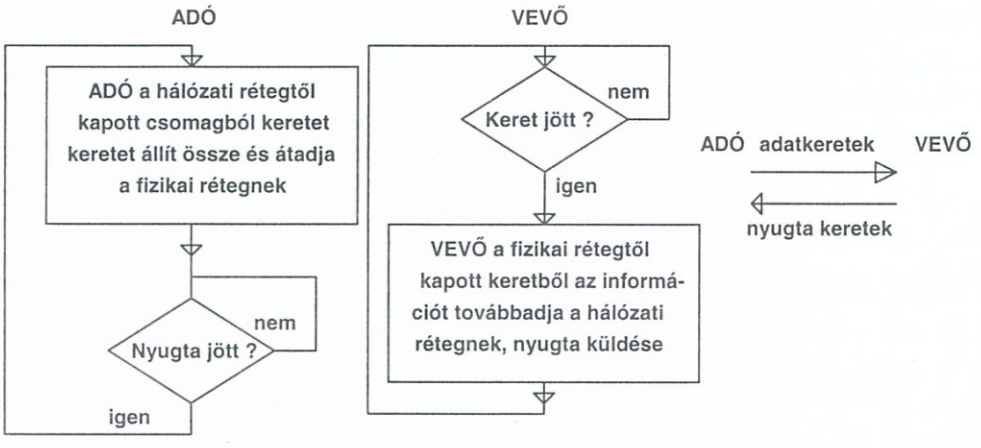
## 3.3.5 Keretek átvitele két pont között




*3-27. ábra Korlátozás nélküli, egyirányú protokoll*

A következőkben röviden az ADÓ és VEVŐ kommunikációs csomópontok között, adatkeretek segítségével megvalósított adatkapcsolati protokollokat mutatjuk be. Az adatkapcsolati réteg tördeli keretekké a bitfolyamot, és látja el fejrésszel, amelyet a VEVŐ oldali adatkapcsolati réteg távolít el, és állítja vissza a bitfolyamot.

### Korlátozás nélküli, egyirányú (szimplex) protokoll



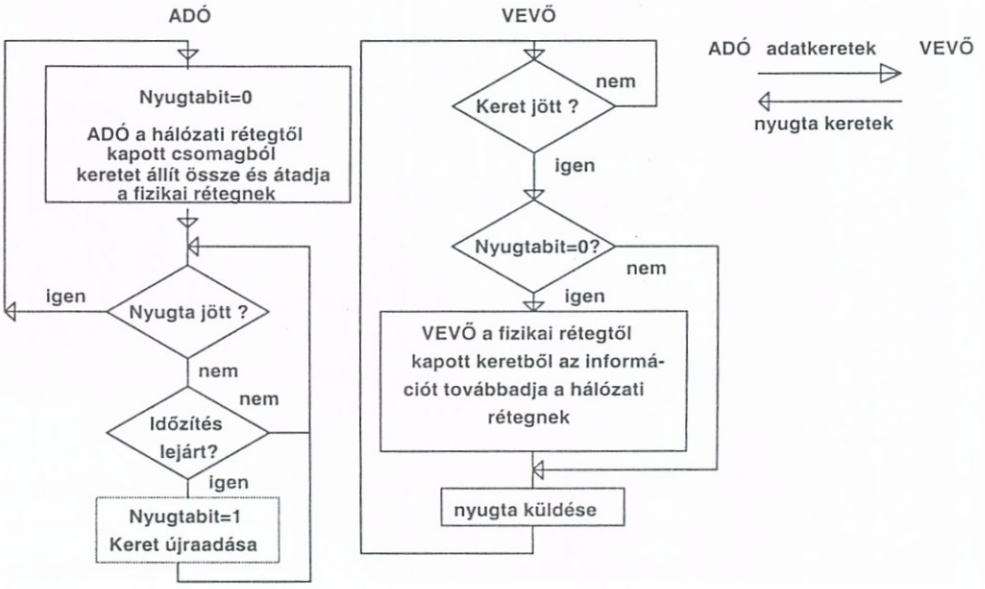

*3-28. ábra Egyirányú "megáll és vár" protokoll blokkvázlata*

Az első vizsgált protokoll a lehető legegyszerűbb: az adatátviteli sebesség, a feldolgozás nincs korlátozva: amilyen sebességgel küldi az ADÓ a kereteket, a VEVŐ ugyanilyen sebességgel képes ezt venni. Ez a gyakorlatban azt jelenti, hogy az ADÓ és a

VEVŐ hálózati rétege mindig készen áll, a feldolgozási idő elhanyagolható, és a keretek esetleges tárolására szolgáló puffer kapacitás végtelen. Az adatkapcsolati rétegek közötti csatorna hibamentes, kerethiba, keretvesztés nem fordul elő. Az átvitel egyirányú.

### *Egyirányú „megáll és vár" protokoll*

A valóságban nagyon sok esetben a VEVŐ nem képes megfelelő sebességgel feldolgozni a kereteket, azaz valahogy az ADÓ-t le kell lassítani olyan mértékben, hogy a VEVŐ küldött kereteket mindig fel tudja dolgozni.




*3-29. ábra Egyirányú összetett protokoll blokkvázlata*

Ez csak egy módon lehetséges: informálni kell az adót arról, hogy mikor küldheti a következő keretet, azaz a vétel és a feldolgozás tényét nyugtázni kell.

Vagyis a protokoll megköveteli az ADÓ-tól, hogy egy keret elküldése után addig várjon, amíg a kis üres (nincs adat!) nyugtakeret meg nem érkezik. Ezt a protokollt szokták „megáll ás vár" (stop and wait) protokollnak nevezni.

A protokoll jól működik az adatkeretek átvitelekor, hiszen a VEVŐ csak akkor küld vissza nyugtát, ha a keret vétele helyes volt. **Mi van azonban akkor, ha VEVŐ által küldött nyugtakeret sérül meg?**

Mivel nyugta nincs, az ADÓ egy bizonyos idő múlva ismét elküldené a nem nyugtázott keretet, amit a VEVŐ ismételten venne, azaz a benne lévő adatok megkettőződve kerülnének a hálózati réteghez.

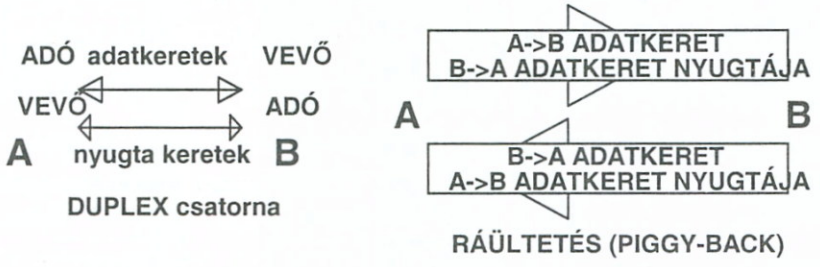
Ez sajnos súlyos hiba. A VEVŐ-nek kell egy olyan módszert alkalmaznia, amely megkülönböztethetővé teszi a számára az először kapott kereteket az újraadásra kerültektől.

### *Egyirányú összetett protokoll*

Ennek egyszerű megoldása az, hogy az ADÓ egy számot helyez el minden elküldendő keret fejrészébe, ezáltal a VEVŐ eldöntheti, hogy először adott, vagy ismételt keretről van-e szó. Mivel a keretek és a nyugták egymás után vannak, ezért elegendő 1 bittel jelezni az újraküldés tényét.

Nézzük: a k-adik keretre (amelynek újraküldési bitje 0 volt, jelezve az első küldést) a VEVŐ nyugtát küld, de az elvész. Az ADÓ mivel a k-adik keretet elküldte, de nem nyugtázták (legalábbis azt hiszi), egy adott időzítés lejárta után ismételten elküldi a keretet, de már 1-es újraküldési bittel).

A VEVŐ ezt véve, a bit alapján már tudja, hogy ezt már vette, ezért nyugtát küld visszza az elveszett helyett, de a keretet eldobja.




*3-30. ábra Kétirányú protokoll*

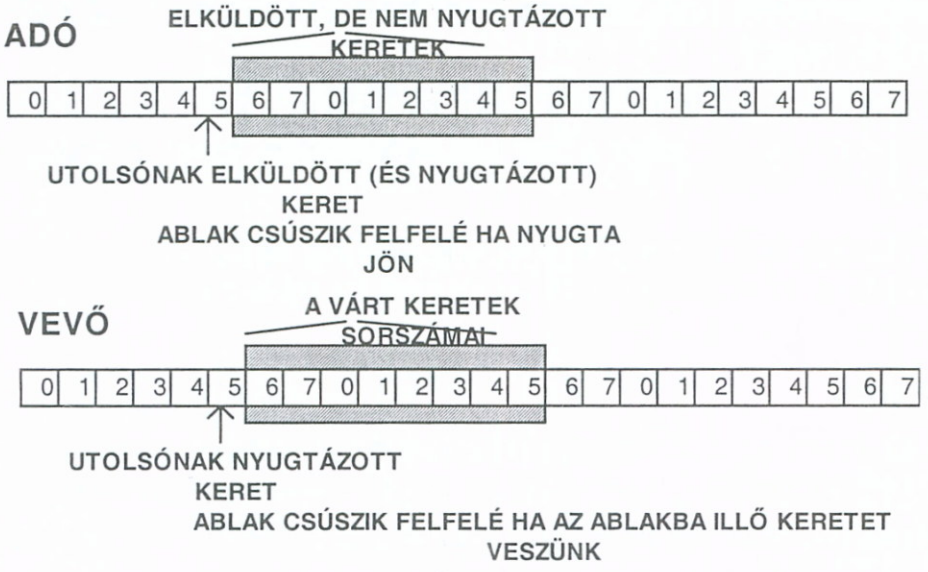
### *Kétirányú protokollok*

Az előző esetekben az adatátvitel egyirányú volt, bár az utolsó két esetnél a nyugtázás miatt az ellenirányú átvitelre is szükség volt az ADÓ informálása miatt. A gyakorlatban az adatátvitel legtöbbször kétirányú, ezért célszerű ezt a kialakítást is megvizsgálni. Jobb megoldás, ha mindkét irány számára ugyanazt a csatornát használjuk, hiszen az adatkereteket a nyugtakeretektől a keret fejrészében elhelyezett jelző meg tudja különböztetni, és ez a keret vételekor azonosítható.

Egy egyszerű megoldással az átviendő keretek számát csökkenthetjük: bármelyik irányba tartó adatkeretre ráültethetjük az előző ellenirányú adatkeret nyugtáját. Ezt szokták **ráültetési (piggy-back) technikának** is hívni. Hogy egy nyugta akkor is visszajusson, ha éppen nincs visszafelé küldött adatkeret, célszerű egy adott időzítés lejártakor a VEVŐ-nek önállóan útnak indítani. Persze, ha az adó eltérő időzítése miatt újra elküldi a keretet, akkor ez problémát jelent.

### *Csúszóablakos protokoll*

Az eddigiekben feltételeztük hogy a csatornán mindig egy adatkeret, majd rá válaszul egy nyugtakeret halad. A legtöbb esetben az adatátviteli vonalak jó minősége miatt, megengedhetjük, hogy a csatornán egymás után több keretet küldünk, és ha az utolsónak küldött keret nyugtáját kapjuk vissza, akkor ez a tény az előzőleg elküldött keretek nyugtázását is jelenti. Ezt lehetővé tévő megoldásokat **csúszóablakos (sliding window) vagy forgóablakos protokoll-**oknak nevezik. A könyvben az első megnevezést fogjuk használni.




*3-31. ábra Csúszóablakos protokoll*

A protokollban minden egyes kimenő keret egy 0-max (példánkban max=7) közötti sorszámot kap. A listában szereplő sorszámú keretek az **adási ablakba** (sending window) esnek. Tehát az ADÓ adási ablakában az elküldött, de még nem nyugtázott keretek vannak. Az ADÓ az adási ablakában lévő mind a 8 keretet elküldi sorban egymás után, és várja a nyugtát. Vagyis az ADÓ nem 1, hanem 8 keretet küld el nyugtázás nélkül. A 8. keret elküldése után kezdi várni a nyugtát.

**Melyik a legrosszabb eset?** Amikor az adó a sorban elküldött keretek közül még az elsőnek küldött nyugtáját sem kapta meg. Ilyenkor mind a 8 keretet ismét el kell küldeni. **A legjobb eset** az, amikor az utolsónak küldött keret nyugtáját kapja vissza az ADÓ, mert akkor mind a 8 keret átvitele sikerült, és újabb 8-at lehet küldeni.

Ezt a megoldást **csővonal**-nak (**pipelining**) hívják, utalva arra a szemléletes képre, hogy a keretek egy „csőbe" haladnak, sorban egymás után

A lényeg az, hogy a sorban elküldött keretek sorszámaiból egy aktualizált listát tart fenn az ADÓ. Mikor egy nyugta megérkezik, az ablak alsó fele feljebb csúszik, lehetővé téve újabb keret elküldését. Nem kell a kereteket egyenként nyugtázni, ha pl. az ADÓ a *3-31-es ábrán* látható módon elküldte a 6,7,0,...5-es sorszámú kereteket, és az 1-es keretsorszámot tartalmazó nyugtát kapja meg, ez azt jelenti, hogy nyugtázott a 6,7,0,1 keret.

A VEVŐ egy **vételi ablakot** (recieving window) tart fenn, amely az elfogadható keretek sorszámait tartalmazza. Bármelyik ablakon kívüli keret érkezésekor az eldobódik. Ha a k-adik keret érkezik, akkor rá a nyugta a következő két feltétel teljesülése esetén lesz visszaküldve:

- A k-adik keret még nem lett nyugtázva.
- Minden keretet az elsőnek várt (az ábrán a 6.) és a k-adik között már vettünk.

### Visszalépés n-el technikájú protokoll

Mi van azonban akkor, ha egy keret a sorban megsérül? Két megközelítés ismert: az egyik a címben már megnevezett **visszalépés n-el** (go back n) protokoll. Ennél a módszernél a VEVŐ, a hibás keret utáni kereteket nyugtázatlanul eldobja, kényszerítve az ADÓ-t az ismétlésre.

### Szelektív ismétlő protokoll

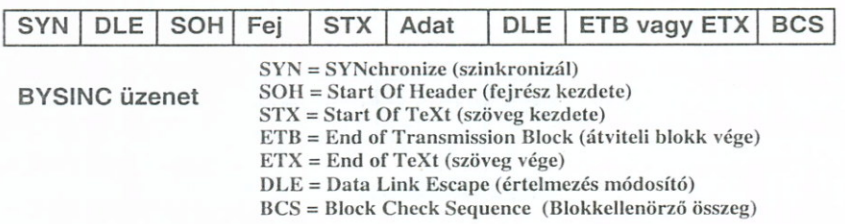
Egy másik használt általános hibakezelési eljárást szelektív ismétlésnek (selective repeat) hívják, és működése már az előzőek és az elnevezése alapján már kitalálható: ennél protokollnál a hibás keretet követő, összes jó keret tárolódik. Természetesen ez a protokoll jóval bonyolultabb, mert kezelni kell a sorban esetleg hibásan érkező keretek újraküldését is.

### *Példák*

A következőkben az előbbiek illusztrálására két konkrét gyakorlati megoldást mutatunk be.

### IBM BISYNC (BInary SYNchronous Communication)

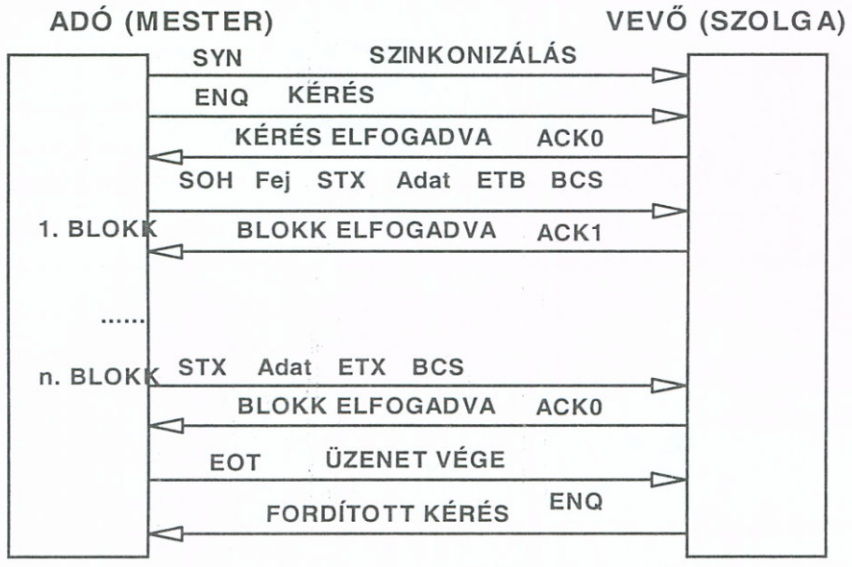
[11] Karakterorientált szinkron eljárás. Hasonlóan a többi ismert karakterorientált eljáráshoz, meglehetős szabadsággal kezeli a vezérlő karaktereket, így ezek az eljárások általában nem teljes mértékben kompatibilisek egymással. Széles körben használják távoli terminálok lekérdezésére, vala-




*3-32. ábra BYSINC keretformátuma*

mint egyéb alkalmazásokra. **Fél-duplex vonalakhoz fejlesztették ki, és egyaránt működik többpontos és kétpontos típusú állomás kapcsolatok esetén is.** A BISYNC üzenetformátumát a *3-32. ábrán* láthatjuk.

A fej(rész) mezőinek tartalma az aktuális hálózattól függ. A protokoll nem definiálja azokat, de fejlécet nem is kötelező használni (Például nem a fejlécben van elhelyezve az adó és a vevő címe.) Az ETB több egymást követő blokk esetén egy blokk lezárását jelenti. Az ETX az utolsó blokkot zárja le. Többpontos vonalon levő állomások megcímzését nem a fejrészben lévő cím, hanem egy külön vezérlőüzenet végzi. Minden blokk végén egy, vagy két karakterből álló blokkellenőrző sorozat (Block Check Sequence = BCS) is átvitelre kerül.

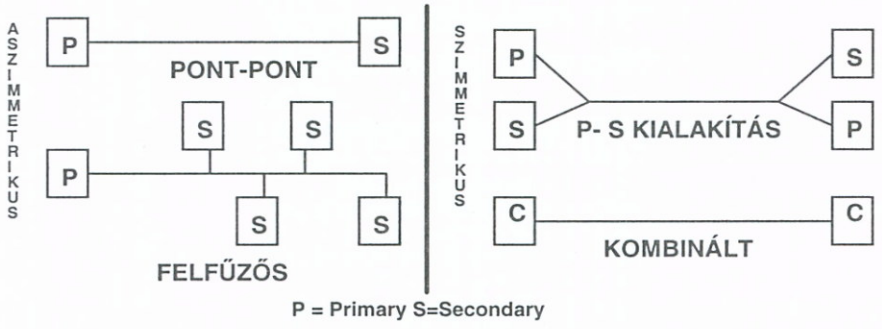



*3-33. ábra BYSINC üzenetváltás*

A *3-33. ábrán* követhetjük végig egy üzenetváltás folyamatát. Egy blokk átvitele után az átvitel irány megfordul, és vevő nyugtát küld az adónak: hibás vétel esetén a NAK karaktert, helyes vétel esetén felváltva kétfajta nyugtát küld vissza az ACK0 és ACK1 jelüt.

Az IBM BYSINC terminológiában az adást kezdeményező és végrehajtó állomást **mester**-nek, míg az üzenetet vevő másik állomást, **szolgá**nak szokás nevezni.

Ha több szolgaállomás egyidőben ENQ karaktert küldene el a mesternek a kommunikáció kezdeményezésére, akkor versenyhelyzet alakulna ki. Ekkor az állomások, érzékelve az ütközést, abbahagyják az adási kisérletet. Az állomásokhoz, ennek a helyzetnek a megoldására, különböző várakozás időket rendelnek. Vagyis ütközés után a rövidebb várakozási idővel rendelkező állomás, az ideje letelte után kommunikálhat, míg a másik még várakozik. Mikor a „nyerő" állomás befejezi az üzenetét, akkor kísérelheti meg a másik a kommunikációt.




*3-34. ábra: Asszimetrikus és szimmetrikus állomás elrendezések*

## HDLC (High level Data Link Control)

Bitorientált eljárás. Az adatkapcsolat szintű asszimetrikus működési módhoz fejlesztették ki, ahol egy mesterállomás

**(főállomás (primary station))** vezérli a szolgaállomásokat **(mellékállomás (secondary station)).**

Mikor ezt a protokollt számítógépek közötti információ cserére is alkalmazták, természetesnek tűnt, hogy bármelyik kezdeményezheti és meg is szüntetheti az adatkapcsolatot. Ilyen esetben pont-pont típusú, szimmetrikus elrendezésre van szükség.

Szimmetrikus kapcsolatot úgy valósíthatunk meg, hogy a vonal mindkét végére főállomást helyezünk el, amelyek a túloldalon lévő mellékállomással kommunikálnak. Az ilyen módon felépített és fizikailag nem különálló fő- és mellék-állomást tartalmazó egységet **kombinált állomásnak** hívják.

Legfontosabb előnyök, (összehasonlítva a karakter alapú eljárásokkal pl. IBM BISYNC) a következők:

- Duplex információcsere lehetőség.
- Vezérlő információk hibavédelme.
- Kötelező ciklikus hibavédelem.
- Kód és bitsorozat független átvitel.
- Több nyugtázatlan adatkeret lehet a vonalon.
- Több csomópontos időben átlapolódó kommunikáció.
- Az alkalmazott adatkeret mezői tetszőlegesen bővíthetők.

A HDLC állomások sok kerettípust adnak és vesznek, amelyek alapvetően két csoportba, a parancsok (command) és válaszok (reply) csoportjába tartozhatnak. Az üzenetek keretekből épülnek fel (frame) és felépítésük a következő:

Tétlen vonal esetén folyamatosan küldik a kerethatároló jelből álló sorozatokat. A **Címmező** többpontú vonalak esetén a pontok címeit hordozza. Pont-pont összeköttetés esetén előfordul, hogy a parancsok és a válaszok megkülönböztetésére használják.

| Kerethatároló | Címmező | Vezérlőmező | Információs mező | Keretellenőrző | Kerethatároló |
|---|---|---|---|---|---|
| 01111110 | 8 bit | 8 bit | Tetszőleges számú bit | 16 bit | 01111110 |

***3-35. ábra A HDLC keretek formátuma***

A **Vezérlőmező** sorszámokat, nyugtákat hordoz, később részletesen ismertetjük.

Az **Információs mező** hordozza az adatokat. Hossza tetszőleges, de túlzott hossz esetén a hibák valószínűsége nő.

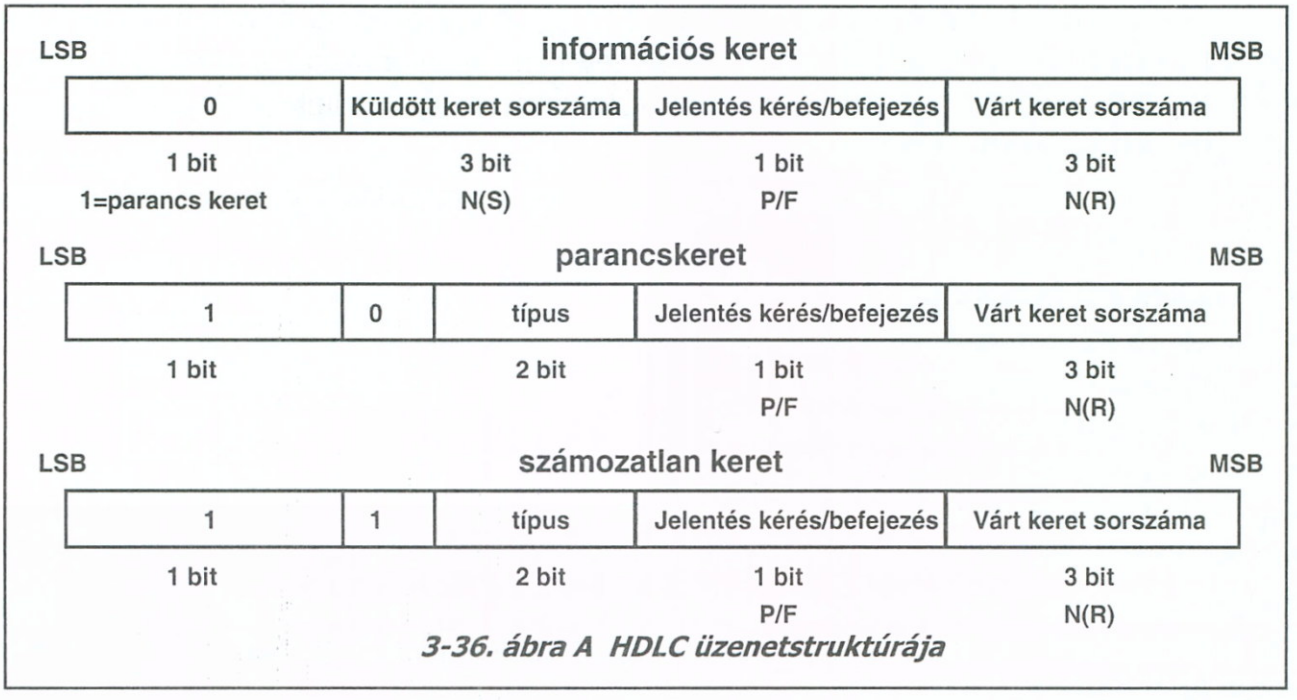
A **Keretellenőrző** mező a hibafelismerésre használható ciklikus redundancia kódot tartalmazza.

A kerettípusok felépítése a *3-36. ábrán* látható.

A kereteknek három típusa van: információs, parancs és számozatlan. 3 bites keretszámmal működő csúszóablakot használ, ami azt jelenti, hogy egyszerre maximum hét nyugtázatlan keret lehet a vonalon. Nyugtaként az első még nem vett, (a várt keret) sorszáma kerül visszaküldésre.

A P/F bitet arra használja a küldő, hogy a címzett küldjön adatot. A válaszában a P/F alacsony szintje jelzi az adatküldést, és ezt akkor állítja a küldő magasra, ha befejezte az átvitelt.

Ha az átviteli közeg jellemzői szükségessé teszik (pl. műholdas átvitel), lehetséges a keretszám hét bitre történő kiterjesztése is (128 hosszúságú csúszóablak). Ilyenkor a vezérlő mező 16 bit hosszúságú, a megnövelt ablaksorszámok miatt. Számozatlan keretek esetén ilyenkor a második nyolc bit kihasználatlan.




*3-36. ábra A HDLC üzenetstruktúrája*

**Működési módok:**

- **Normál válasz üzemmód (NRM: Normal Response Mode):** Lényegében az SDLC által definiált egyetlen üzemmódot takarja: mindig a főállomástól kapott lekérdezésre válaszolnak a mellékállomások. Ilyenkor a mellékállomás a főállomástól olyan parancsot kap, amelyben P=1. Ezután egy sorozat válaszkeretet küldhet, és az utolsó keretben F=1 jelzi a válaszüzenet végét, amivel egyben vissza is adja a vezérlést a főállomásnak. Ha a mellékállomásnak nincs elküldendő adata, egy sorszámozatlan vételkész (RR) keretet küld vissza F=1 bittel, hogy a vezérlést visszaadja a főállomásnak.
- **Aszinkron válasz üzemmód (ARM: Asynchronous Response Mode):** Ebben az üzemmódban a mellékállomás akkor küldhet, amikor akar, nem kell a főállomás felszólítására várakoznia. Jól használható pont-pont szimmetrikus, és üzenetszórásos (felfűzött) elrendezések esetén.
- **Aszinkron szimmetrikus üzemmód (ABM: Asynchronous Balanced Mode):** Ebben az üzemmódban a két állomás egyenrangúnak van deklarálva, kombinált állomások közötti üzemmódot jelenti.

Mindhárom üzemmód 128-as csúszó-ablakkal is dolgozhat (kiterjesztett üzemmód).

## 3.3.6 Üzenetszórásos átvitel: közeghozzáférési módszerek

Üzenetszórásos csatornával rendelkező alhálózatok esetében ténylegesen egy kommunikációs csatorna van, és ezen az egy csatornán osztozik az összes hálózatba kapcsolt számítógép (vagy más néven állomás).

Ehhez az egyetlen csatornához, közeghez kell minden állomásnak hozzáférni. A hozzáférés alatt itt az adást értjük, hiszen a vétel nem probléma: minden állomás veszi a többi adását. Megfelelő azonosítás után (pl. állomáscím-figyeléssel) dönt arról, hogy az üzenet neki szól-e. **A módszerek a következők feltételezésével tárgyalhatók**:

- N számú független ADÓ osztozik egy kommunikációs csatornán, amelyek véletlenszerűen kereteket generálnak, és amíg a keretet sikeresen nem küldte el, blokkolt állapotban marad.
- Ha két keretet egy időben továbbítanak, a jeleik a csatornán egyszerre jelennek meg, és az eredményjel értelmezhetetlenné válik. Ilyen eseményt ütközésnek (collision) nevezzük.
- az állomásoknak van ütközésérzékelő mechanizmusa, (ütközés: két ADÓ ad egy időben, ld. előbb)
- Folytonos idő: A keretek továbbítása bármikor elkezdődhet.
- Diszkrét idő: Az idő intervallumokra (időrés) van osztva. A keretek továbbítását mindig csak az időrés kezdetén tehetjük meg. Egy időrés vagy nem tartalmaz, vagy egy keretet tartalmaz, vagy több keretet tartalmaz (de ez már ütközés!)
- az állomások vagy képesek (carrier sense = vivőjel érzékelés) vagy nem képesek a csatorna foglaltságát figyelni.

Az átviteli közeg hozzáférésére számos eljárást használnak. A hozzáférés módja — amint azt a későbbiekben látni fogjuk — függ a hálózat topológiájától is, vagyis attól, hogy milyen módon vannak az állomások összekapcsolva.

A közeg elérési módja szerint három fő hozzáférési módszer lehetséges:

**Véletlen vezérlés:** akkor a közeget elvileg bármelyik állomás használhatja, de ebben az esetben az esetleges ütközésekre fel kell készülni.

**Osztott vezérlés:** ilyenkor, egy időpontban mindig csak egy állomásnak van joga adatátvitelre, és ez az adási jog (más néven: vezérjel) halad állomásról-állomásra. Ez a módszer az elvében biztosítja az ütközések elkerülését.

**Központosított vezérlés:** van egy kitüntetett állomás, amely vezérli a hálózatot, engedélyezi az állomásokat. A többi állomásnak figyelnie kell, hogy mikor kapnak engedélyt a közeg használatára. Ez a módszer szintén biztosítja az ütközésmentességet.

| VÉLETLEN | OSZTOTT | KÖZPONTOSÍTOTT |
|---|---|---|
| ÜTKÖZÉSFIGYELÉSES ÜTKÖZÉST JELZŐ | VEZÉRJEL TOVÁBBÍTÁSOS VEZÉRJELGYŰRŰ | LEKÉRDEZÉSES |
| RÉSELT GYŰRŰ | VEZÉRJEL TOVÁBBÍTÁSOS VEZÉRJELBUSZ | VONALKAPCSOLÁSOS |
| REGISZTER BESZÚRÁSOS GYŰRŰ | ÜTKÖZÉSFIGYELÉSES ÜTKÖZÉST ELKERÜLŐ | IDŐOSZTÁSOS TÖBBSZÖRÖS HOZZÁFÉRÉSŰ |

*3-37. ábra Közeghozzáférési módszerek csoportosítása*

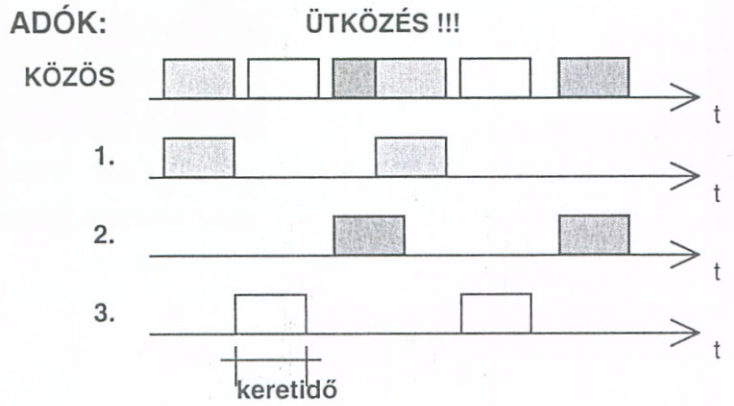
Nézzük részletesebben:

### *Véletlen átvitel-vezérlés: ALOHA*

Mindegyik állomás a csatornán véletlenszerűen kezd adni. A módszer nevében szereplő véletlen kifejezés a döntő jelentőségű: mivel nincs külön eljárás az adási jog megadására, ezért elvileg nem lehet felső időkorlátot adni az üzenettovábbítás időbeli bekövetkezésére.

Az első ilyen típusú átvitelt a Hawai szigeteken próbálták ki: a különböző szigeteken lévő, egy központi helyre adatcsomagokat küldő rádióadók véletlenszerű időpontokban adtak: ha két adási időtartam átfedte egymást, akkor egyik adás sem volt eredményes, mivel az üzenetek összekeveredtek, és ütközés következett be.

Egy keret akkor nem szenved ütközést, ha elküldésének kezdetétől a végéig más állomás nem próbál keretet küldeni. Mivel a keretek küldése véletlenszerű, ezért valószínűségszámítási módszerekkel meghatározható az ütközés (illetve annak elkerülésének) a valószínűsége. Ha egy keret továbbítási ideje alatt legalább egy új keret létrejön, akkor biztosan ütközés következik be. Ezért a keretidőnként keletkező átlagos N keretszám-nak 1-nél kisebbnek kell lennie: $0<N<1$. Kis terhelés esetén $N\sim0$ és kevés ütközés történik, míg nagy terhelés $N\sim1$ esetén sok ütközés.




*3-38. ábra: Aloha típusú véletlen hozzáférés*

Az ütközések száma nyilvánvalóan függ az adók számától, a forgalomtól, és a csomagok időtartamától. Az ütközések száma csökkenthető, ha bármely adó nem akármikor, hanem csak adott időpontokban: időrésben adhat (réselt ALOHA). Réselt Aloha protokollnál van egy időrés kezdetét jelző adó, és csak ezt érzékelve, ebben az időpontban lehet egy keret küldését elkezdeni. Mi van, ha egy időrésben két keret lesz adásra kész? Okoz ez problémát? (igen... miért?)

### Ütközést jelző vivőérzékeléses többszörös hozzáférés (CSMA/CD)

Az előző Aloha módszer-hez képest a változás az, hogy az állomás a küldés előtt először belehallgat a közegbe.

A módszer angol elnevezése: Carrier Sense Multiple Access with Collision Detection =CSMA/CD. Ennél a módszernél, mielőtt egy állomás adatokat küldene, először „belehallgat" a csatornába, hogy megtudja, hogy van-e éppen olyan állomás, amelyik **használja** a csatornát. Ha a csatorna „csendes", azaz egyik állomás sem használja, a „hallgatódzó" állomás elküldi az üzenetét. A vivőérzékelés (carrier sense) jelenti azt, hogy az állomás, adás előtt belehallgat a csatornába. Az állomás által küldött üzenet, csatornán keresztül minden állomáshoz eljut, és véve az üzenetet a bennfoglalt cím alapján eldöntheti hogy az neki szólt (és ilyenkor feldolgozza), vagy pedig nem (és akkor eldobja).

Ennél a módszernél természetesen előfordulhat olyan eset, amikor egyszerre két, vagy több állomás akarja használni a közeget. Ezért adás közben — mivel közben a

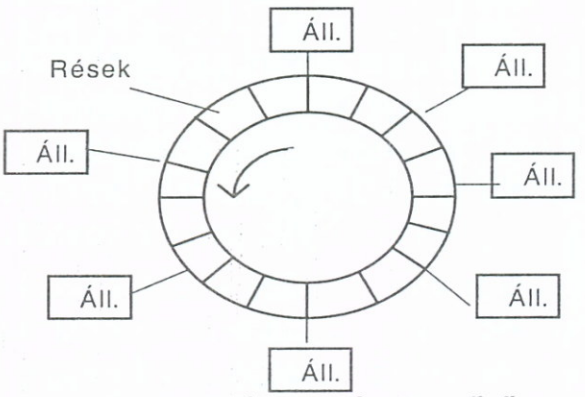
csatornán lévő saját üzenetét is veszi — el tudja dönteni, hogy a küldött, és a vett üzenetfolyam egyforma-e. Ha ezek különbözőek, akkor azt jelenti, hogy valaki más is „beszél", megzavarva a küldött üzenetet, ami ezért hibás lesz. Ezt, ahogy már leírtuk, ütközésnek hívják, és ilyenkor az állomás megszakítja az üzenetküldést.

Az ilyen protokoll szerint működő állomások a következő három állapot valamelyikében lehetnek: **versengés, átvitel,** illetve **tétlen állapot.**

Az ütközés miatt kudarcot vallott állomások mindegyike az újabb adási kísérlet előtt bizonyos, véletlenszerűen megválasztott ideig várakozik. Ezek az idők a véletlenszerűség miatt eltérők, és a versengő állomások következő hozzáférési kísérlete során egy, a legrövidebb várakozási idejű fog adni, mivel a többiek a várakozási idejük leteltével adás előtt a csatornába belehallgatva azt már foglaltnak fogják érzékelni. Az adást a perzisztenciával lehet jellemezni. (perzisztencia = kitartás, sürgetés)

- 1-perzisztens: ha a csatorna felszabadul, azonnal ad
- 0-perzisztens: ha a csatorna felszabadul, véletlen idő múlva ad
- p-perzisztens: ha a csatorna felszabadul, *p* valószínűséggel ad, *q=1-p* valószínűséggel megvárja a következő időrés elejét.

Végiggondolva az eljárást, nyilvánvaló, hogy gyér forgalom esetén a közeghozzáférés nagyon gyors, mivel kevés állomás kíván a csatornán adni. Nagy hálózati forgalom esetén az átvitel lelassul, mivel a nagy csatornaterhelés miatt gyakoriak lesznek az ütközések. A széles körben elterjedt Ethernet hálózat ezt a módszert használja, és részletesebben a következőkben írunk róla.



*3-39. ábra Réselt gyűrű*

Még egy módon lehet befolyásolni az ütközéseket. A fenti esetben, amikor a csatorna szabad lesz, azonnal megkiséreli az adást az ADÓ. Ezt a protokollt 1-perzisztensnek nevezik, mert szabad csatorna esetén azonnal adni kezd. Kevesebb ütközés lehetséges, ha a keretet adó nem ilyen "mohó". Ez azt jelenti, hogy időréses csatornahasználat esetén, a csatorna szabad voltának érzékelésekor *p* valószínűséggel adni kezd, különben *1-p* valószínűséggel megvárja a következő időrés kezdetét. Ez a p-perzisztens CSMA protokoll. Következő részben, mint az IEEE 802.4 szabvány, részletesebben írunk róla.

### Réselt gyűrű (slotted ring)

A gyűrűn felfűzött állomások réseknek elnevezett rögzített hosszúságú kereteket adnak körbe. Minden résben van egy jelző (marker) amelyik jelzi a rés foglaltságát.

Mivel a rés hossza állandó, az állomásnak az üzeneteit akkora darabokra kell vágnia, hogy azok elférjenek a résben (az állomáscímekkel, és egyéb kiegészítő információval együtt.) Ha egy állomáshoz egy nem foglalt (üres) rés érkezik, akkor az elhelyezi benne a saját adatait, és továbbadja az immár foglalt keretet.

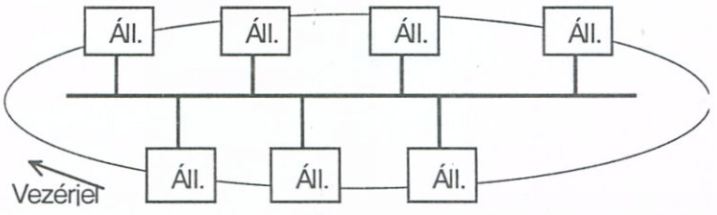
Természetesen az adatot elhelyező állomásnak a feladata a visszaérkezett keret kiürítése, azaz a foglaltságának a megszüntetése. Ha átviteli, vagy egyéb hibák miatt (pl. az állomás elromlik), ez nem történik meg, akkor ez a rés foglaltan tovább kering a

gyűrűben. Ezért kijelölnek egy állomást, amely felügyelői feladatot is ellát: ez figyeli, hogy van-e olyan rés, amely a gyűrűben nem jut alaphelyzetbe, és ha ilyen van, egy idő múlva eltávolítja a gyűrűből.

### *Osztott átvitelvezérlés*

Ennél a közeghozzáférési módszernél minden állomás, amennyiben egy rövid üzenet, a vezérjel (adási jog) birtokosa a közeget adás céljára kizárólagosan használja. Ez a szerep állomásról állomásra vándorol, úgy, a vezérjelet a birtokos egy idő után kötelezően továbbadja a következő állomásnak. Mivel ez a módszer egy önmagában záródó gyűrű kialakítást igényel, ezért a gyakorlatban két módon oldották ezt meg.

### Vezérjeles sín (Token bus — Vezérjel busz)



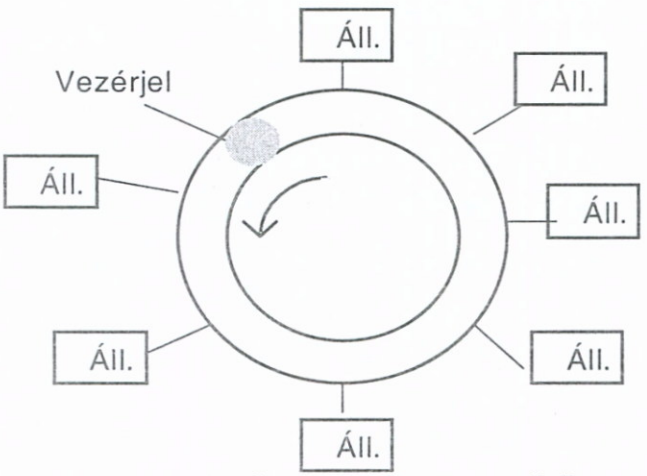

*3-40. ábra Vezérjeles sín*

Az üzenetszórásos eljárást használó, felfűzött busz (sín) topológiájú a hálózat, de az egyes állomások logikai kapcsolata gyűrűt formáz.

A vezérjel busz a közös átviteli közeget úgy vezérli, hogy az állomásról állomásra történő **vezérjel (ún. token)** továbbítása egy logikai gyűrűt képez. Ez azt jelenti, hogy minden ismeri az előtte lévő és utána következő állomásának a címét. A következőnek úgy küldi el a vezérjelet, hogy mivel saját maga az adott időpillanatban az adási jog tulajdonosa (nála van a vezérjel), egyszerűen ezt a közegbe kiküldi, a logikai szomszédjának a címével, amely véve a neki szóló üzenetet ő lesz a vezérjel tulajdonosa, amíg hasonló módon tovább nem adja.

Amikor egy állomás vette a vezérjelet, lehetőséget kap arra, hogy adatblokkokat továbbítson a számára biztosított maximális időn belül a kisajátított közeget felhasználva. Ha nincs adandó adatblokkja, akkor a tokent azonnal továbbadja a logikai gyűrűben sorba következő állomásnak. Következő részben, mint az IEEE 802.4 szabvány, részletesebben írunk róla.

### Vezérjeles gyűrű (Token Ring)




*3-41. ábra Vezérjeles gyűrű*

Fizikailag gyűrű topológiájú hálózatok esetén — mivel lényegében páronként pont-pont összeköttetés valósul meg — a leggyakrabban használt hozzáférési módszer a vezérjel továbbításos eljárás, amelyben egy ún. vezérjel (token) halad körben a gyűrű mentén állomásról állomásra. Minden állomásban egy léptetőregiszterbe lépnek be a bitek, és a másik végén pedig kilépnek. A léptetőregiszter aktuális tartalmát pedig az állomás figyeli.

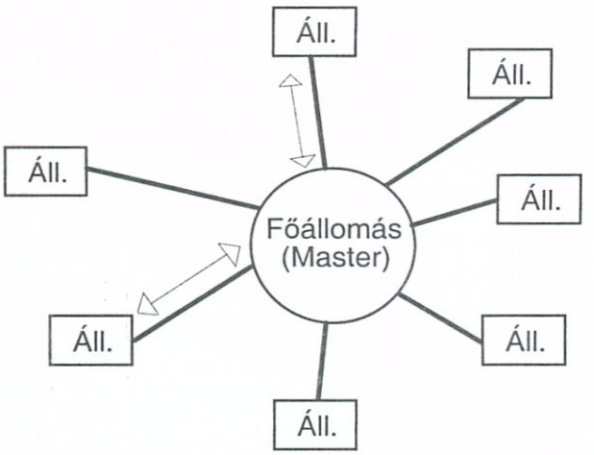
A vezérjel lényegében egy rövid üzenet, ami a gyűrű foglaltságát jelzi. Ha szabadot jelez, akkor a tokent fogadó állomás számára ez azt

jelenti, hogy üzenetet küldhet. A tokent foglaltra állítja, és üzenettel együtt küldi tovább, vagy más megoldásként kivonja a gyűrűből. Az üzenet a gyűrűn halad körben állomásról állomásra.

A bitenként továbblépkedő üzenetet az állomások veszik, megvizsgálják, hogy nekik szól-e, majd továbbadják. Amikor a gyűrűben az üzenet visszaér az elküldő állomáshoz, akkor kivonja az üzenetét a gyűrűből, a tokent szabadra állítja, és továbbküldi az immár szabad jelzésű vezérjelet a fizikailag mellette elhelyzkedő állomásnak.

Elképzelhető, hogy valamilyen hiba miatt, egy üzenet nem kerül kivonásra. A leblokkolás megakadályozására kijelölhetnek egy aktív felügyelő állomást, amely az ilyen „árva" üzeneteket figyeli és kivonja ezeket a hálózatból. A többi állomás ún. passzív felügyelő, és az aktív felügyelő meghibásodásakor egy másik veszi át a szerepét. A módszer előnye a garantált, adott időn belüli üzenetadás.

Az állomások között prioritás is kialakítható, azaz a nagyobb prioritású állomások az alacsonyabb szintű állomások előtt kaphatnak lehetőséget adataik továbbítására. Következő részben, mint az IEEE 802.5 szabvány, részletesebben írunk róla.




*3-42. ábra: Lekérdezéses (polling) eljárás*

### *Központosított átvitelvezérlés*

Ilyen típusú vezérléseknél mindig van egy kitüntetett (központi) egység, amelynek feladata az egyes állomások hálózathoz való hozzáférésének a vezérlése.

### Lekérdezéses (polling) eljárás

Ennél az eljárásnál a főállomás (master), és a többi mellékállomás (slave vagy secondary) alkotja a hálózatot. A főállomás sorban egymás után szólítja fel a mellékállomásokat üzenetek küldésére. Ha a megszólított állomásnak van üzenete, akkor elküldi a főállomáshoz, amely az üzenetben lévő cím alapján meghatározza, hogy melyik másik mellékállomásnak kell ezt elküldenie. A működési elv miatt elsősorban csillag kialakítású hálózatoknál használják.

Azaz a mellékállomások a főállomás közvetítésével tudnak egymással kommunikálni. Ha a megszólított mellékállomásnak nincs üzennivalója, akkor negatív választ küld a lekérdezésre. Ezután a főállomás egy előre meghatározott prioritási sorrend alapján periodikusan folytatja a többi mellékállomás lekérdezését.

Az eljárás előnyös, mert a rugalmas vezérlés lehetőséget biztosít arra, hogy egy mellékállomás több üzenetet is küldjön egymás után, és a lekérdezési sorrendben többször szerepeltetve egyes mellékállomásokat, azok magasabb prioritást kapnak.

Az eljárás sebezhető pontja a mellékállomásoknál bonyolultabb főállomás meghibásodási lehetősége, hiszen ilyenkor az egész hálózat megbénul. Mivel minden két mellékállomás közötti üzenetváltás kétszer megy át a hálózaton, ez növeli az átviteli időt.

### Vonalkapcsolásos eljárás

Az előbb ismertetett eljárásnál a főállomás fő funkciója a postás szerep volt. Mivel az elsődleges feladat a mellékállomások egymással való kommunikációja, ez más módon is megoldható. Ha lehetséges, akkor a két mellékállomást egy vonalon relék vagy elektronikus kapcsolók segítségével összekötjük, és a két állomás üzeneteket válthat egymással a kialakított áramköri úton keresztül. Mikor az üzenetváltást befejezik, a kapcsolat megszűnik, és a kapcsoló felszabadul. Mivel a központban több kapcsoló helyezkedik el, ezért egyszerre több vonalkapcsolat is működhet.

### Időosztásos többszörös hozzáférésű eljárás (TDMA)

Az angol rövidítés a **Time Division Multiple Access** kifejezés első betűiből alkotott betűszó. Elsődlegesen busz felépítésű hálózatoknál alkalmazzák. Ennél az eljárásnál minden a buszhoz kapcsolódó mellékállomás, egy adott időszeletben adhat. Ha nincs üzenete, akkor a szelet kihasználatlan marad

### Ütközést elkerülő, vivőérzékeléses többszörös hozzáférés (CSMA/CA)

A módszer angol elnevezése: Carrier Sense Multiple Access with Collision Avoidance) =CSMA/CA. A véletlen közeghozzáférésekkel foglakozó részben már a módszer alapgondolatát megismertük. Itt is minden állomás, az adást figyelve „belehallgat" a csatornába, és ha nem észlel adást, akkor egy előre meghatározott ideig várakozik. Amennyiben ennek leteltével sem használja más az átviteli közeget, akkor elküld egy RTS (Request To Send) keretet a címzettnek, amellyel lefoglalja a csatornát és megadja a teljes átvitelhez szükséges időt. A címzett egy CTS (Clear To Send) kerettel nyugtázza a fogalalást, amiben szintén szerepel az átvitelhez szükséges idő. A feladó ezután elküldi az adatkeretet, amelyet a címzett nyugtáz. Az ütközést elkerülő RTS/CTS mechanizmus révén kicsi az esély hogy az adatkeretek megsérüljenek.

## *3.3.7 Közeghozzáférés gyakorlatban: IEEE 802-es szabványok*

A hivatalos szabvány kidolgozására az IEEE egy albizottságát kérték fel, amelynek tagjai között a gyártásautomatizálásban érdekelt képviselők is helyet foglaltak. Ők úgy gondolták, hogy a gyártásban részt vevő robotok belső hálózatokon keresztül lesznek összekötve, és pontosan rögzített időzítésekkel dolgoznak, ami a hálózati kapcsolat időbeliségét is meghatározza. Emiatt a hálózat adatátviteli idejének felülről korlátosnak kell lennie, azaz a legrosszabb esetben is, adott időn belül meg kell történnie az információátvitelnek. Sajnos az Ethernet nem rendelkezik ezzel a tulajdonsággal. Úgy is fogalmazhatnánk, hogy nem képes a valós idejű (real time) követelményeknek eleget tenni. Ezért ilyen esetre, két már akkor is létező szabványos megoldást, a vezérjeles sínt és az IBM által kifejlesztett vezérjeles gyűrűt választották. Ilyen módon három szabványt fogadtak el, amelyekre együttesen az IEEE 802-es szabvány részeként hivatkoznak. A szabványokat részekre osztották:

A **802.1**-es szabvány a szabványhalmaz alapjait írja le, és az interfész primitíveket definiálja.

A **802.2**-es az adatkapcsolati réteg felső részét, az ún. LLC (Logical Link Control — logikai kapcsolatvezérlés) alréteget definiálja. Sokáig vita volt arról, hogy az eltérő

közeghozzáférési módszerek miatt hová tartozzon a közeg-hozzáférés: a fizikai réteghez, vagy az adatkapcsolati réteghez. A vita lezárásaként az adatkapcsolati réteget osztották két részre: a közeg-hozzáférési alrétegre (MAC — Media Access Control — közegelérés vezérlés) és az LLC-re.

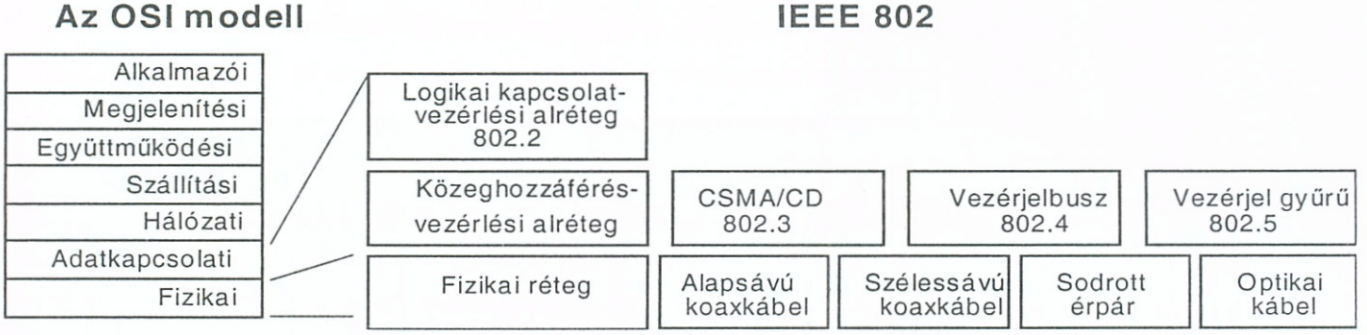
A **802.3**-as szabvány a CSMA/CD (Ethernet) leírása.

A **802.4**-es szabvány a vezérjeles sín, és a

A **802.5**-as szabvány a vezérjeles gyűrű leírása.

Az IEEE szabványok a .5-el nem érnek véget. A továbbiak, csupán felsorolva: 802.6 — Városi hálózatok (MAN), 802.7 — Szélessávú átvitel, 802.8 — Optikai kábelek, 802.9 — Integrált hang és adat lokális hálózatok, 802.10 — LAN-ok biztonsági kérdései. **802.11 – Rádiós hálózatok**

A *3-43. ábra* jól mutatja az IEEE 802 szabvány és az OSI modell kapcsolatát. Ejtsünk most néhány szót a rétegekről.




***3-43. ábra Az OSI modell és az IEEE 802***

### *A fizikai réteg (PHY)*

Az IEEE 802-es három olyan vezetékes fizikai közeget szabványosított, amelyeket az architektúra fizikai rétegében használhatnak: a sodrott érpárt, a koaxiális kábelt (alap és szélessávút) és az optikai kábelt. A fizikai szabvány így megadja a kábel és az átvitel típusára, a kódolás módjára és az adat sebességére vonatkozó előírásokat.

A fizikai réteg felelős a két berendezés közötti fizikai összeköttetés létesítéséért és megszüntetéséért, valamint az átviteli közegen keresztül a bitek átviteléért. Meghatározza még átvitelre alkalmas formában az adatkódolást és dekódolást, vezérli az eszközök időzítését, hogy azokat az adott és vett jelek szinkronizálják.

### *Közeghozzáférés-vezérlési (MAC) alréteg*

Itt a MAC betűszó **M**edia **A**ccess **C**ontrol – Közeghozzáférés vezérlés kifejezés gyakran használt rövidítése. A lokális hálózatban lévő eszközök mindegyike a közös fizikai csatornán a közeghozzáférésért verseng. Mivel a LAN kialakításokban ezen a szinten számos hozzáférés vezérlési módszert használnak ütközésest és ütközés mentest egyaránt, a bizottság ezek közül — ahogy ezt már az előbbiekben is leírtuk — a CSMA/CD, a vezérjel-busz és a vezérjel-gyűrű hozzáférés módszereket választotta ki szabványosításra. A közeghozzáférés vezérlési alréteg szabványa négy funkciót határoz meg:

- **Közeghozzáférés-vezérlés:** A hálózati állomások szabályokat ill. eljárásokat használnak, hogy vezéreljék a fizikai csatorna megosztását.

- **Keretezés:** Kezdeti és záró információ jelzés hozzáadására van szükség ahhoz, hogy azonosítani lehessen az üzenetek elejét és végét, hogy az adó és a vevő szinkronizálódjon, és felismerjék a hibákat.
- **Címzés:** A hálózat címzést használ, hogy azonosítani tudja az üzenet adásában és vételében résztvevő eszközöket.
- **Hibafelismerés:** Célja a helyes üzenetadás és vétel ellenőrzése.

### *Logikai kapcsolatvezérlési (LLC) alréteg*

LLC (= Logical Link Control) funkciói:

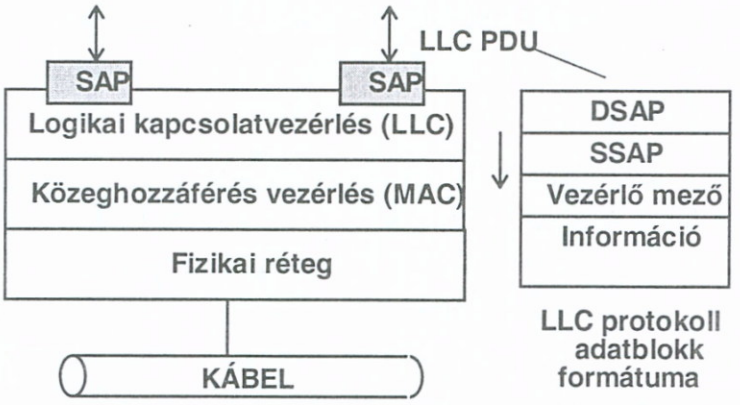
- a MAC rétegtől független egységes adatkapcsolati protokoll nyújtása a hálózati rétegnek,
- adatblokk csere,
- ehhez logikai kapcsolat létesítése

Az adatkapcsolati réteg logikai kapcsolatvezérlés szintjén az IEEE 802.2-es szabványt hozta létre: egységes felületet mutat a felsőbb rétegek felé. Ez a alréteg szervezi az adatfolyamot, parancsokat értelmez, válaszokat generál, a hibákat ellenőrzi, és helyreállítási funkciókat hajt végre. A logikai kapcsolatvezérlés feletti réteg tulajdonképpen a hálózati réteg.

Az LLC/MAC felületek közötti szolgáltatási előírások azokat a szolgáltatásokat rögzíti, amelyeket az LLC, és az alatta levő közeghozzáférés-vezérlési (MAC) alréteg felületei között definiálnak.

A logikai kapcsolatvezérlés felel teljes mértékben az állomások közötti adatblokkok cseréjéért. A lokális hálózatban az adatblokkok cseréjéhez a hálózat állomásai között létesítendő logikai kapcsolatra van szükség.

Ahhoz, hogy meg tudják különböztetni az ugyanazon állomás által létesített különböző cseretípusokat, bevezették a szolgáltatás hozzáférési pont (Service Access Point — SAP) fogalmát, amelyet a hálózati állomásban, az egyetlen adatcserében résztvevő egyedi elem azonosítására használnak. (lásd 1. Fejezet)




*3-44. ábra Szolgálat-hozzáférési pontok*

A *3-44. ábrán* látható két szolgáltatás-hozzáférési ponttal rendelkező eszköz, különböző, más eszközökkel történő adatblokk cserére használhatja a SAP-jait. Az eszköz természetesen számos SAP-ot használhat.

Azt a szolgáltatás hozzáférési pontot, amelyik adatblokkot küld, forrás szolgáltatás-hozzáférési pontnak (Source SAP — SSAP), azt pedig, amelyik adatblokkot vesz rendeltetési szolgáltatás hozzáférési pontnak (Destination SAP — DSAP) nevezik.

Azt az adatblokkot, amely a forrásállomás logikai kapcsolatvezérlési alrétegéből eljut a célállomás logikai kapcsolatvezérlési alrétegéig, logikai kapcsolatvezérlési protokoll adatblokknak (LLC Protocol Data Unit — LLC PDU) nevezzük Az adás folyamán a forrásállomás logikai kapcsolatvezérlő alrétege átadja az adatblokkot a közeghozzáférésvezérlő alrétegnek. Az átadott adatblokk felépítése is a *3-44. ábrán* látható.

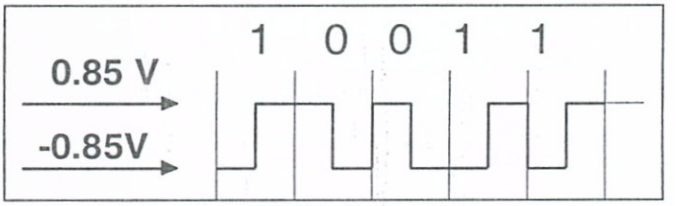
### *IEEE 802.3 szabvány: Ethernet*

Az Ethernet közeghozzáférésének alapgondolatát már bemutattuk. Mielőtt egy állomás adni akar, belehallgat a csatornába. Ha a kábel foglalt, akkor az állomás addig vár, amíg az üressé nem válik, máskülönben azonnal adni kezd. Ha egy üres kábelen két vagy több állomás egyszerre kezd el adni, ütközés következik be. Minden ütközést szenvedett keretű állomásnak be kell fejeznie adását, ezután véletlenszerű ideig várnia kell, majd az egész eljárást meg kell ismételnie.

Az Ethernet hálózatok átviteli sebessége a jeleleg még legtöbbet használt rendszerekben 10 Mbit/s. Ma már 100 Mbit/s sebességü Fast Ethernet vagy 100baseT hálózatok illetve 1 Gbit/s sebességü Gigabit Ethernet hálózatok is vannak.

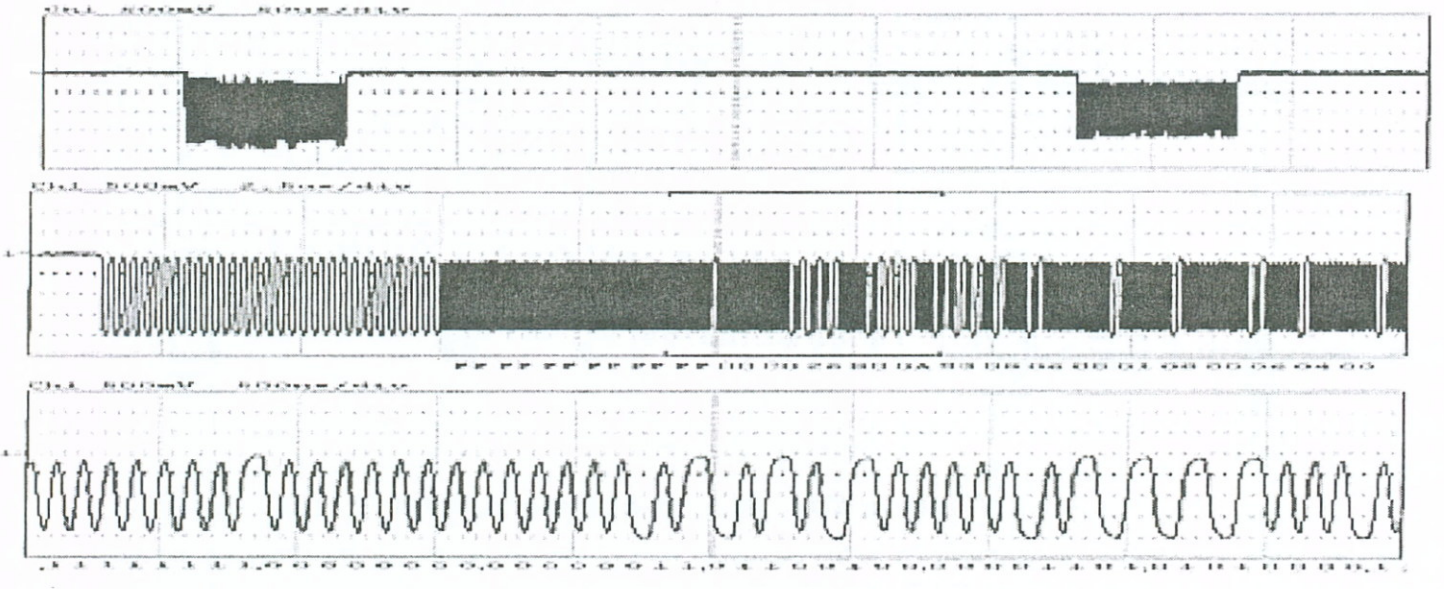
Ez persze nem jelenti azt, hogy egy Ethernet hálózatnak minden körülmények között ez a maximális átviteli sebessége, hiszen egy ilyen hálózat a lehetséges terhelésének csak mintegy 60 %-án üzemeltethető ésszerűen.

Az összes 802.3 implementáció, beleértve az Ethernetet is, manchester kódolást használ, amelyet az *3-45 ábrán* láthatunk. A bitek közepén lévő jelváltás iránya jelenti a 0 vagy 1 információt, és ezen átmenet segítségével a küldő szinkronizálhatja a VEVŐ-t.




***3-45. ábra Jelszintek az Ethernet hálózatban. Manchester kódolás***

A számítógépben lévő interfészkártya a csatlakozó kábeltípustól függő BNC, AUI, telefoncsatlakozó aljzattal van szerelve és tartalmaz egy olyan vezérlő integrált áramkört, amely kereteket vesz, illetve kereteket küld a hálózatra. A vezérlő felelős a kimenő keretek adatokból



***3-46. ábra Ethernet keret a valóságban***

való összeállításáért, a kimenő keretek ellenőrzőösszegének kiszámításáért és a bejövő keretek esetén az összeg ellenőrzéséért.

A bitek közepén levő átmenetek segítségével a küldő szinkronba hozhatja a vevőt.

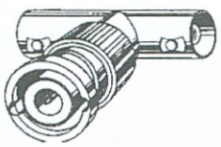
Bármelyik időpontban a kábel a következő három állapot egyikében van:

- 0-ás bit átvitele (ala-csonyból magasba való átmenet),
- 1-es bit átvitele (magas-ból alacsonyba való át-menet), vagy
- tétlen (0 V).

Például a névleges 10 MBit/sec Ethernet hálózat optimális sebessége mintegy 4.5 Mbit/s, ami megfelele kb. 500 Kbájt/sec adatátviteli sebességnek.

Az Ethernet hálózatokban a 2. fejezetben már ismertetett, többféle kábeltípus használható. A koax kábelezés ma már elavultnak tekinthető, de még sok helyen megtalálható.

## Koax kábelezés

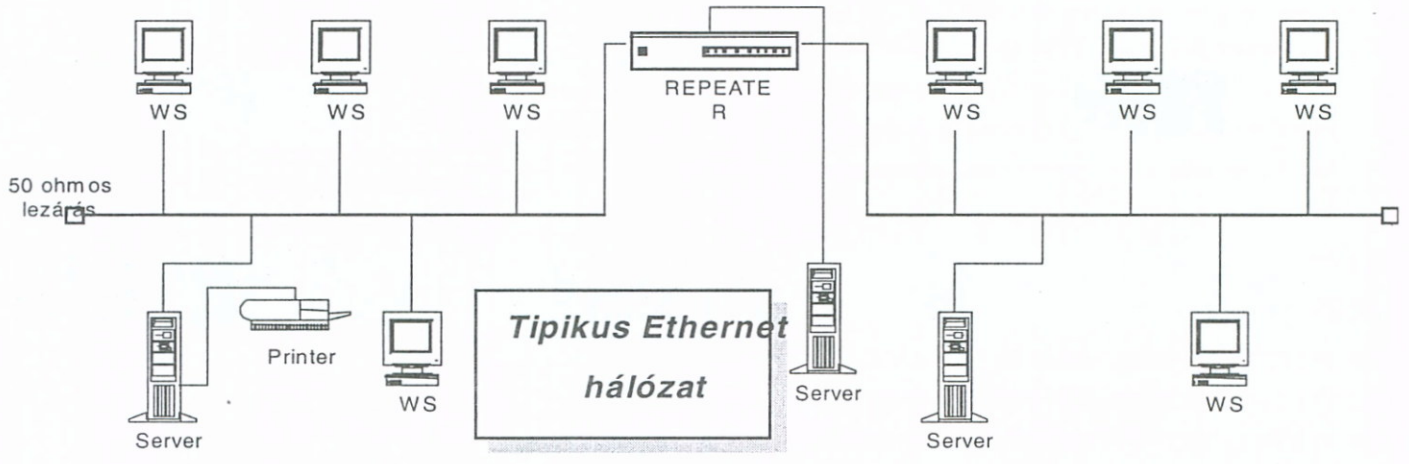


**Vékony koax kábelezés esetén** a jelek visszaverődésének megakadályozására a végpontokat a kábel hullámellenállásával megegyező értékű 50 Ω-os ellenállással kell lezárni. Mivel a számítógépek sorosan fel vannak fűzve a kábelre, a csatlakoztatást oly módon lehet megvalósítani, hogy a koaxiális kábelt egyszerűen kettévágják, a két végére ún. BNC csatlakozót szerelnek, és egy ún. T csatolót illesztenek be, és ez csatlakozik a számítógép hálózati kártyájához.

Az előre kialakított hálózatoknál egy új csatlakozás létesítése egyszerűbb. A felszerelt fali csatlakozásról kell eltávolítani az ún. rövidzáró hurkot és a helyére kötni két darab előre szerelt koaxiális kábelt mindkét végén BNC csatlakozóval, valamint egy T csatlakozás segítségével a számítógéphez illeszteni. Mindkét módszer hátránya, hogy a számítógép-hálózat működésének néhány percre való felfüggesztését kívánja.

**A vastag Ethernet koax kábel** többnyire sárga színű (bár ezt semmilyen szabvány nem rögzíti), ezért gyakran yellow cable -nek is nevezik. A nagyfrekvenciás jelillesztés miatt a kábel borításán azonos távolságokra felfestett jelzések (gyűrűk) jelzik azokat a pontokat, ahol a kábelhez hozzá lehet csatlakozni. Ezt a kábelezési módszert a magasabb költségek, és a különleges szereléstechnikája miatt ritkábban használják.

Ilyen kábelek csak meghatározott íveken hajlíthatók) csak olyan esetekben használják, ahol az erősebb külső zavarok miatt szükséges az erősebb árnyékolás (pl.: ipari felhasználás), illetve nagyobb az áthidalandó távolság.




*3-47. ábra: Koax kábeles Ethernet hálózat*

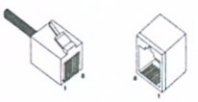
A vastag koax kábeleknél a számítógép csatlakoztatás módja az ún. **vámpír csatlakozó** használata. A vámpír csatlakozókat csak a kábel jelölt, meghatározott pontjain lehet elhelyezni. Ilyenkor a kábelre a vámpír csatlakozóhoz egy adó-vevőt (transceiver vagy **MAU — Media Attachment Unit**) is illeszteni kell,

amihez csatlakoztatott kábel köti össze a adó vevőt a számítógépben lévő illesztő kártyával. Az adó-vevőkábel (**AUI=Attachment Unit Interface**) legfeljebb 50 méter hosszú lehet, és öt különállóan árnyékolt sodrott érpárt tartalmaz.

Ethernet esetén vastag koax kábelhosszúsága max. 500 m, a vékony koaxé 185 m lehet. A hálózat által átfogott távolság növelése érdekében az egyes kábeleket ismétlők (**repeater**) segítségével össze lehet kötni. Az ismétlő egy fizikai rétegbeli eszköz, amely mindkét irányból veszi, felerősíti és továbbítja a jeleket. A hálózat szemszögéből az ismétlőkkel összekötött kábelszegmensek egyetlen kábelnek tekinthetők (eltekintve az ismétlő okozta plusz késleltetéstől).

Egy rendszer több szegmenset és több ismétlőt tartalmazhat, de nem lehet két olyan adó-vevő, amely 2,5 km-nél távolabbra helyezkedik el egymástól, ill. nem lehet olyan adó-vevő közötti út, amely négynél több ismétlőn halad keresztül.

## Csavart érpár



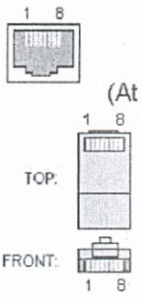
A jelenleg használt, a koax kábelezést leváltó **csavart érpár** alkalmazásakor minden ilyen módon bekötött számítógép lényegében pont-pont kapcsolatot valósít meg az elosztó eszközzel. A kialakítás csillag toplógiájú hálózat, amely küllős kerékhez hasonlít, ahol a küllők végén, a kerék kerületén vannak az egyedi végpontok (munkaállomások, fájlkiszolgálók). A munkaállomás a kábelhez a hálózati adapter kártyán található 8 pólusú RJ45-ös típusú csatlakozón keresztül kapcsolódik, és az összes kábel a középen lévő dobozba — hasonlóan, mint a küllők a kerékagyba — kapcsolódik. Ezt az egységet hívják **HUB**-nak. (Ejtsd: hab. Angolul a hub egyik jelentése: kerékagy.)

A hub fizikailag egy doboz, rajta **port**-oknak nevezett, telefoncsatlakozókhoz nagyon hasonlító 8 érintkezős (neve: RJ45) csatlakozó aljzatokkal. Minden port egy munkaállomástól, fájlkiszolgálótól, vagy egyéb hálózati egységtől érkező kábelt fogad. A hub-ok számos formában és méretben kaphatók: 4 portostól akár 124 portosig. Ha a hub nagy (16 vagy még több port-ot tartalmaz) gyakran rackba (tartóba) szerelhető kialakítású.

Minden hub egymással **uplinkelhető** (összeköthető). Egy uplink port-tal rendelkező hub közvetlenül, egyenes bekötésű kábellel összeköthető egy másik hub-bal. Ha egyik hub-nak sincs uplink port-ja, akkor az összekötés egy szabványos port-on keresztül, keresztbe bekötött kábellel oldható meg.

### Ethernet 10/100Base-T

Same connector and pinout for both 10Base-T and 100Base-TX.




RJ45 FEMALE CONNECTOR at the network interface cards/hubs.
RJ45 MALE CONNECTOR at the cables.

| Pin | Name | Description |
|---|---|---|
| 1 | TX+ | Tranceive Data+ |
| 2 | TX- | Tranceive Data- |
| 3 | RX+ | Receive Data+ |
| 4 | n/c | Not connected |
| 5 | n/c | Not connected |
| 6 | RX- | Receive Data- |
| 7 | n/c | Not connected |
| 8 | n/c | Not connected |

*Note: TX & RX are swapped on Hub's.*

***3-48. ábra Ethernet csatlakozó- és kábelbekötés***

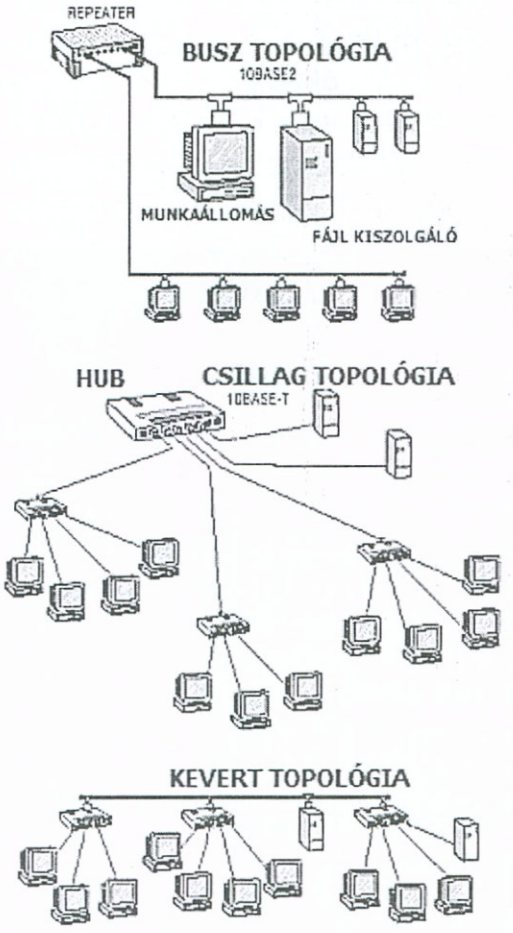
A hub-ok teljesítménye csökken, ha egyre több felhasználót kötünk rá. Ha a hub 100 Mbps sebességű hálózaton működik, és 5 felhasználót kapcsolunk rá, akkor minden felhasználónak csak 20 Mbps sebességű lesz a rendelkezésre álló sávszélessége.

A hub-oknak jelenleg öt típusa létezik:

1. **A szabványos 10BaseT vagy 100BaseTX hub-ok.** A 10BaseT hub 10 Mbps hálózati sebességen

működik, míg a 100BaseTX hub 100 Mbps sebességet használ. Egy szabványos 10 Mbps hub nem köthető össze egy 100 Mbps hub-bal, csak akkor, ha switch-et, vagy automatikus sebességérzékelős hub-ot telepítünk közéjük. Ne felejtsük el, ezek a hub-ok félduplexek, ami azt jelenti, hogy váltakozva, mindig csak az egyik irányban áramlanak rajtuk keresztül az adatok. Ha a teljes duplex kommunikációra, azaz mindkét irányban történő egyidejű adatátvitelre van szükségünk, akkor switch-et válasszunk

2. **Automatikus sebességérzékelésű, vagy két sebességen működő (dual speed) hub-ok.** Ezek a hub-ok egyidejűleg, port-onként vagy 10 Mbps vagy 100 Mbps sebességgel képesek működni, amiért ezek az eszközök nagyon rugalmasan használhatók. Beépítésük minden típusú--akár kicsi, akár nagy, új vagy régi--hálózatba javasolt, különösen akkor, ha a hálózatban vannak 10 Mbps illetve 100 Mbps sebességű részek is.
3. **Sztekkelhető (összekapcsolható) hub-ok.** Mivel egymáshoz speciális módon kapcsolódnak, a hálózat felől egyetlen hub-nak látszanak. Ez a szabványos hub-okkal szemben különösen akkor nagyon előnyös, ha bővíteni akarjuk a hálózatot.

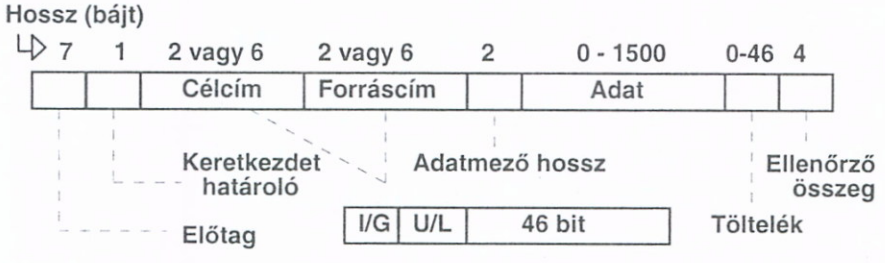



*3-49. ábra Ethernet kialakítások*

4. **SNMP- (távolról felügyelhető (menedzselhető)) hub-ok.** Ezek a hub-ok támogatják a hálózati menedzselhetőséget biztosító szabványos Simple Network Management Protocol (SNMP) protokollt (ld. 10. fejezet) Ez a protokoll lehetővé teszi a rendszergazdák számára, hogy a hub-ot a hálózat bármelyik helyéről távolról konfigurálják (paramétereit, jellemzőit beállítsák), illetve adatait lekérdezzék. Ezek az adatok: a hub hálózati forgalmi adatai, a naplózott hibák, stb. (lásd később)
5. **Kombinált hub-ok.** Ezek a típusok az előbbiekben említett hub-ok két vagy több jellemzőit egyesítik. Például egy 10/100-as automatikus átkapcsolós hub, még sztekkelhető is lehet, és még ismerheti az SNMP protokollt is.

Hálózatépítésnél különféle épületkábelezési megoldás lehetséges. Lehet, hogy csak egyetlen kábel kígyózik át az épület szobáin úgy, hogy az állomások a hozzájuk legközelebb eső ponton csatlakoznak rá.

Lehetséges egy, az alaptól a tetőig futó gerinckábel alkalmazása, amelyre az egyes emeleteken ismétlők segítségével vízszintes futó kábelek csatlakoznak. Egyes megvalósításainál, a függőleges gerincvezeték optikai szál, míg a vízszintesek, csavart érpárú kábelek. Ilyenkor az elosztó eszközök a csavart érpáras kábelek fogadása mellett, optikai kábelek fogadását (csatlakoztatását) is lehetővé teszik.




*3-50. ábra: 802.3 keretformátum*

## A 802.3 MAC-protokollja

A 802.3 keretszerkezetét a *3-50.ábrán* mutatjuk be. Minden keret egy 7-bájtos **előtag**gal (preamble) kezdődik, amely 10101010 mintájú. E minta Manchester-kódolása, amely egy 10

MHz-es, 5,6 usec időtartamú négyszögjel, lehetőséget biztosít a vevő órájának; hogy az adó órájához szinkronizálódjon. Ezután következik a **Keretkezdet** (start of frame) bájt, amely a keret kezdetét jelöli ki az 10101011 mintával.

A keret két címet tartalmaz, egy **Célcím**-et és egy **Forráscím**-et. A szabvány 2- és 6-bájt-os címeket is megenged, de a 10 Mbit/s-os alapsávú szabvány számára kijelölt paraméterek csak 6-bájtos címek használatát engedélyezik. A célcím legfelső helyiértékű bitje (I/G) közönséges címek esetén 0, csoportcímek esetén 1 értékű. A csoportcímek teszik lehetővé több állomás egyetlen címmel való megcímzését. Amikor egy keret csoportcímet tartalmaz célcímként, akkor a keretet a csoport minden tagja veszi. Az állomások egy meghatározott csoportjának való keretküldést többes-küldésnek (multicast) nevezik. A célcímben csupa 1-est tartalmazó keretet az összes állomás veszi. Ez az üzenetszórás (broadcast).

A címzésnél érdekes a legmagasabb helyiértékű bit melletti 46. bit (U/L) használata. Ez a bit a helyi és a globális címeket különbözteti meg. A helyi címeket a hálózatmenedzserek jelölik ki és a helyi hálózaton kívül nincs jelentőségük. A globális címeket ellenben az IEEE jelöli ki azért, hogy a világon ne fordulhasson elő két azonos globális cím. Mivel 48 - 2 = 46 bit áll rendelkezésre, ezért megközelítőleg $7*10^{13}$ globális cím létezik.

Az alapgondolat az, hogy 2+46 bitet használva már a világ bármely két állomása megcímezheti egymást. Ezt a 6*8 bitet megegyezés szerint hexadecimális alakban, bájtonként kettőspontokkal elválasztva adják meg, például:

`3A:12:17:0:56:34`

A hat bájtos fizikai (MAC) cím felső három bájtját a XEROX cég osztja ki az Ethernet kártya- és áramkörgyártók között, míg az alsó három bájtot az adott gyártó adja, az egyediség biztosítása miatt.

**Adatmező hossz** (length field) az adatmezőben található adatbájtok számát adja meg. A minimum 0, a maximum 1500 bájt. **Adatmező:** Az érvényes keretek és a szemét megkülönböztetése érdekében a 802.3 szabvány szerint egy érvényes keretnek legalább 64 bájt hosszúnak kell lennie, a célcímtől az ellenőrzőösszeget is beleértve. Ha tehát egy keret adatrésze 46 bájtnál rövidebb, akkor kitöltő (**Töltelék**) mezőt kell használni a minimális kerethossz eléréséhez. A minimális kerethosszúság alkalmazásának másik oka az, hogy egy rövid keret küldését egy állomás még azelőtt befejezhetné, mielőtt a keret első bitje elérné a kábel legtávolabbi végét, ahol is az egy másik kerettel ütközhet.

Az utolsó mező az **ellenőrzőösszeg** (checksum). Az ellenőrzőösszeg algoritmusa a ciklikus redundancia ellenőrzésen (CRC) alapul.

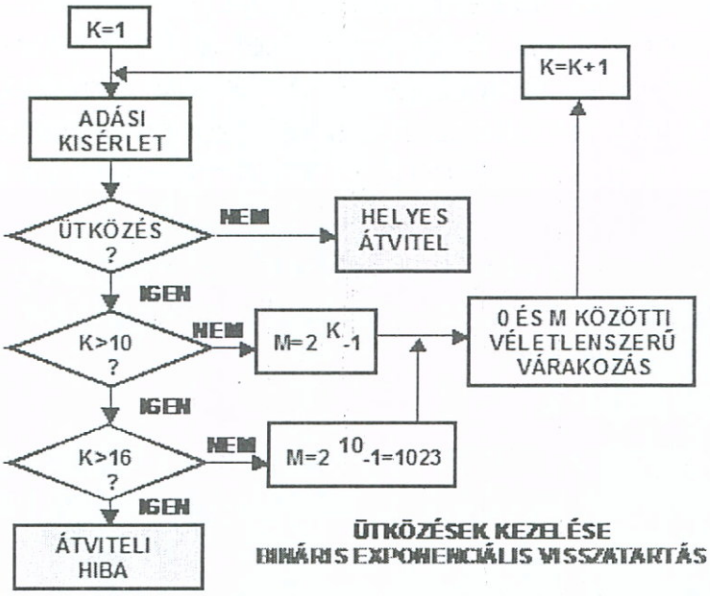
Ahogy már említettük, ütközés bekövetkeztekor minden ütközést észlelő állomás abbahagyja adását, a többi állomás figyelmeztetésére szándékosan zajos jelet küld egy darabig, majd véletlenszerű ideig vár, és csak ezután próbálja ismét megkezdeni az adást.

### Hogyan lehet a véletlenszerűséget biztosítani?

Az ütközés után az időt diszkrét időintervallumokra osztják. Az első ütközés után minden állomás az újabb próbálkozás előtt 0 vagy 1 időintervallumot várakozik. Ha két

állomás ütközik, és mindkettő ugyanazt a véletlen számot kapja, akkor ismét ütköznek. A második ütközés után már a 0, l, 2 vagy 3 számok közül választanak véletlenszerűen, és annak megfelelő ideig várakoznak. Ha a harmadik ütközés is bekövetkezik (amely 0,25 valószínűséggel fordulhat elő), akkor az állomások a 0 és 7 közötti intervallumból választanak véletlenszerűen egy számot.

Általánosan fogalmazva: a **k.** ütközés után az állomásoknak a 0 és $2^k$-1 közötti intervallumból kell egy számot választaniuk, és ennek megfelelő időt kell várakozniuk. Ha azonban elérik a 10. ütközést, akkor a véletlenszám generálás felső határa az 1023-as értéken állandósul: 16 bekövetkezett ütközés után a vezérlő abbahagyja a próbálkozást, és hibajelzést ad a számítógépnek, és a felsőbb rétegek feladata a további hibajavítás. Ezt az algoritmust **bináris exponenciális visszatartásnak (binary exponential backoff)** nevezik.




***3-51. ábra Bináris exponenciáls visszatartás blokkvázlata***

**Összefoglalva**: kevés ütköző állomás esetén viszonylag kis késleltetés következik csak be, ugyanakkor nagyszámú állomás esetén az ütközés még belátható időn belül feloldódik.

A próbálkozások számával exponenciálisan növekvő várakozási idő miatt dinamikusan lehet az adni kívánó állomások számához igazodni. Ha a véletlenszám generálás felső határa minden ütközéskor 1023 lenne, akkor két állomás újbóli ütközésének valószínűsége valóban elhanyagolhatóvá válna, de a várakozási idő átlagos értéke túl nagy lenne, és a hálózat nagyon lelassulna. Ha viszont az állomások csak a 0 és 1 közül választanának, akkor 100 egyszerre adni akaró állomás keretei addig ütköznének, amíg végre 99 állomás a 0-t, míg a maradék egy az 1-est (vagy fordítva) választaná. Ez megint igen nagy lassulást okozna.

Ahogy az eddigiekből kiderült, a CSMA/CD nem biztosít nyugtázást. Mivel az ütközés hiánya nem garantálja azt, hogy a keretek nem sérülnek meg, ezért a megbízható átvitel érdekében a célállomásnak ellenőriznie kell az ellenőrzőösszeget, és ha hibátlan, akkor erről a tényről egy nyugtakeret küldésével értesítenie kellene a keret küldőjét. Általában ez a nyugtázás egy másik keretet igényelne, amelynek elküldése érdekében, akárcsak egy adatkeret esetén, meg kell szereznie a csatornahozzáférési jogot. Ez megoldható lenne úgy, hogy a sikeres adásokat követő közegért történő versengés során a célállomásnak prioritást biztosítunk. Mivel a keretek hibás vételét a felsőbb rétegek is tudják jelezni, ezért ez általában itt valósul meg.

Kétségkívül jelenleg ma a legnépszerűbb hálózat az Ethernet, amelyet rugalmassága és egyszerűsége és olcsósága biztosít.

Az UTP kábelezés alkalmazásakor a telefonhálózattal együtt szerelhető. Megjelent a

100 MHz-es sebességű változata (Fast Ethernet), az 1GHz-es sebességű változata (Gigabit Ethernet) amelyeknél szintén az UTP kábelezést is használhatja.

A koaxiális kábelt alkalmazó megoldások száma csökkenőben van például azért, mert egy sorosan felfűzött kábelrendszerben sokkal nehezebb a hibát behatárolni, mint a csillag kialakítású UTP-s rendszer esetén.

**Struktúrált kábelezés**

Struktúrált kábelezés esetén egy olyan hálózatot alakítunk ki, hogy az kielégítse az adat, hang és egyéb alkalmazásokkal szemben támasztott igényeket és az a struktúrált kábelrendszer magában foglalja a kábeleket, a rendezőket, a csatlakozókat is.

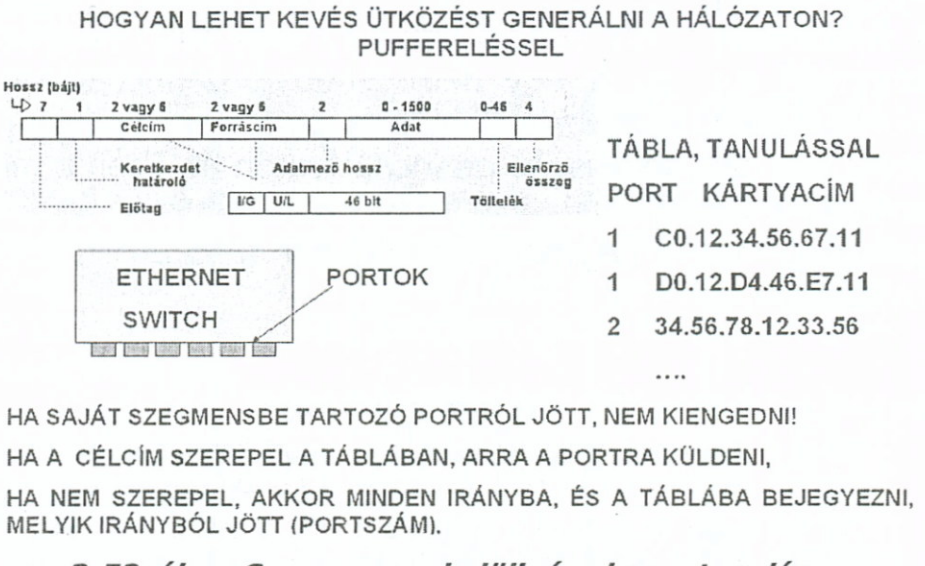
Alapelv az egységesség: minden végpont azonos tulajdonságokkal rendelkezik és funkciója szabadon változtatható. A rendezőtől minden végpont felé (pl. a fali csatlakozókig) azonos érszámú kábel fut, csillag kialakítású hálózatot alkotva, amelynek középpontjában a rendező van.

A jelenlegi Ethernet kábelezési technika az üvegszálas (egyre ritkábban koax) gerincvezetéket részesíti előnyben, amelyhez jelismétlőkön (UTP multiport repeater = HUB ) keresztül csavart érpárokkal csatlakoznak a számítógépek, csillag topológiát formálva. A különféle fizikai kábelezés megfelelő csatlakozáspárokat tartalmazó egységek használatát követeli meg. Ezek lehetnek az üvegszálnál alkalmazott csatlakozók (ST, FC/PC, SMA, SC, FDDI), vékony koax BNC csatlakozója, vastag koax-nál a AUI csatlakozót használnak, míg csavart érpárnál az RJ-45-ös telefoncsatlakozót.

**A struktúrált kábelezés felépítése:** Minden munkahelyhez külön fali csatlakozók tartoznak, amelyek egyaránt alkalmassak a telefon és a számítógép csatlakoztatására is. A fali csatlakozókat a berendezésekkel adapter kábelek kötik össze. A fali csatlakozókat sodrott érpáras kábelen keresztül egy közös helyiségben lévő elosztódobozban lévő csatlakozóaljzatokba vezetik, amelynek **patch panel** a neve.

Az aktív elemek (telefonközpont, hálózati kiszolgáló számítógépek, stb.) szintén ebben a közös helyiségben kerülnek elhelyezésre, amelyeket szintén a patch panelen lévő csatlakozó aljzatokba kötnek. Az itt lévő csatlakozóaljzatok egymással való ún. patch kábellel történő összekötésével a végpontok és az aktív elemek összeköthetők. Így kialakul egy csillag topológiájú végponti hálózat.

Bármilyen is legyen a közeg, a szakadt kábelek, rossz megcsapolások, laza csatlakozók komoly adatátviteli problémákat okoznak. Lelassul a hálózat, sok „hálózati hiba" üzenet jelenik meg a rendszerben.




*3-52. ábra Szegmensen belüli címek megtanulása*

## Intelligens hálózati elosztó eszközök: bridge (híd), switch (kapcsoló)

Az Ethernet rugalmasságát is növelik a már bemutatott hubok. Ezek az egyik portjukon vett keretet bit-ről bitre átmásolják a másik portjukra, mintegy meghosszabbítva ezzel az elektromos jellemzők miatt rövidre korlátozott szegmenst. Több porttal rendelkező (multiport) repeater-ek használata esetén minden portra átmásoljuk a vett keretet.

Természetesen az ütközéseket továbbra is időben észlelni kell, a jelterjedési idő nem lehet több 64 byte elküldésének idejénél, ezért bármely két állomás között maximum 4 repeater helyezkedhet el, ám a kapott hálózat meglehetősen szövevényes lehet. A hálózat mérete ezzel a módszerrel 2.5 km-re növelhető.

A hatékonyságot fokozhatjuk **bridge**-k (hidak) közbeiktatásával. Ezek csupán az egyik szegmensből a másikba irányuló kereteket engedik át, szétválasztva ezzel a szegmensek forgalmát. Így jelenősen csökken az ütközések száma és nő a rendelkezésre álló sávszélesség.

Ezen a hálózat szegmentálásával segíthetünk. Hidak közbeiktatásával a szegmensek belső forgalma nem terheli a többi szegmenst. Minél szegmentáltabb a hálózat, annál kevésbé osztott a közeg, és annál jobb a hálózat hatásfoka. Ennek az irányába történik fejlődés, ha a mikroszegmensekről beszélünk. Ebben az esetben, nagyon kevés munkaállomás csatlakozik a szegmensre. Ekkor az állomás akkor adhat, amikor csak akar. Ezzel azonban a keretek ütközésének problémáját mindössze áttettük a kábelről a hídba, ott ugyanis ütköz(het)nek a keretek.

Erre kínálnak megoldást a **LAN switch**-ek. Ezek funkciójukban tulajdonképpen multiport bridge-k, ám képesek nem blokkoló módon továbbítani a kereteket

Azaz, ha egy switch-nek hét portja van, akkor egyidőben képes keretet továbbítani például az 5. portról a 6.-ra, es a 6.-ról a 7.-re. Csak akkor áll elő ütközés, ha egyszerre többen kívánnak ugyanarra a portra adni, ekkor a switch az egyik keretet puffereli, és a másik után adja le a portra. Igy minden port számára (pl. Ethernet esetén) dedikált 10 Mbit/s-os sávszélesség áll rendelkezésre.

A switch-nek szinte mindig van egy vagy több nagysebességű portja is (FDDI, Fast Ethernet) melyen át egy nem Ethernet gerinchálózatra kapcsolható. Igy az egyes munkacsoportok egymáson belül a switch-en keresztül kommunikálnak, a külvilággal pedig egy nagysebességű gerinchálózaton át. Ezen a módon akkor sem ütközhetnek a keretek, ha több, egy switch-en lévő állomás kíván a gerinchálózaton keresztül forgalmazni, mert a nagysebességű gerinc több Ethernet port forgalmát képes egyszerre továbbítani.

Egy LAN switch alapvetően kétféle elven működhet:

- **Store & forward** működés esetén a kapott keretet letároljuk, ellenőrizzük, hogy ép, majd a célállomás címéből meghatározzuk, hogy melyik porton kell továbbítani, és arra elküldjük.
- **Cut through** állapotban a switch azonnal, a célállomás címének beérkezése után elkezdi a keret továbbítását. Így csökkent a késleltetés, hiszen a címmező a keret elején található. Ha a kimeneti port foglalt, akkor természetesen a keretet puffereljük és a port felszabadulása esetén adjuk le. Sajnos így nem képes ellenôrizni, hogy a keret ép-e.

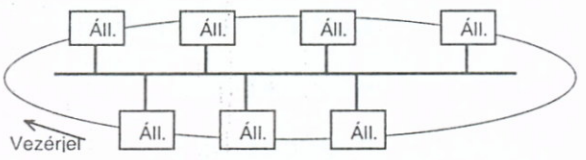
Éppen ezért a switch adaptív működési módjában a hibás keretek számától függően hol store & forward, hol cut through üzemmódban működik. Ha a hibák száma egy szint fölé emelkedik, az előbbire, aztán ha tartósan, egy szint alá csökken, az utóbbira vált

A LAN-ok szervezésében is mutatkozik fejlődés. Minthogy egy porton gyakran csak egy állomas van, egy sor biztonsági funkció implementálható. Megadhatjuk például, hogy ki-kinek küldhet keretet, így az állomásokat, bár egy LAN-on vannak, mégis leválaszthatjuk egymásról, és elkülönített virtuális LAN-okba ún. VLAN-okba szervezhetjük őket. Ez nem csupán biztonságot, de a hatékonyságot is növeli. A VLAN-ok ezen kívül könnyen konfigurálhatóak és rugalmasak, kevesebb terhet rónak a hálózat adminisztrációjára, így üzemeltetésük — ami a költségek tetemes részét képezi — kevesebbe kerül.

### Fast Ethernet

Számos alkalmazás számára kevésnek bizonyult a hagyományos Ethernet 10 Mbit/s-os sebessége. Kisérletek indultak a változatlan elvek melletti nagyobb sebesség elérésére. Ezek egyike a Fast Ethernet, ami mindent érintetlenül hagy, csak a szegmensek mérete csökken a tizedére és az átviteli sebesség nő a tízszeresére. Minthogy némely UTP kábelek képesek 100 Mbit/s átvitelére, ha a korábbi hálózatunk szegmensei elég rövidek voltak, a Fast Ethernetre való áttérés csak az aktív elemeket érinti. A kábelezés és a használt szoftver maradhat a régi. A megengedett távolság kb. 100 méter, szűkös ugyan, de használható, ám ekkor már csak switch-ek közbeiktatásával növelhető a méret. Ez azonban nem jelent komoly megszorítást, mert a HUB-ok egyre inkább kiszorulnak a switch-ek csökkenő ára és komoly előnyei miatt.

### *IEEE 802.4 szabvány: vezérjeles sín (vezérjel-busz)*




[1] Mint azt már az előbbiekben leírtuk, a gyártásautomatizálás sokszor igényel valós idejű, vagy felülről korlátos válaszidejű számítógépes hálózatot. Sajnos ennek az IEEE 802.3 szabvány nem tesz eleget. Egy másik probléma az, hogy ott a kereteknek nincs prioritása, azaz a fontos keretek nem kerülhetnek előnybe a kevésbé fontosakkal szemben. A gyűrű felépítés, ahol az állomások egymásnak küldik sorba körbe a kereteket, ilyen szempontból jó megoldás:

*Ha k állomás alkotja a gyűrűt, és T ideig tart egy keret átvitele, akkor bármelyik állomás k*T időn belül képes kommunikálni (felső korlát).*

Sajnos a gyűrű, mint fizikai topológia kevéssé illeszkedik a futószalagok egyenes vonalú kialakításához. Ezért egy olyan kialakítást szabványosítottak, amely fizikailag lineáris buszkialakítása miatt üzenetszórásos módot használ (azaz a gyűrűtől eltérően, nem pont-pont kapcsolati módon dolgozik), logikailag azonban gyűrű felépítésű. Elnevezése: **vezérjel busz, vagy vezérjeles sín.**

A logikai gyűrűszervezés azt jelenti, hogy minden állomás ismeri a közvetlen bal, és jobb oldali állomásának a címét. Ez a szomszédság nem a fizikai elhelyezkedés, hanem a gyűrűben elfoglalt logikai elhelyezkedés szerinti. Amikor a gyűrűt elindítják, elsőként a legmagasabb sorszámú állomás küldhet üzenetet. A küldés után átadja a küldés jogát a közvetlen szomszédjának, amit egy speciális keret a vezérjel (token) képvisel. Ez a vezérjel a logikai gyűrű mentén jár körbe, állomásról állomásra. **Küldé-**

**si joga csak a tokent birtokló állomásnak van, ezért ütközés nem jöhet létre.** A gyűrűhöz csatlakozó állomások minden üzenetet vesznek, de csak a részükre szólót veszik figyelembe.

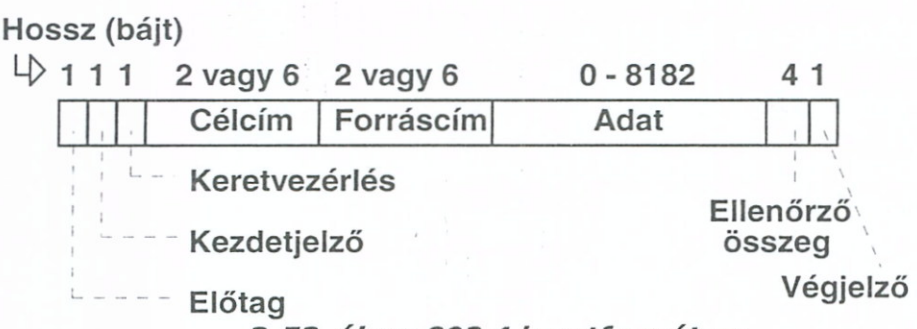
Mivel a gyűrű működése az állomások összehangolt működését igényli, ezért meg kell válaszolni néhány, az eddigiekben mellőzött kérdést:

- Hogyan lép egy állomás a gyűrűbe?
- Mi van akkor, ha egy állomást, akinél a vezérjel van, kikapcsolunk? Mi lesz a vezérjellel?
- Hogyan épül ki a gyűrű?

A kérdéseket protokoll részeként megvalósított gyűrűkarbantartás válaszolja meg. A megoldás lényege, hogy a küldött adatkeretek mellett, az állomások **vezérlőkeret**ekkel kommunikálnak egymással. Fontos megjegyezni, hogy a sínhez való fizikai csatlakozás nem jelent azonnal gyűrűhöz való csatlakozást is: az állomások gyűrűbe illesztése, illetve eltávolítása a vezérjelbusz MAC-protokolljában leírt vezérlőkeretek segítségével valósul meg.

A fizikai réteg a kábeltelevíziózásban használt 75 ohmos szélessávú koaxiális kábel. Mind az egykábeles mind a kétkábeles (irányonként egy kábel) rendszer használható.

### A sín MAC protokollja




*3-53. ábra: 802.4 keretformátum*

A gyűrű üzembe helyezésekor az állomások a gyűrűbe cím szerint csökkenő sorrendbe kerülhetnek be. A vezérjel küldés is mindig a nagyobbtól a kisebb sorszámú állomás felé irányul. Amikor egy állomás megkapja a vezérjelet, azt adott ideig birtokolhatja, és ez alatt az idő alatt — ha a keretei rövidek —, akár több keretet is elküldhet.

Ha a vezérjelet birtokló állomásnak nincs elküldendő kerete, akkor a tokent azonnal továbbküldi. A vezérjeles sín keretformátuma a *3-53. ábrán* látható.

Az **előtag**, a vevő órájának szinkronizálását segíti elő. A **kezdetjelző** és a **végjelző** mező a keret határait jelzik. Mindkét mező analóg kódolású szimbólumokat tartalmaz, amelyek a digitális 0 és 1 kódolásától jelentősen különböznek. A speciális határoló jelek alkalmazása miatt nincs szükség adathossz mezőre.

A **keretvezérlésmező** az adat- és a vezérlőkereteket különbözteti meg egymástól, és adatkeretek esetén a keretek prioritását hordozza. Tartalmazhat olyan jelzést is, amely a célállomást a keret hibátlan vagy hibás vételének nyugtázására kötelezi.

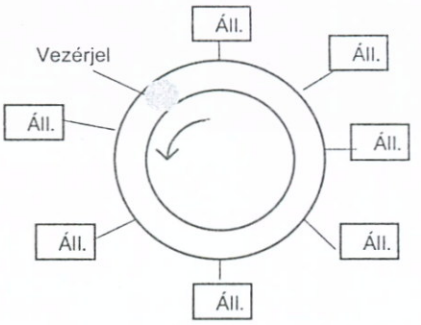
Vezérlőkeretek esetén a keretvezérlés mező a keret típusát jelöli. A megengedett típusok halmaza a vezérjel átadási és a különböző gyűrűkarbantartási keretekből áll. Ez utóbbiak között vannak az állomásokat a gyűrűbe be- illetve kiléptető kerettípusok. Megjegyezzük, hogy a 802.3 szabványban vezérlő keretek nincsenek.

A **célcím** és a **forráscím** mező ugyanolyan, mint a 802.3-ban. Akárcsak a 802.3-ban, egy adott hálózatban vagy csak 2 bájtos, vagy csak 6 bájtos címeket használhatnak az állomások Az egyedi és csoportcímek, valamint a lokális és globális címek kijelölésére ugyanazok vonatkoznak, mint 802.3-asnál.

Az **adatmező** hossza 8182 bájt 2 bájtos címzés, illetve 8174 bájt, 6 bájtos címzés esetén. Ez több mint ötszöröse a legnagyobb 802.3-beli keretnek. (Persze ott azért is választottak rövid kereteket, hogy egy állomás ne tarthassa fel túl hosszú ideig a többi állomást).

A vezérjeles sínen az időzítésekkel lehet korlátozni a hosszú keretek küldését, egyébként viszont nagyon kényelmes hosszú kereteket küldeni akkor, ha követelmény a valós időben történő feladatvégrehajtás. Az átviteli hibák kiszűrésére az **ellenőrző-összegmező** szolgál.

### *IEEE 802.5 szabvány: vezérjeles gyűrű*




Bevezetésként meg kell jegyeznünk azt a tényt, hogy a gyűrű nem igazán alkalmas üzenetszórásos átvitelre, hiszen tulajdonképpen kör alakba rendezett, két pont közötti kapcsolatok halmaza. A gyűrű kiszámítható felső időkorlátos csatornahozzáférést is biztosít. A létező többféle gyűrű kialakítások közül a 802.5 által szabványosítottat vezérjeles gyűrűnek (token ring) nevezik. [1] Emiatt például 1 Mbit/s-os gyűrű, amelynek kerülete 2000 m, csak 10 bitet tartalmazhat egyszerre.

Lényegében minden állomás a digitális technikában megismert shiftregisztert tartalmaz, és a bitek ezeken a gyűrűbe kapcsolt shiftregiszetereken keresztül tolódnak állomásról állomásra.

A gyűrűben zajló átvitel tervezésénél és elemzésénél alapvető kérdés egy bit "fizikai hossza". Ha egy gyűrű K Mbit/s-os adatátviteli sebességgel rendelkezik, akkor 1/K μsec-onként kerül ki egy bit az átviteli közegre. A tipikus 200 m/μs-os jelterjedési sebességgel számolva ez azt jelenti, hogy egy bit megközelítőleg 200/K métert foglal el a gyűrűn, ami meghatározza, egy adott hosszúságú gyűrűben egyszerre tartozkodó bitek számát.

A gyűrűinterfészeknek két üzemmódjuk van: vételi és adási. Minden gyűrűinterfészhez érkező bit az állomás egy ideiglenes regiszterébe (pufferébe) kerül, — ahonnan az adott állomás ismét a gyűrűbe küldi ki. Vétel esetén a pufferben levő bitet a gyűrűbe való kiírás előtt az állomás megvizsgálja, majd továbbadja.

Ha nem az eredetit küldi tovább, akkor adásról beszélünk. A bitek interfészeknél való pufferelése, másolása minden egyes állomásnál 1-bites késleltetést eredményez.

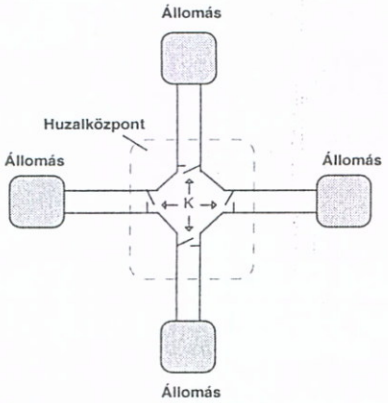
Mivel csak egyetlen vezérjel van, ezért csak egyetlen állomás adhat egyszerre, így tehát a csatorna hozzáférés ugyanúgy ütközésmentesen valósul meg, mint a vezérjeles sín esetén.

**Ha az állomások tétlenek, a vezérjeles gyűrűben, körbejár egy speciális bitminta, az ún. vezérjel (token).** Amikor egy állomás keretet akar küldeni, még a küldés előtt meg kell szereznie a vezérjelet, és el is kell távolítania a gyűrűből.

A vezérjeles gyűrű tervezésének további gondja az, hogy magának a gyűrűnek is elegendő késleltetéssel kell rendelkeznie ahhoz, hogy tétlen állomások esetén is képes legyen a teljes vezérjel befogadására és keringtetésére. A késleltetés két komponensből áll: az állomás okozta 1-bites késleltetésből és a jelterjedési késleltetésből. A tervezőknek majdnem minden gyűrűben számolniuk kell az állomásoknak különböző időkben, különösen éjszakára való kikapcsolásával, és az ebből adódó késleltetések csökkenésével. Ha az állomások gyűrűillesztői a gyűrűtől kapják áramellátásukat, akkor az állomások leállításának nincs ilyen hatása. Ha azonban az interfészek kívülről kapják az áramot, akkor a gyűrű folytonosságának fenntartása miatt úgy kell azokat megtervezni, hogy kikapcsoláskor a bemenetük a kimenetükhöz kapcsolódjon. Ez nyilvánvalóan megszünteti az 1-bites késleltetést

**A gyűrűben körbeterjedő biteket a küldő állomások távolítják el a gyűrűből.** Miután egy állomás az utolsó keretének utolsó bitjét is elküldte, a vezérjelet a gyűrűbe vissza kell helyezni.

Nagy terhelés esetén a hálózat hatékonysága a 100%-ot is elérheti. A keretek nyugtázása nagyon egyszerűen megoldható. A keretformátumnak egyetlen 1-bites mezőt kell tartalmaznia, amely kezdetben nulla. Amikor a célállomás megkapja a keretet, ezt a mezőt 1-be állítja. Mivel a keretet a küldő vonja ki, ezért könnyen tudja ezt a bitet, a nyugtát ellenőrizni.



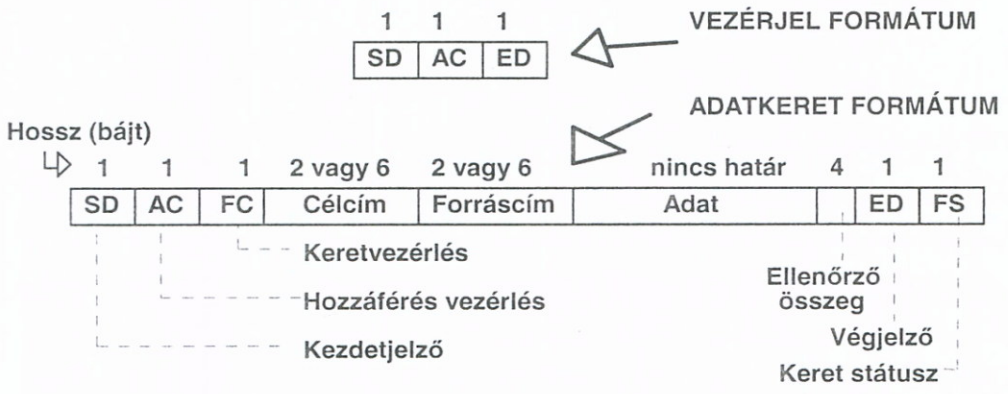

*3-54. ábra: Huzalközpont*

Amikor a forgalom kicsi, akkor a vezérjel a működési idő legnagyobb részében a gyűrűben körbe-körbe fut. Alkalomszerűen egy-egy állomás kivonja a gyűrűből, kereteit elküldi, majd ismét visszahelyezi a gyűrűbe.

Ha azonban a forgalom olyan nagy, hogy az egyes állomásoknál sorok keletkeznek, akkor, ahogy egy állomás befejezi adását, és a vezérjelet visszahelyezi a gyűrűbe. A következő állomás, figyelve azt, azonnal lecsap rá, és kivonja a gyűrűből. Ily módon az adási engedély, szép egyenletesen, körbeforog a gyűrűben.

A 802.5 szabvány szerinti gyűrű a fizikai rétegben a 1, 4, vagy 16 Mbit/s-os sebességre alkalmas árnyékolt sodrott érpárt használ.

Sajnos a gyűrűhálózatokban a kábel megszakadása esetén az egész gyűrű működése megszűnik. A probléma megoldása: a huzalközpont (wire center), amely a *3-54. ábrán* látható. Minden állomás egy bejövő és egy elmenő vezetékkel kapcsolódik hozzá.




*3-55. ábra 802.5 keretformátum*

A huzalközponton belül egy állomás be-

és kimenő vezetékét rövidre záró ún. terelő relék (az ábrán K-val jelölve) vannak, amelyeket az állomások látnak el árammal. Ha a gyűrű megszakad, vagy egy állomás meghibásodik, akkor a tápáram hibája miatt a relé elenged, így az állomás kikerül a logikai gyűrűből.

A reléket szoftver is működtetheti, így lehetőség nyílik olyan diagnosztikai programok írására, amelyekkel az állomások egyenkénti kiiktatása révén hibás állomásokat, vagy gyűrűszegmenseket fel lehet fedezni.

### Vezérjeles gyűrű MAC protokollja

A MAC alréteg alapműködése nagyon egyszerű. Amikor nincs forgalom, akkor a gyűrűn egy 3-bájtos vezérjel körbe-körbe kering addig, amíg valamelyik állomás meg nem szerzi a második bájtja egy adott, 0 értékű bitjének 1-be állításával.

Ezáltal az első két bájt keretkezdet szekvenciává alakul át. Az állomás ezután egy normál adatkeret további részeit kezdi el küldeni.

Rendes körülmények között a keret első bitje a gyűrűn körbeérve még azelőtt visszatér küldőjéhez, hogy az a teljes keretet el tudta volna küldeni. Csak egy nagyon hosszú gyűrű képes egy teljes keretet felvenni. Következésképpen az adóállomásnak már küldés közben el kell kezdeni a gyűrű "lecsapolását", azaz az útjukat befejező bitek kivonását a gyűrűből.

Egy állomás a vezérjelet legfeljebb az ún. vezérjel tartási ideig (token-holding time) birtokolhatja, amelynek alapértéke 10 ms. Ha az első keret elküldése után még elegendő idő marad, az állomás további kereteket is elküldhet. Ha az összes keret elküldése befejeződött, vagy a vezérjel tartási idő lejárt akkor az állomásnak vissza kell állítania a 3-bájtos vezérjelet, és vissza kell helyeznie a gyűrűre.

### Vezérjeles gyűrű karbantartása

A vezérjeles sín protokolljában a gyűrűkarbantartás teljesen decentralizált megoldású. A vezérjeles gyűrű karbantartása ettől teljesen eltérő módon valósul meg. Minden gyűrűben van egy felügyelő állomás (monitor station), amely a gyűrű karbantartásáért felelős. Ha a felügyelő állomás meghibásodik, akkor a helyébe, egy versenyprotokoll alapján gyorsan megválasztott másik állomás lép. (Minden állomásnak megvan az esélye, hogy felügyelő állomássá váljon.) Amíg azonban megfelelően működik, a felügyelő állomás egyedül felelős a gyűrű helyes működéséért.

Amikor a gyűrű feláll és az első állomás — vagy bármelyik állomás — észreveszi, hogy felügyelő állomás hiányzik, egy Claim token vezérlőkeretet küldhet el. Minden állomásba beépítik a felügyelővé válás képességét.

A felügyelő felelős többek között a vezérjel vesztés figyeléséért, a gyűrűszakadáskor elvégzendő teendők elvégzéséért, az összekeveredett keretek eltávolításáért és az árván maradt keretek kiszűréséért. Árva keret akkor keletkezik, amikor egy állomás egy rövid keretet a maga teljességében kibocsát, de annak kivonására már nem képes, mert időközben meghibásodott, vagy kikapcsolták. Ha erre a rendszer nem figyelne, akkor a keret a végtelenségig cirkulálna.

Az összekeveredett, ill. meghibásodott kereteket érvénytelen formátumuk, vagy helytelen ellenőrzőösszegük révén lehet felismerni.

**Hasonlítsuk össze a vezérjeles gyűrű és a vezérjeles sín vezérlési filozófiáját!**

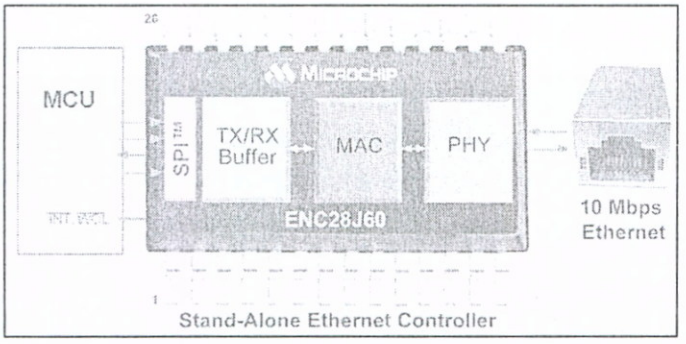
Az utóbbiban nincs olyan központi komponens, amelynek bármilyen véletlenszerűen bekövetkező hibája miatt az egész rendszer működésképtelenné válhat. Ezért olyan rendszert terveztek, amelyben a vezérjel aktuális birtokosának különleges jogai vannak: például: felvehet új állomásokat a gyűrűbe, egyébként azonban semmi sem különbözhet a többitől.

A vezérjeles gyűrű esetében a tervezők úgy érezték, hogy a vezérjel vesztés, árva keretek stb. kezelése sokkal egyszerűbb egy központi felügyelő állomással. Továbbá egy normál rendszerben az állomások csak nagyon ritkán mennek tönkre, ezért az új felügyelő állomás alkalomszerű versengéses megválasztása nem okoz nagy problémát.

## 3.3.8 Az Ethernet sikere

A gyűrűt használó megoldások – mint azt az előzőekből megállapíthatjuk, viszonylag nagy adminisztratív tevékenységet igényelnek. Az Ethernet hálózatok hatalmas karriert futottak be: ma a világon legjobban elterjedt adatkapcsolati protokoll. Vajon mi okozta ezt a hatalmas sikerét? Ennek több oka van:

1. Közeghozzáférési algoritmusa jól alkalmazkodik a hálózatot használó emberi tevékenységhez: a felhasználók nem folyamatosan használják a hálózatot, hanem időben elosztva van hálózati forgalom. A számítógépes illesztőkártyái is olcsók.
2. A jelek kódolásánál hatalmas előny, hogy minden bit önmagát szinkronizálja a jel fel és lefutó éleivel (Manchester kódolás).
3. A technikai fejlődésével, az adatátviteli sebesség növelését folyamatosan lehetett növelni (1-10-100-1000-... Mbit/sec).
4. A protokoll, a viszonylag egyszerű felépítésű adatkeretek továbbítása során nyugtázást nem használ, de ez nem baj, mert a fizikai közegben való továbbítás során, az ütközéseken kívül, hibák nem nagyon keletkeznek.
5. Az Ethernet keretekkel kommunikáló gépet könnyen lehet a rendszerbe beilleszteni: egyszerűen csupán csatlakoztatni kell a fizikai hálózathoz. A vezérjeles gyűrű és sín esetében a gyűrűbe való illesztés, illetve kilépés komoly adminisztrációt igényel a rendszertől.
6. A fizikai közeg kezdetben kialakított sin toplógiája miatt, az átvitel csak félduplex lehetett. A csavart érpáras, külön adó és vevő vonalat tartalmazó megoldások, már teljes dulex átvitelt tettek lehetővé, megduplázva ezzel a rendszer áteresztő képességét.
7. A rádiós hálózatok adatkapcsolati protokolljának szinte kínálkozik az üzenetszórást alkalmazó Ethernet.
8. Az integrált áramköri technika lehetővé tette, hogy a teljes keretkezelést egyetlen integrált




*3-56. ábra Ethernet chip*

áramköri lapka végezze. (Ethernet chip). A mai hálózati kártyákon ez az egyetlen aktív elem, és könnyen lehet ezért a PC-k alaplapjára is integrálni.

9. A hibájának tartott ütközések gyakoriságát lecsökkentették, és ezért a hálózat átviteli teljesítményét (throughput) megnövelték az Ethernet hálózatban elhelyezett intelligens kapcsolóeszközök (bridge, switch).

Bár elvileg üzenetszórásosos hálózaton működik, de képes pont-pont jellegű kapcsolat kiszolgálására is. A kábel- és ADSL modemek gyakran használnak a számítógéppel való adatcserére Ethernet közeget. (sebesség van, ütközés nincs!)

## Ellenőrző kérdések: 3. Fejezet

?

1. Hogyan működik a telefon? Mi a helyi központ feladata? Hogyan épül fel egy hierarchikus telefonrendszer? Mi az a bérelt vonal? Mit jelent a PSTN?
2. Ismertesse a moduláció fogalmát! Szinuszos jel milyen jellemzőit lehet modulálni?
3. Mi az amplitudó moduláció?
4. Mutassa be az FSK átviteli módszert!
5. Ismertesse a fázismodulációt, illetve a fázis-amplitudó modulációt! Mi az a QAM16?
6. Milyen részekből áll egy modem? Hogyan lehet a működésmódját beállítani?
7. Soroljon fel néhány jellemzőt, amit a modemnek adott parancsokkal beállíthatunk!
8. Mik azok voice-, illetve fax modemek?
9. Hogyan működnek a V92 szabványú modemek?
10. Mik azok a szoft modemek?
11. Ismertesse a széles sávú analóg jelátvitelet hasznosító kábel-TV működését!
12. Mutassa be, és hasonlítsa össze, a soros és a párhuzamos adatátviteli elveket!
13. Soros adatátvitelnél Hogyan tudjuk biztosítani az adó és a vevőoldal szinkronizmusát?
14. Mit takarnak a karakter- illetve bitorientált átviteli eljárás fogalmak? Mi a szinkron és aszinkron átvitel lényege?
15. Mi a bitbeszúrás?
16. Az adatátvitelnél mi a bitsebesség, és mi a jelzési sebesség?
17. Melyek a digitális jelek vonali kódolásánál figyelembe veendő legfontosabb szempontok?
18. Mi az NRZ, illetve RZ digitális kódolási módszer lényege?
19. Mi az AMI, a HDB3, illetve a PE digitális kódolási módszer lényege?
20. Mutassa be az ASCII kódrendszert? Hogyan lehet a karaktereit csoportosítani? Mi az UNICODE?
21. Mi az adatkapcsolati protokollok feladata? Hogyan hozunk létre kereteket karakter orientált, illetve bitorientált összeköttetések esetén?
22. Ismertesse a karakter-beszúrásos módszer lényegét!
23. Ismertesse a bitbeszúrásos módszer lényegét!
24. Folglalja össze a hibakezeléssel kapcsolatos legfontosabb ismereteket!
25. Mi a különbség egyedi és csoportos bithibák között?
26. Mi az Error Correctin Code (ECC)? Mi az a Hamming távolság, és hogyan használható hibajelzésre, illetve hibajavításra?
27. Mi az a ciklikus redundancia kód (CRC)? Hogyan használható?

28. Ismertesse, és blokkvázlaton mutassa be a korlátozás nélküli egyirányú (szimplex) protokollt!
29. Ismertesse, és blokkvázlaton mutassa be az egyirányú "megáll és vár" protokollt!
30. Ismertesse, és blokkvázlaton mutassa be az egyirányú összetett protokollt!
31. Miért előnyös a kétirányú protokollok használata? Mi az a piggy-back technika?
32. Mi a csúszóablakos protokoll lényege?
33. Ismertesse a visszalépés n-el technikájú protokollt!
34. Ismertesse a szelektív ismétlő protokollt!
35. Mutassa be az IBM BISYNC protokollt! Ismertesse vázlatosan, hogy milyen részekből áll egy BYSINC keret?
36. Hogyan történik egy BYSINC üzenetváltás?
37. Mutassa be a HDLC protokollt! Milyen állomáselrendezések lehetségesek?
38. Hogyan épül fel és milyen részekből áll egy HDLC keret?
39. Milyen kerettípusokat használ a HDLC protokoll?
40. Milyen működési módjai vannak a HDLC protokollnak?
41. Milyen feltételezésekkel tárgyalhatók a közeg hozzáférési módszerek?
42. Milyen hozzáférési módszerek lehetségesek a közeg elérési módja alapján?
43. Mi a véletlen, az osztott és a központosított átvitelvezérlés lényege?
44. Mi az az ütközés?
45. Hogyan működik az ALOHA protokoll?
46. Ismertesse a CSMA/CD módszert! Hogyan működik a réselt gyűrű?
47. Mi a vezérjel? Hogyan működik a vezérjeles sín?
48. Hogyan működik a vezérjeles gyűrű? Mi a vezérjeles gyűrű és a vezérjles sín között a különbség?
49. Mikor és miért előnyös a lekérdezéses, (polling) eljárás?
50. Foglalja össze a vonalkapcsolásos és TDMA eljárás lényegét!
51. Mutassa be az OSI modell és az IEEE812-es szabványok kapcsolatát! Ismertesse a 802.1-802.5 szabványokat!
52. Miért osztotta a szabvány az adatkapcsolati réteget két alrétegre. Mi ezek feladata? Mi a közeghozzáférési (MAC) alréteg feladata és funkciói? Milyen típusú kábelek használhatók a fizikai rétegben? Mi a logikai kapcsolatvezérlési (LLC) alréteg feladata és funkciói?
53. Mi az a SAP? Milyen réteghatáron helyezkedik el?
54. Ismertesse a 802.3 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Mi a kapcsolata az Ethernettel?
55. Milyen kódolást használ a 802.3? Mi a lényege?
56. Foglalja össze az Ethernet hálózatban használt hálózati eszközöket! (HUB, REPEATER, BRIDGE, SWITCH).
57. Ismertesse a 802.3 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?
58. Hogyan kezeli az Ethernet az ütközéseket? Mi a bináris exponenciális visszatartás módszerének a lényege?
59. Foglalja össze a struktúrált kábelezés tulajdonságait!

60. Ismertesse a 802.4 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Milyen kábelezést használ? Ismertesse a 802.4 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?
61. Ismertesse a 802.5 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Mi a gyűrű felépítés előnye? Milyen kábelezést használ? Ismertesse a 802.5 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?
62. Hogyan történik a vezérjeles gyűrű karbantartása?
63. Hasonlítsa össze a vezérjeles gyűrű és a vezérjeles sín vezérlési filozófiáját!
64. Foglalja össze, hogy minek köszönhető az Ethernet sikere!

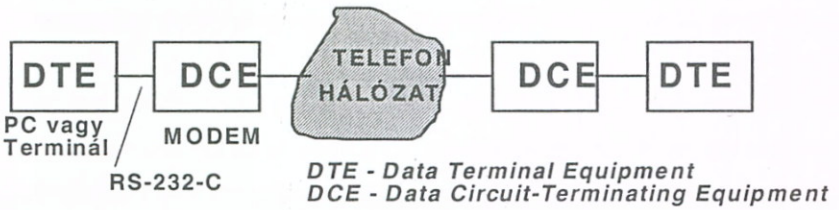
# 4. ADATKERETEK ÁTVITELI MEGOLDÁSAI

*Az embert a hitek vezérlik, de a tények mozgatják.*

Az előző fejezetben már megismerkedtünk az adatkapcsolati rétegben áramló keretekkel, átviteli protokolljaival. Ebben a fejezetben összfoglaljuk a gyakorlatban használt keret átviteli megoldásokat, fizikai megoldásaikkal együtt.

## 4.1 Aszinkron és szinkron soros adatátvitel

A nagyfokú és széleskörű elterjedése miatt egy számítógép és egy modem, vagy terminál közötti illesztés fizikai rétegének megvalósítása nagyon fontos. Ez teljes-, vagy félduplex, pont-pont típusú összeköttetés kialakítását igényli. Részletesen meg kell határozni a mechanikai-, a villamos-, a funkcionális-, és eljárás interfészeket.

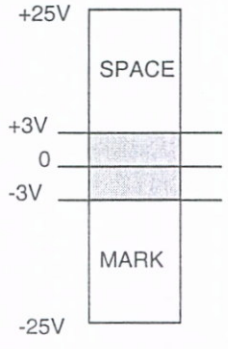



*4-1. ábra DTE és DCE egységek kapcsolata*

Az ezt megvalósító szabvány megalkotója az Electronic Industries Association elnevezésű, elektronikai gyártókat tömörítő szakmai szervezet, így az EIA RS-232-C a pontos hivatkozás. Ennek nemzetközi változata a CCITT V.24. ajánlása, amely csak néhány ritkán használt áramkörben tér el. Az ajánlás (Recommended Standard 232 C), az eredeti ajánlás harmadik („C") változata.

Mivel személyi számítógépek megjelenésével a benne található soros periféria szabványos illesztő felületté vált, ezért a soros vonalat széles körben — eredeti funkcióján túlmenően — kezdték különböző perifériális eszközök illesztésére felhasználni. Az igazsághoz hozzátartozik, hogy jelentősége az USB busz megjelenésével csökkent, de ipari környezetben még ma is széles körben használt.

A szabványleírásban a számítógép és a terminál hivatalos neve: **adatvégberendezés — DTE (=Data Terminal Equipment), míg** a kapcsolódó modemé **adatáramköri-végberendezés—DCE (=Data Circuit-Terminating Equipment),** és a köztük zajló kommunikáció az RS-232 soros vonalon folyik. A legtöbb gyakorlati esetben a DTE egy terminál, vagy egy számítógép, míg a DCE az analóg telefonhálózathoz kapcsolódó modem. (*4-1. ábra*).




*4-2. ábra RS232 jelszintek*

### 4.1.1 RS-232C szabvány

A DTE-DCE egységeket összekötő vezetékrendszer mechanikus csatlakozóját is definiálták: 25 pólusú Canon csatlakozó (szokták DB-25-nek is nevezni). Két, egymásba dugható csatlakozó közül a dugós rész a DTE-n, a hüvelyes részt a DCE-n helyezkedik el. Mégis, a gyakorlatban elterjedt a mindössze 9 pólust tartalmazó DB9-es csatlakozó.

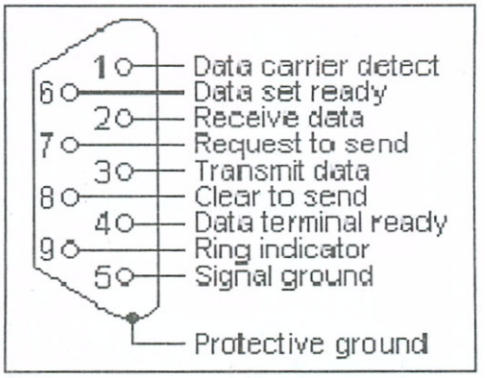
A villamos specifikáció szerint a -3V-nál kisebb feszültség a vonalon a 1-et (MARK), míg a +3V-nál nagyobb feszültség 0 -át (SPACE)

jelent. A legfeljebb 15 m hosszú kábeleken 20 kbit/s-os maximális adatátviteli sebesség a megengedett. A legtöbb gyakorlati esetben (pl. a számítógépek soros vonalánál) a feszültség ±12V.

A funkcionális előírás a 25 ponthoz tartozó vonalakat megjelöli, és leírja azok jelentését. A DB9-es csatlakozóra azt a 9 jelet vezették ki, amelyeket majdnem mindig felhasználnak. (*4-3. ábra*)

| FUNKCIÓ | JEL | DB25 LÁB | DB9 LÁB | DTE | DCE |
|---|---|---|---|---|---|
| ADAT | TxD | 2 | 3 | O | I |
| | RxD | 3 | 2 | I | O |
| KAPCSOLAT | RTS | 4 | 7 | O | I |
| | CTS | 5 | 8 | I | O |
| | DSR | 6 | 6 | I | O |
| | DCD | 8 | 1 | I | O |
| | DTR | 20 | 4 | O | I |
| KÖZÖS | GND | 7 | 5 | - | - |

1. Amikor a számítógépet vagy a terminált bekapcsolják, az aktiválja (MARK-ba állítja) az Adatterminál kész (Data Terminal Ready) jelet (20).
2. Amikor a modemet kapcsolják be, akkor a modem az Adat kész jelet (Data Set Ready) (6) aktiválja.
3. Ha a modem vivőjelet érzékel a telefonvonalon, akkor a Vivőérzékelés (Data Carrier Detect) jelet (8) aktiválja.
4. Az Adáskérés (Request to Send) (4) jelzi, hogy a terminál adatot akar küldeni.
5. Az Adásra kész (Clear to Send) (5) azt jelenti, hogy a modem felkészült az adatok fogadására.
6. Az adatok adása az Adás (Transmit) vonalon (2), vétele a Vétel (Receive) vonalon (3) történik.




*4-3. ábra DB25 ÉS DB9 soros csatlakozó jelei*

A többi, fel nem tüntetett áramkör a gyakorlatban alig használt funkciókkal rendelkezik: adatátviteli sebesség kiválasztása, modem tesztelése, adatok ütemezése, csengető jelek érzékelése, adatok másodlagos csatornán való fordított irányú küldése.

Az eljárásinterfész az a protokoll, amely az események érvényes sorrendjét határozza meg. A protokoll akció-reakció eseménypárokon alapszik. Amikor egy terminál kiadja

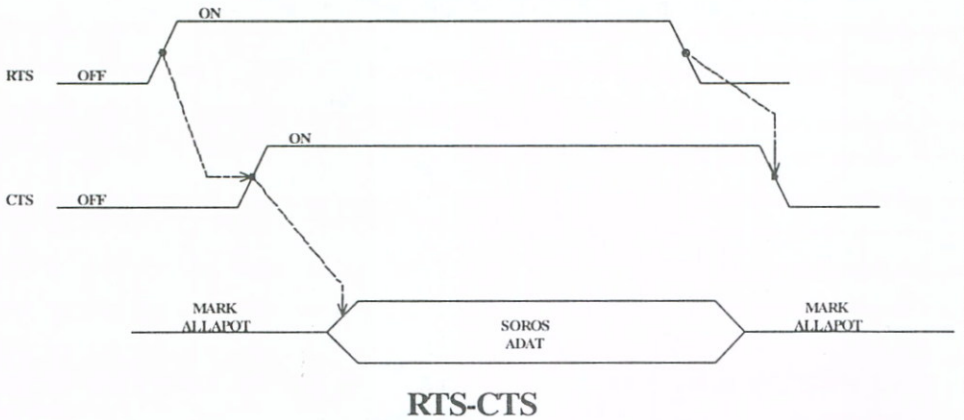

**CTS: DCE kész az adatcserérére a DTE-vel**
**RTS: DTE adatot akar cserélni a DCE-vel**
**DSR: DCE vonalra kapcsolódását jelzi**
**DTR: DTE kész a vonalra kapcsolódott DCE adatait fogadni**

*4-4. ábra: DTE-DCE összekötő vezetékek*

pl. az Adáskérés jelet, a modem egy Adásra kész jellel válaszol, ha képes fogadni az adatokat. Ugyanilyen jellegű akció-reakció-párok léteznek a többi áramkör esetén is. *(4-5. ábra).*

Az aszinkron soros átvitelnél a bitcsoportos átviteli mód biztosítja az ADÓ és a VEVŐ szinkronizmusát. Természetesen ehhez a járulékos információhoz járulékos biteket is

fel kell használni. Ezek a START és a STOP bitek. Ezeket szokták keretező (framing) biteknek is nevezni, mivel a tényleges információt "keretbe foglalják". A START bit jelzi, hogy utána következnek a tényleges információt hordozó adatbitek, míg a STOP bit(ek) ezek végét jelzi.




RTS-CTS

*4-5. ábra: RTS-CTS kézfogásos kapcsolat*

A soros protokoll szerint, ha a soros vonalon nem folyik információátvitel, a vonal állapota aktív (MARK) szintű. Az adatátvitel kezdetekor az ADÓ a vonalat egy bit átvitelének idejéig alacsony (SPACE) szintre állítja (START bit), majd utána történik meg az adatbitek átvitele. Az átvitt adatbitekből álló bitcsoport végére az ADÓ STOP bit(ek)-ből álló aktív (MARK) szintű jelet helyez el. A VEVŐ az adás kezdetéről a vonal MARK-SPACE állapotváltozásából szerez tudomást.


*4-6. ábra: A soros adatátvitel elve*

Ezután a VEVŐ, sorban egymás után, az adatbiteket veszi. A STOP bitek érkezése után már figyelheti a vonalon ismét megjelenő állapotváltozást, ami a következő bitcsoport adásának kezdetét jelöli. Fontos kérdés a vonalon időegység alatt átvitt információ mennyisége, amit bit/sec-ban mérünk.

Az adatátvitel során az esetleges átviteli hibák felderítését megkísérelhetjük olyan módon, hogy az átviendő adatbit csoportot egy paritás bittel egészítjük ki úgy, hogy az így kiegészített adatcsoportban lévő 1 értékű bitek száma páros (páros paritás), vagy páratlan (páratlan paritás) legyen. Ilyen módon, az ADÓ oldalán mindig biztosítható, hogy az 1-es értékű bitek száma mindig páros/páratlan legyen, és a VEVŐ oldalon az egy (ill. páratlan számú) bit változása miatti hiba felderíthető.

Mivel az elsőnek átvitt bit mindig a START bit, ezért az előzőek alapján a soros adatátviteli protokoll konkrét kialakításánál a következő paramétereket kell rögzíteni:

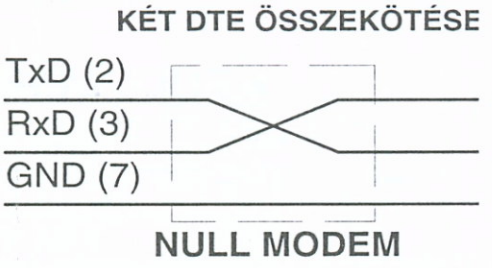
- **Adatbitek száma**: a gyakorlatban 5, 6, 7 vagy 8 bit.
- **Paritásbit**: használunk paritásbitet vagy nem, és ha igen, páros vagy páratlan paritást alkalmazunk.
- **Stop bitek száma**: ez a soros vonalnak a bitcsoport átvitele utáni garantált logikai 1 állapotának az idejét határozza meg, az egy bit átviteléhez szükséges idővel kifejezve. Hossza 1, 1.5, vagy 2 bit lehet. A legrövidebb az egy bit, és ez biztosítja, hogy a VEVŐ a következő bitcsoport vételéhez szükséges szinkronizáló START bit indító élének érzékelésére felkészüljön. Két stop bit használata akkor előnyös, ha valamilyen okból szükséges a vett adatbitek azonnali feldolgozása és az ehhez szükséges hosszabb idő.

- **Adatátviteli sebesség (bit/s):** Igen fontos adat, mert ez határozza meg alapvetően az ADÓ és a VEVŐ szinkronizmusát. Szokásos értékei: 75,150,300,600,....,9600,19200,....115200 bit/sec.

Mivel a soros adatátvitelt széles körben használják, ezért, megvalósítására céláramköröket fejlesztettek ki. Ezeknél az ADÓ oldalán csupán az adatbit csoportot kell párhuzamosan a bemenetekre adni, az áramkör elvégzi a sorossá alakítást, a paritás, START és STOP bitekkel való kiegészítést, valamint az átvitelt. A vevőoldalon a vett soros adatokból vevőáramkör képezi a bitcsoportot. Ezek az áramkörök programozhatóak, azaz vezérlőkódokkal megadhatók az átvitel paraméterei és a soros adatátviteli protokoll. Mivel ezek az áramkörök TTL jelszintekkel működnek, ezért be és kimenetükön 0 és 5 V-os jeleket várnak illetve adnak.

Ezért ezeket a soros perifériа áramköröket mindig ki kell egészíteni egy olyan szintátalakító áramkörrel, amely a TTL szinteket szabványos RS232 jelszintekké, oda- és visszaalakítja, a 0V (SPACE) -> +12V; 5V (MARK)-> -12V szabályok szerint.

Mivel majdnem minden számítógépnek van soros vonala, gyakran előfordul, hogy két számítógépet RS-232-C soros vonalon keresztül kötnek össze. Mivel nem DTE-DCE típusú az összeköttetés, ezért a megoldás egy **null-modem**-nek nevezett „eszköz" (hiszen ez tulajdonképpen egy keresztezett kötésű adatkábel), amely az egyik gép adási vonalát a másik gép vételi vonalával köti össze. A legegyszerűbb esetben ez elegendő, ha azonban a modemvezérlő vonalakat is használnunk kell, akkor hasonló módon néhány más vonal keresztbe kötését is el kell végezni.



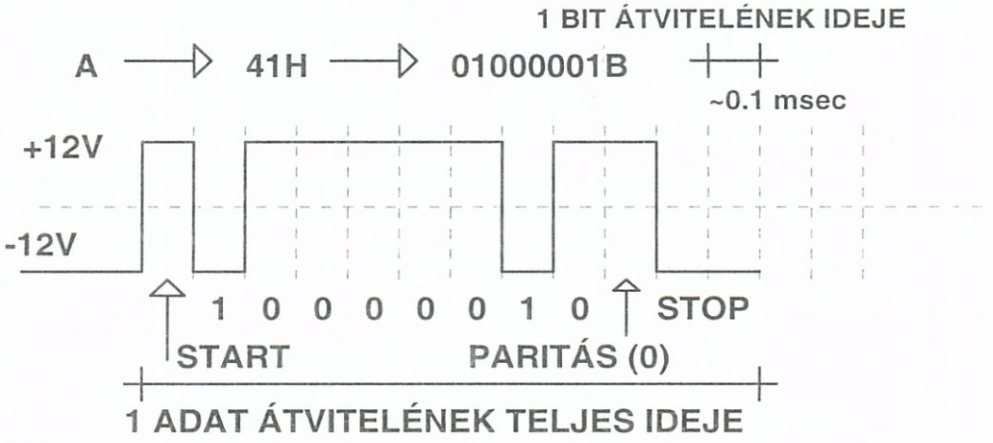

*4-7. ábra Null-modem: Két DTE összekötése*

A tényleges adatátviteli sebesség számítására nézzünk egy konkrét példát: Ha például az adatátviteli sebesség 9600 bit/s, és 8 bites adatokat (bájtokat) viszünk át páros paritásbittel kiegészítve, 2 STOP bittel a végén, akkor például másodpercenként:

`9600/(1 START + 8 ADAT + 1 PARITÁS + 2 STOP bit) = 9600/12 = 800 adat (bájt)`

kerül átvitelre. Ha például az „A" karaktert visszük át (ASCII kódja 41H), akkor a soros vonalon a következő jelformát láthatnánk: (*4-8. ábra*).

A karakter bitjeit fordított sorrendben visszük át (binárisan a legkisebb értékű (LSB) az első). A páros paritásbit ennél a karakternél 0 mert eleve páros (kettő) 1 értékű bitet tartalmazott.




*4-8.ábra Egy soros jelalak*

Amint láttuk, az RS-232-C illesztés elsődlegesen számítógépmodem összeköttetésre tervezték, és mivel alkalmazhatósága miatt más területeken, az iparban is kezdték használni, az adatátviteli

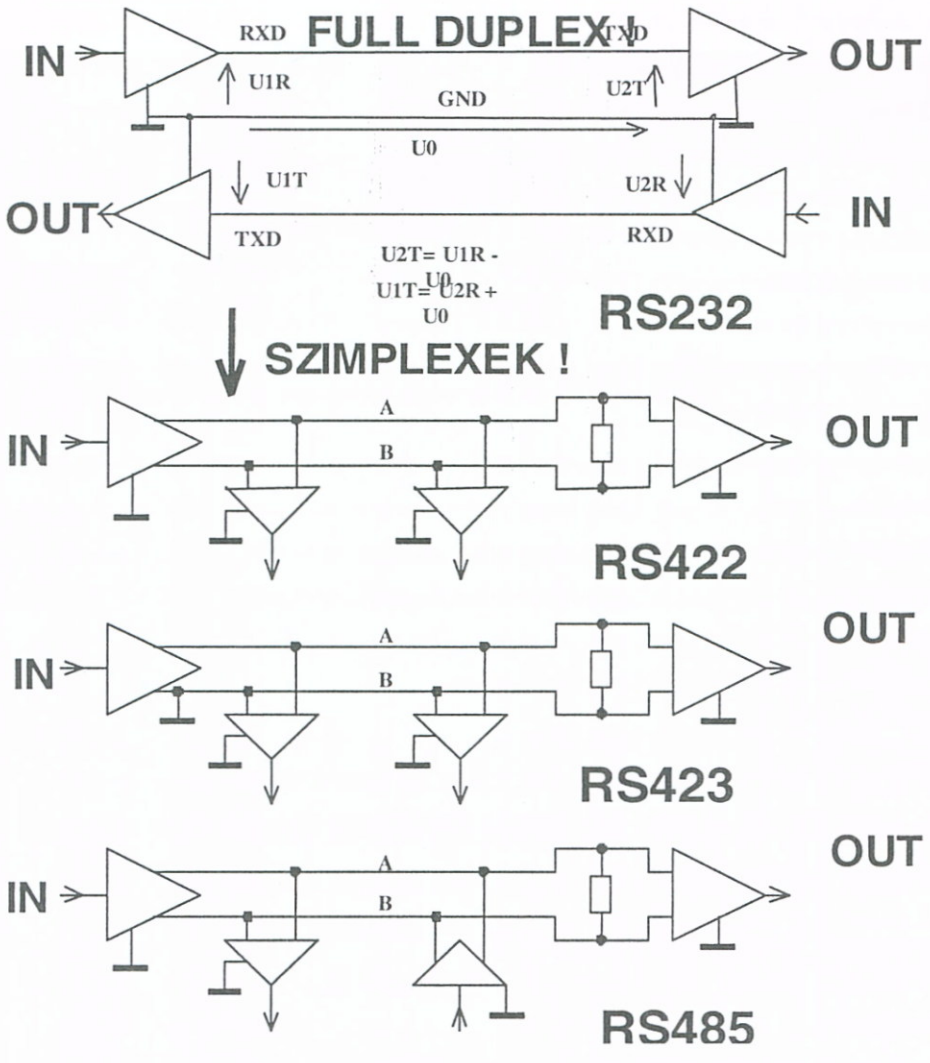
sebességre tett 20 kbit/s-os és a kábelhosszúságra tett 15 m-es korlátozás fokozatosan zavaróvá vált.

Mivel a jeleket egy közös földvezetékhez képest mérjük, ezért a rendszer a villamos zavarokra nagyon érzékeny. Az EIA sokat vitázott, hogy vajon egy olyan új szabványt fejlesszenek-e ki, amely a régi (de technikailag nem túl fejlett) szabvánnyal kompatíbilis, vagy egy olyat, amely a régivel nem kompatibilis, de a korszerű követelményeket maradéktalanul kielégíti. A kompromisszumos megoldás mindkettőt tükrözi.

**Itt fontos megjegyezni, hogy csak a jelet átvivő fizikai réteget cseréljük le! A logikai jelkezelés változatlan maradt.**

## 4.1.2 Az RS-449, -422, -423, és az RS-485-ös szabványok

Az új, RS-449-nek nevezett szabvány valójában három szabvány ötvözése. A mechanikai, a funkcionális és az eljárási interfész az RS-449 szabványban, míg a villamos interfész két további szabványban van megadva.




*4-9. ábra: A soros adatátviteli RS szabványok*

Mindkét villamos szabványnál a jeleket az összekötő vezetékpár közötti feszültségkülönbség hordozza, és a vevők bemenetén lévő differenciálerősítő fogadja ezeket a jeleket. Mivel a zavart indukáló külső villamos zaj hatása mindkét vezetéken megjelenik, ezért a különbségképzésnél ezek hatása kölcsönösen kioltja egymást.

E kettő közül az egyik az RS-423-A, amely az RS-232-C szabványhoz hasonlít abban, hogy áramköreinek közös földje van. Ezt a technikát **asszimmetrikus átvitelnek (unbalanced transmission)** nevezik, mert a jeleket a közös (föld)vezetékhez képest értelmezzük. A másik villamos interfész az RS-422 ellenben a **szimmetrikus átvitelt (balanced transmission)** használja, amelyben minden fő áramkör két jelvezetékkel rendelkezik.

Ennek eredményeképpen az RS-422-A szabvány 2 Mbit/s-os átviteli sebességet enge-

délyez. Ezek a szabványoknál már az egy ADÓ mellett több vevő is lehet a vonalon, de így átvitel csak szimplex. A pont-pont típusú összeköttetés helyett itt már üzenetszórásos összeköttetés van, és ez az ún. **multi-drop** kialakítás. Teljes duplex átvitelhez két egység között még egy vezetékpárt kell alkalmazni, ellentétes VEVŐ-ADÓ áramkörökkel. Ez a **négyvezetékes átvitel**.

Az egyre intelligensebb összekapcsolt eszközök igénylik a kétirányú kommunikációt. Ezért 1983-ban az EIA egy újabb szabványt jelentett meg, az RS-485-öt. Az RS-422-höz hasonló szimmetrikus átvitelt használja, de a vonalpáron már több ADÓ és több VEVŐ is lehet és közöttük az egy vezetékpáron félduplex összeköttetés lehetséges. Természetesen a teljes duplex kommunikációhoz itt is a négyvezetékes kialakítás szükséges.

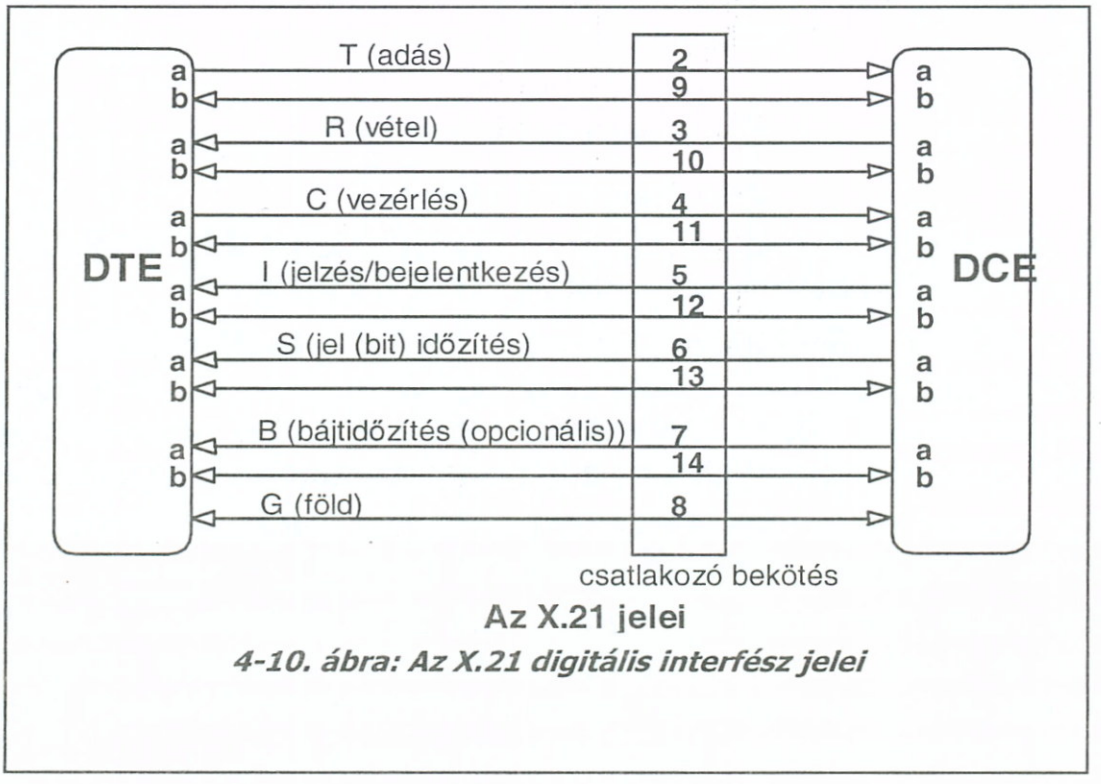
Összefoglalásul nézzük meg a fent részletezett szabványok leglényegesebb jellemzőit:

| Jellemzők | RS 232C (V.24) | RS 423 (V.10) | RS 422 (V.11) |
|---|---|---|---|
| Átvitel | aszimmetrikus | aszimmetrikus | szimmetrikus |
| Kábel típus | sodrott érpár | koaxiális | sodrott érpár |
| Kábelhossz | 15 m | 600 m | 1200 m |
| Adatsebesség (max) | 20 kbit/s | 300 kbit/s | 2 Mbit/s |
| Meghajtó kimeneti szint (terheletlen) | +/- 25 V | +/- 6V | +/- 6V (diff) |
| Meghajtó kimeneti szint (terhelt) | +/- 5...+/- 15V | +/- 3,6 V | +/- 2V (diff) |
| Minimális vételi szint | +/- 3V | +/- 0,2 V | +/-0,2V (diff) |

## *X.21 interfész*

A CCITT egy digitális interfész-ajánlást adott ki 1976-ban, az X.21-et. Ez az ajánlás a felhasználói számítógép, (DTE), és a hálózathoz kapcsolódó készülék, (DCE) közötti hívásokat, valamint az azok kiadásához és törléséhez szükséges jelcseréket rögzíti. Az X.21 által definiált 8 vezeték irányát elnevezését és jelentését a 4-10.*ábra* tartalmazza.

Lényegében úgy tekinthető, mint az RS232 „ipari" változata. Az alkalmazott csatlakozó 15 pontos, de nincs mindegyik kihasználva.




***4-10. ábra: Az X.21 digitális interfész jelei***

# 4.2 ISDN

A klasszikus távbeszélő rendszereket analóg hangátviteli célokra tervezték, és nem alkalmasak modern digitális távközlési igények kielégítésére (adat-, fax- vagy videoátvitelre). Az ISDN célja digitális jeleket vivő hálózat kialakítása, amely a meglévő, analóg távbeszélő hálózatot hivatott felváltani.

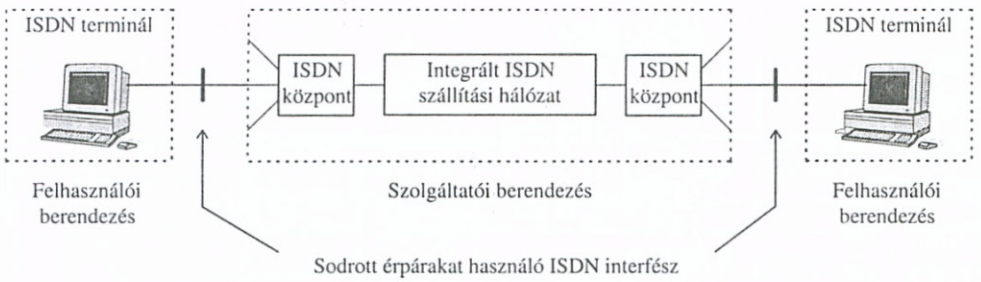
Az új digitális rendszer elsődleges célja az, hogy integrálja a hang-, és nem hangjellegű átviteli szolgáltatásokat. Elnevezése: **ISDN (Integrated Services Digital Network — integrált szolgáltatású digitális hálózat).**

**ALAPGONDOLAT: Ne az analóg működésű hálózaton vigyünk át digitális információt, hanem digitális hálózatokat alakítsunk ki, és itt a digitalizált hang csak az egyik átviendő komponens.**

Régebben az analóg (hang-) átviteli távbeszélőrendszerek, a nyilvános kapcsolt hálózatok, a kapcsolás felépítésére szolgáló vezérlőinformációikat, az ún. jelzéseket ugyanabban a 4 kHz-es csatornában vitték át, mint amelyben az emberi hangot.

Jelenben az átvitelt és kapcsolóhálózatokat az *4-11.ábrán* bemutatott bináris jelátviteli integrált hálózat váltotta fel.

## 4.2.1 Az ISDN szolgáltatásai




***4-11. ábra: ISDN hálózat***

A legalapvetőbb szolgáltatás továbbra is a hangtovábbítás, de számos új tulajdonsággal kiegészítve. Az ISDN telefonokon több, azonnali hívásfelépítésre alkalmas gombot helyezhetünk el, amelyekkel a világ bármelyik telefonját el lehet érni. A telefonok a kicsöngés ideje alatt a hívó telefonszámát, nevét és címét is kijelezhetik.

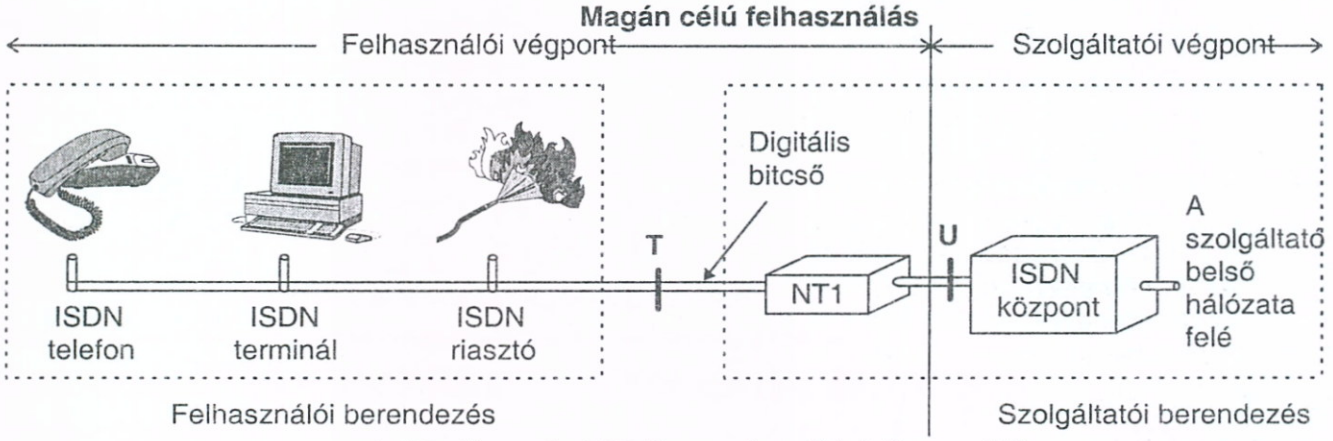
E sajátosság kifinomultabb változata szerint a telefonkészülék egy számítógéphez is hozzákapcsolódik azért, hogy egy bejövő híváskor a hívó adatrekordja képernyőn megjeleníthető legyen. Egy másik fontos adatátviteli sajátosság az, hogy zárt felhasználói csoportok alakíthatók ki, ami magánhálózatok létrehozását teszi lehetővé.

Egy csoport tagjai csak a csoport más tagjait hívhatják, és kívülről sem jöhet be semmiféle hívás (csak szigorúan ellenőrzött módon).

Egy másik, várhatóan népszerűvé váló ISDN szolgáltatás a teletex, amely valójában házi és üzleti célokra átalakított elektronikus levelezési szolgálat. További korszerű hangátviteli szolgáltatások: világméretű konferenciahívások lebonyolítása (kettőnél több partner között).

Számos esetben kézzel aláírt szerződések, ábrák, grafikonok, fénymásolatok, és egyéb grafikus anyagok átvitele válhat szükségessé. Ehhez egy másik ISDN szolgálatot célszerű igénybe venni, a Csoport 4 módban

működő színes, vagy szürkeségi fokozatokat átvivő faxot. Kialakíthatók távmérési (telemetria) vagy riasztó (alarm) szolgáltatások is az ISDN szolgálat segítségével.




*4-12. ábra: Az ISDN magán célú felhasználása*

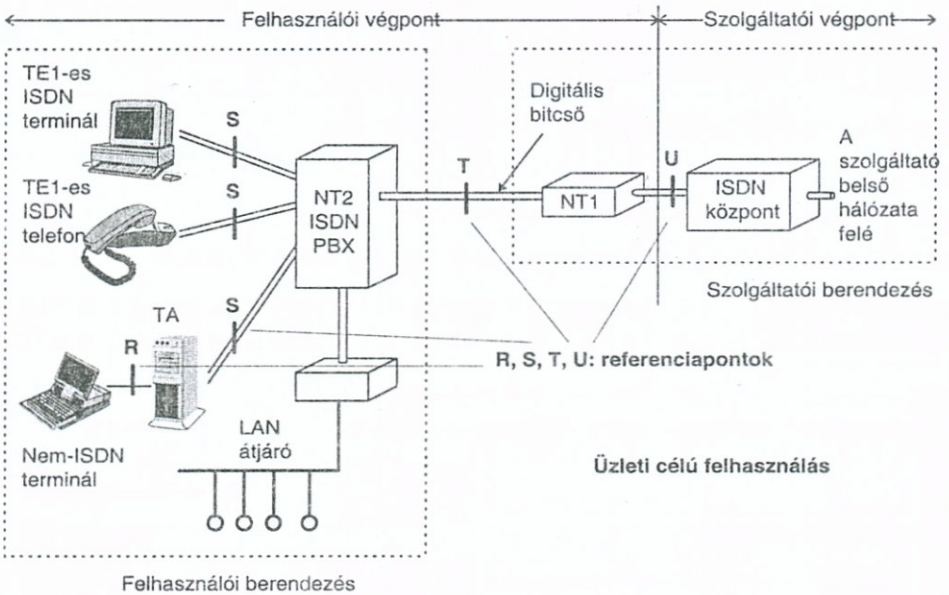
## 4.2.2 Az ISDN rendszer felépítése

Az ISDN alapkoncepciója az ún. digitális bitcső (digital bit pipe). Ezen — a felhasználó és a szolgáltató között húzódó képzeletbeli csövön — áramlanak mindkét irányban az információt szállító bitek. A bitfolyam időosztásos multiplexelésével a digitális bitcső támogathatja a bitcső több független csatornára való felosztását.

Két alapvető bitcső szabványt fejlesztettek ki: egy kisebb adatátviteli sebességűt magán célokra, és egy üzleti célokra tervezett nagyobb sebességűt, amely több csatornát támogat.

A szolgáltató a felhasználói helyszínen elhelyez egy hálózati végződést, amelynek a neve NT1 (Network Termination 1), amelyet ezután ugyanazzal a sodrott érpárral, amellyel a felhasználó telefonja a hagyományos telefonközponthoz volt kötve, most egy ISDN központhoz köti.

Az NT1 dobozán lévő csatlakozóba egy passzív sínkábel illeszthető be. A kábelhez nyolc eszköz — ISDN telefonok, terminálok, riasztók, és egyéb más berendezések — csatlakoztatható.




*4-13. ábra: Az ISDN üzleti célú alkalmazása*

Nagyobb vállalatok számára a *4-13. ábrán* látható konfigurációt alkalmazák. Ebben a modellben az NT1-el összekötve egy 2-es típusú hálózatvégződést, egy NT2-t (Network Termination

2) is találunk. A modellben az NT2-t egy digitális telefon alközpont, a PBX (Private Branch eXchange) testesíti meg, ez nyújtja a telefonok, terminálok és egyéb berendezések számára a valódi interfészt.

A CCITT négy referenciapontot határozott meg a különböző eszközök között: R, S, T, és U.
Ezek a referenciapontok lényegében szabványosan kialakított csatolófelületek. (Rétegszemlélet !)
Ezek a pontok interfészként mintegy szétválasszák az egyes ISDN elemeket.

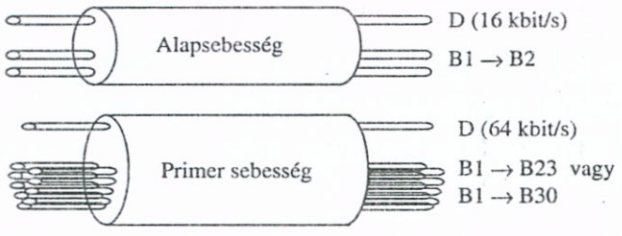
- Az **U referenciapont** a szolgáltatói hivatalban lévő ISDN központ és az NT1 közötti összeköttetést jelenti. Ez jelenleg kétvezetékes sodrott érpár, de a jövőben optikai szálra cserélhető.
- A **T referenciapont** az, amit a csatlakozó az NT1-en a felhasználónak biztosít.
- Az **S referenciapont** az ISDN PBX és az ISDN terminálok közti interfész.
- Az **R referenciapont** a terminál adapterek és a nem-ISDN terminálok közötti összeköttetés.

Az R-interfészfelületnél nagyon sokféle interfész található. A gazdaságosság az NT1 és NT2 közös egységbe integrálását követeli.

## 4.2.3 Az ISDN interfész

| | |
|---|---|
| A | 4 kHz-es analóg telefoncsatorna |
| B | 64 kbit/s-os PCM csatorna hang és adatátviteli célokra |
| C | 8 vagy 16 kbit/s-os digitális csatorna |
| D | 16 vagy 64 kbit/s-os digitális csatorna, a sávon kívüli jelzésre |
| E | 64 kbit/s-os digitális csatorna, a sávon belüli jelzésre |
| H | 384, 1536, ill. 1920 kbit/s-os digitális csatorna |

Az ISDN célja, hogy a felhasználó számára egy digitális bitcsövet biztosítson, akár a T, akár az S referenciapontón keresztül. Új típusú csatlakozót definiál, aminek nyolc érintkezője (tűje) van. Ebből kettő adás és adási föld, további kettő vétel és vételi föld számára van kijelölve. A maradék négy közül kettő a terminálok NT1, ill. NT2 általi tápfeszültség ellátásra, míg kettő ennek fordítottjára használatos. A szimmetrikus átviteli módnak a következtében az ISDN kábel 1km hosszú lehet, és jó zajtűrő képességgel rendelkezik. A bitcsőben több csatorna is kialakítható, amelyek közül az eddig szabványosított csatornatípusokat a táblázat tartalmazza.




*4-14. ábra: Bitcsövek*


*4-15. ábra: Az ISDN keretformátum*

A szabványosításnak nem szándéka, hogy a digitális bitcsövön a csatornák tetszőleges kombinációja létezzen. Eddig három kombinációt szabványosítottak(1994):

**1) Alapsebesség:** **2B + 1D**

**2) Primer sebesség:** **23B + 1D (U.S. és Japán) vagy**
**30B + 1D (Európa)**

**3) Hibrid:** **1A + 1C**

Az alapsebességű D-csatorna 16 kbit/s-os. Híváskéréseket az ezen elküldött üzenetek segítségével lehet kiadni.

A primer sebességű interfészt a T referenciapontoknál való használatra, PBX-el rendelkező üzleti vállalkozások számára tervezték. A 23B + 1D választás lehetővé teszi, hogy egy ISDN keret kényelmesen illeszkedjen az AT&T T1 rendszeréhez. A 30B + 1D választás pedig a CCITT 2,048 Mbit/s-os rendszeréhez való illeszkedést teszi lehetővé.

A *4-15. ábrán* azt a fizikai rétegbeli keretformátumot láthatjuk, amely alapsebességű NT1-től vagy NT2-től a TE1 felé irányuló forgalom esetén érvényes. A keret 48 bitből áll, az adatbitek száma 36. A keret elküldéséhez 250 µsec szükséges, ami 144 kbit/s-os adatátviteli sebességet jelent, de ha a nem adatbiteket is számítjuk, akkor az átviteli sebesség 192 kbit/s-ra emelkedik.

Fontos tudnunk azt, hogy a felhasználói adatok csupán egy **nyers bitfolyamot** alkotnak. **Nincs hibaellenőrzés, nincs ellenőrzőösszeg, nincs redundancia, nincs nyugtázás és nincs újraadás sem.** Ha hiba történik, akkor azt a felsőbb rétegeknek kell javítania. Az ISDN semmi mást nem tesz, mint a B csatornák (és kisebb mértékben a D-csatorna) segítségével a felhasználónak nyers bitfolyamot biztosít.

### 4.2.4 Az ISDN jelzésmód

Minden jelzés (hivószámok), azaz vezérlőcsomag küldése a D-csatornán zajlik. A B-csatornák teljes 64 kbit/s-os kapacitása tisztán a felhasználói adatok átvitelére fordítható, sem fejrészek, sem egyéb más információ elküldésére nincs szükség. Az ISDN nem határozza meg a B-csatornák tartalmát.

A D-csatornával a helyzet alapvetően más. A felhasználó ezt éppen az ISDN rendszerrel való kommunikációra használja. Egy hívás végrehajtásához az ISDN eszköznek egy meghatározott formátumú csomagot kell küldeni NT1-nek.

## 4.3 ATM — Asychronous Transfer Mode

A hálózatok továbbfejlődésében nagy szerepet játszik az olyan átviteli módszerek alkalmazása, amelyek figyelembe veszik az információforrások különbözőségét. A jelenlegi információátviteli rendszerek olyan protokollokat, adatátviteli módszereket használnak, amely az adott típusú információ átviteléhez fejlesztettek ki. Ezért más típusú információ átvitele ilyen csatornán keresztül rossz, és nem hatékony csatorna kihasználást okoz.

Az alkalmazásokat megvizsgálva, azok alapvetően kétféle digitális átvitelt igényelnek:

- **állandó bitsebességet biztosítót (CBR — Constant Bit Rate),** Ilyen például a digitális 64 kbit/s-os telefon, telefax,TV átvitel.
- **változó bitsebességet biztosítót (VBR — Variable Bit Rate).** változó sebességű adatátviteli sebességet igényel az interaktív szöveg és képátvitel. Ilyenkor

| BITEK | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| FLOW CONTROL | | | | | VPI (FELSŐ 4 BIT) | | | |
| VPI (ALSÓ 4 BIT) | | | | | VCI (FELSŐ 4 BIT) | | | |
| VCI (KÖZÉPSŐ 8 BIT) | | | | | | | | |
| VCI (ALSÓ 4 BIT) | | | | | TIPUS | | | |
| CRC ELLENÖRZŐ KÓD | | | | | | | | |

*4-16. ábra: ATM cellafejléc*

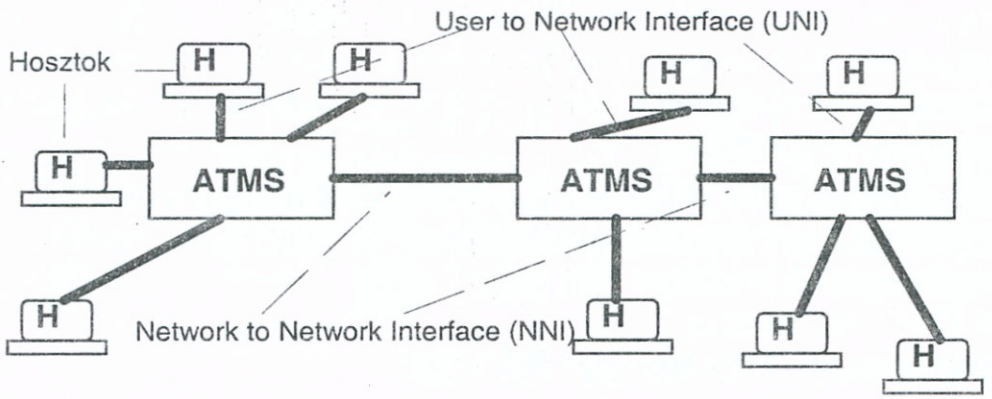
sokszor lökésszerűen egy állandó bitsebességű átvitel zajlik, amit hosszabb szünet követ. Video átvitelnél is elegendő csak a kép teljes változásakor átvinni a képet, közben csak a változásokat.

**A nagy átviteli sebességet biztosító ATM-et, ez utóbbi, VBR típusú adatátvitelre tervezték.** A követelményeket kielégítendő, az ATM kisméretű cellákkal dolgozó, aszinkron időosztásos multiplex adatátvitelt használ. A kis cellaméret előnyös, mert hiba esetén kevés információ veszik el, az átvitelt megvalósító elektronikus kapcsolók egyszerűbbek és hardveresen megvalósíthatók.

A csomagok 53 oktet (oktet=8 bit) hosszúak, ebből mindössze 5 oktet a fejléc és 48 oktet az információ. A fejrész tartalmazza a csomagot vivő virtuális kapcsolat adatait (*4-16 ábra*). Az információs mező rövid, így a kezelő pufferek méretei kicsik lehetnek.

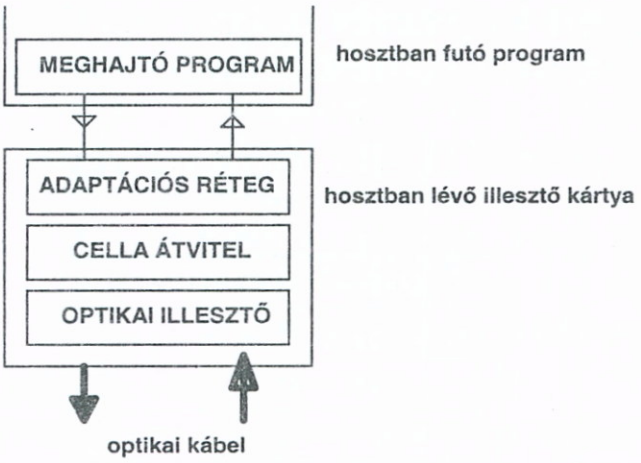
Az ATM architektúra ún. ATM kapcsolókon alapul (továbbiakban: ATMS). A kapcsolókat kétirányú üvegszálas kapcsolat köti össze egymással. A hosztok a kapcsolókhoz szintén üvegszálakon keresztül kötődnek. Az ATMS-ekből kialakított ATM hálózatot felhasználva bármely, a hálózathoz kapcsolódó hoszt tud a másik hoszttal kommunikálni.

Az ATM kapcsolat orientált összeköttetést használ a hosztok összekötésére. Ez azt jelenti, hogy először a két hoszt között egy virtuális áramkört kell kialakítani: ez lehet kapcsolt virtuális áramkör (Switched Virtual Circuits=SVC), vagy állandó virtuális áramkör (Permanent Virtual Circuits=PVC).




*4-17. ábra: ATM hálózat*

Az SVC a szokásos telefonhíváshoz hasonlóan működik. A hoszt a hozzákapcsolt ATMS-el kommunikálva, SVC létrehozását kéri. A hoszt megadja a távoli hoszt címét, a kért szolgálat minőségét, és várja, hogy az ATMS létrehozza a kapcsolatot. Ez az ATM jelzésrendszer segítségével történik. A kialakult útvonal esetleg több ATMS-en keresztül valósul meg.




*4-18. ábra: ATM protokoll szintek*

Az ATMS-hoszt összeköttetés 24 bites címet használ a virtuális áramkör azonosítására, amit az adott hoszthoz kapcsolódó ATMS rendel hozzá a kialakított kapcsolathoz.

PVC esetén a kialakított virtuális összeköttetés állandó lesz, és a rendszerkezelő konfigurálja ezt az ATMS-eken. A 24 bites címet is Ő adja a virtuális kapcsolatnak.

ATM protokoll szintek:

1 szint: fizikai szint - koax, üvegszál

2. szint: ATM datalink - 53 bájtos cellák, CBR VBR

3. szint: ATM Adaptation - bizonyos felhasználások

Az ATMS minden hoszt által kért virtuális kapcsolathoz egyedi áramkör azonosítót rendel hozzá, azonban ez csak a hoszt-ATMS között él. Több ATMS-t tartalmazó rendszerben az ATMS-ek között már más az azonosító. (Vagyis a 24 bites cím csak pont-pont kapcsolatot azonosít) Ezért célszerű a címet két részre bontani:

- 8 bites Virtual Path Identifier (VPI) és a
- 16 bites Virtual Circuit Identifier.

Miért jó ez a szétbontás? Azért mert számos áramkör ugyanazon az útvonalon halad, ezért a VPI részüket azonosra választva könnyű az egyes ATMS-eket elérni.

Az ATMS-ek cellák átvitelét valósítják meg. Az alkalmazói programok számára ez nem látható, mert az ATM szabványos adaptációs rétegével kommunikálnak. A rétegben több dolgot valósítottak meg: pl. a hibás vagy elveszett cellák kezelése, hibakezelés és javítás.

## 4.4 Az USB busz

A soros adatátvitel méltán népszerű kommunikációs megoldás, hiszen kevés vezeték felhasználásával lehet az információt átvinni. Azonban a klasszikus RS232 soros adatátvitelnek van három súlyos hátránya:

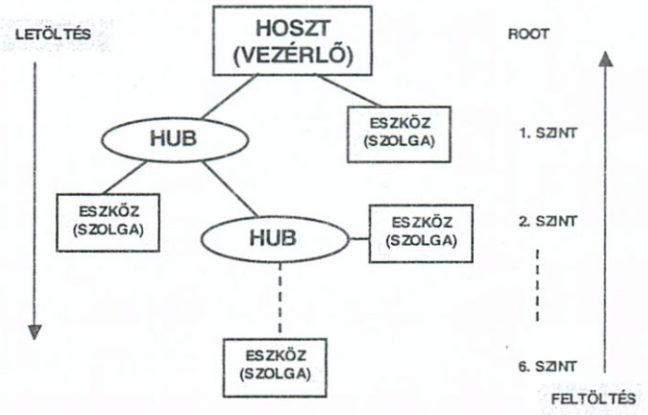
- Az általa nyújtott, jelenleg már viszonylag kicsi adatátviteli sebesség.
- Mindössze egy készülék kapcsolható hozzá, a pont-pont típusú kapcsolat miatt.
- Használata negatív tápfeszültséget is igényel.

Az USB a felsorolt hátrányok orvoslása mellett, lehetővé teszi az összeköttetés létrehozását, illetve a megszüntetését a készülék bekapcsolt állapotában, valamint kábele a csatlakozó eszköz táplálását biztosító vezetékeket is tartalmazhatja (100mA/500mA).

Az USB busz **felfűzött csillag topológiájú, lekérdezés alapú mester-szolga típusú félduplex kommunikációt valósít meg.**

Kiindulópontja mindig egy USB vezérlő (mester vagy hoszt), amelyhez vagy egy készülék, vagy újabb egységek illesztését lehetővé tévő **USB-HUB** csatlakozik.

Egy vezérlőhöz a HUB-okon keresztül maximum 127 készülék kapcsolható, és az egymáshoz kapcsolt HUB-ok szintjeinek száma sem lehet 6-nál több.




*4-19. ábra USB toplógia*

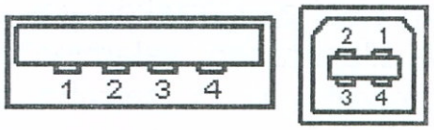
Az USB mester(hoszt) központú: a kommunikáció vezérlését mindig a mester (a hoszt) végzi. A hoszt felől érkező forgalmat a hub-ok üzenetszórásos módon minden eszközhöz továbbítják, azaz ezeket mindenki látja. A perifériák válaszai csak a hoszt felé haladnak. A kapcsolódó eszközök önállóan beszélgetést nem kezdeményezhetnek. Két sebességen képes a busz működni: a 12 Mbit/sec teljes, illetve a 1.5 Mbit/sec alacsony sebességen. Az USB2.0 szabvány már 480 Mbit/sec adatátviteli sebességet is lehetővé tesz.

### USB adatforgalom típusai

Az USB busz átviteli protokoll a PC perifériák különféle igényeinek megfelelően, négy működésmódot valósíthat meg:

- **vezérlésátvitel:** rövid, a hoszt által kezdeményezett, kétirányú forgalom, általában parancs kiküldés<->állapot visszajelzés formájú, a periféria konfigurálására és állapotának lekérdezésére szolgál;
- **állandó sávszélességet megkövetelő átvitel:** ilyen például valós idejű digitális hangtovábbítás;
- **megszakításos átvitel:** itt olyan valós idejű egyirányú átvitelről van szó, amelyben időről időre viszonylag nem túl sok adatot kell továbbítani;
- **nagy tömegű adatátvitel:** nagy adattömeg egyirányú, nem időkritikus átvitele, szükség esetén félbeszakítható és folytatható, például: folyamatos, egyirányú, nyomtatónak küldött vagy lapolvasótól érkező adatok.

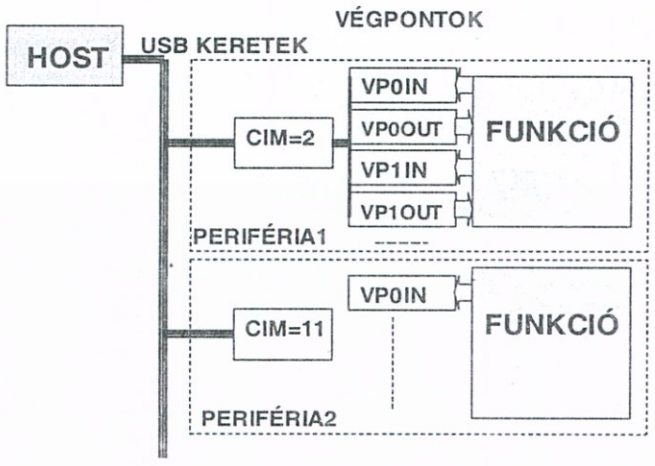
| # | Vezeték szín | Funkció |
|---|---|---|
| 1 | Piros | Vbus(5V) |
| 2 | Fehér | D- |
| 3 | Zöld | D+ |
| 4 | fekete | Gnd |




*4-20. ábra USB csatlakozó kiosztás*

### Az USB fizikai felépítése:

Az USB busz fizikailag négy vezetékől áll: +5V, GND tápvezetékek, illetve a csavart érpárú D+, és D- jelvezetékekből. Minden buszról táplált eszköz 100 mA-el terhelhető, de az eszköz – a hosztnak szóló külön kérésre – 500 mA-es tápáramot is kérhet, ha ezt a hoszt tudja biztosítani. Ha nem tudja, akkor a kérő eszközt letiltja. A hub és a periféria, illetve a hubok közötti maximális kábelhossz 1,5 Mbájt/mp sebességű perifériák esetén 5 méter lehet, míg nagysebességű átvitel esetén csak 3 méter. Két csatlakozó típust használnak: a téglalap alakú **A típust** a hoszt oldalon, illetve a négyzetes kialakítású **B típust** a perifériáknál. A szabvány még a vezetékek színéről is rendelkezik.




*4-21. ábra USB végpontok*

A bitek fizikai átvitele **NRZI** (Non Return to Zero Inverted) kódolással továbbítódnak a kábelen.

A megfelelő jelszinkronizáció érdekében öt egymást követő 0 után a küldő automatikusan beszúr egy 1-t az adatfolyamba, s azt a fogadó oldal az adat tárolásakor - szintén automatikusan - figyelmen kívül hagyja. (bit stuffing). Az USB busz nem 0, +V feszültségszinteket használ. Az, hogy egy bit 0, vagy 1, azt a D+ és D- vezetékek közötti feszültség hordozza. Ha (D-)-hoz képest D+ feszültsége nagyobb, mint 200 mV, akkor egyik állapot, fordított esetben a másik állapot áll fenn. Ezeket szokták **J ill. K állapotoknak** nevezni. Speciális események (pl. RESET) jelzésére szolgál egy olyan speciális vezeték állapot (neve: **SE0**), amikor mindkét vonalat, bizonyos időre 0 feszültségre húzzuk le.

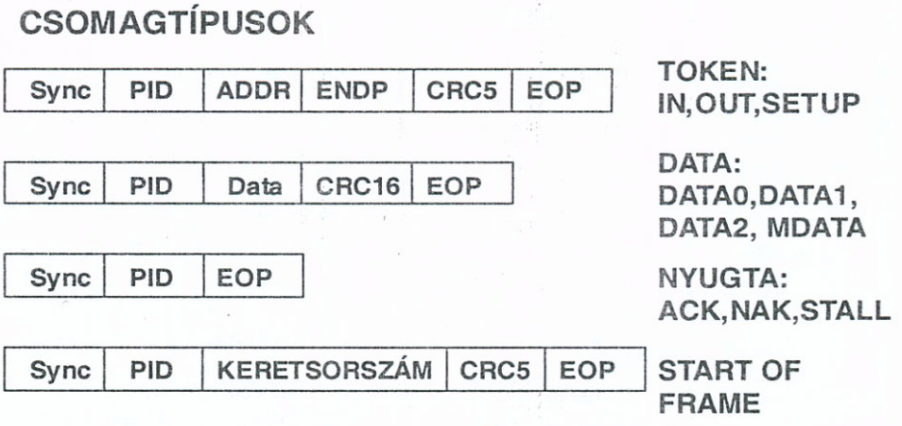
### USB végpontok

A perifériák a megszokott portoknak tekinthető, ún. végpontokon keresztül küldenek és fogadnak adatokat. Egy perifériának maximum 16 számozott (0-15) be- ill. kimeneti végpontja lehetséges, ezek közül a 0-ás címün keresztül történik a vezérlésátvitel, vagyis ezt mindig ki kell alakítani.

Az USB hoszt minden végponthoz definiál egy, a végpont által kezelt adatforgalom típusának megfelelő kommunikációs csatornát; ezt az USB szóhasználatában csőnek (**pipe**) nevezik. A cső tulajdonképpen a rendelkezésre álló sávszélességből (a másodpercenként átvitt bitek számából) az átvitel típusának lefoglalt rész. Az USB eszköz inicializálásakor, a 0. végponton át folytatott párbeszédben közli a hoszttal többi végpontjának jellemzőit, köztük a nekik szükséges sávszélességet is.

### Csomagtípusok

Az USB buszon megvalósított átvitel jellege kétféle átviteli módszer használatát igényli: **az adatokat csomagokban (packet) továbbítja, és 1 ezred másodperc (1msec) hosszúságú időtartományokra - keretekre - osztja fel a működési időt.** A következőkben összefoglaljuk a keretben küldött csomagtípusokat, amit a csomagban lévő **PID** mező (ld. később) azonosít. Mivel ez 4 bit hosszúságú, ezért 16 fajta csomagtípus lehetséges. A csomagokban használt mezők:




*4-22. ábra USB csomagtípusok*

**Sync** – szinkronizáló bitsorozat. Minden csomag ezzel kezdődik: 8/32 bit hosszú lassú/gyors átvitelnél.

**PID** – packet identifier: csomagtípus azonosító. 4 bit hosszúságú, de 8 bitessé van alakítva: az alsó 4 biten elhelyezett PID-nek a felső négy bit az inverze.

A következő PID típusok vannak: **token** – kommunikáció létrehozása, **data** - adatátvitel, **nyugta** – adó/vevő közötti megbízható átvitel, speciális – pl. **EOP** - a csomag vége a keret utolsó csomagja.

**ADDR** – 7 bites eszköz cím, a bejelentkezéskor, még címet nem kapott eszköz ADDR=0-át küld vissza

**ENDP** - végpont sorszáma (4 bit), vagyis egy eszközben maximum 16 végpont lehetséges.

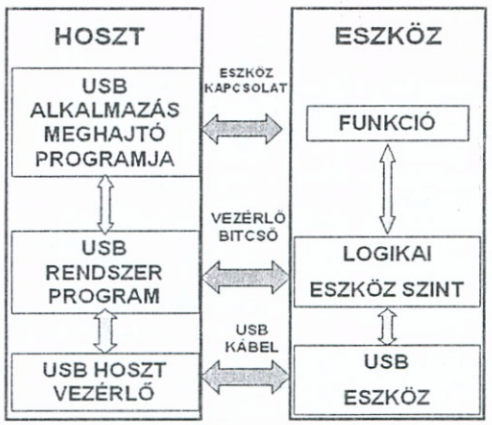
**CRC** – ciklikus redundancia kód, hibaellenőrzésre. 5 bit hosszúságú token csomagnál 16 bit hosszúságú adat csomagnál.

**EOP** – End Of Packet=csomag vége, **SE0** vonalállapot 2 bit idejére

### Időkeretekben történő csomagátvitel

Mivel az 1 msec hosszúságú keretben kell a csomagoknak elférni, ezért a hoszt az adatforgalom szervezését két fontos jellemző alapján végzi: minden átviteltípusra nézve korlátozva van az egy csomagban, illetve az egy keretben küldhető adatok mennyisége. Például:

- Vezérlésátvitelkor az adatcsomag mérete legfeljebb 64 bájt lehet, és egy keretben legfeljebb 13 ilyen csomag, azaz 832 bájt küldhető, vagyis másodpercenként 832 ezer bájt.
- Valós idejű átvitelben a maximális adatcsomagméret 1023 bájt, és egy keretben egy ilyen csomag lehet, emiatt a legnagyobb adatátviteli sebesség tehát 1023 ezer bájt másodpercenként.
- A szintén valós idejű megszakításátvitel esetén a maximális csomagméret 64 bájt, és ilyenből 19 lehet egy keretben; vagyis az elérhető sebesség 1216 ezer bájt/másodperc.




*4-23. ábra USB kommunikáció*

Mivel több, valós átvitelt használó periféria csatlakoztatásakor elfogyhat a sávszélesség (azaz „betelhet" a keret), ezért ilyenkor a hoszt, miután megkapta a paramétereket az új végpontoktól, nem definiál nekik csövet, és hibajelzést küld a képernyőre: az új periféria használatához valamelyik régit ki kell kapcsolnunk.

A sávszélességgel való gazdálkodást a hoszt a végpontra küldött engedélycsomaggal (tokennel) tartatja be: az egyes perifériák csak ezzel a kapott engedéllyel használhatják az USB buszt.

Ahogy azt már a csomagtípusokat bemutató ábrán láttuk, az 1 msec-os időkeretben küldött minden csomag, szinkronizáló **Sync** mezővel kezdődik, majd egy, a csomag típusát azonosító **PID** (packet identifier) mező következik. **Token csomag** küldése esetén ezt követi a periféria címe 7 biten (**ADDR**), a periférán belüli végpont sorszám (**ENDP**) 4 biten, mad egy 5 bites hibavédelmi CRC kód.

Például egy **IN token** érkezése után, a végpontnak a hosztnak adatokat kell küldenie. A hoszt, minden időkeret kezdetét, a hubok és a perifériák időzítésére szolgáló **SOF** (start of frame) csomag kibocsátásával kezdi, majd a végpontok átviteltípusa szerint sorban küldi nekik az **IN** (a hoszt adatot vár) vagy az **OUT** (a hoszt adatot küld)

tokent. **SETUP** token után, a hoszt a végpontot beállító parancsot küldi el adatcsomagként.

Minden adatcsomagot - leszámítva a valós idejű átviteléit - háromféle nyugtacsomag valamelyikével nyugtázni kell:

**ACK** - az adat problémamentes fogadását jelzi,

**NAK** - jelzi, hogy a periféria nem tud adatot küldeni vagy fogadni, mert például a nyomtató még nem kész,

**STALL** - a periféria nem tudja kezelni a hibát, ahhoz a hoszt - vagy a felhasználó - beavatkozása szükséges.

A nyugtacsomagot kivéve, minden csomag végén van hibaellenőrző CRC kód; ha azt a címzett vételkor hibásnak találja, nem küld vissza nyugtacsomagot. Mivel összesen 10-féle csomag létezik, a PID mezőt a CRC képzésnél nem használjuk: a PID négy bites, a második négy bitben az előzők negáltja van, azaz a PID önellenőrző.

Az esetleges nyugtavesztés felismerése miatt az adatcsomagoknak kétféle PID azonosítójuk van: ez a **DATA0** és a **DATA1**. Ezek, ha több csomagból álló sorozatról van szó, felváltva kerülnek bele a csomagokba. Ha például, egy DATA0 adatcsomag rendben megérkezik, de a visszaküldött ACK nyugtája valamilyen hiba miatt elvész, ekkor a küldő, mivel nem jött nyugta, újra elküldi ugyanazt a csomagot, de megint DATA0 jelzéssel. Mivel a fogadó tudja, hogy most DATA1-nek kellett volna következnie, rájön, hogy ez nem más, mint az előző ismétlése. Ezért eldobja, és újra küld egy ACK nyugtát.

## 4.5 CAN busz

Az autókban egyre jobban terjedő A „Controller Area Network" (CAN) kommunikációs rendszert, a gépjármű kiegészítőket gyártó német Robert Bosch cég definiálta az 1980-as évek közepén azzal a céllal, hogy növeljék a gépjárművek megbízhatóságát, üzemagyagtakarékosságát, komfortfokozatát úgy, hogy eközben a kábelezés is egyszerűsödjön.

Az eredeti CAN protokoll az OSI modell alsó két réteg (fizikai és adatkapcsolati) funkcióinak csak egy részét valósítja meg.

A buszra kapcsolódó egységeket csomópontnak (NODE) hívják. Az azonos buszon elhelyezkedő csomópontok azonos jogokkal rendelkeznek.

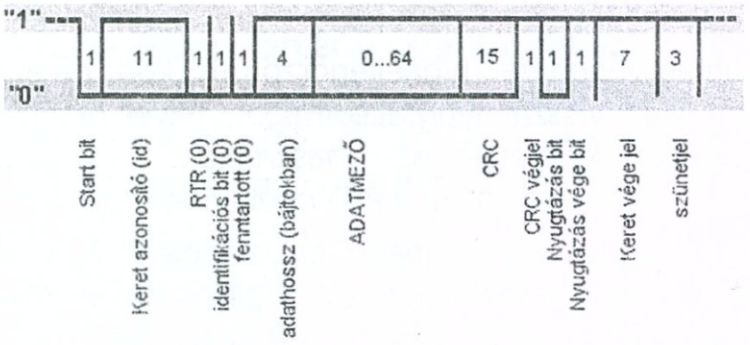
A CAN-buszon **üzenetalapú kommunikáció** folyik. Címek nincsenek, ha egy csomópont üzenetet küld a buszra, azt minden másik csomópont megkapja, és az üzenet fejlécében lévő azonosító (ID) alapján eldönti, hogy feldolgozza-e az adott üzenet adatbájtjait, vagy figyelmen kívül hagyja az üzenetet. Mivel az azonosító bitjei képzik az üzenetek első bitjeit, egyidejű adáskezdés esetén a versenyt az a csomópont nyeri, amelyik kisebb értékű azonosítóval rendelkezik.

Legyen két csomópont, A és B. A azonosítójának első 4 bitje: 1011. B-jé: 0111. Elindítják a kommunikációt: A és B elsőnek küldött két bitje 11, és mindegyik NODE ezt is olvassa. Harmadik bitként A 0 küld ezt is olvassa, mert a 0 az erősebb. B 1-et küld, és 0-át olvas. Érzékelve ezt B befejezi a kommunikációt, és A küldheti a több bitjeit tovább.

Ez a kommunikáció lehetővé teszi, hogy egy korábban kiépített buszra bármikor újabb

csomópontot fűzzünk fel, vagy egy régi csomópontot távolítsunk el a rendszer átprogramozása nélkül.

A CAN-busz aszinkron soros adatátvitellel rendelkezik, tehát nincs órajel. A bitek szinkronizálása az üzenetek start/stop keretezése mellett egy ún. „bit-stuffing" (bitbeszúrás) módszerrel történik, ami azt jelenti, hogy egymás után maximum öt bit lehet azonos polaritású. Ha egymás után ötnél több azonos bitet kell átküldeni, akkor az adó az ötödik után beilleszt egy ellentétes polaritású bitet, a vevő ehhez szinkronizálja a belső órajelét. A beillesztett bitet persze a vevő eltávolítja a hasznos adatbitek közül (destuffing).




*4-24. ábra CAN adatkeret felépítése*

Ha egy csomópont egy üzenetet kíván a buszra juttatni, akkor először egy meghatározott ideig figyelnie kell a buszt, hogy egy másik adó nem kezd-e adni, tehát az adás megkezdése előtt a busznak egy meghatározott időre passzív állapotban kell lennie (vivő figyelés). Ha ez az idő letelt, elkezdheti az adást, az összes többi csomópont pedig vevővé válik, és az üzenet vételének befejezéséig nem kezdhetnek adásba.

A CAN rendszer előnye, hogy lehetnek olyan csomópontok, amelyek rendszeresen küldenek üzeneteket, és lehetnek olyanok is, amelyek csak akkor küldenek üzenetet, ha azt egy másik csomópont kéri. Például egy gépjármű biztonsági rendszerében a kritikus érzékelők (mint pl. a légzsák) periodikusan küldenek üzeneteket, melyeket a központi egység dolgoz fel. A kevésbé kényes érzékelőknek (pl. olajnyomás, akkumlátor állapot) pedig a központi irányítóegység kérdezi le az állapotát.

**Az üzenet-alapú kommunikáció egyik legnagyobb előnye, hogy egy korábban kiépített buszra bármikor újabb csomópont fűzhető fel, vagy távolítható el a rendszer átprogramozása nélkül.**

## 4.6 Széles sávú átvitel telefon- és kábel-TV hálózaton

A **hagyományos réz érpáron,** a meglévő telefonhálózat felhasználásával is lehet **széles sávú hálózati szolgáltatásokat** nyújtani. Ilyen megoldás az **ADSL** *(Asymmetrical Digital Subscriber Line = Aszimmetrikus digitális előfizetői vonal)*, mely kétirányú, nagysebességű, digitális hálózati kapcsolatot biztosít.

Az ADSL, mint technológia, a meglévő rézvezetékes hálózaton nyújt emelt szintű szolgáltatást a már bemutatott hagyományos távbeszélő szolgáltatás (PSTN) illetve annak digitális változata, az ISDN mellett. A telefonvonalon történő széles sávú átvitelkor kétirányú adatáramlás folyik a felhasználó és az összeköttetés másik végpontja (a hálózat) között. Mivel az információt a felhasználó (az előfizető) igényli, ezért a két irány eltérő sebességű:

- az előfizető felé (Downstream - lefelé irány) Mbit/s nagyságrendű,

- míg a hálózat felé (Upstream - felfelé irány) néhány száz kbit/s-os az adatátviteli sebesség.

Ismétlésképpen nézzük meg, hogy a nagysebességű adatátvitel mellett melyik két alapszolgáltatás működhet, (vagy az egyik, vagy a másik).

**Hagyományos távbeszélő szolgáltatás (PSTN)**

Az analóg távbeszélő hálózat a 300Hz-3400 Hz közötti frekvenciatartományban képes beszéd, illetve kis sebességű modemmel (legfeljebb 56kbit/s) adatátvitelt biztosítani. Ehhez az előfizetőnél hagyományos telefonkészülék, fax vagy modem szükséges, a hálózatban pedig többek között telefonközpontok.

**ISDN2 alapcsatlakozás (ISDN)**

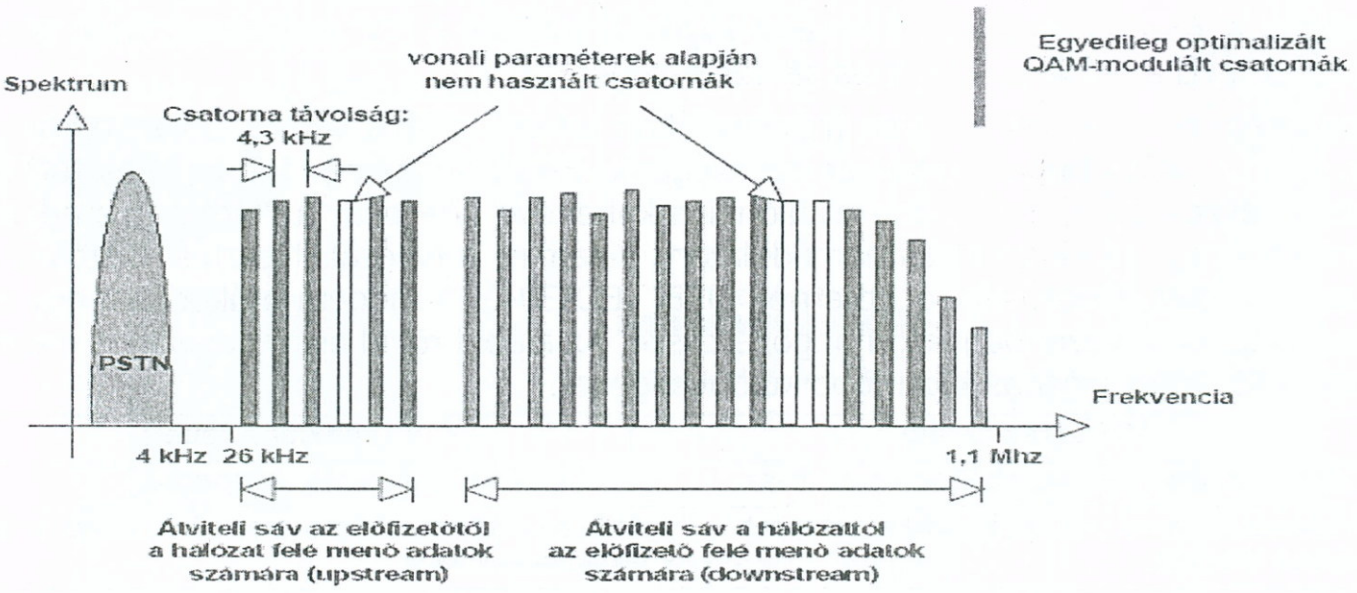
Az ISDN alapcsatlakozás a már ismertetett digitális átviteli módszer segítségével biztosít:

- két 64 Kbit/s sebességűfelhasználói (2 B),
- valamint egy 16 Kbit/s sebességű jelzésátviteli (D) csatornát.

Mindkét **B csatorna** alkalmas kapcsolt beszéd- és adathívások bonyolítására. A két csatorna, egymástól függetlenül, egy időben is használható (például két telefonhívásra, telefonálásra és faxolásra, telefonálásra és adattovábbításra, stb.), és együttesen is igénybe vehetők (képtelefonálásra, 128 Kbit/s sebességű állománytovábbításra). A **D csatorna** szállítja a jelzéseket a végberendezések és a központ között.

**Tehát az ADSL lényege, hogy a meglévő távbeszélő (vagy ISDN) alapszolgáltatással párhuzamosan, ugyanazt az előfizetői érpárt felhasználva, kapcsolatot biztosítson a nagysebességű hálózatokhoz (pl. Internet-hez).**

Hogyan lehet egyszerre, egymástól függetlenül telefon (vagy ISDN) szolgáltatást és kétirányú nagysebességű adatátvitelt biztosítani? A megoldás a már tanult frekvencia multiplexelés. Az alapsávi analóg (PSTN) vagy digitális átvitel mellett, a vonalon ma-



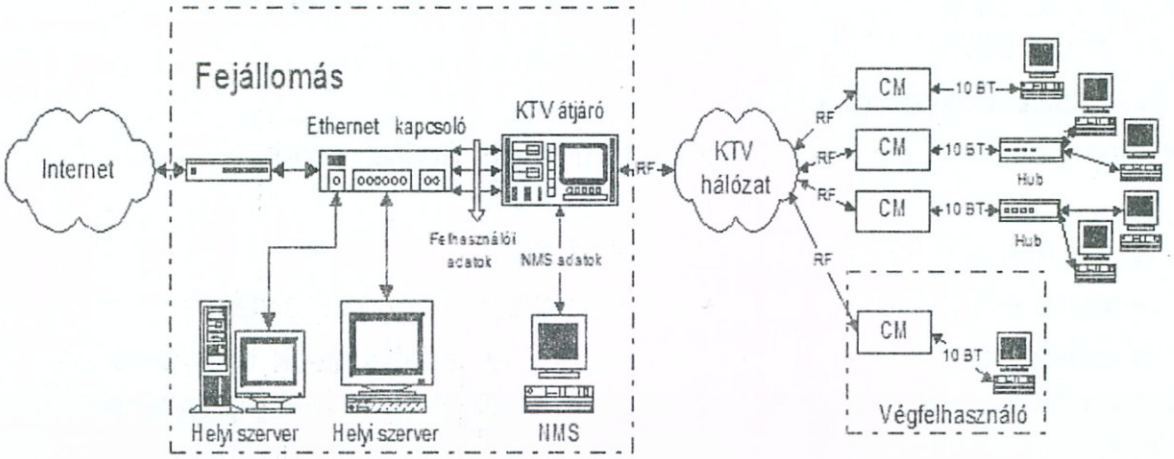

*4-25. ábra ADSL frekvencia kiosztás*

gasabb frekvenciájú jeleket is átviszünk. Az alapsávi, és a magasabb frekvenciájú jelek szétválasztását az előfizetőnél elhelyezett elektromos szűrő biztosítja.

Ez a szűrő, (angolul splitter) választja el egymástól a távbeszélő, és a gyors adatátviteli szolgáltatást.

## 4.6.1 Az ADSL és a kábeltelevízió kapcsolata

Adatátvitelre nem csupán a távbeszélő szolgáltatást lehetővé tévő rézvezető, hanem a kábeltelevíziós hálózat koaxiális kábele is felhasználható.




*4-26. ábra* ***KTV – Internet*** *(CM-kábelmodem, NMS hálózatkezelés)*

Kábeltelevíziós internet hozzáférés, kétirányú adatkommunikációra alkalmassá tett kábelhálózaton érhető el. A kábeles tévéadások a teljes, koaxiális kábelen továbbítható frekvenciatartományon belül, sávokat foglalnak el. Egy-egy TV-adás videó- és hangjeleivel a csatorna vivőfrekvenciájának az amplitudóját modulálják, és minden csatorna az 54 MHz-890 MHz közötti tartományban mindössze egy 8 MHz-es sávszélességet foglal el.

A kábeles internetezés is, ebben a tartományban foglal el a TV adásoknak szánt, de kihasználatlan sávokat. A feltöltés céljaira 5-65 MHz tartományt használják, míg a letöltés a 65-850 MHz-es tartomány zajlik. A letöltés a TV adásokhoz hasonlóan 8 MHz-es, míg a feltöltés 2 MHz-es sávszélességen zajlik.

Hogy képesek legyünk az ilyen frekvenciákon érkező adatok vételére, illetve küldésére, egy ún. **kábelmodem**-re van szükségünk. Ez tartalmaz egy tunert (illesztőt), egy jelszétválasztót (splittert), ami elkülöníti a kábeltévé adásokat az internetes adatcsatornáktól, egy modulátort és demodulátort, valamint a hálózati illesztést biztosító hálózati csatolót (lehetőségek: Ethernet, USB, IEC1394). A modem működését egy mikroprocesszoros vezérlőegység hangolja össze. Az alábbi rövid összehasonlítás megmutatja a két megvalósítás közti főbb különbségeket.

| ADSL (réz érpár) | KTV (koaxiális kábel) |
|---|---|
| Célja a kétirányú telekommunikáció biztosítása. | Célja a televíziós műsorok szétosztása. |
| A távbeszélő vonal (réz érpár) sokkal szélesebb körben elterjedt. | Kevésbé elterjedt, elsődlegesen sűrűn lakott területekre összpontosul. |
| Vonali sávszélesség minden egyes előfizetőnél teljesen kihasználható. | Csak néhány csatornát használhatunk adatátviteli célokra. |
| **A központ felől minden előfizető adott sávszélességű külön csatornát kap.** | **Többen osztoznak a sávszélességen.** |

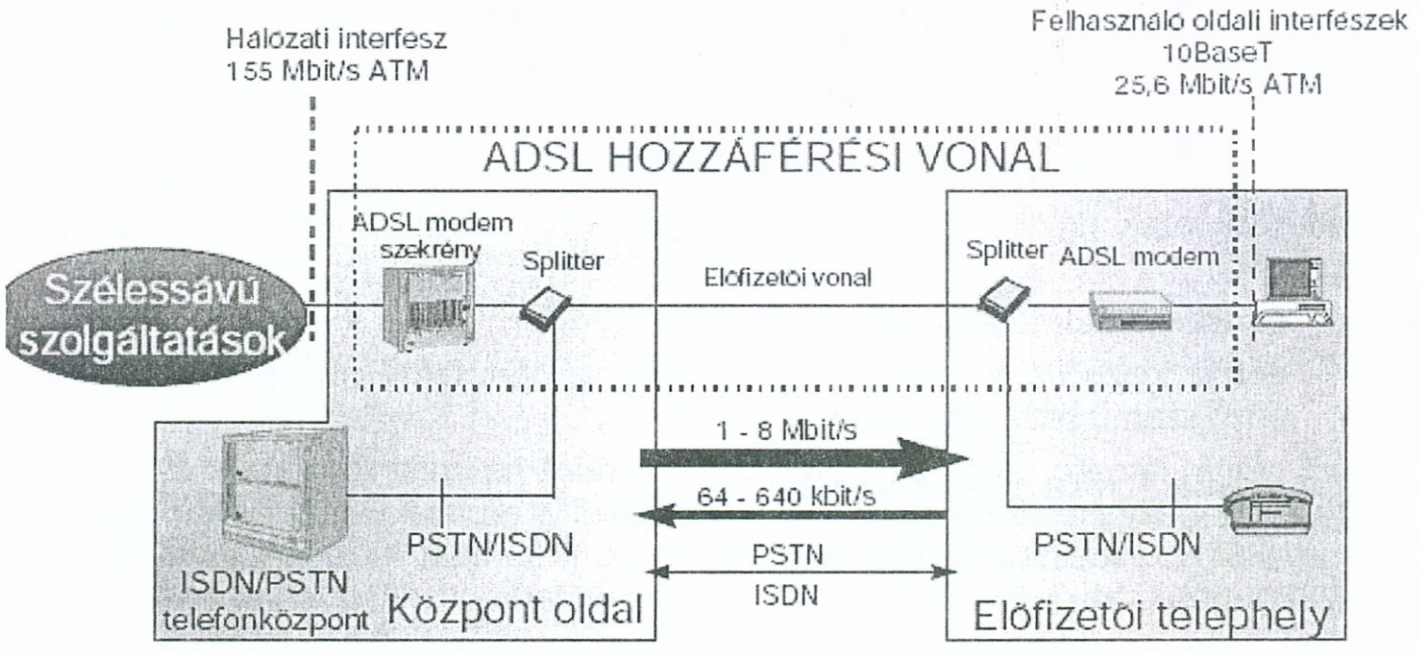
A utolsó sorban leírtak döntik el, hogy melyik megoldás az előnyősebb: a központhoz külön-külön kapcsolódó telefonelőfizetők külön, adott sávszélsségű csatornákat kapnak, míg a KTV esetén osztozni kell a sávszélességen!

## 4.6.2 ADSL rendszertechnika

Az ADSL átviteli rendszer egy olyan nagysebességű digitális hálózati hozzáférés, mely átviteli közegként a sodrott réz érpárt használja, és a két irányban eltérő adatátviteli kapacitású. A hálózattól a felhasználó felé (Down-stream) maximálisan 6-8 Mbit/s, míg a másik irányban (Up-stream) 1Mbit/s átviteli kapacitású csatorna áll rendelkezésre. A *4-27. ábrán* az ADSL hozzáférés rendszertechnikai képe látható. A rendszer minden esetben egy modem párból tevődik össze, ahol az egyik modem az előfizetőnél, míg a másik a helyi telefonközpontban található.

Az előfizetői vonalon, frekvenciában külön választva, viszik át a hagyományos PSTN/ISDN jeleket és a számítógép adatait. Ehhez a vonal mindkét végén egy frekvenciasáv szétválasztó, un. splitter eszközre van szükség. A szétválasztás frekvenciája (tehát a splitter típusa), attól függ, hogy hagyományos telefon vagy ISDN szolgáltatást nyújtanak az előfizetőnek. A splitterhez csatlakoznak: az ADSL modemek (ezek felelnek a kétirányú gyors adatátvitelért), a távbeszélő/ISDN szolgáltatás nyújtásához szükséges telefonközpont / távbeszélő készülék.

Az előfizető oldali ADSL modemhez számítógépet (vagy speciális kiegészítővel ellátott televíziót), lehet csatlakoztatni hagyományos 10 Mbit-es Ethernet, vagy 25,6 Mbit-es ATM porton keresztül. Az előfizető felől jövő illetve felé menő adatok a szélessávú hálózaton keresztül továbbítódnak. Ezek az adatok hordozzák a speciális szolgáltatá-

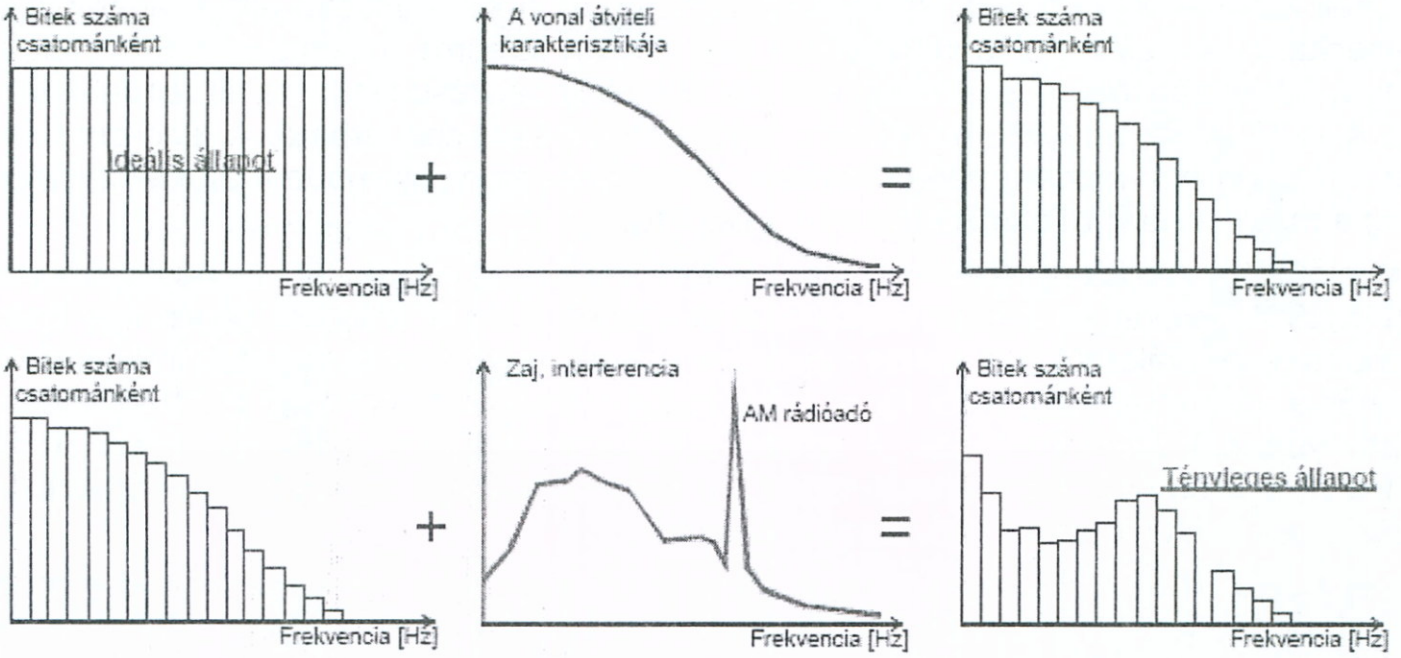



*4-27. ábra ADSL rendszer felépítése*

sok jeleit (gyors Internet elérés, vállalati hálózat kihosszabbítás, otthoni interaktív videózás, stb.).

## 4.6.3 ADSL vonali átvitel

Az ADSL rendszerben alkalmazott kódolási módszer a **DMT moduláció**. Már a 19,2 kbit/s-os modemekben is használt DMT (Discret MultiTone - Diszkrét több vivős) moduláció a fejlett digitális jelfeldolgozás eredménye. Lényege, hogy az átvitelre használt




*4-28. ábra ADSL frekvencia kiosztás - alkalmazkodás*

frekvenciasávot több, egymás utáni kis sávszélességű csatornára osztjuk (4 kHz), és azokban külön-külön viszünk át hasznos információt. A *4-28.ábra* mutatja, hogy a rendszer két dologhoz is alkalmazkodik:

- vonal átviteli karakterisztikájához
- a vonalon megjelenő zajokhoz, zavarokhoz.

A modemek ezek figyelembe vételével állítják be a csatornánként elérhető átviteli sebességet:

- minden csatornánál lemérik a jel/zaj viszonyt,
- ehhez igazodva a modemeknél már megismert **QAM-moduláció**-val különböző számú bitet visznek át.

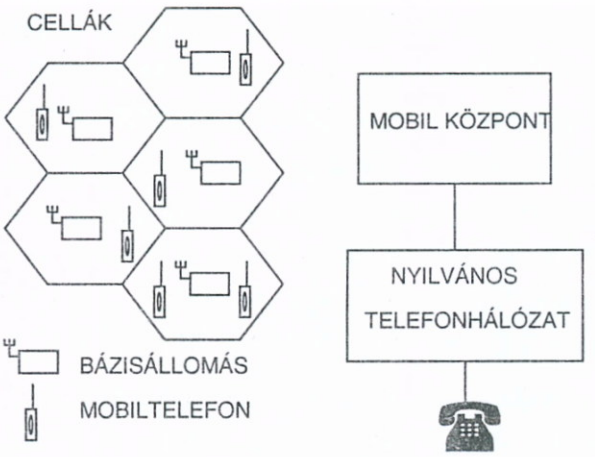
A *4-25. ábrán* látható, hogy az egyes, adatátvitelre használt csatornák és a bennük átvitt adatok, milyen módon töltik ki a rendelkezésre álló frekvenciasávot. Az is látható, hogy PSTN távbeszélő szolgáltatás esetén 26 kHz fölött kezdődik az ADSL spektrum (ISDN-nél ez az érték 130 kHz), és 1,1 MHz-ig tart.

A rendszer felépítése olyan, hogy az ADSL átviteli rendszeren belül a hasznos információt ATM cellákba csomagolva viszik át.

Hálózati oldalon a 155 Mbit/s ATM jelfolyam jelenik meg. Az ebbe a jelfolyamba statisztikusan multiplexált ATM cellákat egy menedzsment rendszer segítségével a megfelelő kiszolgálóhoz továbbítják előre meghatározott PVC-k vagy SVC-en keresztül.

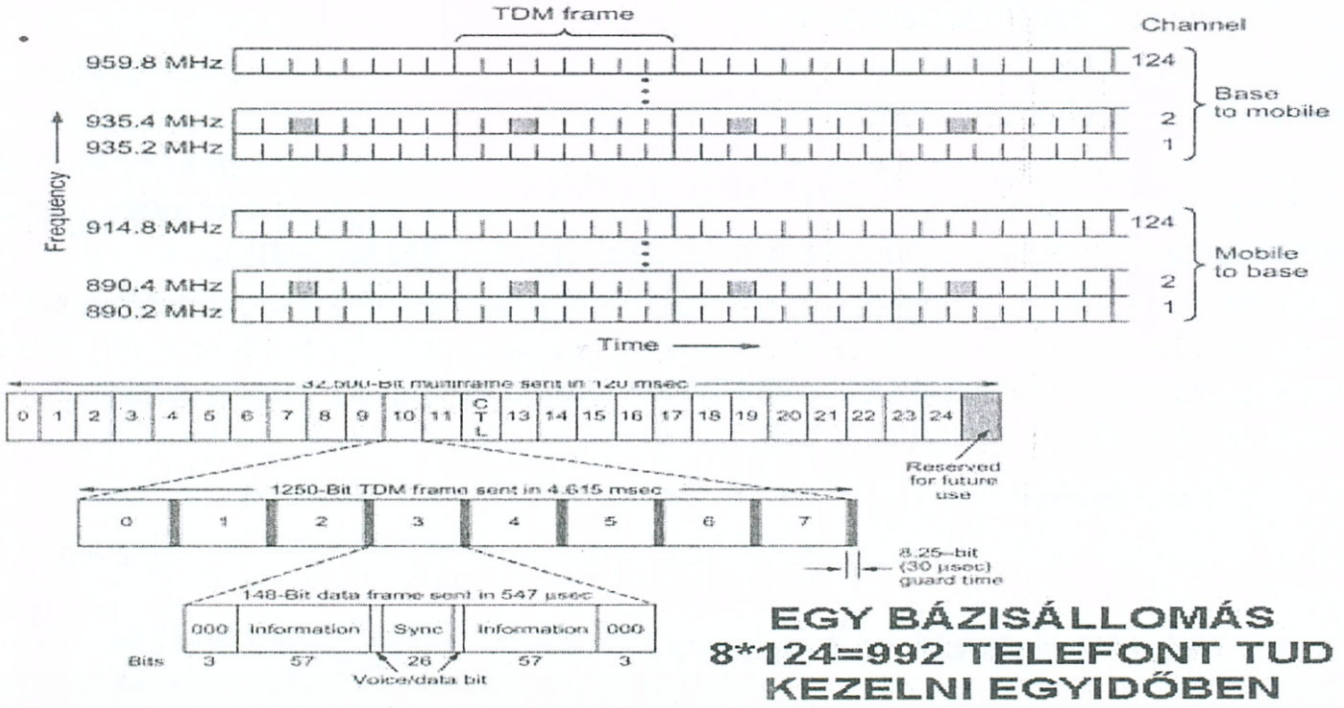
# 4.7 Analóg+digitális átvitel: mobil telefonok (GSM)

GSM=Global System for Mobile Communication (mobil kommunikációs világrendszer) Vivőfrekvencia: 450, 900 vagy 1800 MHz-et (Usa,Japán: 1900MHz), **több frekvenciasávon időmultiplexelést végző csatornákat használ** (egyidejűleg többen is beszélhetnek ugyanazon a csatornán).




*4-29. ábra: A cellás rádiótelefon rendszer felépítése*

A cellás szerkezetű rádiótelefon rendszerek az igényeket a rendelkezésre álló frekvenciatartomány kihasználtságának növelésével elégítik ki. A cellás technika a cellaosztáson és a frekvenciák ismételt felhasználásán alapszik. A területet kisebb részekre osztják. A cellákon belül egy központi rádióállomás tartja a mozgó előfizetőkkel a kapcsolatot. Az URH sávban a hullámterjedés sajátosságai lehetővé teszik, hogy egy bizonyos távolság felett újra fel lehessen használni a frekvenciasávot. Így ugyanaz a frekvencia egyidejűleg több, egymástól megfelelő távolságban lévő cellában is kioszthatő.




*4-30. ábra Mobiltelefon csatorna kezelés*

Például az ábrán a 935.4 MHz-es frekvenciasáv besötétített időszeleteit használja egy mobiltelefon a báziállomástól történő vételre, és a 890.4 MHz –es frekvenciasávot adásra. Az időszeletben lévő 148 bites információ tartalmazza az adat vagy hanginformációt. 8 időszelet együtt egy TDM csoportot (keretet alkot), amelyben 1250 bit van.

A felhasználók egy cellán belül a helyi bázisállomáson keresztül tartják a rádiós kapcsolatot. A bázisállomás hálózat a mobil központhoz csatlakozik rádiós vagy vezetékes összeköttetéssel. A mobil központ feladata a cellás rendszer működésének vezérlése, és a nyilvános postai távbeszélő hálózathoz való illesztése.

Előfordulhat, hogy éppen a folyamatban lévő beszélgetés közben lép át a felhasználó egy cellahatárt. A modern rendszerek gondoskodnak arról, hogy ilyenkor az összeköttetés ne szakadjon félbe. A hívást átkapcsolják a következő cella egy csatornájára. Ennek feltétele, hogy a fogadó cella rendelkezzen kioszthatő beszédcsatornával. Ezt a váltást handovernek vagy handoffnak nevezzük.

A cellák 200 kHz széles frekvenciasávú duplex csatornával rendelkeznek, amelyek száma maximálisan 200 lehet. A *4-30. ábrán* egy ilyen, 124 csatornát tartalmazó bázisállomás frekvenciakiosztása látható, amely csatornák mindegyike időosztásos multiplexeléssel egyidejűleg 8 összeköttetést támogat. Ilyen módon a cellában összesen 124*8=992 mobil lehet aktív, de ezek közül számos a szomszéd cellák közelsége miatti zavarás miatt nem használható.

A GSM rendszer három alrendszerből tevődik össze:

1. Hálózat alrendszer
2. Bázis állomás alrendszer
3. Üzemeltetést támogató alrendszer

### *Előfizető- és készülékazonosítás*

A mobil készülékek folyamatosan sugároznak egy, csak az adott készülékre jellemző kódot. Ezt a Berendezés Azonosító Regiszter dolgozza fel.

Az előfizetők azonosítására az ún. SIM-kártyák szolgálnak (Subscriber Identity Module, Előfizetői Azonosító Kártya). Ez az előfizető és az érvényben lévő szerződés adatait tartalmazza (díjosztály, megengedett bolyongási határ, titkos kód, stb.).

Hiba esetén a PIN kód megadását, rendszertől függően, párszor meg lehet ismételni, majd az Illetékességet Meghatározó Központ letiltja a készüléket, amit csak szerviz tud újraéleszteni. A készülék és a SIM-kártya tulajdonlása ilyen módon elválasztható, maga az előfizetés a kártyához kötött.

### *1. Hálózat alrendszer*

Ez az alrendszer végzi a kapcsolásokat, és az adatbázisok kezelését. A tárolt adatbázisok:

**Honos Helyzetmeghatározó Regiszter (HHR):** Az ország területekre van osztva, minden területnek van egy HHR-e, amely az adott régióban lakó előfizetők készülékeinek adatait tartalmazza. Így minden mobil készülék egy meghatározott HHR-hez tartozik (ha az előfizető elköltözik és viszi a készüléket is, az információk mindaddig itt maradnak, amíg az előfizető ezt be nem jelenti — ami érdeke, mert a távoli kapcsolások igen megdrágíthatják a telefonálást).

**Látogató Helyzetmeghatározó Regiszter (LHR):** Az adott területen tartózkodó, de nem ott honos előfizetőkről, készülékeikről tartalmaz ideiglenes adatokat.

**Illetékességet Megállapító Központ:** A biztonsági, és titkosítási célokat szolgáló kódokat kezeli.

**Berendezés Azonosító Regiszter:** A legálisan regisztrált mobil készülékekről rendelkezik információval.

Az alrendszer része a Mozgó Kapcsoló Központ, amely a mobil állomásokat kapcsolja egymással, illetve a hagyományos telefonrendszerrel. Ezek a központok hívásfelépítéskor használják az előbb említett adatbázisokat.

### *2. Bázis állomás alrendszer*

A Bázis Adó-Vevőket (gyakorlatilag a telepített adó-vevő antennák) és azok vezérlőit foglalja magában. A vezérlők felügyelik az antennák és a Mozgó (MOBIL) Kapcsoló Központok közötti összeköttetéseket, valamint figyelemmel kísérik a mobil állomások mozgásait, amikor egyik celláról a másikra kell kapcsolni ezeket. Egy vezérlő több antennát is elláthat.

A Bázis Adó-Vevők úgy vannak telepítve, hogy méhsejtszerűen lefedjék a teljes területet (természetesen ezért a teljesítményeket korlátozni kell, különben zavarnák egymást). A méhsejteket cellának nevezzük.

### *3. Üzemeltetést támogató alrendszer*

Ez a program, amely vezérli és felügyeli a teljes GSM- rendszert.

**Hálózat felügyelet:** Ha egy készülék meghibásodik, vagy illetéktelenül akarják használni, riasztást küld a központnak. Az operátor egy interaktív grafikus kijelzőn figyelheti az eseményeket.

**Konfigurációs management**: A GSM-rendszer konfigurálását végző software. Újrakonfigurálást kell végezni minden új előfizető belépésénél, minden új antenna, Mozgó Kapcsoló Központ üzembe helyezésénél.

**Szolgáltatási minőség:** A rendszer állandóan gyűjti és kijelzi a hálózat információit: a kikapcsolt készülékek számát, az elveszett hívásokat, a vonalak terhelését, a forgalmas órákat, eredménytelen híváskísérleteket, egyszóval a hálózat rendelkezésre állását.

**Személyazonosító központ:** Az előfizetők azonosítását, a hálózatban való belépés engedélyezését végzi a SIM-kártyák alapján.

**Számlázó és adminisztrációs rendszer:** A Mozgó Kapcsoló Központoktól kapott forgalmi adatok alapján, valamint a SIM-kártyák adataiból elkészíti az előfizetőknek küldendő számlát.

**Biztonságkezelés:** Az előfizetők személyes adatainak, illetve a rendszer belső adatainak a védelme tartozik hozzá.

**Mobil készülékek**: Az előfizetőknél levő készülékeket adóteljesítmény szerint öt osztályba sorolják: 20, 8, 5, 2 és 0.8 W teljesítményűek vannak (az első két osztályt a gépkocsiba szerelt, az utolsó hármat a hordozható készülékek alkotják).

### *Gsm — hívásfelépítés*

**Hagyományos telefon hív mobil telefont:** A hívás ahhoz a Mozgó Kapcsoló Központhoz érkezik, amely a hívóhoz legközelebb esik. Ez a hívást elirányítja ahhoz a

Kapcsoló Központhoz, amelynek Honos Helyzet-meghatározó Regiszterében a hívott előfizető szerepel.

- Ha ez a Regiszter érzékeli, hogy az állomás a területén van, a kapcsolat létrejöhet.
- Ha nem, a Központ a hívást átirányítja ahhoz a Kapcsoló Központhoz amelynek Látogató Helyzetmeghatározó Regiszterében szerepel a hívott állomás. Az megkeresi azt a Bázis Adó-Vevő állomást (méhsejtet), amelyiknek a területén van a hívott fél. A beszélgetés idejére a hívó és a hívott fél területén levő Mozgó Kapcsoló Központok között közvetlen összeköttetés létesül.

**Mobil telefon hív mobil telefont:**

A hívás annál a Mozgó Kapcsoló Központnál kezdődik, amelyben a hívó fél (vagy honosan, vagy csak látogatóként) szerepel. Az információ eljut a hívó Honos Kapcsoló Központjáig, amely az illetékesség megállapítása után visszaadja azt. Innen kezdve a hívásfelépítés folyamata azonos az előzővel.

**Mobil telefon hív hagyományos telefont:**

A hívás (az esetleges Látogató Központon át) eljut a Honos Központig, amely az illetékesség megállapítása után visszaadja azt oda, ahol a hívó fél tartózkodik. Ott a hívás kimegy a hagyományos rendszerbe, ahonnan az ottani szabályoknak megfelelően folytatja útját. Tehát a hívásfelépítés és a beszélgetés útvonala nem mindig egyezik meg. A hívó és a hívott félről is meg kell állapítani, hogy jogosan használja-e a hálózatot, és ezt csak a Honos Kapcsoló Központ tudja megtenni. Így a hívás felépítésekor ezt a Központot mindenképpen meg kell keresni. A beszélgetés idejére mindig a két legközelebbi (Honos vagy Látogató) Kapcsoló Központ között létesül összeköttetés.

**Bolyongás (roaming)**

A bolyongás (roaming) az előfizető Mozgó Kapcsoló Központok közötti mozgása. Ez lehet országon belül, de országok között is. A bolyongás műszaki feltétele, hogy a GSM-rendszer ki legyen építve. Kereskedelmi és jogi feltétele, hogy a honos és az

***Mobillemtan***

*A Westikett nyomán*

- Ne hivalkodjon mobiltelefonjával, ne hordja a készüléket kihívóan!
- Legyen mindig elérhető – közvetlenül, vagy hang, illetve írott üzenettel, de őrködjön mások nyugalmán! Társaságban csak akkor kezdeményezzen és fogadjon hívást, ha megbizonyosodott róla, hogy ez másokat nem zavar!
- A hívó fél mutatkozzon be, és kérdezze meg, alkalmas-e a hely és az időpont a beszélgetésre!
- Legyen tekintettel mások érzékenységére! Ha közlendője nem tartozik másra, vonuljon félre!
- Ne használja a telefont temetésen, egyházi szertartáson, egészségügyi intézményben - ha az a gyógyító munkát zavarhatja - színházban, moziban, kulturális rendezvényeken, konferenciákon, tömegközlekedési eszközökön! Ha fontos hívásra vár - például étteremben -, állítsa a telefont néma üzemmódba! A beszélgetést egy csendes, félreeső sarokban folytassa le, miután vacsorapartnereitől elnézést kért!
- Készüljön fel arra, ha a beérkező hívásokat nem tudja fogadni! Udvariatlanságnak számít, ha a készüléket akkor kapcsolja ki, amikor az meglepetésszerűen csörögni kezd. Ezt megelőzendő használja a hangpostát!
- Üzenethagyáskor ne feledje el a nevét és a hívószámát bemondani. Fogalmazzon tömören, pontosan!

illető rendszer között legyen érvényes megállapodás, valamint, hogy az előfizető rendelkezzen megfelelő SIM-kártyával.

### *Aktuális mobilos fogalmak: SMS, GPRS, WAP*

Ismeretes, hogy a mobil telefon képes rövid (max. 160 karakter) üzenetek küldésére és fogadására. Lehetséges ennél hosszabb üzenetek küldésére is, ilyenkor a telefon automatikusan darabolja az üzenetet, a másik oldalon pedig az üzenet összeillesztésre kerül.

Az **SMS (SMS=Short Message Service)** üzenetek a mobil hálózaton keresztül a rendszerben található központba (SMSC=Short Message Service Center) kerülnek, amely a cím alapján felveszi a küldeményt, majd a GSM kapcsolóközpont segítségével, az megpróbálja a megcímzett mobilra elküldeni.

Ha van elég hely a mobilban (két tárolóhely lehet: a készülék és/vagy a SIM kártya, akkor az üzenet letöltése megtörténik. Ha nincs hely, vagy ha a mobil nincs bekapcsolva, akkor az SMSC egyre ritkuló ütemben megpróbálja letölteni a levelet. Ha ez nem sikerül, akkor egy idő múlva azt jelzi, hogy nem tudja letölteni. Az SMS-el vezérlések kialakítása, rövid információk automatikus lekérdezése is megvalósítható, aminek nagy távlatai vannak.

Az előzőekben bemutatott összeköttetés alapú mobil kommunikációt kezdi felváltani a többszörös adatátviteli sebességet biztosító **GPRS (General Packet Radio Service)** csomagkapcsolt átvitel. Az ilyen elven működő hálózatban a telefonok mindig folyamatosan (on-line) kapcsolódnak a hálózatra, és csak egy-egy csomag vételéig terheli a hálózatot. Ilyen módon soha nem kell a kapcsolat kialakulására várni.

A **Wireless Application Protocol (WAP)** révén lehet az új generációs mobiltelefonokra az interneten lévő adatokat továbbítani. Elsődlegesen grafikai megjelenítést nem, vagy igen kis mértékben igénylő alkalmazások használhatók, többszörös menürendszer felhasználásával. Lehetséges banki információk, szolgáltatások használata, adatlekérdezések, információs adatbázisok használata.

## *Ellenőrző kérdések: 4. Fejezet*

1. Mutassa be az RS232C soros adatátvitelt! Mi a DCE és DTE?
2. Milyen adatokkal jellemezhető egy RS232C soros adatátvitel?
3. Mi a null modem?
4. Rajzolja le az A karakter átvitelekor kialakuló jelalakot!
5. Mutassa be, hasonlítsa össze a Az RS-449, -422, -423, és 485-ös szabványokat!
6. Mi a szimmetrikus, illetve aszimmetrikus átvitelek között a különbség?
7. Mi az X.21 interfész?
8. Mi az ISDN? Mi az alagondolata? Milyen szolgáltatásai vannak?
9. Mi az a bitcső, hálózati végződés? Milyen referenciapontok vannak a különféle eszközök között?
10. Milyen részekből áll egy ISDN interfész? Milyen keretformátumot használ?
11. Mi az ATM? Mi az alapgondolata?
12. Foglalja össze az USB busz legfontosabb jellemzőit!

13. Foglalja össze a CAN busz legfontosabb jellemzőit! Mi az üzenet alapú kommunikáció lényege? Miért nincs ütközés a CAN buszon?
14. Foglalja össze az ADSL átvitel legfontosabb jellemzőit! Miért aszimmetrikus?
15. Hogyan lehet egyszerre, egymástól függetlenül telefon (vagy ISDN) szolgáltatást és kétirányú nagysebességű adatátvitelt biztosítani?
16. Hogyan működik a kábelTV hálózaton történő hálózati adatátvitel?
17. Foglalja össze a mobiltelefon rendszer működését! Hogyan kezeli a beszédcsatornákat? Mi az a bázisállomás, illetve mobil központ?
18. A GSM rendszernek milyen alrendszerei vannak? Hogyan történik a kapcsolat létrehozása két előfizető között? (Hívásfelépítés különféle esetei.)

# 5. HÁLÓZATI RÉTEG

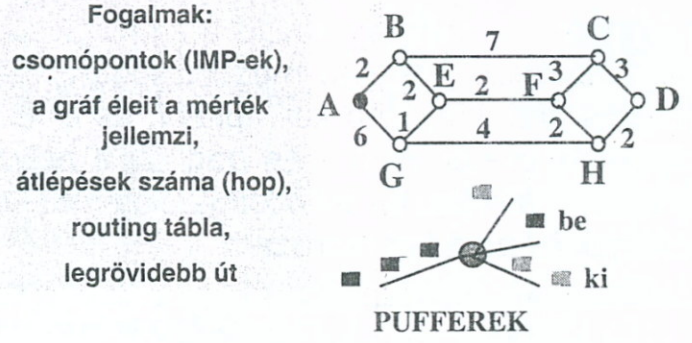
*Felelsz azért, hogy a sorban utánad álló is fel tudjon még szállni az autóbuszra.*

**Egy fizikailag összekötött hálózaton, az adatkapcsolati réteg kereteiben becsomagolva utazik az információ két szomszédos csomópont között.** Mivel a hálózatok fizikai rétege más és más lehet, ezért távoli csomópontok közötti kommunikációhoz egy – biztosítva minden hálózat számára az egységes adatáramlást - egy új réteg, és általa szállított adategység bevezetését igényelte, ez a **hálózati réteg** és az általa kezelt **csomag**ok. A hálózati réteg feladata a csomagok eljuttatása a forrástól a célig. A célig egy csomag valószínűleg több csomópontot is érint. Ehhez természetesen ismerni kell az átviteli hálózat felépítését, azaz a topológiáját, és ki kell választania a valamilyen szempontból optimális útvonalat.

A megvalósításnál figyelembe kell venni azt a tényt, hogy **alapvetően két eltérő hálózatszervezési módszer létezik: az egyik az összeköttetés alapú, a másik az összeköttetés-mentes.** Az összeköttetés alapú hálózatoknál az összeköttetést **virtuális áramkörnek (VÁ)** szokták nevezni. A forrás és a cél között felépült állandó úton vándorolnak a csomagok, de egy fizikai közeget egyszerre több virtuális kapcsolat használhat.

**Virtuális áramkörök** használatakor nem kell minden egyes csomagra forgalomszabályozási döntést hozni. A forgalom szabályozása az összeköttetés létesítésének a része, vagyis kiválasztásra kerül a forrást és a célt összekötő útvonal, amelyen lezajlik az összeköttetés forgalma. Az ilyen módon felhasznált virtuális áramkör az összeköttetés bontásakor megszűnik.

A virtuális áramkörök kialakításához minden csomópontnak fenn kell tartani egy olyan táblázatot, amely bejegyzései a rajta keresztül haladó éppen használt virtuális áramkörök jellemzőit (honnan jött – hova megy) tartalmazzák, és az azonosításukra egy sorszámot használnak. Minden hálózaton keresztülhaladó csomagnak tartalmaznia kell az általa használt virtuális áramkör sorszámát.




***5-1. ábra Összeköttetés mentes hálózat fogalmai***

Amikor egy csomag megérkezik egy csomóponthoz, az tudja, hogy melyik vonalon jött, és mi az általa használt virtuális áramkörének sorszáma. A tárolt táblázatából ezek alapján ki tudja olvasni, hogy melyik csomópont felé kell továbbküldeni.

**Összeköttetés-mentes hálózatokban az átvitel csomagok segítségével történik** (ezeket tárolják és továbbítják a hálózati csomópontok). Az áramló csomagokat **datagram**-oknak nevezik. Elvileg minden egyes csomag, különböző útvonalakat követhet, mivel a csomagok útválasztása egymástól független. Ilyenkor a csomagoknak tartalmazniuk kell mind a forrás, mind a cél teljes címét. A célcím alapján az adott irányba való küldéséért a küldő IMP-n futó program a felelős.

A hálózatok matematikai modellje **gráf**-okkal írható le, ahol a **csomópontok** felelnek meg az IMP egységeknek, és a csomópontokat összekötő gráfélek az IMP-k közötti csatornák. A gráfélekhez (a csatornákhoz) hozzárendelünk egy költséget (**mértéket**) ami a csatorna minőségét (használhatóságát) fejezi ki. Minél gyorsabb a csatorna, és olcsóbb a használata, annál kisebb a mértéke. (Persze a mértéket még egyéb jellemzők, pl. hibaarány is befolyásolja.)

A feladat a kommunikációban szereplő két végpont között a **legrövidebb út** (ahol a bejárt élekhez rendelt mértékek összege a legkisebb) **megtalálása, úgy, hogy az érintett csomópontok száma** (amit a csomóponti átlépések száma jellemez), **minimális legyen.** (pl. az *5-1. ábrán* a G és a H csomópontokat összekötő csatorna (gráf éle) mértéke: 4.

Minden csomópont, mielőtt egy érkező csomagot továbbküldene, a csomag fejlécében szereplő címből, egy **irányító (routing) táblában** lévő bejegyzések alapján megállapítja, hogy melyik irányba küldje tovább. Ehhez a csomópontokban csomagtároló **puffer**eket (tárhelyeket) kell kialakítani. Mivel a tárhelyek száma véges egy csomópontban, ezért a tárhelyek száma és a továbbítás sebessége meghatározza egy csomópont terhelését, ami a hálózati forgalom szabályozásával korlátozható. A leírtakat az *5-1. ábrán* foglaltuk össze.

***Milyen előnyei, illetve hátrányai vannak e kétféle módszernek?***

Először is le kell szögeznünk, hogy egyik mellett sem szól olyan döntő érv, ami az alkalmazásának a győzelmét jelentené. Nézzük először az üzenetek hosszát! Ha a csomagok nagyon rövidek, akkor a teljes célcím használata — ami a csomagküldéshez kell általában jóval hosszabb, mint a virtuális áramkört azonosító kód — csökkenti a hasznos adatátviteli sebességet.

A legnagyobb gond a virtuális áramkörök biztonsága: egy virtuális áramköri táblázatokat tartalmazó IMP gép meghibásodása miatt az összes rajta átmenő nyilvántartott virtuális áramkört újra kell építeni, és a félbeszakadt üzeneteket újra adni. Csomagkapcsolás esetén nem ilyen tragikus a helyzet, hiszen csak azokat a csomagokat kell újra adni, ami éppen továbbítás alatt volt. A következőkben, egy táblázatban [1] hasonlítjuk össze a két módszert:

| Tárgy | Datagram hálózat | Virtuális áramkörös hálózat |
|---|---|---|
| Áramkör létesítése | Nincs | Szükséges |
| Címzés | Minden csomagban forrás és célcím | Csak egy rövid, virtuális áramkört azonosító cím |
| Állapotinformáció | Az alhálózat nem hordoz | Táblázatokban tárolt |
| Forgalomirányítás | A csomagok útvonala egymástól független | A VÁ létesítése meghatározza az útvonalat |
| Csomóponti hibák hatása | Csak az IMP-ben lévő csomagokra | Összes, az IMP-n átmenő VÁ megsemmisül |
| Torlódásvezérlés | Nehéz megoldani | Könnyű, ha elegendő puffer van |
| Alkalmas | Összeköttetés-alapú és összeköttetés-mentes szolgálathoz is. | Összeköttetés-alapú szolgálathoz |

A táblázat utolsó sorához a következő megjegyzést kell fűzni: **A kétfajta szolgálat csupán az információ továbbítás módját különbözteti meg.** Ezért lehetséges a hálózat egyik részén összeköttetés alapú szolgálattal átvinni csomagokat (VÁ kialakítása, majd ezen a csomagok átvitele), míg a hálózat másik részén datagramm szolgálatot használunk csomagtovábbításra.

## 5.1 Forgalomirányítás

**A forgalomirányítás (routing) feladata a csomagok hatékony (gyors) eljuttatása az egyik csomópontból a másikba, illetve a csomagok útjának kijelölése a forrástól a célállomásig.**

Ahogy már leírtuk, a hálózatot célszerű gráfként modellezni, ahol a csomópontok a csomagtovábbító IMP egységek és a csomópontokat összekötő élek az IMP egységek közötti információs adattovábbító csatornák.

A csomagok a hálózati vonalakon keresztül jutnak egy IMP-be, majd az, valamilyen irányba továbbküldi a csomagokat. Mivel az ilyen hálózati csomópontok irányítási, továbbküldési kapacitása véges, elképzelhető a csomagok sorban állása a bemenő illetve a kimenő oldalon.

- Vonalkapcsolt hálózatoknál az útvonal kijelölése a hívás felépítésének fázisában történik.
- Csomagkapcsolt hálózatokban az útvonal kijelölése vagy minden csomagra egyedileg történik, vagy kialakítunk egy olyan útvonalat, amelyen egy sorozat csomag megy át.

Ezért a csomópontoknak ún. **routing táblákat** kell tartalmaznia, amiben a vele kapcsolatban álló csomópontokra vonatkozó adatok (pl. távolság) be van jegyezve.

A forgalomirányítás összetettségét alapvetően meghatározza a hálózat topológiája. Például egy csillaghálózatban, mivel a csillag központjában lévő csomóponton keresztül történik az adatátvitel, kizárólag ennek kell rendelkeznie a forgalomirányításhoz szükséges minden információval.

Általában is elmondható, hogy szabályos elrendezések esetében általában könnyebb az optimális forgalomirányítási algoritmus kidolgozása.

A legtöbb valóságos hálózat lényegesen bonyolultabb topológiájú, szabálytalan szövevényes és sokszor állandóan változó szerkezettel rendelkezik.

A forgalomirányító módszerek (algoritmusok) osztályozásának alapjául a következő négy irányítási főfunkciót tekinthetjük:

- vezérlésmód; (hogyan történjen?)
- döntésfolyamat; (milyen esetben kell?)
- információ-karbantartó folyamat; (hálózati forgalmi ismeretek frissítése)
- továbbító eljárás (hogyan jut el a vezérlési információ a csomópontokhoz)

Ezen funkciók feladata a forgalomirányítási információk áramlásának szabályozása, kerülő utak választékának kialakítása, az irányítási információk felújítása valamennyi csomópontban és az útvonalválasztás az adatcsomagok részére.

A forgalomirányítási algoritmusoknak két osztálya van:

- Az **adaptív (alkalmazkodó) algoritmusok,** amely a hálózati forgalomhoz alkalmazkodik, és a
- **determinisztikus (előre meghatározott) algoritmusok,** ahol az útvonal választási döntéseket nem befolyásolják a pillanatnyi forgalom mért vagy becsült értékei.

Ezek alapján alapvetően négy lehetséges vezérlésmód különböztethető meg:

- **determinisztikus forgalomirányítás;** olyan rögzített eljárás, amelyet a változó feltételek nem befolyásolnak;
- **elszigetelt adaptív forgalomirányítás**, amelynél minden csomópont hoz irányítási döntéseket, de csak helyi információk alapján;
- **elosztott adaptív forgalomirányítás**, amelynél a csomópontok információt cserélnek azért, hogy az irányítási döntéseket a helyi és a kapott információkra együtt alapozhassák;
- **központosított adaptív forgalomirányítás,** amelynél a csomópontok a helyi forgalmi információikat egy közös irányító központnak jelentik, amely erre válaszul forgalomirányítási utasításokat ad ki az egyes csomópontok részére.

### *A legrövidebb út meghatározása*

Nyilvánvaló hogy a forgalomirányítás során két pont között meg kell találni a legoptimálisabb útvonalat, amely még egyéb csomópontokat tartalmaz.

Az optimális útvonal nem feltétlenül jelenti a fizikailag legrövidebb útvonalat, mivel számos egyéb tényező is befolyásolhatja az optimális választást: lehet például mértéknek a csomópont-átlépések számát tekinteni, lehet azt az időt, hogy mennyi idő alatt jut el a csomag a kezdőponttól a végpontig, vagy a vonalhasználat költségeit.

Az objektív mérték megállapításához lehet olyan teszteket futtatni az adott szakaszokon, amely magadja az átlagos sorbaállási és átviteli időt, és ezt tekinti a mértéknek. Általánosan egy adott szakasz mértékét a távolság, az adatátviteli sebesség, az átlagos forgalom, a kommunikációs költség, az átlagos sorhossz vagy más egyéb tényezők alapján határozzák meg.

Matematikailag a probléma a gráfelmélet segítségével tárgyalható, és megoldható. A feladat a gráf két csomópontja közötti olyan élekből álló útvonal meghatározása (shortest path), amelyre az érintett élek mértékeinek összege minimális. A megoldási módszer Dijsktrá-tól (1959) származik [1].

## *5.1.1 Determinisztikus forgalomirányítás*

Vannak olyan forgalomirányító módszerek, amelyeknél nincs szükség semmilyen forgalomirányítási táblára, a hálózati topológia ismeretére, minden csomópont autonóm módon, azonos algoritmus alapján dolgozik.

### *Véletlen forgalomirányító eljárás*

Ez alapján működő rendszerben a továbbítandó csomagot a csomópont egy véletlen-

szám generátor segítségével kiválasztott, az érkező vonaltól eltérő más vonalon küldi tovább. Mivel a hálózat által ilyen módon szállított csomagok véletlen bolyonganak, ésszerűnek látszik, ha a csomagokhoz hozzárendeljük a mozgásuk során átlépett csomópontok számát és töröljük azokat a csomagokat, amelyek lépésszáma elér egy előre meghatározott értéket. Ez az eljárás nem garantálja a csomagok kézbesítését, de nagyon egyszerűen realizálható, és nem túl bonyolult hálózatokban jól működhet.

***Elárasztásos forgalomirányító eljárás***

A csomópontok mikor egy csomagot továbbítanak, a bejövő csomagot minden vonalra kiküldenek, kivéve ahonnan érkezett. Nem igényel semmi ismeretet a hálózatról. A lépések száma itt is korlátozva van. Jelentős érdeme a módszernek, hogy a csomag legalább egy példányban mindenképp a legrövidebb úton ér célba. Ez azonban jelentősen terheli a rendszert, mivel nagyszámú másolat (redundancia) van, és sok felesleges továbbítás történik. Az algoritmus rendkívül megbízható, és még megsérült rendszer esetén is működőképes.

## 5.1.2 Adaptív forgalomirányítás

A probléma a hálózat elosztott jellegéből ered. Amikor a csomópontok irányítási döntéseket hoznak, olyan eseményeket kell figyelembe venniük, amelyek a hálózat távoli részében történtek, és amelyekről vagy egyáltalán nem rendelkeznek semmiféle információval, vagy a meglevő információjuk már időszerűtlen.

A csomaghálózatokban a forgalomirányítási információ ugyanazon a közegen és ugyanolyan sebességgel halad, mint a felhasználói információ. Nem volna értelme a csomagkapcsolt hálózatban az irányítási és egyéb vezérlő információkat egy külön nagy adatátviteli sebességű rendszerben, a felhasználói forgalmat pedig kis sebességű vonalakon továbbítani.

A csomaghálózat szempontjából is jó lenne az egész hálózatra kiterjedő forgalomirányítási információ azonnali elérhetősége. Bár a gyakorlatban ez megvalósíthatatlan, a szimulációs modellezés módszerével mégis analizálták az ilyen módon működő hálózat elméleti teljesítőképességét. A szimuláció során minden egyes csomópont úgy hozta irányítási döntését, hogy ehhez a hálózat többi részéről is teljes körű és közvetlen áttekintése volt. Az irányító algoritmus — ismerve az összes többi csomóponton a sorok hosszát és minden egyes vonalon az áthaladó csomagok számát — az irányítás alatt álló csomagja részére azt a következő, optimális adatátviteli vonalat választotta ki, amelyen az áthaladva minimális késleltetési idővel érkezhetett célba. Ennek a szimulációs kísérletnek teljesen váratlanul az volt az eredménye, hogy itt az átlagos késleltetési idők nem voltak lényegesen kisebbek, mint a rögzített forgalomirányító eljárásnál, amelynél a forgalomirányítási táblákat a legrövidebb utakra állították be.

Ennek oka lehetett az, hogy bár a forgalomirányítás a pillanatnyilag lehető legpontosabb információn alapult, az időközben megváltozott forgalom miatt a döntés pillanatában optimális útvonal még a kérdéses csomag célba érkezése előtt már nem volt optimális.

Ez szabályozástechnikai analógiával egy lengő rendszernek felel meg. Az ideális algoritmus sem tudja előre figyelembe venni a jövőben bekövetkező eseményeket. A szi-

muláció jól jellemzi a különböző, ténylegesen működő forgalomirányító algoritmusok egyik lehetséges nagy hátrányát; azt a tényt, hogy a hálózat egy bizonyos részéről a hálózat többi részei esetleg úgy értesülnek, hogy az pillanatnyilag alig van terhelve, és tartalék kapacitással rendelkezik. Ha ezek a részek ugyanakkor éppen torlódással küszködnek, valamennyien egyszerre fognak arra törekedni, hogy ebbe az alig terhelt zónába tereljék a forgalmat, amivel ott még súlyosabb torlódást idézhetnek elő.

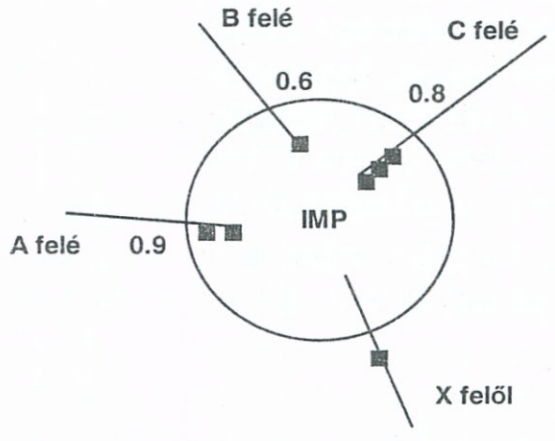
A valóságos hálózatokban alkalmazott adaptív forgalomirányító eljárásoknak vagy a helyileg rendelkezésre álló információt (izolált adaptív irányítás), vagy a hálózatban terjesztett információt kell felhasználniuk.

### *Központosított adaptív forgalomirányítás*

Minden egyes csomópont helyzetjelentést állít össze, és abban a folyó sorhosszakat, a hálózat elemeinek meghibásodásait stb. elküldi a hálózat forgalomirányító központjába (RCC = Routing Controll Center). A központ ezek alapján átfogó képet alakít ki a hálózatról, és valamennyi forgalmi áramlat részére meg tudja határozni a legkedvezőbb útvonalat. A legjobb utakat a hálózat csomópontjai forgalomirányítási táblák formájában kapják meg.

A központnak szóló helyzetjelentéseket és a csomópontoknak szóló új irányítási táblákat szabályos időközönként (periodikusan) vagy csak jelentős változás hatására (aszinkron módban) küldik. Ha a periodikus üzemmódot választják, akkor - az irányító algoritmus működtetése érdekében - a hálózatban áramoltatott vezérlő információ hatalmas menyiségű lehet. Különösen, ha a hálózat maga nagy, akkor a túlzott mértékű irányítási funkció jelentős többletterhelést okoz. Aszinkron módban viszont csak elfogadható mennyiségű vezérlő információ áramlik a hálózatban.

Azt várhatnánk, hogy a hálózat forgalomirányító központja az optimális utak kiválasztásához a lehető legjobban hasznosítja a hálózat kapacitását. Az elkerülhetetlen időkülönbségek miatt a csomópontokból elinduló állapotjelentések eleve késve érkeznek a központba és a távoli csomópontokból, ez a késés már jelentős lehet. Megfordítva, miután a központ elvégezte a forgalomirányító funkció által igényelt tekintélyes idejű számításokat, további időhátrány származhat abból, hogy a csomópontok késve kapják a módosított forgalomirányítási táblákat. Így azután a központ olyan információk alapján dolgozik, amely részben már elavultak, és a csomópontok részére is olyan utasításokat ad ki, amelyek még inkább elavultak, amikor célba érnek.




*5-2. ábra: "Forró krumpli" algoritmus*

### *Elszigetelt adaptív forgalomirányítás*

Ilyenkor a forgalomirányítási döntéseket a helyi körülmények alapján hozza a csomópont. Egyszerű algoritmus az ún. **„forró krumpli" algoritmus.** Ennek az a lényege, hogy a beérkezett csomagot abba kimeneti sorba rakja, amely a legrövidebb, legrövidebb ideig „égeti a kezét", gyorsan megszabadul tőle. Lényeges, hogy nem foglalkozik az irányokkal.

Érdekes kiterjesztése az algoritmusnak, amikor ennél a döntésnél az irányokhoz tartozó mértékeket is figyelembe veszi. Ez azt jelenti, hogy nem küldi automatikusan a legrövidebb sorba, hanem figyelembe veszi a kiválasztott sor mértékét is.

Például a 5-2. *ábrán* látható X jelű csomópont felől érkező csomag az eredeti algoritmus szerint B felé lenne elküldve. A módosított algoritmus szerint ez már nem biztos, hiszen a mértéke (jósága) csak 0.6, ezért talán jobb lehet az A irányt választani. A korrekt döntéshez kell egy sorhosszt jellemző mérőszámot is választani (1-ha üres a sor, 0 — ha nagyon sok csomag van előtte) és így pl. a két szám szorzatának nagysága alapján hozni meg az irányra vonatkozó döntést.

Egy másik lehetséges algoritmus a **fordított tanulás módszere**. A hálózatban minden csomópont egy csomagot indít, amely tartalmaz egy számlálót és az elindító azonosítóját. A számláló értéke minden csomóponton történő áthaladáskor eggyel növekszik. Amikor egy csomópont (IMP) egy ilyen csomagot vesz, akkor ezt elolvasva tudja, hogy a csomagot küldő hány csomópontnyi távolságra van tőle.

Természetesen az optimális út keresése érdekében, ha ugyanarra a távoli csomópontra egy kedvezőbb értéket kap (van rövidebb út is), akkor az előzőt eldobva ezt jegyzi magának. Ha azonban meghibásodás következik be, vagy az optimális útvonal valamelyik része túlterhelődik, akkor ezt az algoritmus nem veszi észre. Ezért célszerű, időnként „mindent felejteni", törölni a feljegyzéseket, hogy az ilyen változó körülményekre is működjön az algoritmus.

### *Elosztott adaptív forgalomirányítás*

A megvalósított hálózatokban mindeddig legnépszerűbb az elosztott adaptív forgalomirányító eljárás. **Lényege: a csomópontok környezetében kévő forgalmat kell figyelnie minden csomópontnak.**

Az algoritmus fő célkitűzése az adatforgalom részére a legkisebb késleltetéssel járó útvonalak keresése. E célból minden egyes csomópontban egy táblázatot hozunk létre, ami minden egyes célállomáshoz megadja a legkisebb késleltetésű útvonalat, s ezzel együtt a továbbításhoz szükséges idő legjobb becsült értékét. A hálózat működésének kezdetén a késleltetések a hálózat topológiája alapján becsült értékek, később azonban, mihelyt a csomagok célba értek, a becsült késleltetési időket felváltják a hálózatban ténylegesen mért továbbítási idők.

A szomszédos csomópontok között a késleltetési táblák cseréje természetesen sok vezérlőcsomag továbbításával történik, ami jelentős többletterhelést ró a hálózatra. Ha a táblákat túl gyakran, pl. 2/3 másodpercenként tartják karban, a hálózati mérések azt mutatják, hogy a kis adatátviteli sebességű vonalak kapacitásának 50 százalékát a késleltetési táblák továbbításával járó forgalom foglalja le.

A táblák ilyen, periodikus karbantartása helyett, az aszinkron karbantartás a célravezetőbb. Ez utóbbi azt jelenti, hogy a csomópontoknak csak akkor kell továbbítaniuk a késleltetési táblákat, ha számottevő változást észlelnek a forgalom intenzitásában, vagy a hálózat elemeinek működési körülményeiben. Késleltetési táblák újraszámítására csak akkor kerül sor, ha jelentősebb helyi változás történt, vagy ha módosított késleltetési tábla érkezik valamelyik szomszédos csomóponttól.

## *5.2 Torlódásvezérlés*

Azt hihetnénk, hogy ha a vonalak és csomópontok kapacitása elegendő az adatforga-

lom lebonyolításához, akkor a szabad információáramlás minden esetben garantálható. A tényleges helyzet azonban más. Előfordul, hogy a rendeltetési helyen a csomagoknak a hálózatból való kiléptetése akadályba ütközik, mert a hálózat legfeljebb azzal a sebességgel tudja kézbesíteni a csomagokat, amilyen ütemben a felhasználó hajlandó azokat elfogadni. A csomagok küldőjére ekkor minél előbb át kell hárítani ezt az akadályt, ellenkező esetben a csomagok a hálózatban felhalmozódnak. Ez jelenti azt a forgalomvezérlési funkciót, amelynek segítségével a hálózati forgalmat folyamatosan mozgásban lehet tartani.

Bár a hálózat adatátviteli kapacitását általában a várható igényeknek megfelelőre tervezik, mégis a forgalom statisztikus változásai, még ha alacsony bekövetkezési valószínűséggel is, de túlterhelést idézhetnek elő. A jó hálózati forgalomvezérlési algoritmus megoldást ad a túlterhelések elviselésére is. Fel kell használnia beépített forgalomvezérlő mechanizmusát arra, hogy a túlzott forgalmi igényeket visszautasítsa. Mindaddig fenn kell tartania a korlátozó intézkedéseket, ameddig a normális, korlátozásmentes üzem ismét vissza nem állítható.

Ha egyes hálózatrészek túltelítődnek, akkor a csomagok mozgatása lehetetlenné válhat. Azok a várakozási sorok, amelyeknek a csomagokat be kellene fogadniuk, állandóan tele vannak. **Ezt a helyzetet nevezzük torlódásnak (congestion).**

A torlódás szélsőséges esete a **befulladás (lock-up).** Ez olyan, főként tervezési hibák miatt előálló eset, amelyben bizonyos információfolyamok teljesen leállnak a hálózatban.

A torlódás a csomaghálózatokban olyan állapot, amelyben a hálózat teljesítménye valamilyen módon lecsökken, mert a hálózatban az áthaladó csomagok száma túlságosan nagy.

A torlódás lehet helyi jellegű, amikor a jelenség a hálózatnak csak bizonyos részét érinti, vagy súlyosabb, mikor az egész hálózatra kihat.

A közúti forgalom viselkedése még közelebb áll a csomagkapcsolt hálózatok viselkedéséhez. A közutak hálózatot alkotnak, amelyben szállítási csatornák, utak, kereszteződések stb. találhatók. A forgalom az úthálózaton át a nagyszámú forrás-rendeltetés pár között járműfolyamok formájában áramlik. E folyamok minduntalan összefolynak majd szétválnak a különböző keresztezési és elágazási pontokban. A célba érési arány (az időegység alatt célba érkező járművek száma) akkor maximális, ha az utakon közlekedő járművek száma nem halad meg egy bizonyos szintet. Olykor, pl. csúcsforgalomban, amikor az útra kelt járművek száma nagyon nagy, forgalmi dugók jönnek létre, és az egyes járművek előrehaladása sokkal lassúbbá válik. Sőt túlságosan is könnyű olyan feltételt teremteni, amelynek fennállása esetén aligha éri el a jármű rendeltetési helyét. És ez a járműfolyamok és az egyes járművek áramlása egymásra hatásának a következménye.

A torlódás olyan állapot, amely a legtöbb szállítmányozó rendszerben előfordul. Például a folyóban úszó farönkök akadálytalanul sodródnak mindaddig, amíg az egyes darabok mozgása nincs hatással a többiek előrehaladására. Ha a rönkök számát eddig a szintig növeljük, akkor ezzel együtt az átbocsátott mennyiség is növekszik, de ezzel elérkeztünk egy olyan ponthoz, amelynél a rönkök már akadályozzák egymás mozgását, és a teljes átbocsátott mennyiség lecsökkenhet. További rönkök bedobása esetén a rönksűrűség oly mértékben megnőhet, hogy már szilárdan egymáshoz ékelődnek, és így a csatornában az áramlási sebesség nullára esik vissza.

Visszatérve a hálózatokhoz, általánosan a torlódás okainak az IMP-k viszonylagos lassúságát tekinthetjük, valamint azt a lehetséges okot, hogy a kimenő vonalak kapacitása kisebb, mint a bemenő vonalaké. Ezért kidolgoztak stratégiákat a torlódás elkerülésére:

**Pufferek foglalása:** virtuális áramkörök esetén használható, hiszen itt az információ áramlását megelőzi a hívásfelépítés. Az IMP eszközökben az adott virtuális áramkörhöz tárolóterület (puffer) foglalható. Az IMP csak akkor nyugtázza a bejövő csomagot, ha tovább tudta küldeni (és így van szabad puffer). A nyugta egyben jelzi, hogy jöhet a következő csomag.

**Csomageldobás módszere:** Itt nincs előzetes pufferfoglalás. Ha a datagram szolgálatnál alkalmazzuk, akkor a csomagot egyszerűen eldobjuk, ha nincs hely. Virtuális áramkör esetén ez nem tehető meg, a csomagot újraadásig valahol tárolni kell. Mivel az adatcsomagok általában ráültetett nyugtákat is tartalmaznak, ezért eldobásuk nem célszerű. Érdemes egy külön „nyugtázott csomagok puffer-területe" részt fenntartani, és a csomag, ha nyugtát tartalmaz, vizsgálat után eldobás helyett ide kerülhet.

**Izometrikus torlódásvezérlés:** Mivel a hálózaton jelenlévő túl sok csomag okozza a torlódást, ezért célszerű a csomagok számát korlátozni. Ezt úgy lehet megtenni, hogy a hálózatban engedélycsomagok járnak körbe. Ha egy IMP adni kíván, egy ilyen engedélyt kell vennie, és annak továbbadása helyett egy adatcsomagot küldhet tovább. Mivel a hálózatban az engedélyek száma korlátozott, így az ezeket helyettesítő csomagok száma is korlátozva lesz. Persze ez nem garantálja, hogy egy IMP-t ne árasszanak el csomagok. Másik probléma az engedélyek kiadásának és elosztásának megoldási nehézségei.

**Lefojtó csomagok használata:** A módszer alapfilozófiája: a torlódáskiküszöbölő algoritmus csak akkor kezdjen működni, ha a hálózaton torlódásveszély kezd kialakulni. Erre a megoldás a következő: minden IMP figyeli a kimeneti vonalainak átlagos kihasználtságát (K), és ezt mindig újraszámítja a pillanatnyi f vonalkihasználtság, és egy 0 és 1 közötti **a** felejtési tényező alapján:

$$K_{uj}=a*K_{régi}+(1-a)*f.$$

Ha **K** értéke egy küszöböt elér, akkor a kimeneti vonal „figyelmeztetés" állapotba kerül. Az IMP minden beérkező csomag elküldése előtt — ha ezt ilyen állapotú kimeneti vonalon kell továbbküldenie — elküldi, de a forráshelyre visszaküld egy lefojtó csomagot a beérkezett csomagban talált célcímmel együtt. Amikor a forrás IMP egy ilyen lefojtó csomagot kap vissza, akkor adott mértékben csökkentenie kell az ilyen irányú forgalmát.

A torlódások legsúlyosabb esete a holtpont. Ez azt jelenti, hogy az egyik IMP valamire vár, ami a másik IMP-től függ, az pedig egy olyan eseményre, amely a rá várakozótól függ. Ebből nincs kiút. Ilyen eset következhet be, ha például mindkét IMP puffere, a másik felé irányuló csomagokkal van tele. Ahhoz hogy fogadni tudjon az egyik, ki kellene ürítenie a pufferét, de nem tudja, mert a másik azt jelzi, hogy foglalt. Másik irányban is azonos a szituáció. Ez az eset a **közvetlen tárol és továbbít holtpont**.

Ez az eset természetesen nem csak két szomszédos csomópont, hanem egy hálózat egészében vagy egy részében is létrejöhet, ha egyik IMP-nek sincs szabad helye a csomagok fogadására.

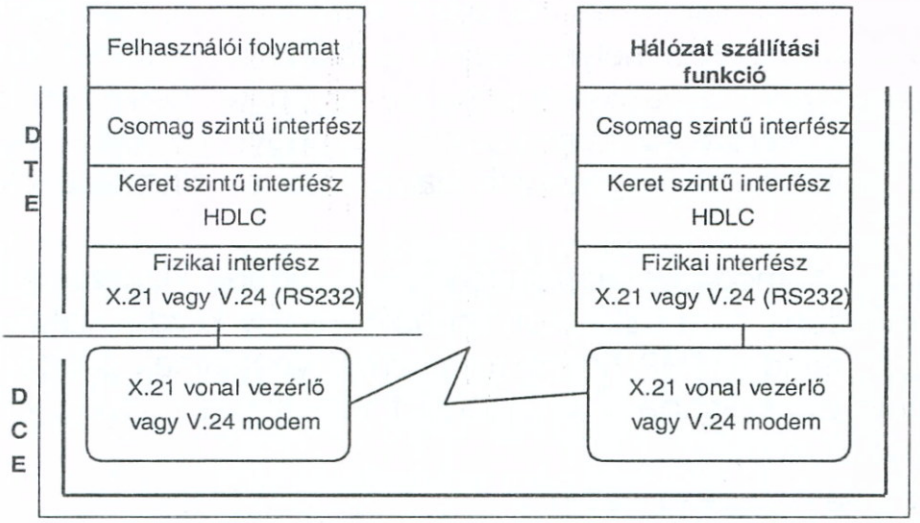
Ez a **közvetett tárol és továbbít holtpont.** Az ilyen és hasonló holtpontok kialakulásának kiküszöbölésére számos, itt nem részletezett módszert fejlesztettek ki. [1]

## 5.3 X.25 hálózat

Ez egy CCITT ajánlás, amely a felhasználó (az adatvég-berendezés, DTE=Data Terminal Equipment) és a hálózat (adatáramkör végződő berendezés, DCE= Data Circuit terminating Equipment) közötti interfészt definiálja.

| DTE (user) oldal | Kapcsolat típus | DCE(node)oldal |
|---|---|---|
| Packet Level | - - - - - - - - - - - (virtuális kapcsolat) | Packet Level |
| *Frame Level* | - - - - - - - - - - - (virtuális kapcsolat) | Frame Level |
| Fizikai Szint | ---------- tényleges kapcsolat | Fizikai Szint |

Az X.25 ajánlás alapján elkészített berendezések az ISO OSI 7 szintű referencia modell alsó három szintjét valósítják meg, miközben maga az ajánlás végső soron a network (3.) és transport (4.) szint közti interfészt definiálja, vagyis az X.25 a 4-7 szinten elhelyezkedő alkalmazások számára nyújt hálózati szolgáltatást.




DTE+DCE EGYÜTT= X.25 INTERFÉSZ

*5-3. ábra: X.25 interfész*

Csomagkapcsolású hálózatokban alkalmazzák. Ezekben három alapvető csomag típusú szolgálatot határoztak meg.

- Az első a **datagram (DG)** szolgálat, amely lehetővé teszi, hogy a felhasználó a hálózatban független csomagokat bárhová elküldjön, vagy bárhonnan fogadjon.
- A második az **állandó virtuális áramkör (PVC=Permanent Virtual Circuit),** amely két DTE-t állandóan összeköt logikai csatornával. Ez biztosítja a csomagváltások során a sorrendhelyességet.
- A harmadik szolgálat a **virtuális hívás (VC=virtual call)** ami az előbbi PVC rövid időre kapcsolt összeköttetés, ideiglenesen kialakított megfelelője.

Fontos szolgáltatás a nem csomagkapcsolt hálózatokkal való illesztést biztosító a csomagösszeállítás és felbontás **PAD (Packet Assembly-Disassembly)** funkció. Ez a szolgáltatás az előfizető bit és karakterfolyamait (pl. egy terminál jeleit) csomagokká alakítja, illetve visszaalakítja. Ez teszi lehetővé, hogy a karakter üzemmódú terminálok csomag üzemmódú DTE-kel kommunikáljanak.

A **PAD** valójában egy célszámítógép, amely egyszerű aszinkron soros vonalon keresztül képes egy terminállal kapcsolatba lépni Pl. PC soros port - és az onnan érkező karakterek-ből X.25 ajánlásnak megfelelően adatcsomagokat képezni és az X.25 hálózatba továbbítani, majd a visszaérkező adatokat karakterekre bontva a "buta" terminálhoz továbbítani.

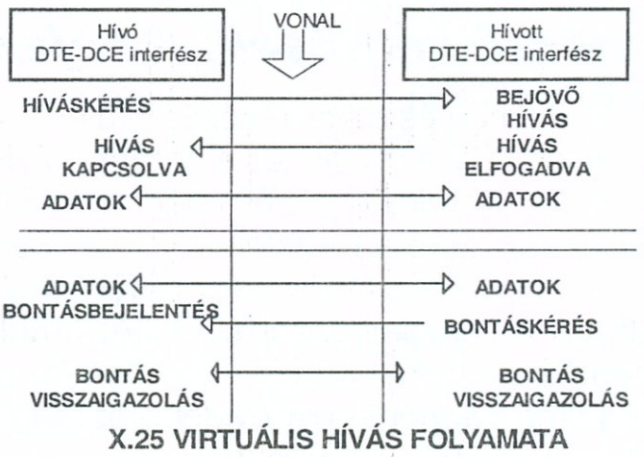
Az X.25 három protokollszintet különböztet meg.

A **fizikai szint** a csomagkapcsoló központhoz való kapcsolódást biztosítja adatáramkörökön keresztül. Az adatáramkör lehet bérelt áramkör, vagy kapcsolt összeköttetés, de akár analóg távbeszélő áramkör is. Digitális átvitel esetén ez az X.21, amely egyaránt gondoskodik mind az összeköttetés gyors felépítéséhez a digitális címzésről és a bérelt áramkör működéséről. Analóg áramkör esetén a modemes összeköttetés V.24 ajánlása használható, amit X.21bis szabvány néven is emlegetnek.

A **második szint** egy HDLC szerinti adatkapcsolat, amely a DTE és a DCE közötti hibamentes adatcserét biztosítja. A HDLC keretek az X.25 interfészen keresztül csak egy-egy csomagot hordoznak. A protokoll neve LAP-B (Link Access Protocoll-Balanced), ami egyenrangú állomásokat (kombinált állomás) definiál a két végponton.

Sorrendtartó, hibamentes adatátvitelt biztosít. Az adatátvitel bájtokba csoportosított bitek segítségével zajlik, és annak érdekében, hogy bármilyen bájtérték az átvitt adatban szerepelhessen, speciális megoldást kell használni. Ez a HDLC keret, amely a már tanult bit-beszúrással (bit stuffing) biztosítja az előbbi feltételt. A frame-ek elejét/végét egy speciális kód szinkron karakter — úgynevezett FLAG, (07Eh, 01111110b) — jelzi. Ha nincs átviendő adat, akkor a vonalon folyamatosan FLAG-ek mennek.

Ahogy ezt már leírtuk, a HDLC bit-orientált protokoll, vagyis nem szabja meg, hogy az adat milyen struktúrájú, még azt a kikötést sem teszi, hogy az adat hossza 8-al maradék nélkül osztható legyen. (A bit-beszúrás miatt úgyis változik az átviendő adat, és a ténylegesen átvitt adat bitszáma.)




***5-4. ábra: X.25 Virtuális hívás***

A **harmadik szint** a csomagszint, amely az előbbiekben felsorolt (DG, PVC, VC) csomagtípusú szolgálatokat biztosítja. A virtuális áramkörön a forgalom vezérlését ablaktechnika biztosítja. Reset és újraindítás lehetséges hibaállapot fellépése esetén. A hívások lebonthatók, és a felszabaduló csatornák újra felhasználhatók.

Az *5-3. ábrán* X.25 DTE-DCE interfész elrendezés látható. A modemes átvitelnél megismert DCE itt kissé más jelentésű. A távbeszélő áramkörökben használt DCE nem más, mint a felhasznált modem. Részletes leírásuk a [2] irodalomban megtalálható.

Az *5-4. ábrán* látható a virtuális hívás folyamata. A három fázis: a hívás felépítése, adatátvitel és lebontás csomagok segítségével történik. Forgalomvezérlés, ami megakadályozza, hogy az egyik oldali gyorsabb DTE-DCE interfész elárassza csomagjaival

a másik oldalt, a már megismert, és csomagszinten alkalmazott csúszóablakos átviteli technikával történik. A vételi és adási ablakok mérete, 8 illetve 128 lehet.

## 5.4 Keretrelézés

Mint a nevéből következik nem a csomagokat, hanem az adatkapcsolati szint kereteit viszik át a megfelelő minőségű hálózaton. A keretrelézés egy X.25-höz hasonló módszer, bár az X.25-höz eltérően nem megbízható összeköttetést biztosít, nincs a sebességet és vevő fogadóképességét figyelembe vevő áramlásvezérlés (flow control).

Az átvitelhez HDLC kereteket használ, ahol az adatrész akár 4 kbájt is lehet.

A keretek egy vagy több, állandó kapcsolatra beállított virtuális áramkörön (Data Link Connection Identifier = DLCI) keresztül haladnak. Mivel a hibamentes keretátvitelt nem figyelik, ezért a felette lévő réteg (HDLC IPC, TCP/IP) feladata a hibák felismerése, és a hibás keretek megismételtetése.

Ez azonban nem akkora probléma, mert a keretrelézést általában nem analóg (pl. telefon) vonalakon, hanem a kis hibaaránnyal működő digitális optikai átviteli vonalakon keresztül valósítják meg. Mivel nincs áramlásvezérlés, a vevő azokat a kereteket, amelyeket nem képes venni, egyszerűen eldobja. Alkalmazása előtt meg kell adni a használni kívánt a maximális átlagos adatátviteli sebességet (pl. 56 kbit/s). Észak Amerikában az "európai" X.25 átvitel helyett használják.

?

## Ellenőrző kérdések: 5.Fejezet

1. Mi a csomag és a keret közötti különbség? Mi a hálózati réteg feladata?
2. Mi a virtuális áramkör és a datagram? Milyen előnyei és hátrányai vannak?
3. Mi a forgalomirányítás és miért van rá szükség? Mi a routing tábla?
4. Mikor egyszerű a forgalomirányítás megvalósítása? Függ a topológiától?
5. Mi a forgalomirányítás négy fő funkciója?
6. Milyen vezérlésmódokat különböztetünk meg? Mi a determinisztikus és az adaptív forgalomirányítás közötti különbség?
7. Fogalmazza meg a legrövidebb út meghatározásának célját és módszerét!
8. Mi az a mérték, és mitől függ?
9. Ismertesse a véletlen forgalomirányítás módszerét! Ismertesse az elárasztásos forgalomirányító eljárás módszerét!
10. Ismertesse a központi adaptív forgalomirányítás módszerét!
11. Ismertesse az elszigetelt forgalomirányítás módszerét! Mi az a „forró krumpli" algoritmus? Mi a fordított tanulás módszere?
12. Ismertesse az elosztott adaptív forgalomirányítás módszerét!
13. Mi a torlódás, és mi a torlódásvezérlés célja? Mi a befulladás?
14. Mutasson be néhány módszert a torlódás elkerülésére!
15. Mutassa be a lefojtó csomagokat használó módszert!
16. Foglalja össze az X.25 hálózat legfontosabb tulajdonságait! Ismertesse a három alapvető csomagtípusú szolgálatot! Mi az a PAD?
17. Ismertesse az X.25 hálózat három (fizikai-, keret- és csomagszintű) protokollszintjét!

# 6. FELSŐBB RÉTEGEK

*A győzelem legfőbb feltétele a megfelelő ellenfél kiválasztása. (Moldova György)*

Az OSI modell tervezése előtt már sok hálózat működött, amelyek általában a hálózati rétegig jól átgondoltak voltak, és ezért jelentős mennyiségű működési tapasztalat és ismeret halmozódott fel.

A következőkben röviden összefoglaljuk az OSI modell felső négy rétegének feladatait. Ezek a rétegek már a hoszt-hoszt közötti kapcsolatok felépítését és működését írják le.

Az OSI modellben a szállítási, viszony, a megjelenítési és az alkalmazási rétegek alkotják a felső rétegeket.

## 6.1 Szállítási réteg

A rétegek közül a szállítási réteg még az alsó három réteg logikai folytatásának tekinthető, hiszen gondoljuk meg: ha egy hoszt üzenetet küld a másiknak, akkor az üzenet továbbítása előtt ezt általában csomagokra kell darabolni, ezeket a hálózati rétegnek átadva át kell vinni a hálózaton és a célhosztnak átadni, ahol az üzenet összerakásra kerül. Az üzenetben leírt különféle fajtájú tevékenységet végre kell hajtani. Az üzenetkapcsolást használó rendszerek pontosan ebből a csomag-darabolásból és összerakásból adódó problémákat kerülik ki a teljes üzenet egyszerre történő átvitelével.

**A szállítási réteg feladata: megbízható adatszállítás biztosítása a forráshoszt és a célhoszt között, függetlenül az alatta lévő rétegek kialakításától. Ez a réteg biztosítja, hogy a kommunikáló két hoszt egymást úgy lássa, mintha pont-pont összeköttetés lenne közöttük.**


***6-1. ábra: Adatkapcsolati és szállítási réteg hasonlósága***

A használt protokollok sok esetben hasonlítanak az adatkapcsolati réteg protokolljaira, de itt az IMP-ket összekötő fizikai csatornát, a két hoszt közötti teljes alhálózat jelenti. Fontos eltérések azért vannak:

- Adatkapcsolat esetén a pont-pont összeköttetés miatt nem kell címzés, míg szállítási rétegnél kötelező.
- Az összeköttetés létesítése adatkapcsolati szinten egyszerű: a másik oldal mindig ott van (ha nincs, akkor tönkrement). Szállítási réteg esetén a kezdeti összeköttetés létesítés bonyolult.
- A csomagok átvitele is eltéréseket mutat az adatkapcsolati keretátviteltől. Az alhálózat tárolókapacitása miatt elképzelhető, hogy egy csomag eltűnik (valahol tárolódik) majd egyszer hirtelen előkerül. A csomagok duplázódása miatt felmerülő problémákat is kezelni kell. Ez speciális protokoll használatát igényli.

Mivel a szállítási réteg a hálózati rétegre épül, ezért a hálózati szolgálat minősége alapjaiban meghatározza a szállítási protokoll kialakítását.

Szállítási protokollok esetén az üzenetek átviteléhez szükséges összeköttetés kezelése, a létesítés, és lebontás összetett folyamat.

A szállítási réteg számára a szállítási szolgálat-elérési pontok (TSAP = Transport Service Access Point) címei azonosítják a forrás- és a célhosztot. A szállítási réteg a hoszthoszt kapcsolat többféle konkrét megvalósítását biztosítja: például fájlok átvitele a két hoszt között, vagy az egyik hoszt a célhoszt termináljaként kíván működni, stb.

## Hálózati szolgálatok minőségi osztályai:

A különféle szállítási protokollok teljesítőképsségének a vizsgálata miatt a hálózati szolgálatokat minőségük alapján három típusba sorolták:

- **A típus:** ez lényegében tökéletes, hibamentes szolgálat. Nincs elveszett, sérült, kettőzött csomag (vagy elhanyagolhatóan kevés.) Ilyenkor a szállítási protokoll az adatkapcsolati protokollhoz hasonló feltételekkel, nagyon könnyen és egyszerűen működik. LAN-ok esetén ez már sokszor teljesül
- **B típus:** Egyedi csomagok csak nagyon ritkán vesznek el, de a hálózati réteg időnként kiad egy N-RESET-et. (*N-RESET: a hálózati réteg által kiadott, alaphelyzetbe állító, az összes függő csomagot, illetve kapcsolatot törlő parancs (Net-Reset)*) Ekkor a szállítási protokoll feladata az, hogy összeszedje a hálózatban a maradékot, új összeköttetést létesítsen, újraszinkronizálja az átvitelt, és úgy folytassa az abbamaradt összeköttetést, hogy a felhasználó ebből semmit se vegyen észre. WAN-okra ez jellemző, és jóval összetettebb szállítási protokollt igényelnek.
- **C típus:** Rossz minőségű, nem megbízható szolgálat, elveszett vagy kettőzött csomagokkal, gyakori N-RESET-el. Ilyenek a csak datagram szolgálatot nyújtó WAN-ok, és pl. a rádiós csomagszóró hálózatok. Ezek bonyolult összetett szállítási protokollt igényelnek.

## Címek a szállítási rétegben

A különféle szállítási szolgáltatásokhoz különféle TSAP címek tartoznak. Míg a hosztok címei általában ismertek, addig ezeket a szolgáltatásokhoz tartozó hoszton belüli címeket a küldő hoszt nem ismeri. Ezért, általában három féle a módszert alkalmaznak:

- a küldő hoszt által igényelt szállítási alkalmazást mindig a célnál egy **alkalmazásszolgáltató (process server)** fogadja, és az üzenet tartalma alapján rendeli hozzájuk a TSAP címeket.
- Egy **speciális névszolgáltatót (name server)** használnak. Egy adott szolgáltatásra vonatkozó TSAP-cím megtalálásához, a felhasználónak összeköttetést kell létesítenie a névszolgáltatóval, amely egy ismert TSAP címen várakozik. Ide elküldi a szolgáltatás nevét megadó üzenetet, amely visszaküldi a szolgáltatás TSAP címét.
- A harmadik megoldás a TCP/IP protokoll sajátossága. A hosztokon futó alkalmazások egy olyan címmel rendelkeznek, amely **jól ismert**, definiált, kiosztott **szál-**

**lítási címek** (más néven: portcímek). Például egy böngészőt kiszolgáló webszerver szállítási címe: 80H. Ezért a hosztokon nem probléma a szállítási címek megszerzése, mert ezek jól ismert címek. Ez a megoldás kiküszöböli a szállítási címek felderítéséhez szükséges kommunikációt.

### Elveszett, vagy kettőzött csomagok kezelése

Az összeköttetés létesítése nem egyszerű C típusú hálózatszolgálatok esetén, mert az elveszett, vagy kettőzött csomagok sok problémát okozhatnak. Ezekre, és hasonló problémákra alkalmazott megoldások:

- A csomagok élettartamának korlátozása.
    - Csomópontátlépés számláló alkalmazása a csomagban, amelynek értéke minden csomópont átlépésekor eggyel növekszik. A csomag eldobásra kerül, ha ez az érték egy adott korlátot elér.
    - A csomag létrehozásának időpontját a csomagban tároljuk; a csomagot vevő IMP-k ezen időpont alapján a csomag korát meg tudják állapítani. Ha a csomag „túl öreg", eldobják.
- Az összeköttetés létesítése során a felek tetszőleges, véletlenszámként előállított kezdősorszámmal kezdik az adásukat, így kiszűrhető egy korábbi összeköttetésből származó, esetleg késve érkezett csomag.

Az összeköttetés lebontásakor biztosítani kell az adatvesztésmentes lebontást. Nem lehet addig törölni az összeköttetést, amíg az összes elküldött adat meg nem érkezik.

## 6.2 Viszonyréteg

Például egy több városra kiterjedő helyfoglalási rendszerben, amikor valaki terminálján keresztül egy helyfoglalást intéz, akkor a helyi terminálokat összefogó számítógép szállítási kapcsolatot létesít a központi géppel, és a viszonykapcsolat segítségével lebonyolítja a helyfoglalást, majd a viszony lezárul.

A szállítási összeköttetést nem célszerű megszüntetni, mert egy újabb innen érkező helyfoglalási viszonynak is ezt kell használnia.

**A viszonyréteg a szállítási réteg felhasználásával szolgálatokat nyújt a felette lévő megjelenítési rétegnek. A fő funkciója az, hogy lehetőséget biztosítson a viszonyt használóknak adatokat cserélni a viszonyokon keresztül. A viszonyok a szállítási összeköttetések felhasználásával valósulnak meg. Egy viszony használhat egy vagy több szállítási összeköttetést is.**

Az előbbi példában a viszonyrétegen keresztül egy adatcsere valósult meg. Nagyon lényeges, hogy míg a szállítási rétegben kiadott bontási parancs azonnal megvalósul (a DISCONNECT szállítási primitív felhasználásával). Ez azt jelenti, ha egy szállítási kapcsolat megszakítási kérés érkezik a réteghez, azt a réteg mindenféle megerősítés nélkül elfogadja és végrehajtja. A viszonyréteg a rendezett bontást támogatja: azaz a viszonyrétegben a bontási kérelmet meg kell erősíteni. Egy viszony csak a két fél egyetértése alapján fejeződik csak be.

Mi történjék, ha a szállítási rétegben létesített összeköttetés megszakad? Ha például

ez egy adatátvitel volt, akkor a megszakadáskor az eddig átvitt adatok törlése után, az összeköttetést újra indítjuk. A viszonyréteg az adatfolyamban elhelyez a másik viszonyrétegnek szóló szinkronizációs pontokat, és a megszakadt összekötettés, a viszonyréteg által szolgáltatott információk alapján, a megszakadástól folytatódik.

Fontos feladata a viszonyrétegnek a párbeszédes kapcsolatok kezelése. Ez azt jelenti, hogy bár a szállítási réteg teljes duplex kapcsolatot biztosít, de egy kérdés, és rá a felelet fél-duplex kapcsolat használatát igényli. Ha például a szállítási réteg képes több kérdés fogadására, akkor a viszonyréteg feladata a soron következő kérdések számontartása és rájuk a válaszok kikényszerítése.

A gyakorlati megoldása ennek az **adatvezérjel (token)** bevezetése: mindig csak a vezérjelet birtokló küldhet adatot, addig a másik félnek hallgatni kell. Az adatküldés befejezésekor az adatvezérjelet átadja a másik oldalnak, és így a helyzet megfordul.

## 6.3 Megjelenítési réteg

A megjelenítési réteg felelős az információ megjelenítéséért és egységes értelmezéséért, a feladata a szállított **információ jelentéséhez** kapcsolódik:

- adatábrázoláshoz
- adattömörítéshez
- hálózati biztonsághoz, védelemhez.

### 6.3.1 Adatábrázolás

Az adatábrázolás gyakorlati megvalósításakor, a küldőnek, és a fogadónak egységes adatábrázolásban kell megállapodnia.

Például több bájtot igénylő számábrázolásnál, nem mindegy, hogy a számot a nagyobb értéktől kezdve tároljuk, és a kisebb van a végén (little endian), vagy éppen fordítva (big endian).

Vagy gondoljunk az arab, vagy héber írásmódra: az írás (és az olvasás) is balról jobbra történik, és a könyvek eleje és hátulja is fordítva van.

A különféle számítógépek különböző adatábrázolási módokat használnak. Ez karakterek esetén lehet különböző kódrendszerek használata (az IBM nagy gépek EBCDIC-kódja vagy az ASCII kód), de lehetnek a számábrázolásban is különbségek.

A karakterábrázolással kapcsolatosan pozitív dolog, hogy szerencsére csaknem egységes az **ASCII kódrendszer** használata, és - ahogy ezt már az előzőekben leírtuk - a nemzeti karakterek használatát a kódlapok, illetve ma már a **UNICODE** használata teszi lehetővé. Ezek leírását a 3. fejezetben szerepeltettük.

Ha két gép között ilyen eltérések vannak, akkor a hálózati kapcsolat során átvitt adatokat a megfelelő reprezentálás érdekében átalakítani, konvertálni kell. Struktúrált adatok esetén pl. rekordok esetén a helyzet bonyolultabb, mivel egyes mezőket kell konvertálni, míg másokat nem.

Az adatábrázolásból adódó problémák kezelése nem egyszerű: a küldőnek vagy a vevőnek kell biztosítania az átalakítást? Célszerű-e valami általános hálózati formátu-

mot használni, és erre átalakítva lehetne adatot a hálózaton átküldeni? Az ide kapcsolódó kérdések tárgyalása meghaladja a könyv kereteit, többet az [1] irodalomban olvashatunk róluk.

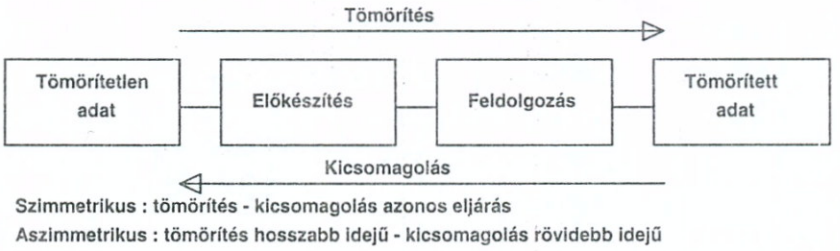
## 6.3.2 Adattömörítés

Mivel a hálózatok használatáért általában fizetni kell, és az információátviteli sebesség sem közömbös, egyáltalán nem mindegy, hogy egy időegység alatt mennyi információt viszünk át rajta. Az adatok ábrázolása általában **redundáns**. Ez azt jelenti, hogy ugyanazt az információt különböző mennyiségű adat hordozhatja anélkül, hogy megjelenítése megváltozna. Redundáns egy adathalmaz, ha mennyisége több mint amennyi az információ hordozásához és megjelenítéséhez szükséges lenne. Ilyen módon azok az információk, melyek az információ hordozásához és megjelenítéséhez nem kellenek, elhagyhatók, anélkül hogy a megjelenítendő információ megváltozna.

Tömörítéskor az adathalmaz különböző típusú redundanciáira alapozva csökkenthető az információt hordozó adatok mennyisége. A redundancia több módon keletkezhet, például a **kódolási redundancia** esetén az adathalmazban az adatkódok összes variációjából, annak csak töredékét használjuk fel. Másfajta redundancia van akkor, ha egy mozgófilm képkockáit egymástól függetlenül tároljuk, hiszen az egymás utáni képkockák, csak kis mértékben térnek el egymástól, ilyenkor elég lenne csak az eltérés megadása.

A csatornán elküldött információt szimbólumsorozatként is felfoghatjuk, amelyek egy adott szimbólumkészletből származnak, pl. decimális számjegyek készlete, karakterek készlete, stb. Az adattömörítés elvét a *6-2.ábra* mutatja.

A tömörítés során – ha jó tömörítést akarunk elérni – az információ egy, lényegtelen részét, eldobhatjuk (pl. képtömörítésnél). Ezt nevezzük **veszteséges tömörítésnek**. Ilyenkor a kicsomagolás csak az információ jelentős részét állítja vissza. Természetesen adattömörítésnél ez az eljárás nem használható, csak az ún. **veszteségmentes tömörítés**. Két fontos mennyiségi jellemzőt kell ismernünk:




***6-2. ábra: A tömörítési módszer***

**Tömörítési arány:** Azt fejezi ki milyen mértékben csökkent a tömörített állomány mérete az eredeti állapothoz képest. Például 1:10 azt jelenti, hogy az új állomány mérete az eredeti tizede. Ez megfelel a 10-szeres tömörítésnek.

**Veszteségi tényező:** Veszteséges tömörítésnél a tömörítő bizonyos adatokat eltávolít az állományból, ezért nem állítható vissza az eredeti állápot. A mutató megmutatja mekkora adatmennyiség veszett el az eredeti adatmennyiségből. A következőkben néhány tömörítési eljárást ismertetünk.

### *Tömörítési eljárások*

**Darabszám kódolás:** Ha egy adathalmazban sok egymás után következő azonos

szimbólum fordul elő, célszerű egy külön szimbólumot fenntartani az ismétlődés jelzésére, és utána következik az ismétlődő szimbólum, míg az azt követő számérték jelzi az ismétlődő szimbólumok számát:

Például a felkiáltó jel legyen az ismétlődés jelző: !

! (szimbólum) <x> azt jelenti, hogy a szimbólum <x>-szer ismétlődik (x számérték 0-255 között)

!A112 -> 112 A betű egymás után

!! - > maga a ! jel

**Szimbólumsor helyettesítés:** gyakori **azonos** szimbólumsor helyett egy speciális szimbólum. Például ilyen a tabulátor (TAB) jel is (több betűközt helyettesít)

**Mintahelyettesítés:** gyakori szimbólumsorozat helyettesítése speciális szimbólummal. Például a BASIC nyelvben szereplő definiált alapszavak (INPUT, FOR, stb.) a program futtatása előtt bájtokban tömörítik, úgy, hogy a változókat, adatokat leíró ASCII karakterrel ábrázolt részeket változatlanul hagyják, (legfelső bit =0), míg az alapszavakhoz a maradék 128 kód valamelyikét rendelik hozzá (legfelső bit=1). Például: 80H->FOR, 81H->INPUT, stb. Ezzel az ún. tokenizálással a szöveges BASIC program mérete jelentősen csökkenthető.

**Sorozathossz kódolás (Run Lenght Encoding = RLL):** Sok nullát tartalmazó bináris sorozatokban a nullák számát bináris számként adjuk meg. Például:

```
000100100000100001      -> a nullák száma: 3 2 5 4
     18 bit             -> 3 bittel kódolva : 011 010 101 100  és ez csak 12 bit
```

**Statisztikai kódolás:** a kódhossz a kód előfordulási gyakoriságától függ. Ennek az a lényege hogy a információt leíró kódhalmazban a kódok hosszát azok gyakorisága alapján állapítjuk meg. Statisztikai kódolásra jó példa Morse ABC. Itt az angol szövegek leggyakoribb betűjének, az „e"-nek a kódja a legrövidebb: a pont. A szintén gyakori „t" — nek pedig a vonás. Ilyen vonatkozásban, az ASCII kódrendszer - ahol minden karakter hossza egyformán 7 bit - nem jó megoldás.

HUFFMANN-KÓDOLÁS

Eredeti szöveg: KEREKES SZEKEREK MENNEK

| Statisztika, | kódolás: |
|---|---|
| 8db E | 00 |
| 5db K | 01 |
| 2db R | 10 |
| 2db S | 1100 |
| 2db N | 1101 |
| 2db BETŰKÖZ | 1110 |
| 1db M | 11110000 |
| 1db Z | 11110001 |

Hatékonyság:

Eredeti szöveg: 184 bit

Kódolt szöveg 70 bit

01 00 10 00 01 00 1100 1110 1100 11110001 ..... KÉT RÉSZ: BETŰSZÓTÁR

K E R E K E S sp S Z KÓDOLT SZÖVEG

*6-3. ábra Példa: Huffmann kódolás*

**Huffmann kódolás:** egyes jelek, vagy bájtsorozatok előfordulási gyakoriságát figyeli, és generált kód hossz ettől függ. (*6-3 ábra*). Figyeljük meg az ábrán a kódolást: nem kell elválasztó jel a kódok között!

**Transzformációs kódolás**: Ilyen például a Fourier transzformáció: egy periodikus időfüggvényt adott amplitudójú és kezdeti fázisszögű szinuszhullámok összegével írunk le.

Általánosabban fogalmazva nagyon sok folytonos összetett jelenség írható le vektorizációval: egy kép leírható alapalakzatok (körök, vonalak, és koordinátáik) segítségével (vektorgrafika). Egy zeneszámot MIDI formátumban szintén alap hang és ritmusegységek írnak le, stb.

**Subband kódolás:** csak bizonyos frekvenciatartományba eső jeleket transzformáljuk (pl. telefon: 0-4 kHz)

**Predikció vagy relatív kódolás:** ha az egymást követő jelek nem sokban térnek el egymástól, akkor elég a kis különbségeket kódolni.

Ezek az előbbiekben felsorolt módszerek a gyakorlatban mind használhatók, a be- és kitömörítést programok, vagy jelenleg már egyre inkább hardver (egy chip) segítségével oldják meg.

**Video és hang tömörítések:** Az emberi képi és hangi érzékelés tökéletlenségének kihasználásával is tömöríthetünk. Noha ez információvesztéssel jár, de mivel az elhagyott információt úgysem érzékelnénk, ezért elhagyható.

**MP3: „Nem azt halljuk, ami elhangzott, de azt halljuk, amit hallottunk volna"** Az MP3 a fület csapja be, működése az emberi hallás ismeretén alapul. Egy zajtömeg érzékelésekor csak bizonyos hangokat fogunk fel. Az erősebb hangok elnyomják a gyengébbeket, és bizonyos magasságtartományban lévő hangokat sokkal jobban hallunk, mint a rajta kívül esőket. Az, hogy egy pillanatban sok hang közül melyek jutnak el tudatunkig, leírható szabályokkal. A tömörítés kitöröl minden részletet az eredeti hanganyagból, amit fülünk, agyunk amúgy sem érzékelne, így az adatokat összezsugorítva is jó minőségű hangzást észlelünk, körülbelül ugyanazt, mintha az eredeti szólna. Az MP3 nem szimmetrikus: a betömörítés tovább tart, mint a kicsomagolás.

Finomabb fülű emberek észreveszik a különbséget: az eredeti felvétel teltebb, gazdagabb, árnyaltabb hangzású. Az MP3 a legkevésbé hűen azokat a hangokat adja vissza, amelyek hangmagasságban közel állnak egymáshoz.

**Állókép tömörítések** esetén, a képpontokat tartalmazó képeket blokkokra osztjuk fel, és a blokkok változásait tároljuk képinformációként.

**Mozgókép tömörítések:** Ezek az egyes képkockákat állóképként tömörítik. Lejátszás során, ezek gyorsan, egymás után jelennek meg, ezért az egyes képkockák tömörítése kisebb igényű, mint az állóképeknél, de a feldolgozási időnek gyorsabbnak kell lennie. Lejátszáskor kb. 30 képkocka/másodperc lejátszási sebesség szükséges. Az MPEG tömörítés kihasználja, hogy az egymás utáni képkockák általában kis mértékben térnek el egymástól, és ezért, csak a szomszédos képkockák megváltozásait tárolja.

### *Példa tömöritésre: a fax kódolás.*

Egy átlagos letapogatott (beszkennelt) szöveges oldalon sok redundáns információ van, hiszen igen sok ismétlődő egymás utáni fekete és fehér képpontot tartalmaz. Faxon történő továbbítás előtt érdemes valamilyen módon a küldendő adathalmazt tömöríteni. A letapogatásnál kétfajta felbontást lehet használni: normált és finomat. A következő technikai adatok jellemzik a fax letapogatást:

- Függőleges felbontás: normál: 3.85 vonal/mm, finom 7.7 vonal/mm.
- Vízszintes felbontás: soronként 8 képpont/mm. Egy sor 216 mm hosszú, ez 1728 képelemet jelent soronként.

Ez azt jelenti, hogy 1 négyzetmilliméteres felületen normál esetben 32, finom felbontás esetén 64 pixel van. Bár az átvitel fekete-fehér képpontokkal történik, de választhatóan a képpontok sűrítésével/ritkításával szürkeségi fokozatokat utánzó féltónusos átvitel is lehetséges.

A sorok kódolása az adott sorban egymás után lévő azonos állapotú (fekete vagy fehér) képpontok, ún.

futamhosszak leírásával történik. A vonalszakaszok hosszát a képelemek számával adjuk meg, mindegyikhez egy kódszót rendelve két táblázat szerint. Mindkét táblázatnak két oszlopa van: az egyik oszlopban a fehér, a másik oszlopban a fekete képpontokhoz tartozó kódok vannak.

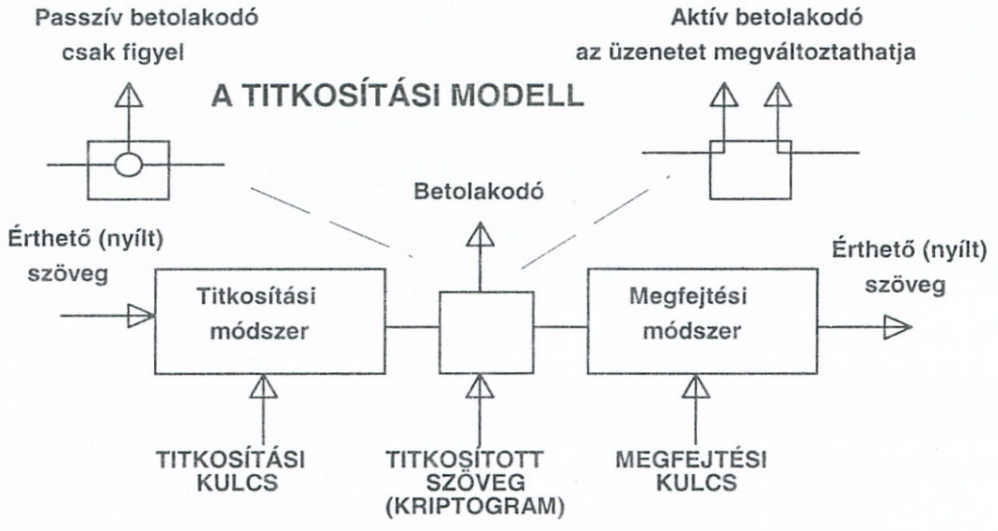
Az ún. **kezdő kódtáblában** az egyik oszlopban a sorok a 64 és többszörösei képpontszámhoz tartozó kódokat tartalmazzák, míg a másik ún. **befejező kódtáblában** pedig a 0-tól 63-ig terjedő képpont sorozati hossz, illetve a hozzá tartozó kód található.

Ha a sorban lévő azonos képpontok hossza 64 egész számú többszöröse, akkor ennek kódolása a kezdő kódtábla megfelelő kódszava, és a befejező kódtábla nulla hosszúságú kódszavával történik. Ha a kódolandó futamhossz 0-63 között van, akkor a befejező kódtábla megfelelő kódszavát használjuk. A kódszavak önhatárolók, azaz egymás mellé írásuk esetén is szétválaszthatók. A különböző futamhosszakhoz az előfordulási valószínűségük alapján rövidebb (gyakori előfordulás) vagy hosszabb (ritkább előfordulás) kódot rendelnek. Minden sort egy speciális kód a sorvég (End Of Line =EOL) zárja. Egy letapogatott oldal átvitelekor, elsőként szinkronizálási célból egy EOL kódot továbbítanak, majd következnek az első sor kódszavai. A sor végét egy EOL zárja. Ha egy sor rövidebb (mert a maradék részen nincs változás), akkor 0 hosszúságú futamhosszakat iktat be (töltelékbiteket) iktat be a rendszer az EOL kódig.

A képtartalom nem hibavédett. Ha egy sorban hibás kódszó érkezik, akkor a vevő kieshet a szinkronból. Az újra-szinkronozás a következő EOL hatására következik be. Az oldal végét 6 egymást követő EOL jelzi. Illusztációként a kezdő és befejező kódtábla néhány eleme:

| FEHÉR KÉPPONTOK SZÁMA | KÓDSZÓ | FEKETE KÉPPONTOK SZÁMA | KÓDSZÓ |
|---|---|---|---|
| EOL | 000000000001 | | |
| KEZDŐ KÓDTÁBLA | | | |
| 64 | 11011 | 64 | 0000001111 |
| 128 | 10010 | 128 | 000011001000 |
| ... | | | |
| 1728 | 010011011 | 1728 | 0000001100101 |
| | | | |
| BEFEJEZŐ KÓDTÁBLA | | | |
| 0 | 0011010 | 0 | 0000110111 |
| 1 | 000111 | 1 | 010 |
| 2 | 0111 | 2 | 11 |
| 3 | 1000 | 3 | 10 |
| 4 | 1011 | 4 | 011 |
| ... | | ... | |
| 63 | 00110100 | 63 | 000001100111 |

## 6.3.3 Adattitkosítás




*6-4. ábra: A titkosítási modell*

A következő témakört itt csak röviden, elméleti szempontból foglaljuk össze, és a 9. fejezetben, a hálózati biztonsággal foglakozó részben térünk ki a gyakorlati részletekre. Gyakran előfordul, hogy bizalmas vagy titkos információt, banki átutalásokat kell továbbítani a hálózaton keresztül. Megoldandó,

hogy az arra jogosulatlan személyek, ne férhessenek hozzá a titkosított adatokhoz. Megfelelő titkosítási algoritmus felhasználásával elérhető, hogy a titkosított adatok csak nem, vagy csak igen nehezen legyenek megfejthetők.

A titkosítástan (kriptológia) alapvető szabálya az, hogy a titkosítás készítőjének feltételeznie kell, hogy a megfejtő ismeri a titkosítás általános módszerét *(6-4. ábra)*. A módszernél a titkosítási kulcs határozza meg a konkrét esetben a titkosítást.

A titkosítási-megfejtési módszer régen nem lehetett bonyolult, mert embereknek kellett elvégezni. **A titkosításnak, vagy rejtjelezésnek két általános módszere ismert:**

- **helyettesítéses rejtjelezés:**
- **felcseréléses rejtjelezés**

A következőkben néhány példával illusztrálva bemutatjuk a klasszikus rejtjelezési megoldásokat, módszereket.

### *Helyettesítéses rejtjelezés*

A helyettesítéses rejtjelezések és kódolások **a nyílt szöveg szimbólumainak sorrendjét változatlanul hagyják, csak valamilyen módszerrel álcázzák.** Néhány példa:

**Egyábécés helyettesítés:** Első híres alkalmazójáról Julius Ceasar-ról elnevezve szokták Ceasar-féle rejtjelezésnek is hívni. Az eredeti abc-t egy három (általános esetben: k) karakterrel eltolt abc-vel helyettesíti, és így írja le a szöveget. Bár a lehetőségek száma nagy, de a nyelvi-statisztikai alapon könnyen fejthető. (betűk, szavak relatív gyakorisága alapján.)

**Többábécés rejtjelezés:** 26 Ceasar-abc sort tartalmazó négyzetes mátrix. Nyílt szöveg fölé egy kulcsot (egy szöveget) írunk, és a kulcsban lévő betű dönti el, hogy melyik sort használjuk az adott nyílt szövegbeli betű titkosítására. A megfejtés alapja: a kulcs hosszának jó megsejtése.

```
ABC....XYZ      KULCSOCSKAKULCSOCSKA
BCD...YZA       EZ A NYILT IRAT
...             OT.......
ZAB...WXY
```

- E betűt az O betű helyettesíti, mert a K-val kezdődő sor 5.-ik (E betű az 5-ödik!) tagja O.
- Z betűt az T betű helyettesíti, mert a U-val kezdődő sor 26.-ik (Z betű a 26-odik!) tagja T.

**Porta-féle rejtjelezés**-nél 26*26-os mátrixot használunk, amelynek minden eleme betűpár. A nyílt szöveg sorban egymás után álló két karaktere a mátrix egy sorát és oszlopát határozza meg, a metszéspontban lévő betűpárt írjuk az eredeti betűpár helyére.

Gyakran használt megoldás volt a **kódkönyves rejtjelezés.** Egy kódkönyv tulajdonképpen nem más, mint szavak és kifejezések listája, ahol az egyes címszavak mellett - akárcsak egy szótárban - megtalálhatók a hozzájuk tartozó betű- illetve számkombinációk. Tulajdonképpen ez is egy helyettesítéses eljárás, csak jóval nagyobb halmazo-

kon elvégezve. Tipikus alkalmazási területe volt a diplomácia, ahol nem túl hosszú üzeneteket kellett sokszor kézzel rejtjelezni, illetve visszafejteni. Sajnos – mivel a kódkönyvnek minden olyan szót tartalmaznia kell, amelyre eseteg szükség lehet – a gyakorlatban ennek csak a töredékét használták fel. Természetesen a kódkönyv megszerzése esetén a feltörés már nem probléma.

### *Felcseréléses rejtjelezés*

**A titkosított szöveget a nyílt szövegben szereplő betűk sorrendjének megváltoztatásával hozzuk létre.**

A helyettesítéses rejtjelezések és kódolások a nyílt szöveg szimbólumainak sorrendjét változatlanul hagyják, csak álcázzák. A felcseréléses rejtjelezések a betűk sorrendjét változtatják, de nem álcázzák.

Módszer: A kulcsban egy betű csak egyszer fordulhat elő. A szöveget kulcsnyi szélességű sorokra tördelve egymás alá írjuk, a titkosított szöveget megkapjuk az oszlopok egymás után fűzésével. Az oszlopok leírási sorrendjét a kulcs betűinek abc-beli sorrendje határozza meg.

```
Nyílt szöveg:      eztmostkodoljuk
Titkosítva:        odkmoutkjesoztl
PROBA        <-    kulcsszó
45321        <-    kiolvasási sorrend
eztmo
stkod
oljuk
```

Ez a rejtjelezés is megfejthető. Betűgyakoriságok vizsgálata alapján eldönthető, hogy felcseréléses rejtjelezésről van szó. Majd az oszlopszámokat kell megsejteni, majd az oszlopok sorrendjét.

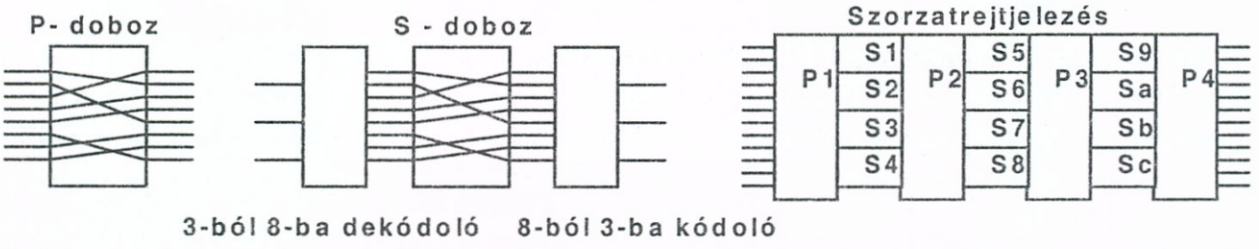
### *DES (Data Encyption Standard) — Adattitkosítási szabvány*

Régen emberek voltak a titkosítók, ezért a készítők egyszerű, emberek által jól megtanulható algoritmusokat és hosszú kulcsokat használtak. A számítógépek megjelenésével a hagyományos módszerek (helyettesítés és felcserélés) tovább élnek, de a hangsúly máshová került. Mivel a kódolt információt bicsoportok hordozzák ezért ezeket a bitcsoprtokat kell titkosítani.

A számítógépek megjelenésével felmerült az igény olyan titkosítási algoritmusok iránt, amelyek olyan komplikáltak, hogy még egy számítógép se tudja megfejteni. Manapság a titkosítási algoritmus a nagyon bonyolult (hiszen a számítógép végzi), és a betolakodó (megfejtő) - még számos titkosított szöveg birtokában sem - tudja megfejteni.

**A DES módszer egy 64 bites nyílt szöveget, 64 bites titkosított szöveggé alakít, egy 56 bites titkosítási kulcs segítségével.** Vagyis van egy olyan áramkör, amelynek 64 bemenete és 64 kimenete van. Ezen kívül, van még 56 bemenete, aminek 0-1 állapotait a csatlakoztatott kulcs határozza meg. Feldarabolva a nyilt szöveget 64 bites részekre és a DES doboz bemenetére juttatva, a 64 bemeneti bit 64 kimeneti bitté alakul (titkosítás), ami az 56 bites kulcstól függ. A titkosított szöveget visszafejtés előtt szintén 64 bites részekre darabolva a DES doboz bemenetére küldve, és ugyanazt a kulcsot felhasználva, visszakapjuk az eredeti nyilt szöveget. Ezt a mód-

szert szokták **egykulcsos titkosítás**-nak is nevezni. Fontos megjegyezni, hogy tetszőleges számú összetartozó nyílt-titkos blokk elemzése alapján sem lehet a kulcsot megfejteni!




*6-5. ábra Bitcsoport felcserélése és helyettesítése*

A felcserélések és helyettesítések egyszerű digitális jombinációs áramkörök segítségével valósíthatók meg *(6-5. ábra).* A felcseréléseket a P doboz, a helyettesítéseket az S doboz végzi. A P doboz nem más, mint egy 8 bemenetű és 8 kimenetű áramkör, egy bemenő paraméter által meghatározott módon, a bemeneti 8 bitet felcserélve állítja elő a 8 bites kimenetet.

A helyettesítést az S doboz végzi, ez a doboz a bemenetére adott 3 bit nyílt szöveget alakítja át 3 bit titkos szöveggé. (3 ->8 dekóder és 8->3 dekóder).

A titkosítás első lépésben egy kulcstól független felcserélés történik, az utolsóban, pedig ennek az inverze. Az utolsó lépésben egyszerűen az első 32 bitet felcserélik az utolsó 32 bittel. A közbülső 16 fokozat ugyanúgy működik, de a kulcs más-más része határozza meg az alkalmazott P és S dobozok konkrét felépítését. Ez természetesen logikai függvényekkel is leírható, és ez alapján titkosító program is készíthető. Egy olyan digitális áramkör is elkészíthető, amelynek 64 bemenete és 64 kimenete van. A további 56 kulcsbemenetre adott titkosító bicsoport fogja meghatározni azt, hogy konkrétan hogyan képződik a 64 bemenetből a 64 kimenet.

DES megfejtése – mivel az algoritmus közismert — az összes kulcskombináció kipróbálásával oldható meg. A feltörési folyamat automatizálható, és ez - egyszerre sok számítógépen – a kiosztott kulcs-csoportok párhuzamosan történő végigpróbálásával lehetséges.

Számítógépes környezetben, ha a két fél egymástól fizikailag távol van, akkor nehéz a kulcsot megbízható csatornán eljuttatni a másik félnek (például személyesn átadni).

### *Nyilvános kulcsú titkosítás*

Az egykulcsos titkosítást végző legbonyolultabb algoritmusnak semmi haszna, ha a használt kulcsot megszerzik a betolakodók. **A megoldás a két kulcsot: egy nyilvános (publikus), és egy titkos kulcsot használó titkosítási eljárás.**

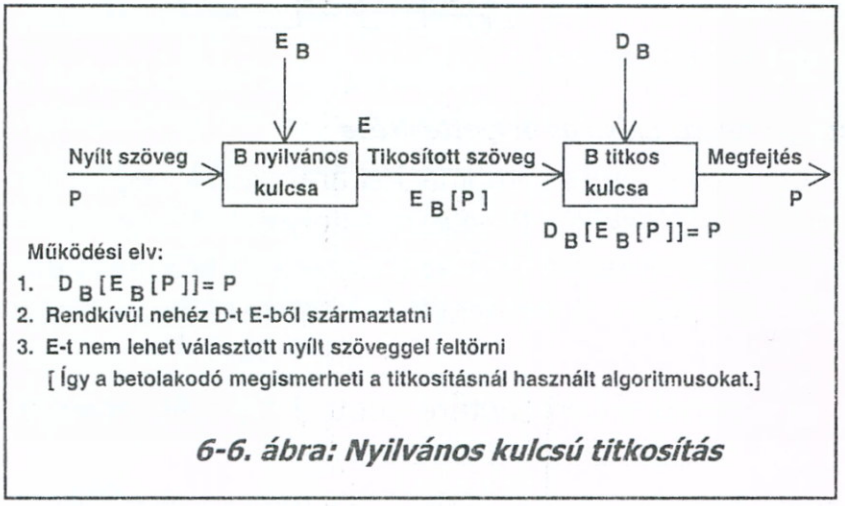
E probléma megoldása egy olyan **E** titkosítási algoritmus, és olyan **D** megfejtési algoritmus használata, amelyeknél a D kikövetkeztetése gyakorlatilag akkor is lehetetlen marad, ha E teljes leírása hozzáférhető. Ha P a nyílt szöveg, akkor E(P) - azaz az E titkosítási algoritmust alkalmazva a P nyílt szövegre – hozza létre a titkosított szöveget.

Az algoritmusnak a következő követelményeknek kell megfelelni:

- D(E(P))=P (Itt P a nyílt szöveget jelenti.)
- Rendkívül nehéz D-t E-ből leszármaztatni
- E-t nem lehet választott nyílt szöveggel feltörni

Az első követelmény azt mondja ki, ha D-t egy titkos szövegre E(P)-re alkalmazzuk, akkor a nyílt szöveget, P-t kapjuk vissza. A második követelmény magáért beszél. A harmadik követelményre azért van szükség, hogy a betolakodók az algoritmussal megismerkedhessenek.

Innen származik a **nyilvános kulcsú titkosítás** elnevezés. Ilyen feltételek mellett valóban nem volna oka E eltitkolásának, mert bármely két személy, akik titkos üzenetet akarnak váltani egymással, először kidolgoznak két, a fenti követelményeknek megfelelő algoritmust, E-t és D-t. A titkosítási algoritmusokat ezután nyilvánossá teszik. Az algoritmus, és az ezen alapuló számítógépes módszer a **Pretty Good Privacy (Kellemes biztonság), röviden: PGP.** Az algoritmus azon a matematikai tényen alapul, hogy igen nehéz két sokszámjegyű szám szorzatának ismeretében a két összeszorzandó számot meghatározni, vagyis nincs ezt a feladatot megoldó algoritmus, azaz a megfejtés a gyakorlatilag reménytelen próbálkozással lenne lehetséges.




*6-6. ábra: Nyilvános kulcsú titkosítás*

### *Hogyan működik PGP?*

A hagyományos titkosítási módszereknél, (mint pl. az előbbiekben bemutatott DES) ugyanazt a kulcsot kell használni a kódoláshoz és a dekódoláshoz. Ha valakinek egykulcsos rejtjelzéssel küldünk titkosított üzenetet, a kódoláshoz egy kulcsot használunk, amit a címzettnek is használnia kell az üzenetet visszafejtéséhez.Ez viszont azzal jár, hogy a kulcsot először egy megbízható csatornán el kell juttatni a fogadó félhez, és csak utána lehet kezdeni, a titkosított üzenetek küldését.

**A nyilvános kulcsos rejtjelezésnél mindenki két, egymással szorosan összefüggő kulccsal rendelkezik.** Az egyik kulcs nyilvános, azaz mindenki által hozzáférhető, míg a másik a titkos.

A program, amivel a kulcsokat generáljuk, mindenki által elérhető, a neve: PGP.

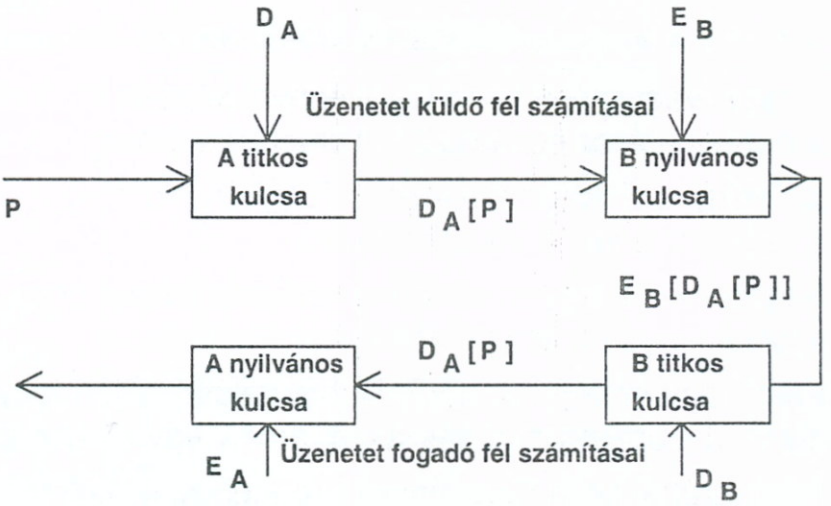
Mindkét fél, „A" és „B", aki egymással titkos üzenetvátást akar megvalósítani, **a nyilvános PGP program felhasználásával generál egy kulcspárt**. Az „A" személy által generált kulcspár közül az egyik tetszőlegesen választott lesz az „A" személy tikos kulcsa: ATK, a párja az „A" személy nyilvános kulcsa: ANK. Ezek után, **„A" elküldi a nyilvános kulcsát (ANK-t) „B"-nek, és „A" pedig megkapja „B" nyilvános kulcsát BNK-t.** „A" úgy ír titkosított levelet „B"-nek, hogy elküldése előtt azt BNK-val titkosítja. Az elküldött levelet csak „B" tudja elolvasni, miután a kizárólag nála lévő BTK kulcsával visszafejtette. „B" pedik „A" nyilvános kulcsával (ANK) titkosítja „A"-nak szóló leveleit, amit „A" saját titkos kulcsával (ATK) tud visszafejteni, és elolvasni.

A PGP program a kulcspárok generálásán túlmenően, képes a nyílt szöveget egy kulcs segítségével titkosítani, illetve a titkosított szöveget (jó kulcsot felhasználva), vissza-

fejteni. Mivel a kulcsok a valóságban hosszú, meg nem jegyezhető karaktersorozatok, ezért minden kulcshoz tartozik egy rövid, választható azonosító, amit a kulcs helyett használhatunk, és a PGP program egy fájlban tartja nyilván az azonosító-kulcs összerendeléseket. Mivel általában nem csupán egy személlyel akarunk titkos levelezést folytatni, ezért minden partner vonatkozásában létre kell hozni a kulcspárokat. A kapcsolatok során az elküldött nyilvános kulcsokhoz tartozó titkos kulcsokat egy fájlban **„titkos kulcskarikán"** tároljuk, míg a kapott nyilvános kulcsokat egy másik fájlban a **„nyilvános kulcskarikán"** tároljuk. A kulcsokra a rövid azonosítóval hivatkozunk. A kulcsok speciális struktúrában vannak tárolva, mely tartalmaz egy azonosítót (userId - a személy neve), a kulcspár generálásának dátumát, és magát a kulcskódot.

A nyilvános kulcsot széles körben el lehet terjeszteni a kommunikációs hálózatokon. Egy nyilvános kulccsal bárki kódolhat üzeneteket. Ezeket a fogadó a saját titkos kulcsával tudja csak kibontani. Más nem, hiszen a titkos kulccsal senki más nem rendelkezik. A kódolt üzenetet még a feladó sem tudja dekódolni. Lényeges, hogy a nyilvános kulcs sem nyújt lehetőséget a titkos kulcs kitalálásához.

Megoldható az üzenetek biztonságos sértetlenség- és eredetigazolása (autentikációja) is. Ekkor küldő saját titkos kulcsával "aláírja" az üzenetet (hozzáír egy részt a saját titkos kulcsával titkosítva.) Az aláírás eredetiségét bárki ellenőrizheti az illető nyilvános kulcsának felhasználásával. A fogadó a dekódolás után bizonyos lehet a küldő személyében, és abban hogy az üzenet tartalma nem változhatott meg. Mindezt azért, mert ehhez a feladó titkos kulcsa szükséges, mellyel rajta kívül senki nem rendelkezik. A hamisítás kizárt, és a küldő nem tagadhatja le magát utólag.




*6-7. ábra: Nyilvános kulcsú titkosítás aláírással*

**Az eredetigazolás és a biztonság egyszerre is megoldható.** Ehhez először alá kell írni az üzenetet a saját titkos kulccsal, majd az egészet kódolni a fogadó nyilvános kulcsával. A fogadó pedig először a saját titkos kulcsával dekódol, majd a küldő nyilvános kulcsával ellenőrzi az eredetet. *(6-7. ábra)* Jelenleg – ezek az eljárások – már a korszerű levelező rendszerekbe épülve – is használhatók.

## 6.3.4 Lenyomatkészítő függvények (hash algoritmusok), MD5

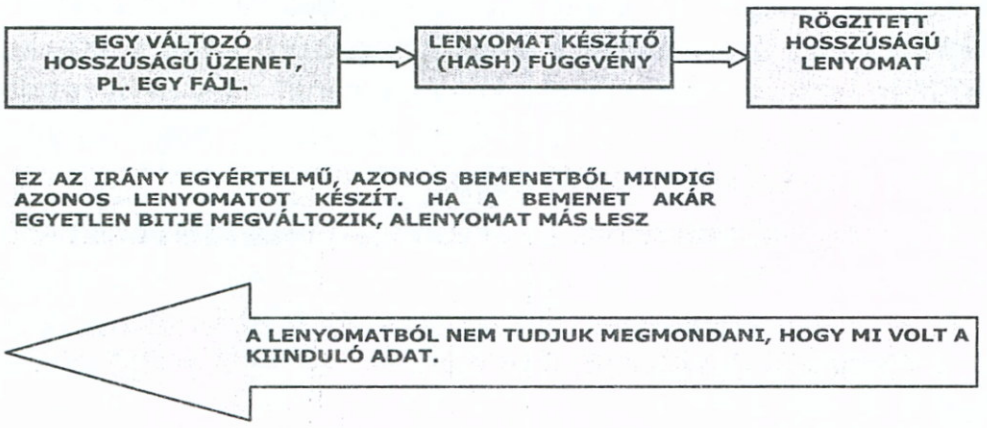
Nagyon sok esetben elegendő az adatok változatlanságának az igazolása, vagyis azt kell bizonyítanunk, hogy egy adatátviteli csatornán átjövő anyagot nem változtatta meg senki. Ezt a feladatot megoldó **lenyomatkészítő függvények** (más néven hash függvények) tetszőleges méretű bemenő adatsorból adott méretű kivonatot képeznek úgy, hogy magából a kivonatból a gyakorlatban ne lehessen következtetni a

kiindulási adatra, illetve nagyon nehéz legyen olyan másik adatsort készíteni, aminek ugyanaz lesz a kivonata.

A lenyomatkészítő függvényeket több titkosítási módszer is felhasználja, alapvető szerepe van például az **üzenet sértetlenségét** bizonyító eljárásokban.

Matematikai értelemben a lenyomatkészítő függvények (hash függvények) egy korlátlan halmazt (tetszőleges szöveget) képeznek le egy korlátos halmazra (fix hosszúságú bájt-sorozatra).

Elméletileg bizonyítható, hogy egy adott kivonat több különböző kiinduló szövegből is előállítható, ennek ellenére a hash függvények a gyakorlatban mégis alkalmasak lehetnek egy szöveg azonosítására, ugyanis a kiindulási szövegek általában „értelmesek" (azaz valamely emberi nyelven értelmes mondatokat tartalmaznak).




*6-8. ábra A lenyomatkészítés (hash algoritmus) módszere*

Annak a valószínűsége gyakorlatilag elhanyagolható, hogy egy másik értelmes szövegnek is ugyanez legyen a lenyomata. Vagyis egy hash függvénnyel képzett kivonat gyakorlati értelemben egy az egyben meghatároz egy adott dokumentumot még akkor is, ha jóval rövidebb annál.

Ahhoz, hogy egy lenyomatkészítő függvény kriptográfiai célokra jól alkalmazható legyen, teljesülnie kell a következőknek:

- Nagyon kicsi legyen annak a valószínűsége, hogy két tetszőleges bitsorozatnak azonos legyen a lenyomata.
- Igen nagy számításigényű, illetve valószínűtlen legyen az, hogy valaki olyan kiindulási dokumentumot készítsen, amely egy megadott lenyomatot eredményez.

Egy kriptográfiai hash függvénynek rendelkeznie kell tehát az alábbi tulajdonságokkal:

- **Méretcsökkentő tulajdonságú**, azaz változó hosszúságú bemeneteket rövidebb, fix hosszúságú kimenetekké alakít át.
- **Egyirányú függvény**, azaz olyan $h(x)$ függvény, hogy *x forrás bitsorozat* ismeretében a lenyomat $y=h(x)$ *könnyen* számolható, de adott $y$-hoz a megfelelő $x$ meghatározása *nehéz probléma*.
- **Ütközésmentes**, azaz *nehéz* olyan egymástól különböző $x$ és $x'$ üzeneteket találni, amelyekre $h(x)=h(x')$ teljesül.

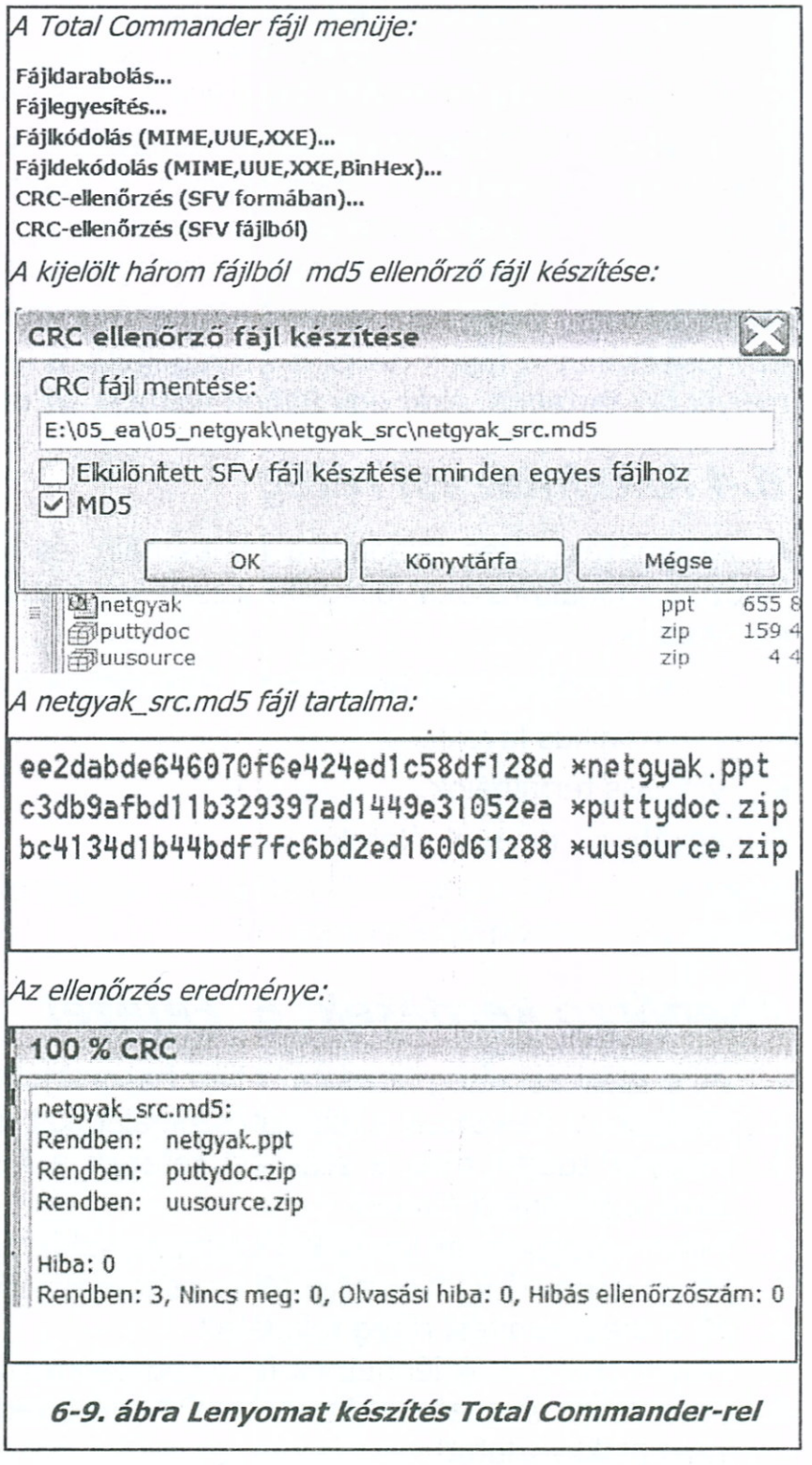
### *A lenyomatkészítő függvények felhasználási területei*

**Üzenethitelesítés:** az üzenet megváltozása felismerhető, ha külön kezeljük az üzenetet és a lenyomatát. A későbbiekben tárgyalt digitális aláírás titkosítási módszerénél használják.

**Jelszó alapú távoli partnerazonosítás:** a jelszavaknak csak a lenyomata kerül tárolásra, illetve nyilvános csatornán továbbításra. Jelentős szerepet kap hálózati bejelentkezéseinknél, hogy jelszavakat ne kelljen olvasható formában átvinni a hálózaton. Ilyenkor titkosítás helyett használható, mert felhasználva, biztonságos jelszóátvitelt tesz lehetővé. Hogyan történik ez a gyakorlatban?

1. A munkaállomás bekéri a felhasználó jelszavát bejelentkezéskor.
2. Elkészíti ennek a lenyomatát.
3. A végeredményt a felhasználónévvel együtt elküldi a szervernek, ahol megvan a felhasználó éppen érvényes jelszavának lenyomata,
4. Összehasonlítja a hálózaton keresztül kapott lenyomattal. Ha a kettő egyezik, akkor a jelszó érvényes.

**Gyorsítótárak (cache) működésénél.** Kiszámítjuk, és eltároljuk a gyorsítótárban elhelyezett blokkokba szervezett adatok lenyomatát, amelyek jóval kisebb helyet foglanak el, mint az eredeti blokk, amiből származtattuk. Ezek után annak eldöntéséhez, hogy a keresett blokk bent van-e a gyorsítótárban nem a teljes blokkra, hanem annak lenyomatára keresünk, vagyis hasonlítjuk össze a keresett blokkok lenyomatával.




***6-9. ábra Lenyomat készítés Total Commander-rel***

### *Az MD5 lenyomatkészítő függvény*

Bár több, a szakma által kriptográfiai célokra is kellő erősségűnek elfogadott kriptográfiai lenyomatkészítő függvény létezik: SHA-1, MD2, MD4, MD5, ezek közül az utolsónak említett MD5-az egyik legtöbbet alkalmazott.

*Ronald Rivest* által tervezett MD5 algoritmus a bemenetből 16 bájtos (128 bites) kriptográfiailag erős lenyomatot készít.

Az algoritmus elsőként **hozzáfűzi az üzenet hosszát** az üzenethez, amit 64 biten ábrázol. Amennyiben az üzenet hosszabb lenne, mint $2^{64}$ bit, azaz a hossza nem ábrázolható 64 biten, akkor a hossznak csak az alsó 64 bitjét fűzi hozzá, így az MD5-nek nincs elvi üzenethossz korlátja.

Az algoritmus működéséhez biztosítani kell, hogy a bemenet hossza (a hossz-információval együtt) 512 egész számú többszöröse legyen. Ennek érdekében az üzenet és a hosszúság információ közé egy 1-et és szükséges mennyiségű 0-t szúr be.

A közkedvelt Total Commander-Fájl menüjében is megtalálható többek között az MD5 algoritmus megvalósítása: segítségével a kiválasztott fájlokból tudunk lenyomatot készíteni, illetve a lenyomat ellenőrzésével meg tudjuk állapítani, hogy a fájl megváltozott-e (módosította például egy vírusprogram).

Amint látható, hexadecimálisan ábrázolt bájtokból állnak a lenyomatok, amelyek az azonosításához a fájlok nevét is szerepeltetik a lenyomatokat tartalmazó fájlban. Bármelyik fájl tartalma, akár egy bitben változik, az md5 ellenőrzés ezt felfedi. (*6-9. ábra*)

## 6.4 Alkalmazási réteg

Az alkalmazási réteg feladata a felhasználó, és a felhasználói programok számára a hálózati szolgáltatásokat biztosító illesztések biztosítása. A fő hálózati szolgáltatások a következők:

- állományokhoz való hozzáférés, állományok továbbítása,
- elektronikus levelezés,
- virtuális terminálok,
- egyéb, pl. névszolgálatok.

Az ezekre vonatkozó konkrét példákkal, megoldásokkal a TCP/IP és Internet-tel kapcsolatos fejezetben foglalkozunk.

## Ellenőrző kérdések: 6. Fejezet

?

1. Mi a szállítási réteg feladata? Miért hasonlítjuk az adatkapcsolati réteghez?
2. Hogyan osztályozzuk a hálózati szolgálatokat minőségük alapján?
3. Hogyan tudják meg a szállítási rétegek a kommunikációhoz való címeiket? Mik azok a jól ismert címek?
4. Milyen megoldások vannak a csomagtovábbítás hibáinak a kezelésére?
5. Mi a viszonyréteg feladata? Mi az adatvezérjel?
6. Mi a megjelenítési réteg feladata?
7. Milyen problémák léphetnek fel az adatábrázolással kapcsolatosan a hálózatokon?
8. Miért lehet az adatokat tömöríteni? Mi az a redundancia? Rajzolja fel az adattömörítés blokkvázlatát!
9. Mit jelent hogy egy tömörítés veszteséges, vagy veszteségmentes?
10. Ismertessen néhány tömörítési eljárást! Mi a darabszám kódolás? Mi a mintahelyettesítés? Mi a sorozathossz kódolás lényege?
11. Mi a statisztikai kódolás? Mi a relatív kódolás?

12. Hogyan tömöríthetünk az emberi képi és hangi érzékelés tökéletlenségének kihasználásával?
13. Rajzolja fel a titkosítási modellt!
14. Ismertessen néhány helyettesítéses rejtjelezési megoldást!
15. Mi a felcserélési és helyettesítéses rejtjelezés közti alapvető különbség? Mutassa be a felcserélési rejtjelezést egy példán!
16. Mi az a DES? Mi az egykulcsos titkosítás?
17. Milyen digitális áramköri elemekkel lehet egy bitcsoport felcserélését és helyettesítését megoldani?
18. Hogyan működik a nyilvános kulcsú titkosítás algoritmusa?
19. Mi az a PGP, és hogyan lehet használni?
20. Mik azok a lenyomatkészítő függvények? Mire és hogyan használjuk? Mi az MD5?
21. Mi az alkalmazási réteg feladata?

# II. HÁLÓZATOK A GYAKORLATBAN

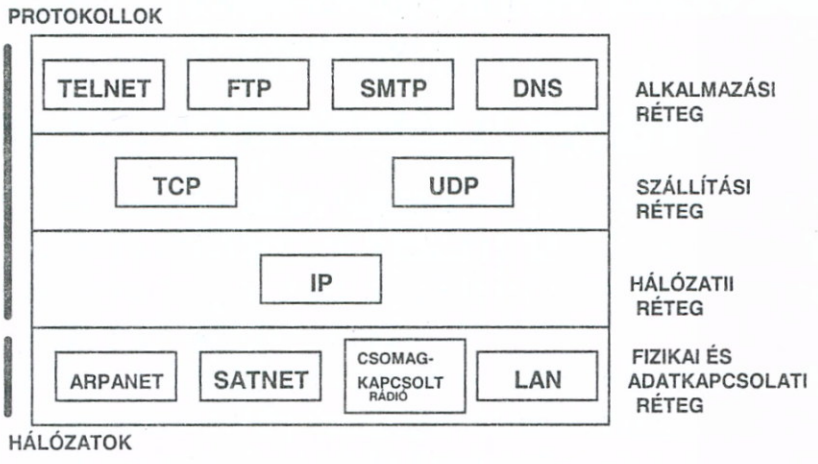
A könyv első részében összefoglaltuk a számítógép hálózatokkal kapcsolatos alapismereteket: rétegek felépítését, kapcsolódási módokat és megoldásokat. A most következő rész – az előző fejezetekben leírtak felhasználásával –, bemutatja a mindennapi gyakorlatban megjelenő megoldásokat, amelyek megértését az első részben megismertek teszik lehetővé. Elsőként az internet működését biztosító, TCP/IP és a kapcsolódó protokollokat ismertetjükl. A következő fejezet tárgya a szervezeti csoportok és személyek közös munkáját lehetővé lokális hálózatok. Ezt követi az igen fontos, biztonsággal foglalkozó rész, majd a hálózatok kezelését összefoglaló fejezet zárja a könyvet.

## 7. TCP/IP PROTOKOLL ÉS AZ INTERNET

*Egy szakember akkor mondható valóban jónak, ha már pontosan tudja, hogy mit nem tud...*

A TCP/IP elnevezés két protokoll rövidítéséből származik. Ez a hálózati réteg feladatát ellátó **Intrenet Protocol (=IP)**, és a hálózatokban lévő gépek összeköttetését biztosító **Transmission Control Protocol (TCP)**. Az Internet kisebb kiterjedésű számítógépes hálózatok (LAN-ok) összekapcsolásából álló globális számítógépes rendszer, „a hálózatok hálózata".

A rendszer alapjait a hatvanas évek elején az USA-ban alakították ki a Védelmi Minisztérium támogatásával indított ARPA kutatási program keretében, ezért kezdetben a neve ARPANET volt. Azt vizsgálták, hogy milyen módon valósítható meg az egymástól távol lévő számítógépeken keresztül történő adattovábbítás. A cél egy olyan kommunikációs rendszer kialakítása volt, amely akkor is tovább működik, ha egy-egy része valamilyen ok miatt kiesik. Az adatok átvitelére csomagkapcsolt átvitelt használtak. Az egyszerű kommunikációt megvalósító levelezésen vagy adatbázisok elérésén túl ma már különféle multimédia-alkalmazások is használhatók.




*7-1. ábra TCP/IP alapú protokollok*

A későbbiekben már Internet-nek nevezett hálózat – bár rétegmodell szemlélettel tervezték az 1960-as években – nem követi az OSI hét rétegű felépítését. A tényleges hálózati kapcsolatot biztosító TCP/IP protokollt UNIX operációs rendszerhez fejlesztették ki, és alapvetően az OSI modell két rétegének a funkcióját valósítja meg: a hálózati és szállítási rétegét. A hálózati modell összesen négy rétegből áll:

**Alkalmazási réteg (Application layer)**: Itt vannak a felhasználói és a hálózati kapcsolatot biztosító programok.

**Szállítási réteg (Transport layer):** Az OSI modell szállítási hálózati rétegének felel meg. A hosztok közötti létesített, és fennálló kapcsolat fenntartását biztosítja.

Két egymástól függetlenül használható rétegprotokollból áll: az egyik a **Transmission Control Protocol (TCP)** azaz a továbbítást szabályozó eljárás, míg a másik az összekötetésmentes szállítási protokoll **User Datagram Protocol (UDP).**

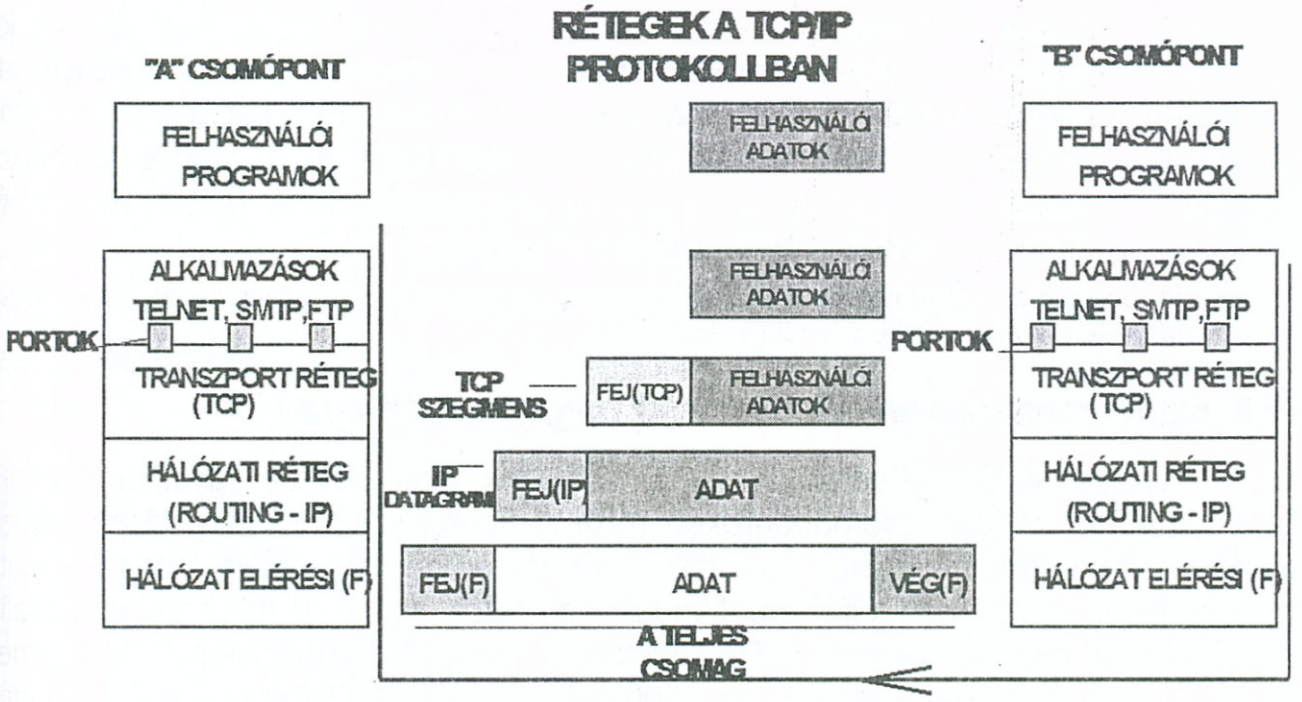
**Hálózatok közötti réteg (Internet layer):** Az OSI modell hálózati rétegének felel meg, ez a réteg végzi a csomagok útvonal kijelölését a hálózatok között. Ennek a rétegnek a protokollja az **Internet Protocol (IP)**. Feladata a hálózati csomópontokon (IMP-ken) keresztül a csomagok (packet-ek), más néven a datagramok továbbítása. A rétegben előforduló események, és hibák jelzésére szolgál az Internet **Control Message Protocol (ICMP)**, az Internet Vezérlőüzenet Protokoll.

**Hálózat elérési réteg (Network interface layer):** Az OSI modell két alsó szintjének felel meg, és ez biztosítja a kapcsolatot a szomszédos csomópontok között. (Pl.: Ethernet, Token-Ring, Token-Bus, PPP)).

Az információ áramlása két (legtöbbször távoli) csomópont között a *7-2. ábrán* látható. Jól követhető az előző fejezetekben leírt elv: az üzeneteket a rétegek kisebb részekre bontják, fejinformációkkal látják el, majd ezeket adják tovább.

## 7.1 RFC dokumentumok

Az Internet szabványokat RFC-knek hívják, ami a Request For Comments (Hozzászólásra, megvitatásra készített anyag) kifejezés rövidítése. Ha megszületik egy szabványtervezet, akkor azt először ajánlásként teszik közzé, és kap egy RFC számot. Ha végül az ajánlást elfogadják, akkor Hivatalos Internet Protocoll (Official Internet




*7-2. ábra: A TCP/IP csomópontok közti információáramlás*

Protocols) válik belőle, de továbbra is az RFC számmal hivatkoznak rá. Megállapodás szerint minden RFC új számot kap, ha átdolgozzák.

Mivel a szabványok nyiltak, bárki tanulmányozhatja ezeket, és biztos lehet benne, hogy a szabványok alapján készített alkalmazás, vagy készülék helyesen fog működni.

Magyarországon pédául a következő címen érhetők el a különböző RFC dokumentumok: **http://wigwam.sztaki.hu/rfc/**

A következőkben többször fogunk hivatkozni az RFC dokumerntumokra, gyakran a kibocsátási évszámokkal együtt.

## 7.2 A TCP/IP protokoll hálózati rétege: az IP (RFC791/1981)

A protokoll összeköttetés-mentes. A szállított csomagok - más néven a datagrammok - a forrás hoszttól a cél hosztig kerülnek továbbításra, esetleg több hálózaton keresztül. Ez a protokoll a „postás."

A hálózati réteg megbízhatatlan, összeköttetés-mentes szolgálatot biztosít, vagyis a csomagok elveszhetnek, többszöröződhetnek, hibás sorrendben érkezhetnek. Ennek ellenére ez a szint, a hibákat nem javítja, és nem is jelzi a magasabb szintek felé. Csomagvesztésre viszont csak ritkán, és indokolt esetben kerül sor. Így, **az összes megbízhatósági mechanizmust a szállítási rétegben kell megvalósítani, ami a két végállomás közötti megbízható összeköttetést biztosítja.**

Az IP protokoll — definiálja az adatátvitel legkisebb egységét, annak pontos formáját, az **útválasztást (routing-ot),** valamint néhány további olyan fontos szabályt, amelyek meghatározzák, hogy a hosztok, IMP-k hogyan dolgozzák fel az IP csomagokat, mikor és hogyan kell jelzéseket generálni, mikor kell csomagot eldobni.

Az IP működése a következő: A szállítási réteg az alkalmazásoktól kapott üzeneteket maximum 64 kbájtos datagramokra tördeli, amelyek az útjuk során esetleg még kisebb darabokra lesznek felvágva. **Amikor az összes datagram elérte a célgépet, ott a szállítási réteg ismét összerakja üzenetté.** A datagram két részből áll: egy fejrészből és egy szövegrészből. A fejrészben 20 bájt rögzített, és van egy változó hosszúságú opcionális rész is *(7-3. ábra).* A bájt szinonímájaként a TCP/IP terminológia gyakran használja az **oktet** (=**bitnyolcas**) kifejezést.

Az adategység, amelyet a TCP/IP protokoll használ: duplaszó, vagyis 4*8=32bit. Ezért például a 20 bájtos adathosszúság esetén a leíró mezőbe 5 kerül (5*4=20 bájt).

### 7.2.1 Az IP csomagok tördelése (fregmentációja)

Mivel a csomagok az átvitel során mindig az adatkapcsolati keretbe ágyazva utaznak, ezért a különféle hálózati csomópontokon történő átvitel során esetleg kisebb darabokra kell szétbontani (tördelni, vagy más néven fregmentálni), azért, hogy az adatkapcsolati réteg kereteiben elférjen. Minden hálózatra van egy jellemző maximális adatátviteli keretméret: network **MTU** (**Maximal Transfer Unit**). Például Ethernet keretben a maximális adathossz MTU = 1500 oktetes (bájtos) IP csomagot lehet elhelyezni. X.25 esetén lehet hogy az MTU csak 128 bájt. A széttördelt darab neve: fragment, és 8 bájttal oszthatónak kell lennie.

Ha például három hálózaton megy keresztül egy IP csomag, ahol pl. MTU1=1500, MTU2=620, MTU3=1500, akkor az MTU2 miatt 616 bájtos fragmentekre kell tördelni a csomagot a második hálózaton történő átvitelkor. Hosztok és ICMP-ék esetén a minimális MTU 576 oktet.

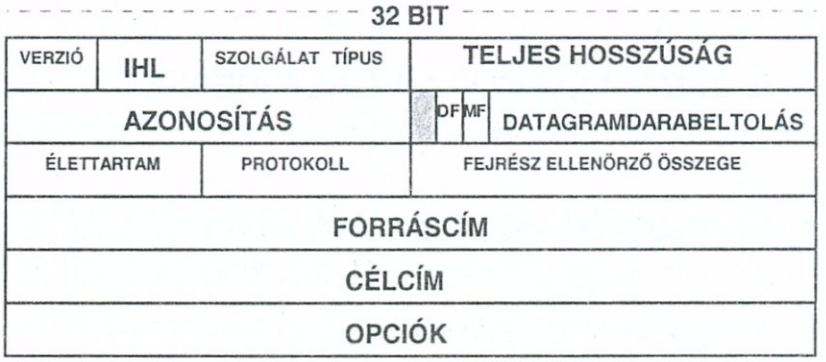
A fregmentált IP csomagok egyesítése (reassembly) — csak a végponton történik meg. A csomagokot egyébként is összerakó TCP protokoll fogja ezt megtenni, egy bizonyos ideig várakozva az érkező részekre. Ha az időzítés lejárt, akkor a fragmenteket (és ilyen módon, a tördelt csomagot) eldobja.

## 7.2.2 Az IP csomag fejlécének a mezői

A mezőket a *7-3. ábra* alapján ismertetjük. Látható, hogy a csomag 4 bájtos duplaszavakból áll.

A **VERZIÓ** mező a protokoll verzióját azonosítja, így a protokoll módosítását is ezzel figyelembe lehet venni. Jelenleg az értéke: 4. Az új IPV6-os protokoll (ld. később) esetén ez az érték: 6.

Az **IHL** (=Ip Header Lenght) adja a fejrész hosszát 32 bites egységekben (20bájt+opció rész). Minimális értéke: 5.




*7-3. ábra: IP csomag fejrésze*

A **SZOLGÁLAT TÍPUS** mező teszi lehetővé a hoszt számára, hogy kijelölje az alhálózattól kívánt szolgálat típusát. Különféle sebességek, és megbízhatósági fokok különböző kombinációi között lehet választani. Ez azért fontos, mert különféle optimális átvitelt lehet megvalósítani. Például digitalizált kép- vagy hang továbbításakor a gyors átvitel sokkal fontosabb, mint az esetleges átviteli hibák javítása. Ha azonban adat- vagy programfájlokat továbbítunk, akkor a pontos átvitel a fontosabb, és nem a gyorsaság.

A **TELJES HOSSZÚSÁG** mező a teljes datagram hosszát tartalmazza (fejrész+adat). A maximális hosszúság 65 536 bájt.

A következő 32 bites mező a tördeléshez (fregmentáláshoz) kapcsolódik.

Az **AZONOSÍTÁS** mező alapján állapítja meg a célhoszt, hogy egy újonnan érkezett csomag (fragment) melyik datagramhoz tartozik. Egy datagram minden egyes darabja ugyanazzal az Azonosítás mező értékkel rendelkezik.

Ezután egy nem használt bit, majd két 1-bites mező következik. A **DF** bit a Don't Fragment (ne tördelj!) kifejezés rövidítése. Ha ez a bit 1 értékű, akkor az átjárók nem tördelhetik a datagramot, mert a célállomás képtelen azt ismét összerakni. Ha a datagram nem vihető keresztül a hálózaton, akkor vagy kerülő utat kell választani, vagy el kell dobni.

Az **MF** bit neve a More Fragments (több darab) rövidítése. A széttördelt datagramdarabokat MF=1 értékkel jelzi, kivéve az utolsót, ahol MF=0. Értelemszerűen a nem tördelt csomagoknál, ez az érték mindig 0. A TELJES HOSSZÚSÁG mező mintegy második ellenőrzésként használható, vajon nem hiányzik-e datagram darab, és hogy az egész datagram összeállt.

Ehhez kapcsolódik a **DATAGRAMDARABELTOLÁS** mező, ami azt jelöli ki, hogy az adott darab hol található a datagramban. Minden datagramdarab hosszúságának (kivéve az utolsót), 8 bájt egész számú többszörösének kell lennie.

Mivel a mező 13 bit hosszú, ezért maximálisan 8192 8 bájtos darabból állhat egy datagram, amelyből a maximális datagramhossz 8*8192=65 536 bájt. Ez a megoldás biztosítja, hogy a maximális hosszúságú IP csomagot is fregmentálni tudjuk.

Az **ÉLETTARTAM**, más néven **TTL (Time To Live)** mező lényegében egy 8 bites számláló, amely a csomagok élettartamát tartalmazza másodpercben. Amikor értéke nullává válik, akkor az adott csomag megsemmisül. Így a maximális élettartam 255 másodperc lehet.

Amikor a hálózati réteg összerak egy teljes datagramot, tudnia kell, hogy mit tegyen vele. A **PROTOKOLL** mező kijelöli, hogy a datagram a különféle szállítási folyamatok közül melyikhez tartozik. A TCP a leggyakoribb választás, de léteznek egyebek is (UDP, ICMP...).

A **FEJRÉSZ ELLENŐRZŐ ÖSSZEGE** csak a fejrész ellenőrzésére szolgál. Egy ilyen ellenőrzőösszeg azért hasznos, mert a fejrész a darabolások miatt változhat az átjárókban.

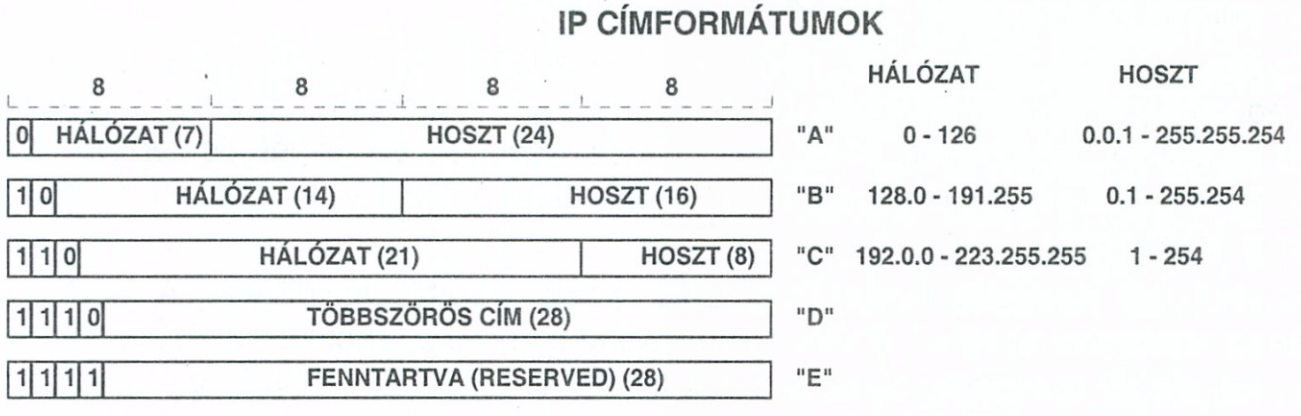
A **FORRÁSCÍM** és a **CÉLCÍM** mezők 32 biten a hoszt címet tartalmazzák, amit a következő részben szereplő IP címzési rendszer leírásánál mutatunk be.

Az **OPCIÓK** mező rugalmasan alkalmazható biztonsági, forrás általi forgalomirányítási, hibajelentési, hibakeresési, időpont megjelölési és egyéb információs célokra. A mező biztosításával elkerülhető, hogy a fejrészben levő biteket és mezőket ritkán használt információk számára kelljen lefoglalni.

## 7.2.3 Címzési rendszer (RFC 1166/1990)

Mivel az Internet lényegében hálózatok összekapcsolása, a címzési rendszer kialakításánál ezt figyelembe vették: **a címzés hierarchikus; azaz vannak hálózatok, és ezen belül gépek (hosztok).**

Így célszerű a címet két részre bontani: egy **hálózatot azonosító**, és egy, az adott hálózaton belül **gépet azonosító** címre. A hálózatok közötti kapcsolatot az útvá-




*7-4. ábra: IP címek felépítése*

lasztók (routerek) biztosítják. A routerek a teljes cím hálózati címrésze alapján irányítanak.

A 32 bites címek két részre történő osztását – figyelembe véve a különféle nagyságú hálózatok létezését – a címtartomány több, különböző nagyságú részre bontásával valósították meg. Öt különböző felosztást hoztak létre, ahogy ezt a *7-4. ábrán* bemutatjuk. A felosztások típusát a cím felső, maximum 4 bitje határozza meg, és az ABC első 5 betűjével jelöljük.

A felosztásnál a tervezők feltételezték, hogy lesz a világban néhány nagyon sok gépet tartalmazó, igen nagy hálózat, lesznek közepes méretű hálózatok, és sok kicsi, kevés gépet tartalmazó hálózat.

A felosztásban szereplő első három címforma ezt tükrözi: az A osztályú címmel 126 hálózat hálózatonként ~16 millió hoszttal címezhető, B osztályúval 16 382 hálózat hálózatonként ~65000 gépet címezhetünk. A C osztályú cím kb. 2 millió hálózatot, (amelyek feltételezhetően LAN-ok), egyenként 254 hoszttal azonosít.

Az utolsó előtti címforma (D osztályú cím) többszörös címek (mulicast address) megadását engedélyezi, amellyel egy datagram egy hosztcsoporthoz irányítható. Az utolsó címforma (E) fenntartott.

**A cím négy bájtját szokásos közéjük pontokat írva, a bájtok decimális megfelelőjével leírni. (dotted decimal notation).**

**A címzéseknél a hálózat, és a hálózaton belüli gép (hoszt) címének szétválasztására címmaszkokat (netmask) használnak.** Alkalmazásakor, a hálózati cím leválasztására bitenkénti ÉS műveletet végeznek az IP cím és a cím maszk között. Ha a címmaszk negáltjával végezzük a maszkolást, akkor a hálózaton belüli gép címét kapjuk meg. C osztályú címek esetén ezért a maszk: 255.255.255.0, míg B osztálynál: 255.255.0.0, A osztálynál pedig 255.0.0.0.

Példa: Legyen az IP címünk a következő: **193.224.1.23**

Ez binárisan felírva: **11000001.11100000.00000001.00010111** Mivel a 32 bites érték felső két bitje egyes, ezért ez egy C osztályú cím.

A maszk **255.255.255.0** Binárisan: **11111111.11111111.11111111.00000000**

Az IP cím és a címmaszk bitenkénti ÉS(AND) kapcsolata adja a hálózati címet:

**193.224.1.23 .AND. 255.255.255.0 = 193.224.1.0.**

A hálózaton belüli gépcím is kiszámítható: **193.224.1.23 .AND. 0.0.0.255 = 0.0.0.23**

**Egy hálózaton belül, egy csomagot küldő gép a célgép IP címével a leírt hálózatmeghatározási műveletet végzi el. Eredményül vagy a saját hálózat címét kapja, (és akkor a csomag a hálózaton belül az adott című gépnek küldi el), vagy idegen hálózati címet és akkor annak az útválasztónak routernek (vagy átjárónak) hívott, más hálózatokkal kapcsolatban álló gépnek kell elküldenie, amely majd továbbítja a külső hálózatok felé, amelyek között a címben szereplő idegen hálózat is szerepel. Az átjáróban ezek szerint, minimum két hálózati kártya található, az egyik a belső, másik a külső hálózathoz kapcsolódik.**

Vagyis, ahhoz hogy egy számítógépünk egy másik hálózaton lévő gépet el tudja érni, a helyi hálózatunkban kell lennie egy alapértelmezés szerinti átjárónak (default gate-

way) amely tudja azt, hogy a külső hálózatoknak szóló csomagokat hova kell küldeni.

A címzésnél bizonyos címtartományok nem használhatók:

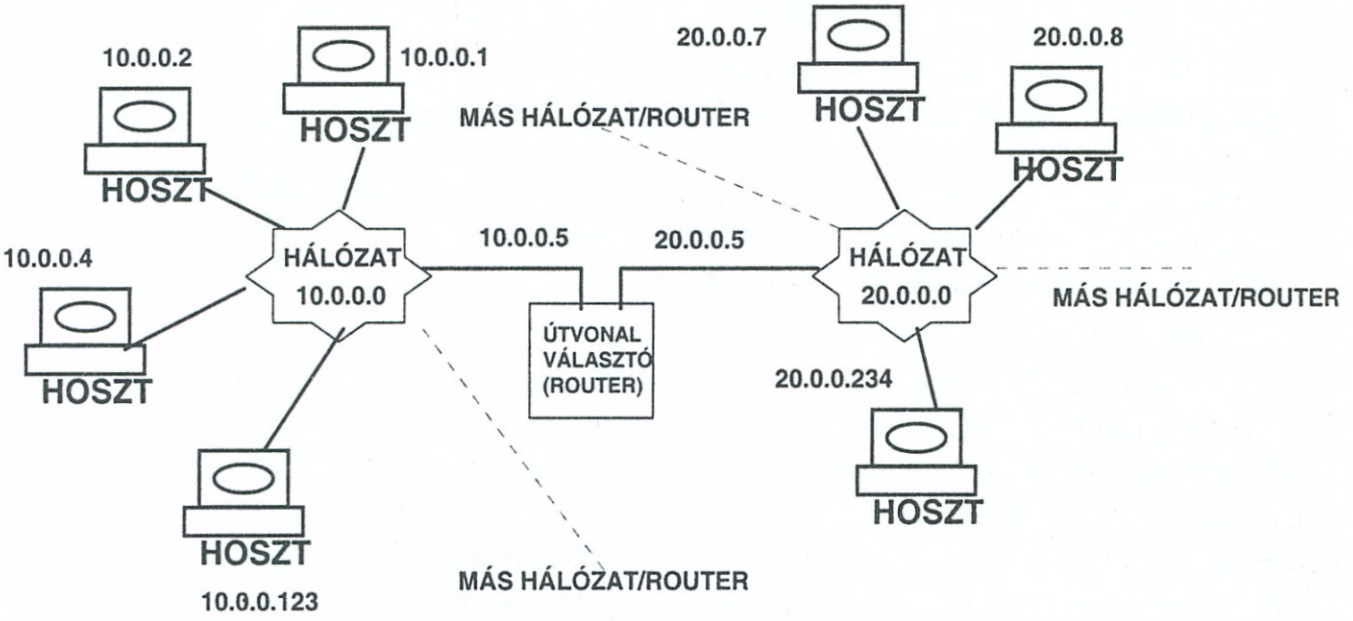
- A 127-el kezdődő címek a „loopback" (visszairányítás) címek, nem használhatók a hálózaton kívül, a hálózatok belső tesztelésére vannak fenntartva.
- A hoszt címrészbe csak 1-eseket írva lehetséges az adott hálózatban lévő összes hosztnak üzenetet küldeni (broadcast). Például a 195.13.2.255 IP címre küldött üzenetet a 193.13.0 című hálózatban lévő összes gép megkapja.
- Ha a hoszt címrésze 0, az a aktuális hálózatot jelöli. Ha a hálózati cím 0, akkor az aktuális hálózatot jelenti. Például a saját gépről 0.0.0.0 címre küldött üzenet a saját gépre érkezik.

### *Belső hálózati címtartományok (RFC1918/1996)*

A TCP/IP alapú hálózatokban néhány címtartományt lefoglaltak olyan (belső) hálózatok számára, amelyek közvetlenül nem kapcsolódnak az internetre. Ilyen címeket az átjárók sem továbbítanak, ezért védelmi célokra is alkalmazhatók.

```
10.0.0.0        -   10.255.255.255
172.16.0.0      -   172.31.255.255
192.168.0.0     -   192.168.255.255
```

Az első egy A-osztályú hálózati címtartomány, a második 16 darab egybefüggő B-osztályú hálózati címtartomány, a harmadik pedig 255 darab egybefüggő C-osztályú hálózati címtartomány. Ezeket magunk kioszthatjuk, belső hálózatokon használhatjuk, alkalmazásukhoz külön engedély nem szükséges. Az ilyen című csomagokat az útválasztók sem továbbítják.




*7-5. ábra Hálózatok összekapcsolása, címzés*

### *Alhálózatok létrehozása*

Vannak olyan esetek, amikor például biztonsági, vagy forgalomkorlátozási okokból egy szervezetnek kiosztott címet úgy szeretnénk használni, hogy a szervezeten belül önál-

ló részhálózatokat alakítunk ki. Ekkor a különböző részhálózatokban lévő gépek egymással csak útválasztókon keresztül tudnak kommunikálni.

Ennek megértéséhez nézzünk egy konkrét példát. Van egy vállalat, négy önálló osztállyal. A vállalat által igényelt C osztályú (192.223.5.x x=0...255) hálózatot szeretnénk 4 alhálózatra bontani.

Az előzőekben már leírtuk, hogy hálózat és gépazonosítókban nem szerepelhetnek a a csupa 0 („saját") és csupa 1 („üzenetszórás") bitkombinációk.

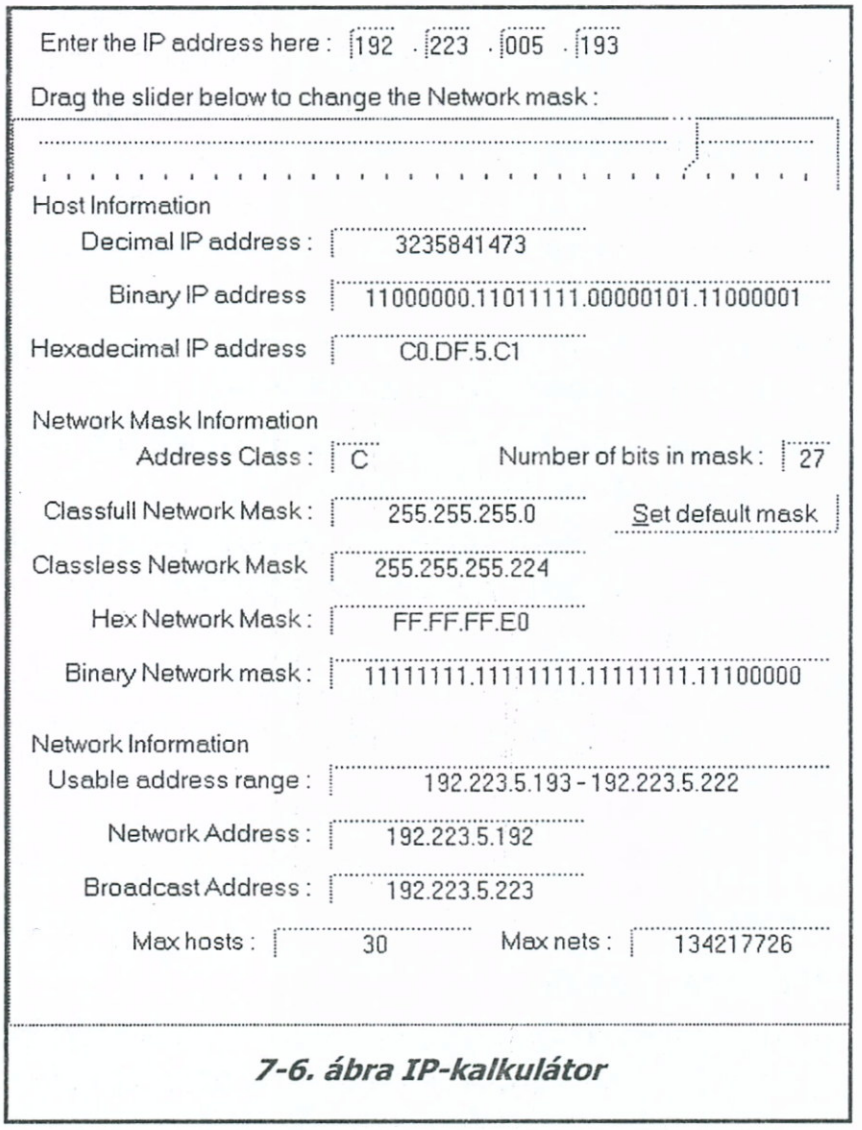
A négy alhálózat kialakításához, az alhálózatok azonosítására 3 bitet kell felhasználni, mert 2 bit esetén – mivel a 00 és 11 kombináció nem használható - csak két alhálózat kialakítása volna lehetséges. Ha három bittel címezzük az alhálózatokat, akkor nem négy, hanem hat alhálózat kialakítása lehetséges. (000 és 111 hálózati című hálózat nincs). A C osztályú hálózat alsó 8 bitje hhh ggggg alakú, ahol **h** az alhálózati cím **g** a gépcím bitjeit jelöli.

Mi lesz az alhálózatokban alkalmazott alhálózati maszk?

```
11111111.11111111.11111111.1110
0000 -> 255.255.255.224
```

Hat alhálózat jött létre. Az alhálózatok címének utolsó bájtja: .32, .64, .96, .128, .160, .192 kezdőcímek lesznek.

0-31 értékek esetén a hálózati cím 000 bithármas, és 224-255 cím esetén a hálózati cím 111 bithármas, és ezek nem használhatók.




***7-6. ábra IP-kalkulátor***

Egy alhálózatban csak 30 gép lehet, mert a csak nullát, illetve csak 1-et tartalmazó gépcím sem választható.

Milyen címtartományt kapnak az 110 alhálózati azonosítójú gépek? Csak az utolsó bájtot kiírva: 110 00001-től 110 11110-ig. Kiírva a teljes címtartományt:

```
192.223.5.193-től 192.223.5.222-ig
```

*7-6. ábrán* egy, az előzőkben leírtak alapján működő alhálózat IP-cím kalkulátor képernyőképét láthatjuk.

## 7.2.4 DNS - Domain Name Service (RFC883/1983 – RFC1591/1994)

Mivel a számszerű IP címek megjegyzése – hasonlóan a telefonszámokhoz – az emberek számára nehézkes, ezért kifejlesztettek egy más jelllegű azonosítási módszert a hálózatokban létező gépek azonosítására. Ez az interneten használt **tartomány név adatbázis, a DNS-t (Domain Name Service)** (domain–ejtsd: domén (=tartomány) és a továbbiakban, ha használjuk ezt így írjuk majd).

Minden weblap letöltésnél, levél közvetítésnél szerepe van, nélküle megbénulna a hálózat, és mivel a szolgáltatás csendesen dolgozik a háttérben, sokan még a létezéséről sem tudnak. **DNS a hálózatot egymásra épülő szintekből álló, névvel jelölt tartományokra – doménekre osztja.**

A TCP/IP kommunikáció az IP címek alapján történő datagram továbbításon alapul, vagyis a továbbításhoz ismerni kell a hálózati cél IP címét, hogy a csomagokat el tudjuk oda juttatni.

**DNS egy** – funkciójában a telefonkönyvhöz hasonló, de más felépítésű – **osztott, hierarchikus felépítésű adatbázis.**

**Adatbázis**, mert lényegében az IP címek, és könnyen megjegyezhető nevek egymáshoz tartozó párjait tartalmazza.

**Osztott**, mert nem egyetlen *névszolgáltató (name server)* számítógépen lévő állományban található, hanem az adatbázist ilyen névszerverek százezrei szolgáltatják nevek millióiról.

**Szintekre osztott, azaz hierarchikus**. A DNS rendszer nemzetközi és többszintű, a legfelső szinten van a biztonság miatt több példányban létező néhány tucat root (gyökér) szerver.

A DNS rendszer legfontosabb feladata a név – IP cím **névfeloldás**, de – ahogy azt látni fogjuk – egy sor más információt is szolgáltat az IP címekhez rendelt *domén-nevek*-ről.

### *IP címek, nevek*

Az Interneten levő hálózati eszközök, számítógépek mindegyikének egyedi azonosítója, azaz IP címe van. A felhasználók azonban olyan neveket szeretnek használni, amelyek könnyebben megjegyezhetők, mint egy ilyen hosszú szám, és a névből következtetni tudnak a gép, a szolgáltatás helyére, a szolgáltatás típusára is.

Ezért kezdettől fogva neveket rendeltek az IP címekhez. Amikor az Internet még csak pár ezer számítógépből állt, ezt a név-cím hozzárendelést egy folyamatosan bővülő fájl, a *host táblázat* tartalmazta. Ezt a táblázatot minden számítógépen lokálisan tárolták és egy központi helyről rendszeresen frissítették. Ennek nyoma mind a mai napig megvan: pl. XP alatt a system32/drivers /etc/hosts fájl éppen ilyen.

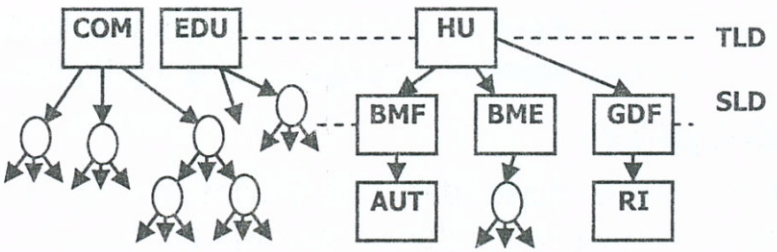
Például, tegyük fel, hogy a 196.123.23.4 IP címhez a www.kkdf.hu név tartozik. Ezt egy szövegszerkesztővel megnyitott hosts fájlba értelemszerűen, az ott látható bejegyzés, mint minta alapján, a következő sorba begépeljük, és mentjük. Ha ezután a böngészőben **www.kkdf.hu** címet adjuk meg, akkor a névfeloldás, vagyis a kapcsolat létesítéséhez szükséges IP cím megszerzése, ebből a hosts fájlból történő kiolvasással történik meg.

Az Internet növekedésével azonban ez a megoldás tarthatatlanná vált: a fájl hatalmasra nőtt, egyre sűrűbben kellett módosítani, és egyre több gépre kellett letölteni, mivel minden hálózati szereplőnek szüksége volt erre a táblázatra. Ezért jött létre a Domain Name Service (DNS), ami az Internetes kommunikáció egyik alapja. A nevek feloldása a háttérben, a hálózati kommunikáció segítségével történik. A névszerverek feladata kettős:

1. **Látni** - azaz az elosztott DNS adatbázis többi részét kérdezni, a hálózati szolgáltatások számára az érvényben levő név-cím hozzárendelésről információt adni.
2. **Láttatni,** mutatni - az elosztott adatbázis ide kiosztott részére információforrásként viselkedni, azaz a nevek egy bizonyos halmazáról a többi név szerver számára - mint illetékes - adatokat szolgáltatni.

Ha egy név esetében egy szerver az Internet számára elsődleges információforrás, azaz illetékes, azt úgy szokás kifejezni, hogy az ő adata **autoritatív.**

Mindenki ismer Internet neveket, pl.: atmel.avr.com, www.cocacola.hu, stb. Ezek a nevek fa struktúra szerint szerveződő hierarchiát alkotnak. A név végén van a fa gyökere.




*7-7. ábra Domén nevek hierarchiája*

A fa **fordított**, mert a gyökér a hierarchia legmagasabb foka. A nevek feloldása a gyökértől kezdődik, és fokról fokra halad előre. A név-fa különböző elágazási pontjaiért és ágaiért különböző szerverek felelősek. Egy-egy szerver több ágért is felelős lehet. A név-fa egy-egy pontját **domén**-nek (tartománynak), **domén névnek** vagy egyszerűen **név**-nek nevezzük.

### *A nevek hierarchiája*

A hierarchia csúcsát 'root'-nak, gyökérnek nevezzük. Az ez alatti neveket **top level domén**-eknek, **TLD**-knek mondjuk. Ezek a TLD-k két módon szerveződnek és kapták nevüket:

- A jelölt szervezet tevékenysége szerint: `com` – vállalatok, `edu` – egyetemek, oktatási intézmények, `mil` – katonai szervezetek, `gov` – kormányhivatalok, `net` – hálózati szervezetek, `org` – mindenféle más szervezet, `arpa` – az Internet ősében, az Arpanetben levő gépek neveire szolgált kezdetben. Az inverz nevek feloldásánál (ld. később) mind a mai napig fontos szerepe van.
- Az USA-n kívüli domén-ok számára az **ISO 3166** szabványban meghatározott kétkarakteres országkódokat használják. Például: `IT` – Olaszország, `PL` – Lengyelország, `HU` – Magyarország, stb.

A hierarchia nagyon hasonlít az operációs rendszerek hierarchikus fájlstruktúrájához (pl. C:\temp\halozat\tematika.txt), csak az alá-fölérendeltség itt éppen fordítva, jobbról balra olvasható le. Pl. szamitogep.csoport.tanszék.intezet.hu. A hierarchia szintjeit az elnevezések közé írt ponttal választjuk el, és elvileg a TLD domén név után is kellene egy pont, utalva a gyökérre, de a hétköznapi gyakorlatban általában nem használják.

A TLD elnevezés mellett használatos még az **SLD (Second Level Domain)** kifejezés is, a hierarchia második szintjén levő doménekre.

### *Zónák*

A név-fa zónákra oszlik: egy-egy zóna a **fa egyben kezelt része.** Sokszor - de nem feltétlenül, - egybeesik egy aldoménnel. Például, egy zóna lehet a:

**tanszek.iskola.hu**

és minden név, ami a hierarchiában ez alatt van. Egy zónának tekinthető például az összes TLD-t tartalmazó root zóna is.

Egy zóna a 'láttató', az 'autoritatív' szerver szempontjából egységként kezelt, rendszerint egy fájl. Egy-egy zónát több szerver is láttat(hat). Ezek közül az egyik az elsődleges, a többi (ha van) másodlagos.

- Az **elsődleges szerveren** az adatok a zónakezelő munkájának a nyomán ténylegesen változnak.
- **A másodlagos szerver(ek)** a zóna adatait meghatározott rend szerint az elsődleges szervertől tükrözi(k). A tükrözés rendjét az elsődleges szerveren a rendszeradminisztrátor a zóna konfigurációjával határozza meg.

### *Delegálás*

A hierarchia egyes darabjait a zóna kezelője tovább delegálhatja más szerverekre. Például az iskola.hu domén gazdája az tanszek.iskola.hu aldomén láttatását, autoritását az illető tanszék egy meghatározott gépére bízhatja a konfigurációban: mindenki felelős és úr lehet a saját illetékességi körében (ennek elnevezése: szubszidiaritás elve). A root zóna, sőt még a TLD-k (edu, gov, hu stb.) is általában mást sem tartalmaznak, mint ilyen delegálást. Így jön létre a hierarchikus, osztott adatbázis. A delegálás azonban nem feltétele a több szintű név megadásának. Például lehetséges, hogy a **tanszek.iskola.hu** nincs delegálva, nem különálló zóna, mégis létezik a **gep.tanszek.iskola.hu** domén, mert az **iskola.hu** zóna gazdája bevezette a pontot (.) tartalmazó **gep.tanszék** nevet. Ezt éppen úgy megteheti, mint a **geptanszek**, vagy az **tanszekgepe** nevek bevezetését, melyeknek hatása a **geptanszek.iskola.hu**, illetve a **tanszekgepe.iskola.hu** nevek létrejötte.

### *Domén nevek*

A hierarchia következtében **minden név egyedi.** Lehet, hogy egy gépet több helyen a világon, ugyanazon névvel látnak el, de a teljes domén nevük miatt azonosíthatóan különböznek.

Pl.: **engepem.gdf-ri.hu engepem.aut.bmf.hu**

A domén neveknek azt a teljes alakját, ami a nevet a gyökér doménig tartalmazza **FQDN**-nek (Fully Qualified Domain Name), a domén név pontokkal elválasztott darabjait pedig **szegmenseknek** nevezzük. Annak jelzésére, hogy a domén név teljes, a név végére pontot teszünk. Valójában a TLD-re (hu, edu) való végződés nem garantálja, hogy a név FQDN, mert egy aldomén elnevezése is lehet TLD név (bár nem szokás).

Pl.: **engepem.kando.edu.bmf.hu**

Domén nevekben megengedett karakterek a **latin ABC** betűi [a-z], a **számjegyek** [0-9] és a **kötőjel** (-). Kis- és nagybetű egyformán használható, és nem jelent különbséget. Nem állhat domén névben ékezetes karakter. A törekvések megvannak ennek a problémának a megoldására, de pl. egy angol netező, hogyan gépeli be az árvíztűrő-tükörfúrógépgyár.hu nevet? Gyakori hiba, hogy aláhúzás (_) karaktert adnak meg domén nevekben. Az eredeti definició (RFC1035) az egyes szegmensek elején csak betűt engedett meg, a későbbi (RFC1123) megengedi a számmal kezdődő szegmenst is. Például szabályos az 2inone.hu domén. Viszont továbbra sem állhat kötőjel a szegmens név elején, vagy végén.

### *IP cím -> név hozzárendelés*

Az Interneten nemcsak arra van szükség, hogy név alapján az IP címet kapjunk, hanem a fordítottjára is, azaz hogy **IP címekből domén neveket kapjunk.** Ez a szolgáltatás az **inverz, vagy reverz feloldás**. A hálózati biztonság szerepének növekedése miatt, jelentősége egyre nő. Sok FTP, vagy levelező szerver csak olyan gépekről fogad el kéréseket, amelyek címéből a hozzájuk tartozó domén nevet ki lehet deríteni.

A névszerverbe való regisztrálás már nyilvánosságot jelent, nyomon követhető. Nem véletlen, hogy a varez oldalak többsége, csak IP cím alapján érhető el.

A cím-név feloldás érdekében bevezették az **in-addr.arpa** domént. Így aztán **193.225.161.101** a címhez tartozó nevet úgy kapjuk meg, hogy a domén rendszertől megkérdezzük a **101.161.225.193.in-addr.arpa** névhez tartozó rekordot. (Az IP cím fordított sorrendben szerepel benne!)

Az **in-addr.arpa** doménban éppen úgy delegálják az egyes aldoméneket mint minden más zónában.

### *Névfeloldás: Rezolverek és DNS szerverek*

Most pedig kövessük végig, hogyan is zajlik a névfeloldás. Tételezzük fel, hogy a **www.aut.bmf.hu** nevet kell feloldani, mert azt gépeltük be a böngészőbe. Meg kell állapítania, hogy milyen IP cím is tartozik ehhez a domén névhez. A programot, amelyik ezt a tevékenységet végzi, **rezolvernek, feloldónak** nevezzük.

Első lépésként a rezolver WINDOWS-t használó gépeken megvizsgálja a **system32/drivers/etc/hosts** fájlt. Ha itt megtalálja a keresett IP címet, akkor a névfeloldás véget ért.

Gépünkön a TCP/IP szoftver telepítésekor, konfigurálásakor meg kellett adni egy vagy több DNS szervert. Utána ezekhez fordul a rezolver. A rendszergazdák fontos kötelessége az erre vonatkozó információt a felhasználóiknak megadni. A rezolver rendszerint néhány konfigurációs fájlból és könyvtári szubrutinból áll. Gyakorlatilag minden TCP/IP-t használó, Internetbe kapcsolt számítógépen szükség van rá. A rezolver tehát nem végez közvetlenül névfeloldást, hanem bizonyos általa ismert névszervereket kér meg arra, hogy a feloldást elvégezzék.

A rezolver konfigurációban a DNS szerverek megadásánál értelemszerűen IP címeket kell használnunk. Amikor a rezolver a konfigurációjában megadott névszerverhez fordul, hogy például a **www.aut.bmf.hu** névhez tartozó IP címet megtudja, akkor a szerver általában nem válaszol azonnal.

Példánkban legyen a kérdezett névszerver a **ns.bmf.hu**. Az **ns** adatbázisában szerepelnek a **gyökér névszerverek IP címei**. Ezek valamelyikét kérdezi az ns név-

szerver. Egy root névszervert kérdezve például a **www.aut.bmf.hu** névről, az nem ad mást, mint a **.hu** zónáért felelős név szerverek listáját. Az **ns** névszerver ekkor egy újabb kérdést intéz a **.hu** névszerveréhez, aki újra csak arra vonatkozóan ad információt, hogy melyik névszerverhez lehet fordulni az **bmf.hu** nevek feloldásáért. Ilyen módon a **ns** rekurzív módon oldja fel a nevet, melynek végén a kérdező gép rezolverének megadja a választ.

Ez az osztottság biztosítja, hogy egy gép DNS-be, és ilyen módon a hálózatba való léptetéséhez, csak a hozzá tartozó névszerverben kell az adatbázisba felvenni.

### *A nevek időleges tárolása*

A névszerverek az általuk megtudott neveket egy átmeneti tárolóban (cache-ben) tárolják azzal a céllal, hogyha újra megkérdezik tőlük, akkor ebből az adatot kiolvasva, azonnal tudjanak válaszolni. Ennek többszörös haszna van: csökkenti a hálózati forgalmat, és gyorsítja a névfeloldást.

A cache-ben minden megtudott nevet csak egy bizonyos ideig tárolnak. Ha az idő lejárt, akkor a nevet törlik. Így, ha esetleg a névhez tartozó információ változik, akkor az ismert lesz. Azt az időt, ameddig a cache-ben van egy-egy információ, a név bejegyzésekor adják meg az autoritatív szerverben: minden rekordhoz tartozik egy **TTL (Time To Live)** érték. Ennyi másodpercig tárolják a szerverek a cache-ükben az információt.

### *Forwarder szerverek*

Egy névszerver gyakorlatilag kiegészítheti a cache-ét más szerverek cache-ével, ha a forwarder opciót használják a konfigurálásánál. Ha pl. egy gépen a DNS-konfigurációban megadják, hogy a **ns.bmf.hu** forwarder legyen számára, akkor a **ns.aut.bmf.hu**-n történő névfeloldás úgy zajlik, hogy ha a saját gyorstárában nincs benne a kért név, akkor a DNS szerver, mielőtt a világban a név-fa hierarchiának megfelelő módon elkezdene érdeklődni , megkérdezi a **ns.aut.bmf.hu**-t. Ha annak a cache-ében megtalálható a keresett rekord, akkor válaszol, és így a kérő gyorsan megkapja a választ. Elképzelhető, hogy egy-egy intézménynél több kisebb szerver használ egy közös nagyobb forgalmú forwardert.

### *Slave szerverek*

Az olyan szervert, ami csak forwardert (esetleg többet) használ a nevek feloldására, slave szervernek nevezzük.

### *Zónafájlok*

A névszerverek az egyes zónák adatait általában egy-egy fájlban tárolják. A 'master' szerveren az adminisztrátor személy közvetlenül, vagy valamilyen program közvetítésével maga módosítja ezt a fájlt. A 'slave' szervereken a fájl a tükrözés eredménye.

A zónafájl rekordokból, RR-ekből (resource record) áll. Az RFC1035-ben megadott definíciók számos típusú rekord használatát teszik lehetővé.

A rekordok formáját az RFC1035 határozza meg, és ez a következő:

```
cimke ttl osztály típus adatok
```

A **'cimke'** a domén rekord neve. Lehet üres, ilyenkor az előtte levő rekord cimkéje érvényes. A **'ttl'** a rekordhoz tartozó time to live időt adja meg másodpercben. Nem kötelező paraméter. Ha elhagyjuk, akkor a zónára vonatkozó alapértelmezés lesz a rekordhoz tartozó érték. A következő paraméter értéke gyakorlatilag mindig **IN,** azaz Internet osztály. Ez is elhagyható. A **'típus'** mondja meg, hogy milyen fajta információról is van szó. Pl. IP cím („A" rekord), name szerver információ (NS rekord) stb. Az **'adatok'** mező a rekord típusától függő információt tartalmaz.

### *Néhány fontosabb rekordtípus*

**SOA - Start of Authority rekord**: zóna kezdő rekord: A SOA rekord adja meg egy zónára vonatkozó közös információkat.

**A - Address, cím rekord**: Ez a leggyakrabban használt rekord, amely arra szolgál, hogy egy domén névhez IP címet rendeljünk. Például: `gepem A 190.111.222.3` . Nem írjuk ki a teljes domén nevét, csak annak első részét. A végére oda kell érteni azt a vonatkoztatási rendszert, ahol éppen vagyunk (aktuális zóna). Pl. ha a **`gdf-ri.hu`** zónáról van szó, akkor a fenti bejegyzés úgy értendő, mint *gepem.gdf-ri.hu*

**NS - Name Server, névszerver rekord**: Ez a rekord szolgál arra, hogy egy domén névszervereit megadjuk. Ilyen módon a domén egy delegálási pont. Pl.: ***tanszek*** `NS ns.gdf-ri.hu.`

**CNAME - Canonical Name, kanonikus név rekord:** Ez a rekord arra való, hogy egy hostnak más nevet is adjunk. Például: `www CNAME gepem`. Ha ez a rekordbejegyzés szerepel, akkor `www.gdf-ri.hu` egy másik neve a `gepem.gdf-ri.hu`-nak.

**MX - Mail eXchanger, levelező szerver rekord:** Ez a rekord szolgál arra, hogy egy doménba érkező levelek levelező szerverét kijelölje.

**Hasznos segédprogram:** XP alatt a DNS adatok lekérdezésére az **nslookup** parancs szolgál.

**Regisztrálás a .hu TLD alatt:** A .hu alatti domén név igényléséről részletes tájékoztató olvasható a **www.nic.hu** web lapon. A lényeg az, hogy van egy sor választható szolgáltató, akiknél egyformán be kell tartani néhány formai és technikai szabályt.

**Két fő technikai szabály:**

- A bejegyzendő domént legalább két, független hálózati elérésű névszerverrel kell szolgáltatni.
- A postmaster@domain.hu levelezési címnek - ahogy azt az RFC822 is előírja - működnie kell.

## *7.2.5 IP útválasztás (routing)*

A csomagkapcsolt rendszerekben az **útválasztás (routing)** azt a folyamatot jelöli, amivel kiválasztjuk az **útvonalat (path)**, amin a csomagot továbbküldjük és az **útvonal választó (router)** az a számítógép (IMP), amely ezt végrehajtja.

**Az útválasztó (más néven: router) olyan eszköz, amelynek több (legalább kettő) hálózati csatolója van, és mindegyik más (helyi) hálózathoz csatlakozik.**

Az útválasztó csomagokat fogad el a hozzá csatlakozó hálózatok gépeitől, és továbbítja ezeket valamelyik hálózati csatolóján. Azt hogy melyiken küldje tovább a memóriájában lévő útválasztási tábla (**routing table**) alapján határozza meg.

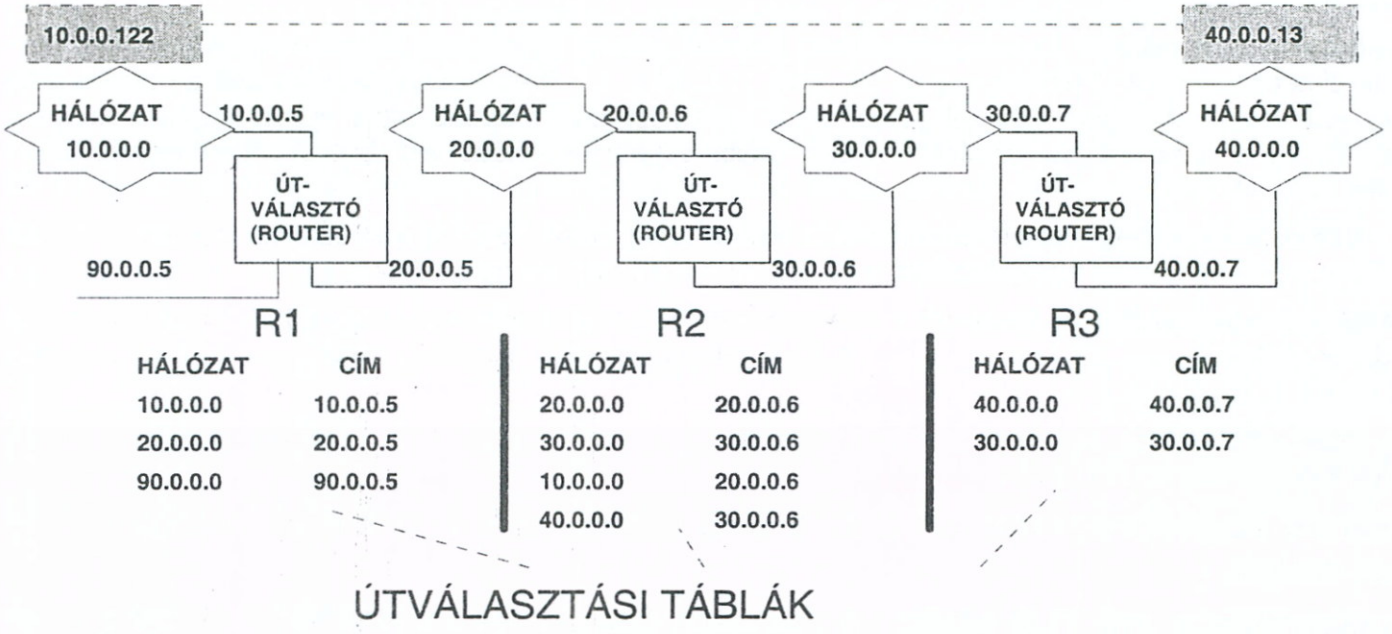
A *7-8. ábrán* látható négy önálló hálózat három routeren keresztül van összekötve.

Az útválasztási táblák tárolják az információt az elérhető csomópontokról, és azok elérési útvonalairól.

Tipikusan a táblázat (N,G) párokból áll, ahol N az elérhető hálózat IP-címe és G annak a útválasztónak a címe, amelyiken keresztül ez a hálózat elérhető.

Pl. az R3 jelű routernél a táblázat:

| Elérhető hálózat | Hálózati csatoló, amelyen keresztül a hálózat elérhető: |
|---|---|
| **40.0.0.0** | **40.0.0.7** |
| **30.0.0.0** | **30.0.0.7** |




***7-8. ábra: Példa az útvonal választásra***

Ha két gép egyazon hálózatban van, akkor útvonal-kiválasztás nélküli, **közvetlen (direkt)** összeköttetés létesíthető közöttük. Különböző hálózatok közötti **közvetett (indirekt)** útvonal-kiválasztásnál először a feladó a saját hálózatában lévő útválasztónak, küldi el a csomagot (datagramot). Az útválasztó fogja - esetleg újabb útválasztókon keresztül - a célhálózatba továbbítani a datagramot.

Mivel a pédában szereplő IP címek A osztályúak, a **255.0.0.0** hálózati cím maszk alkalmazásával állapíthatjuk meg a címzett hálózat címét:

Pl. **10.0.0.112** cím bitenkénti ÉS kapcsolata **255.0.0.0** – hálózati maszkkal adja meg a hálózati címet: **10.0.0.0**

Ezért például, ha a **10.0.0.122** című gép kommunikálni akar a **40.0.0.13** című géppel, akkor a következő történik:

A küldő gépen az IP protokoll megállapítja, hogy a csomag egy másik hálózatnak szól: a cél, maszkolással megállapított hálózati címe nem egyezik a saját hálózat címével.

Ezért a csomagot a hálózatban lévő útválasztó **10.0.0.5** címére küldi el. Az útválasztó megállapítja a címzett hálózati címét (**40.0.0.0**), és mivel az R1 jelű útválasztó táblájában nem talál táblabejegyzést az adott hálózatra vonatkozólag, akkor három eset lehetséges:

1. Az útválasztó hibaüzenetet küld a feladónak („Nem ismerek ilyen hálózatot") és a csomagot eldobja.

2. Felvesszük a nem szereplő hálózatot az útvonalon lévő útválasztók táblázataiba, hogy a routolás megvalósulhasson.
3. Elküldjük a csomagot egy ún. alapértelmezett (default) hálózati címre, ott talán majd ismerik ezt a címet.

Vizsgáljuk meg a 2. pontban említett, útválasztó táblázat bővítési megoldást.

Az előbbiek szerint a **10.0.0.122** című gép a **10.0.0.5** című (**R1**) routernek küldi el a csomagot. Az megnézi a táblázatát, de sajnos nem talál az adott hálózati címre (**40.0.0.0**) vonatkozó bejegyzést. A problémát a következő módon oldjhatuk meg:

Az útválasztó táblákba fel kell venni – ha még nem szerepel - egy újabb bejegyzést, hogy az adott hálózatnak szóló csomagot merre küldjük tovább. Esetünkre a megoldás:

**R1** táblázatába felvesszük:

**40.0.0.0** **20.0.0.5** bejegyzést,

**R2** útválasztó táblázatában a szükséges bejegyzés már szerepel:

**40.0.0.0** **30.0.0.6**

Természetesen mivel a **40.0.0.0** hálózatból választ is akarunk küldeni a **10.0.0.0** című hálózatba, a következő bejegyzések szükségesek:

**R3** táblázatába felvesszük:

**10.0.0.0** **30.0.0.7** bejegyzést,

**R2** útválasztó táblázatában a szükséges bejegyzés már szerepel:

**10.0.0.0** **20.0.0.6**

Számos útválasztóban megadható egy alapértelmezés szerinti átjáró (**default router**). Ha nincs a táblázatban létező útvonal, akkor a szoftver az alapértelmezett útválasztónak küldi el a datagramot.

Az előző példánál maradva, nem kell **R1** és **R3** táblázatát módosítani, ha R1 default routere **20.0.0.5** cím, és **R3**-é pedig: **30.0.0.6**.

Az alapértelmezett utak használata különösen hasznos, ha a hálózatban sok a helyi cím, és csak egyetlen kapcsolat van az internet további helyeihez. Például, az alapértelmezett utak használata jól működik olyan hosztokon, amelyek fizikailag egyetlen hálózathoz kötődnek, és csak egyetlen útválasztóval kapcsolódnak az internet többi részéhez.

A routolás nem csak hálózati, hanem hoszt szinten is megvalósítható: A hálózat kezelője bizonyos hosztokat kitüntethet, az ezekre utaló bejegyzéseket az útválasztó **hosztspecifikus** táblázatában helyezi el. Ez sokszor biztonsági, tesztelési célokat szolgál.

**UTVONAL (IP_Datagram,routing_table)**
A cél IP-cím kiemelése a datagramból: $I_D$
A cél hálózat IP-címének kiszámítása: $I_N$
**Ha** az $I_N$ egyezik valamelyik közvetlenül összekötött lokális hálózattal, akkor a datagram elküldése.
**egyébként ha** az $I_D$ előfordul mint hosztspecifikus útvonal
a datagram átirányítása a táblázatnak megfelelően
**egyébként ha** $I_N$ előfordul az útvonal táblában
a datagram átirányítása a táblázatnak megfelelően
**egyébként ha** van alapértelmezési útvonal
a datagram átirányítása az alapértelmezett útválasztónak
**egyébként** útvonal kiválasztási hiba deklarálása

Az IP útvonal kiválasztás nem könnyű, mert a hálózati pontok között többszörös útvonalak létezhetnek. Sok esetben az útválasztó szoftver olyan dolgokat is figyelembe vehet, mint a hálózat terhelése, a datagram hossza vagy a szolgáltatás típusa a datagram fejrészében, amikor eldönti a legjobb útvonalat. (A hálózati réteget tárgyaló 5. fejezet-ben erről, mint mértékről írtunk.)

### *Az útvonal kiszámítás algoritmusa*

A fentieket összefoglalva létrehozhatjuk az útvonal-kiválasztás algoritmusát. Két paraméter szükséges: a címeket (és adatokat) tartalmazó IP datagram, és az útválasztó (útvonal) táblázat.

### *Statikus és dinamikus útválasztás*

Az útvonal választó táblák kézi módosítása (alapértelmezett átjárók beállítása) jól működik kevés útválasztót tartalmazó összekapcsolt hálózatok esetén.

Bonyolultabb hálózatok esetén a megoldás az, ha az egymással kapcsolatban lévő útválasztók egymásnak tovább tudják adni a rendelkezésükre álló útválasztási információkat, amelyeket felhasználva módosítják saját útválasztási táblázataikat. Az elsőnek leírt kézi, állandó táblákat használó módszer a statikus útválasztás, míg az információk alapján változó a dinamikus útválasztás módszere.

Amikor a dinamikus útválasztásssal működő routereket elindítjuk, akkor a táblában először azok a bejegyzések jelennek meg, amelyek az útválasztókkal fizikai kapcsolatban lévő hálózati utakat írják le. Ezután az útválasztók egymással kommunikálni kezdenek, és az így szerzett ismeretek alapján megkezdik táblázataik frissítését.

Az útválasztási információk cseréjére speciális kimondottan erre a célra kifejlesztett, az RFC dokumentumokban részletesen leírt protokollokat használnak. Például: RIP — Routing Information Protocol (az útválasztás információ protokollja), OSPF — Open Shortest Path First (a legrövidebb utat elsőként megnyitó protokoll).

### *Forgalomirányítás*

Az ARPANET eredetileg az elosztott forgalomirányítási algoritmust használta. Ez - egyrészt néhány csomag, tartós hurokba kerülését okozta, - másrészt nem használt alternatív utakat. Mikor a hálózat mérete megnövekedett, akkor a forgalomirányító táblák kicserélésével előálló terhelés már olyan nagy volt, hogy akadályozta a normál forgalmat is.

Ezért a megváltoztatott jelenlegi algoritmusban minden egyes router belsőleg fenntart egy adatbázist, amely az egyes vonalakon való késleltetéseket tartalmazza. Erre az adatbázisra alapozva, minden router kiszámolja a közte és a környezetében lévő többi router közötti legrövidebb utat. A számítás mértékéül a késleltetést használja.

Mivel minden egyes IMP a legrövidebb út algoritmust (majdnem) ugyanarra az adatbázisra alapozva futtatja, ezért az utak konzisztensek és kevés hurok alakul ki. A forgalom és a topológia változásaihoz való alkalmazkodás érdekében minden útválasztó 10 másodperces átlagolási idővel méri vonalain a késleltetést. E mérések eredményét egy aktuális sorszámmal ellátva minden router megkapja. Az információ köröztetéséhez az útválasztók az elárasztásos algoritmust használják.

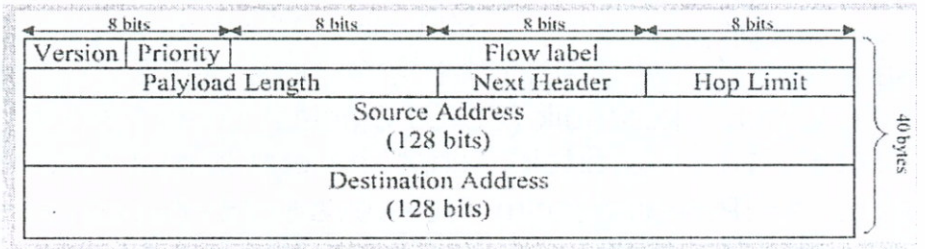
## 7.2.6 IPV6

### Az IPv4 problémái

Az IP protokoll legnagyobb problémája, hogy 4 bájtos címeket használ. Ezért a hálózatba kapcsolható gépek maximális száma $2^{32}$, (kb. 4 milliárd). A valóságban ennél jelentősen kevesebb, a címosztályokra osztás, és a fenntartott speciális címek miatt.

A kioszthatő IP címek elfogyásának orvoslására több megoldás született. Ezek többsége azon az elven működik, hogy nem szükséges egy belső hálózaton minden felhasználó számára külön IP címet adni, elég egy belső lokális cím is. Ezek a gépek az internetre csatlakozó útválasztókon keresztül csak egy közös külső IP címen fognak látszani a hálózaton. Ez a megoldást a későbbiekben még részletesebben tárgyaljuk. Ez a **NAT (Network Access Translation).** A módszer hátránya, hogy ezek a bizonyos felhasználók kifelé bármit elérnek az interneten, viszont őket egyenként kívülről nem lehet megcímezni.

Egy másik probléma az, hogy az internet a mai formájában nem alkalmas adatfolyamok kezelésére. Minden információ feldarabolt, kis csomagok formájában, más és más utakon közlekedik, a csomagok nem sorrendben érnek célba. Ez a tulajdonság általában előnyös, de főleg a valós idejű videó-, és hang adatok továbbításakor hátrányos.




- VERSION (4 BIT): 0110 AZAZ 6
- PRIORITY (4 BIT): VALÓSIDEJŰ VAGY NORMÁL ADATFOLYAM
- FLOW LABEL (24 BIT): ADATFOLYAM AZONOSÍTÓ A GYORSABB TOVÁBBÍTÁSHOZ
- PAYLOAD LENGTH (16 BIT): A CSOMAG MÉRETE (MAX. 65535)
- NEXT HEADER (8 BIT): A FEJRÉSZ UTÁN KÖVETKEZŐ ADATMEZŐ TÍPUSA
- HOP LIMIT (8 BIT): MINDEN ROUTER CSÖKKENTI EGGYEL, AMINT ZÉRÓ ELDOBJUK

*7-9. ábra: IPV6 fejléc*

A problémák megoldása érdekében kidolgozott **IPv6 protokoll** nagyobb, 128 bites cím tartományt használ, ami több mint elegendő. A fejléc felépítése a *7-9 ábrán* látható. Az internet fejlődése miatt az IPv6 protokoll szinten, mintegy beépítve támogatja a titkosítást. Támogatja a mobil eszközöket, és tartalmazza a "Mobile IP" nevű szolgáltatást. A folyamatos jelfolyamok kezelésére is ad megoldást.

### Az IPv6 címzése

Egy IPv4-es cím 4 bájtos. Ez leírva négy decimális alakú szám ponttal elválasztva.

Egy IPv6-os cím 128 bites, azaz 16 bájtos. Ezt kettősponttal elválasztott hexadecimális számokkal írjuk le, és minden szám 16 bitet reprezentál. Tehát elvileg 8 darab hexadecimális szám kettősponttal elválasztva. pl.: `cd73:0:0:0:0:3e6:de12:ab`

**Leírási szabályok:**

- Ha egy címben egymásután több nulla van, akkor azt két kettősponttal lehet helyettesíteni, olyan módon, hogy a címben csak egyszer fordulhat elő, így egyértelműen megmondható, hogy hány nullát helyettesít. pl.: `cd73::3e6:de12:ab`

- A címen az is látszik, hogy nem kötelező minden hexadecimális számot négy számjegyre kiegészíteni. Az 'ab' valójában '00ab', de a bevezető nullák elhagyhatóak.

Az IPv6 címek - hasonlóan a klasszikus IP címekhez - alapvetően két részre vannak osztva. Az első fele a hálózatot címzi, míg a második fele a host-ot. Az előzőekben már bemutatott címmaszkal tudjuk a hálózati és gépcímet szétválasztani. Pl.: `255.255.255.0`-ás hálózati maszk azt jelenti, hogy az első 24 bit a hálózatot határozza meg.

Az eddig létező hálózati címosztályok megszünnek, és az Ipv6-ban így adjuk meg egy hálózat címét:

`1028:2e5:fa01:125b::/64`

Tehát az első 64 bit a hálózatot határozza meg.

Ha esetleg alhálózatokra akarunk felbontatni egy hálózatot, akkor szokásos megoldás, hogy 48 bit azonosítja a hálózatot, 16 bit az alhálózatot, és 64 bitet a gép azonosítására használunk. Nagy ötlet: a 64 bites gépcím a gépben lévő 48 bites hálózati kártya MAC címéből képződik (EUI64 ajánlás) olyan módon hogy a közepébe beszúródik egy "`fffe`". 16 bites érték. Tehát, ha egy kártya hardver címe: `00:50:8d:a2:a9:57`, akkor az IPv6-os gépcím `00:50:8d:ff:fe:a2:a9:57` lesz. Ezért a gép saját maga meg tudja határozni a saját gépcímét, és a hálózatot is megtudhatja a legközelebbi átjárótól/routertől, mivel IPv6-ban a routerek bizonyos időközönként "hirdetik" magukat, a hirdető üzenetben pedig benne van a hálózati címük. Ideális esetben a gépek minden beállítás nélkül ki tudják találni a saját IP címüket.

### *Névfeloldás*

A névfeloldás az IPv4-ben is hasonlóan működik. Általában kétféle módszert használhat a rendszer. Vagy megtalálja a keresett névhez tartozó címet az "`/etc/hosts`" állományban, vagy egy DNS szerverhez fordul. Az IPv6 megvalósítástól függ, hogy az "**/etc/hosts**" fájlban csak IPv6-os címek lehetnek-e, vagy keverve IPv4-esekkel. A DNS szerverek közül sem mindegyik képes IPv6-os címekhez neveket szolgáltatni, csupán a legújabbak tudják ezt.

### *Kompatibilitás*

Az IPv6 – mivel teljesen más csomagformátumot, használ, mint az előző verzió, - nem kompatibilis az IPv4-el. Mivel a legtöbb internet protokoll is az IP-re épül, kérdés az, hogy mi történik akkor, ha alattuk lecseréljük az IP-t. A rétegszemléletből következik, hogy a felső rétegeket az alattuk lévőtől függetlenül tervezték. Tehát újraírni őket nem kell (teljesen), de bizonyos módosítások szükségesek, hiszen számos program, a régi formátumú, és méretű IP címekre, régi szolgáltatásokra számít.

## *7.2.7 Mobil kommunikáció*

A mobilitást az internet jelenlegi felépítése nem támogatja, mivel az IP protokollt eredetileg helyhez kötött gépekhez fejlesztették ki, ezért nem tudja kezelni az alhálózatok között mozgó, állandó IP című gépeket.

A mobilitás tárgyalásakor könnyű két fogalom összekeverése: ez a **hordozhatóság** – amikor az ember a számítógépét különböző helyeken kívánja használni, és magával viszi a gépét – illetve az igazi, a mobiltelefonoknál már megszokott **mobilitás**, amikor a felhasználó utazás közben is fenntartja az Internetes kapcsolatát. Ehhez természetesen vezeték nélküli, rádiós kapcsolat szükséges, és mozgás közben egy terület kapcsolatát biztosító bázisállomás váltásokra is szükség van.

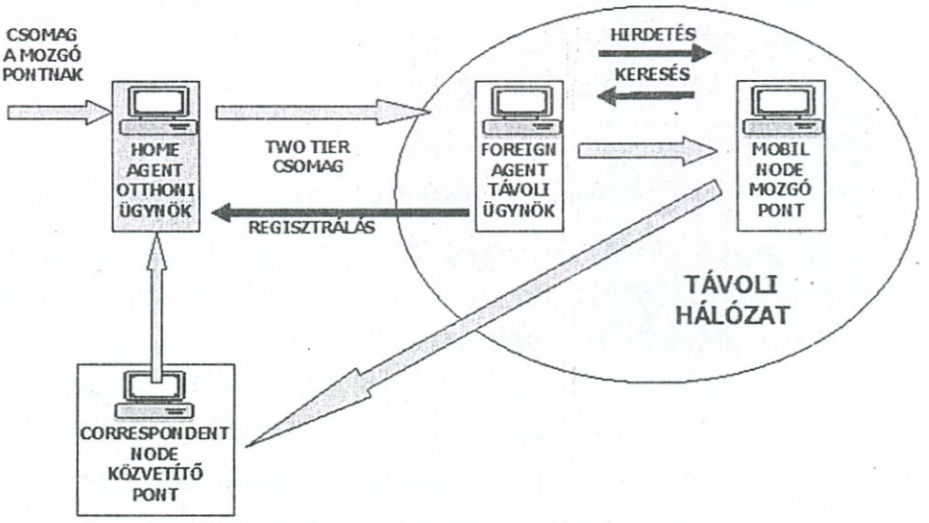
Az eddigi tanulmányaink alapján nyilvánvaló, hogy ezt a problémát IP címek váltásával nem lehet megoldani, mert az összeköttetést kezelő TCP összeköttetéseknek is változni kellene.

A megoldás erre a problémára az ún. **„two-tier addressing" (=kettős címzés)**, aminek az a lényege, hogy minden mozgó számítógéphez logikailag két IP címet rendelünk: Az egyik az állandó, a gép globális azonosítója (**Home Address**), ami mindig állandó, míg a másik ideiglenes, (aktuális), a mozgás során változik (**Care-of Address**).

Mivel a mobilitás egy mobil útválasztóval is megvalósítható (gondoljunk egy repülőgépre, és a rajta lévő, ehhez az útválasztóhoz kapcsolódó számítógépekre) a mobil eszközre célszerűbb a **mobil node (mobil pont, azaz mozgó (csomó)pont)** kifejezést használni.

Ebből következik, ha egy mobil pontnak adatcsomagot akarunk küldeni, annak csak állandó címét (**Home Address**) kell ismernünk. Egy dinamikusan változó, a mozgások során folyamatosan karbantartott adatbázisból mindig hozzárendelhető, az állandó címhez a mindenkori aktuális ideiglenes cím, és ennek alapján fogja a hálózat a gépre a csomagot irányítani.

Ilyen hálózatban minden mobil állomásnak küldött csomag kettős címzést használ: azaz mindkét cím szerepel benne. A csomag irányítása során az ideiglenes cím játssza a meghatározó szerepet: ennek alapján jut el a csomag a mobil géphez. A csomag feldolgozása már az állandó IP cím alapján történik, vagyis a belső folyamatok már ezt a címet használják.




*7-10. ábra Mobil kommunikáció*

Ha egy mobil gépről küldünk csomagot egy fix hálózatba kötött gépnek, erre a megoldásra nincs szükség, hisz ekkor a hagyományos útválasztás segítségével valósulhat meg a csomagküldés.

A működéshez, néhány további kiegészítő elemre, illetve szereplőre is szükség van:

- **Home Agent (otthoni ügynök):** minden mobil pontnak „van otthona": ez a **Home Network**. Úgy képzeljük el, hogy mozgás nélkül egy olyan alhálózatban

van, ahonnan az állandó IP címét kapja. Ebben az alhálózatban van egy gép, a home agent, amely észleli a mobil node-nak küldött csomagokat, és eljuttatja a mobil node aktuális tartózkodási helyére. A Home Agent rendelkezik azon információval, hogy a mobil gép az internetes hálózatban éppen hol tartózkodik, az előbbiekben említett folyamatosan karbantartott adatbázis alapján.

- **Foreign Agent (távoli (külföldi) ügynök):** az a számítógép, amelyhez megérkeznek a mobil pontnak szánt csomagok. Ez a gép abban az alhálózatban van, amelybe a mobil gép is bekerült, és így annak a csomagokat továbbítani tudja.
- **Correspondent Node (Közvetítő pont):** a mobil ponttal kommunikáló számítógépek gyűjtőneve, ami akár egy másik mobil pont is lehet.
- Azért, hogy az átvitel során ez a kettős címzés ne igényeljen speciális csomagformát, az eredeti IP csomagot fejlécestől együtt beteszik egy másik IP csomagba, ami lehetővé teszi, hogy a csomag a mobil pont aktuális címe alapján haladjon a hálózatban. Ez a módszer a **beburkolás** (encapsulation). Ilyen módon a csomag nem az eredeti címzése szerint, hanem külön „alagúton" halad a hálózatban. Ezt a módszert hívják **tunneling**-nek. (tunnel=alagút).
- **Foreign Network:** minden, Home Network-től különböző hálózat.
- **Regisztáció:** A mozgó pontok egy üzenettel jelzik az otthoni ügynöküknek, hogy éppen melyik alhálózatban vannak. Ezt az üzenetet (*Registration Request*), először a tartózkodási hely hálózati ügynökének (Foreign Agent) küldik el, majd az továbbítja az otthoni ügynöknek, amire az ügynök, egy nyugta csomaggal (*Registration Reply*) válaszol.
- A mobil kommunikáció megvalósításához a résztvevők kapcsolódását két fontos periodikus üzenetcsomag biztosítja: **Agent Advertisement Message** – az ügynökök ezzel jelzik a hálózatban a jelenlétüket, **Agent Solicitation Message** – ezzel a üzenettel keresi az idegen hálózatban a mobil pont a kapcsolatot az ügynökökkel.

Az előző fogalmak felhasználásával már bemutatható a Mobil IP Protokoll működése:

Ha egy mobil pont, mozgás közben egy újabb hálózathoz akar csatlakozni, küld egy *Agent Solicitation* üzenetet a hálózat távoli ügynökének (persze közben az is hirdeti magát az *Agent Advertisement* üzenetekkel. Ezek az üzenetek egy üzenetszórásos csatornára küldve juthatnak el minden érintetthez. A mobil pont ettől a távoli ügynöktől kapja meg ideiglenes címét. Ezt a címet az otthoni ügynökkel is közli.

Ha most egy mobil pontnak, annak állandó IP címére csomagot küldünk, akkor azt a csomagot az otthoni ügynök veszi, majd kiegészíti a megkapott távoli hálózat címével, és továbbítja a hálózat távoli ügynökének, aki ezt az ott tartozkodó mobil állomásnak továbbítja.

Ha valaki figyelmesen elolvasta az eddigieket, annak nem nehéz észrevenni a nagyfokú hasonlóságot a már bemutatott mobil telefonhálózat működése, és a most megismert mobil kommunikáció között.

## 7.3 A TCP/IP protokoll szállítási rétegei: TCP, UDP

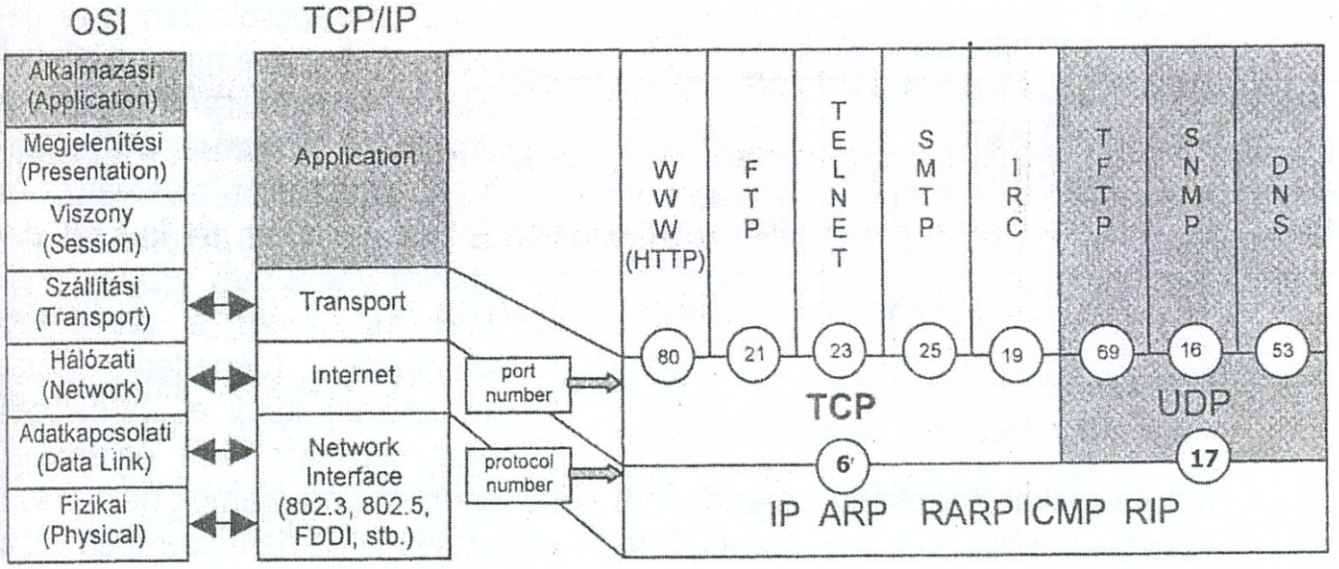
Mivel az interneten, már számos helyi hálózat, rádiós csomagszóró alhálózat; valamint

több műholdas csatorna is működik, végpontok közötti átviteli megbízhatóság csökken. Ezért az eredetileg bevezetett helyett egy új szállítási protokollt,- a TCP-t vezettek be 1980-ban. A TCP tervezésénél már figyelembe vették azt, hogy megbízhatatlan (az OSI terminológia szerint C típusú) alhálózatokkal is tudjon együttműködni. A TCP-vel együtt fejlesztették a hálózati réteg protokollját (IP) is.

## 7.3.1 Portok

Mivel egyszerre egy gépen futó több alkalmazás is folytathat más gépeken futó alkalmazásokkal TCP vagy UDP kommunikációt, ezért feltétlenül gondoskodni kell arról, hogy a különböző alkalmazások által küldött, illetve fogadott információk ne keveredhessenek össze.

Ennek megvalósítására szolgál a **portok rendszere**. A 16 bites címmel magadott portokat legegyszerűbben úgy képzelhetjük el, mint virtuális postafiókokat. **Minden kommunikációra képes programhoz hozzárendel az operációs rendszer egy-egy (vagy esetleg több) portot**. Ha a beérkező csomagokban a megfelelő portszám található a célport mezőben, a csomag bekerül a megfelelő postafiókba, azaz bekerül az adott porthoz rendelt várakozási sorba, melyből csak a portot lefoglaló alkalmazás olvashatja ki. Küldéskor fordított a helyzet, a csomag forrásport mezőjébe kerül be az alkalmazáshoz rendelt portszám, így a kommunikációs partner válaszát már az adott alkalmazás portszámára „címezheti".




*7-11. ábra Prokollok, portok rendszere*

Minden alkalmazásnak, mielőtt adatokat küldetne, vagy fogadhatna, egy, még szabad portot kell kérvényeznie az operációs rendszertől. Ettől kezdve küldéskor ennek a portnak a száma kerül majd a TCP vagy UDP fejléc forrásport mezőjébe, illetve fogadáskor azokat a csomagokat fogja az operációs rendszer az alkalmazásnak továbbítani, amelyeknek a célport mezőjében az általa lefoglalt port száma szerepel.

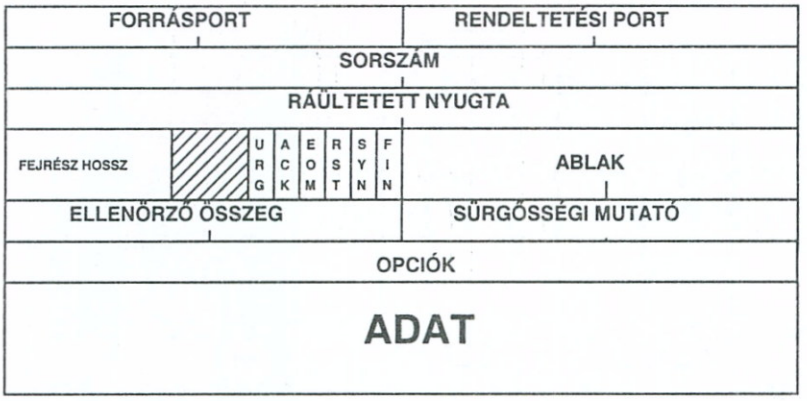
A portszámok okos kiosztása létfontosságú a hatékony információcsere szempontjából. Ha a küldő oldal nem ismeri a fogadó oldalon a megcímezni kívánt alkalmazás által használt portszámot, akkor képtelen lesz annak bárminemű információt küldeni. Ennek a problémának a feloldására kétfajta megoldás terjedt el.

- A portokat egy általánosan elfogadott elv szerint osztják ki az alkalmazások között, így előre tudható, melyik alkalmazást melyik porton lehet megtalálni.
- A portok dinamikusan is kioszthatóak, ekkor nincs előre eldöntött portkiosztási rend. Ha valamelyik alkalmazásnak szüksége van egy portra, azt dinamikusan foglalhatja le, a kommunikáció végeztével pedig „fölszabadíthatja", azaz visszaszolgáltathatja az operációs rendszernek, további felhasználás céljából.

A TCP/IP esetében **hibrid megoldást** alkalmaznak. A teljes, 65536 darabból álló portszám tartomány első 1024 portját közmegegyezés alapján előre meghatározott alkalmazások számára tartják fenn. Ezek a „jól ismert" kiszolgáló alkalmazások, azaz főleg bejövő kapcsolatok fogadása, és feldolgozása a fő feladatuk.

A fennmaradó portszám tartományt dinamikusan (és véletlenszerűen) osztják ki azok között az alkalmazások között, amelyeknek szükségük van rá. Így egyrészt megoldható, hogy a későbbiekben bemutatandó, elterjedt internet szolgáltatások (pl. HTTP, SMTP, FTP, DNS, TFTP, DHCP, stb.) egyszerűen címezhetőek legyenek, mivel egységes megállapodás, hogy mely alkalmazás mely porton várja a bejövő kapcsolatokat. A kliensalkalmazások, melyeknek nincs szükségük állandó és előre definiált portszámra, dinamikusan foglalhatnak portokat, hiszen jellemzően amúgy is kezdeményezőként létesítenek kapcsolatot más számítógépeken futó alkalmazásokkal.

## 7.3.2 TCP (RFC793/1981)




*7-12. ábra: TCP csomag*

A TCP a kapcsolatban lévő gépeken futó folyamatok kommunikációját biztosítja. Fogadja a tetszőleges hosszúságú üzeneteket a felhasználói folyamattól, és azokat maximum 64 kbájtos darabokra vágja szét. Ezekhez fejlécet fűz, majd a darabokat, egymástól független datagramokként küldi el. A hálózati réteg sem azt nem garantálja, hogy a datagra-mokat helyesen kézbesíti, sem a megérkezett datagramok helyes sorrendjét. Figyeljük meg: a TCP fejrészében a forrás és célgép címe nem szerepel: ezt a TCP csomagot beburkoló IP csomag fejléce tartalmazza. A portok valójában az adott gépeken futó kommunikáló programok ki- és bemenetének a címei.

A TCP protokoll segítségével a két kommuikáló folyamat „látja" egymást, a köztük lévő távolságtól függelenül. A TCP feladata az, hogy időzítéseket kezelve, szükség szerint újraadja őket, illetve, hogy helyes sorrendben rakja azokat össze az eredeti üzenetté.

Minden TCP által elküldött bájtnak saját sorszáma van. A sorszámtartomány 32 bit széles, vagyis elegendően nagy ahhoz, hogy egy adott bájt sorszáma egyedi legyen. A TCP által használt 32 bites egységekből álló fejrész a *7-12. ábrán* látható. A minimális TCP fejrész 20 bájt hosszúságú.

A **FORRÁSPORT** és a **CÉLPORT** mezők az összeköttetések végpontjait (TSAP-címek az OSI terminológia szerint) azonosítják. Minden egyes hosztnak magának kell eldöntenie, hogy miképpen allokálja (osztja ki) a portjait.

A 32 bit hosszúságú **SORSZÁM** jelöli az üzenetben a bájt sorszámát. A TCP az üzenet minden bájtját külön sorszámozza.

**RÁÜLTETETT NYUGTA** a csúszóablakos protokolloknál ismertett módon a kapott TCP csomagok nyugtáit tartalmazza.

A **FEJRÉSZ HOSSZ** kijelöli, hogy a TCP fejrész hány 32 bites szót tartalmaz. Erre az információra az OPCIÓK mező változó hossza miatt van szükség. Alapértelmezés: 5.

Ezután az egész **kommunikációt alapvetően meghatározó hat jelzőbit** következik.

Az **URG** jelző akkor 1, ha a protokoll használja a **SÜRGŐSSÉGI MUTATÓ**-t (Urgent pointer). Ez valójában egy eltolási értéket ad meg, amely az aktuális sorszámtól számolva kijelöli a sürgős adatok helyét.

A **SYN** és **ACK** biteknek összeköttetés létesítésekor van funkciója. Összeköttetés kérésekor SYN=1, valamint ACK=0 annak jelzésére, hogy a ráültetett nyugta mező nincs használatban. Az összeköttetés válaszban van nyugta, így SYN=1 és ACK= l.

A **FIN** az összeköttetés lebontására használható, azt jelzi, hogy a küldőnek nincs több adata. A hoszt hibák miatti nem jó állapotba került összeköttetéseit az **RST** bit használatával lehet megszüntetni. Az **EOM** bit az Üzenet vége (End Of Message) jelentést hordozza.

A TCP-beli forgalomszabályozás változó méretű forgóablakot használ. 16-bites mezőre van szükség, mivel az **ABLAK** azt adja meg, hogy hány bájtot lehet még elküldeni.

Az **ELLENŐRZŐÖSSZEG** képzési algoritmusa egyszerű: 16-bites szavakként az adatokat összegzik, majd az összeg 1-es komplemensét veszik. Vételkor az összegképzést - az ellenőrző összeget is beleértve - ismételten elvégzik. Hibátlan átvitel esetén ez az összeg nulla lesz.

Az **OPCIÓK** mező különféleképpen használható fel, pl. összeköttetés létesítése során a puffer-méret egyeztetésére.

Az alkalmazások a TCP protokoll szolgáltatásait megválaszthatják, vagyis azt, hogy az információátvitel milyen módon történjen:

- **Stream Orientation:** az átvitt adat bitsorozat, nem tételezünk fel semmilyen belső struktúrát, ha kell, 8 bit-re ki kell egészíteni.
- **Virtual Circuit Connection:** duplex átvitel előtt erőforrásokat kell allokálni (lefoglalni) mind a két oldalon, a kapcsolatot mindkét oldalon meg kell nyitni, adatátvitel során az adatokat nyugtázni kell.
- **Buffered Transfer:** a protokoll úgy tördeli az adatot, ahogy neki megfelel, de sorrendhelyesen adja át a másik oldali alkalmazásnak.
- **Unstructured Stream:** nem lehetünk biztosak abban, hogy a beérkezett adat pont rekordhatáron érkezett meg, a protokoll semmit se tud a rekordokról.

- **Full Duplex Connection:** a kapcsolatra úgy tekinthetünk, mintha két egymástól független ellentétes irányú csatornánk lenne, lehetséges, hogy az egyik irányt már lezártuk, de a másik irányba még áramlanak adatok (piggybacking).

Az adatávitel hibatlanságának biztosítására a TCP a pozitív nyugtázást használja, hiánya esetén újraadja a csomagot. Az adó elküldi az adatot (de még tárolja), elindít egy időzítőt, majd pozitív nyugtára vár. Ha megjön a nyugta, akkor ezzel vevő jelzi, hogy átvette az adatot, ha lejár az időzítő, akkor az adó újraküldi a csomagot.

Duplikált csomagok ellen védelmül az újrasorszámozást alkalmazza, csúszóablakos technikával, vagyis egyidőben egynél több nyugtázatlan csomag is úton lehet.

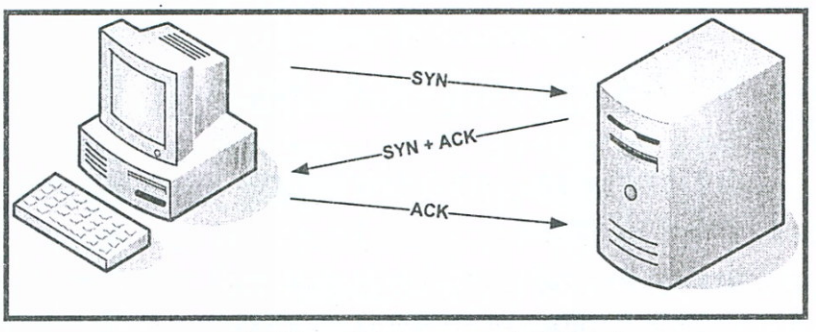
### *Háromutas kézfogás a TCP protokollban*

A TCP kapcsolat létesítésénél három alapproblémát kell megoldani:

1. A küldött csomagok sorrendje megváltozhat az átvitel során.
2. A csomagok eltűnhetnek.
3. A csomagok megkettőződhetnek.

A megoldás az első és második problémára egyszerű: **sorszámmal kell a csomagokat ellátni**. Ez a második problémára is azért megoldás, mert a nyugtázás elmaradása jelzi a küldőnek a hiányzó csomagok ismételt elküldésének igényét.

A harmadiknak említett probléma – a megkettőződés - úgy jöhet létre, hogy például, mikor egy útválasztóba érkezik a csomag, az útválasztót valamiért leválasztjuk egy rövid időre a hálózatról, majd ismét visszakapcsoljuk. Mivel a vevő nem kapta meg a csomagot, a nyugta segítségével ezt jelzi, és a küldő ismét elindítja a hiányzó csomagot, akkor, ha a csúszóablakos protokoll ezt még lehetővé teszi. Ha nem, akkor egy új összeköttetést kell létrehozni. Hogy ne okozzon a kettőződés problémát: van egy új összeköttetés, és benne csomagok, meg van egy kószáló, a régi összekötettésből származó csomag. Ezért **a küldendő és fogadott csomagok kezdeti sorszámát (Initial Sequence Number = ISN) az összeköttetés kezdetekor véletlenszerűen hozza létre a kommunikáló két oldal.**




*7-13. ábra Háromutas kézfogás*

Ezek alapján a kapcsolat létesítésekor a K (=kezdeményező) és F (=fogadó) betűkkel jelölt felek három csomag segítségével indítják a kapcsolatot. Mivel a csomagokat az adott IP című gépeken futó alkalmazások portcímeire kell küldeni, ezért a kapcsolatot kezdeményezőnek ismernie kell a fogadó alkalmazás port címét. Ezt a TCP/IP protokollban az RFC1700-ban definiált, a jól ismert alkalmazásokhoz tartozó port címek rendszerével oldották meg. A kapcsolat létesítése előtt a kezdeményező a kapcsolathoz egy saját, nem lefoglalt KP portcímet rendel, és ismeri a fogadó jól ismert FP portcímét.

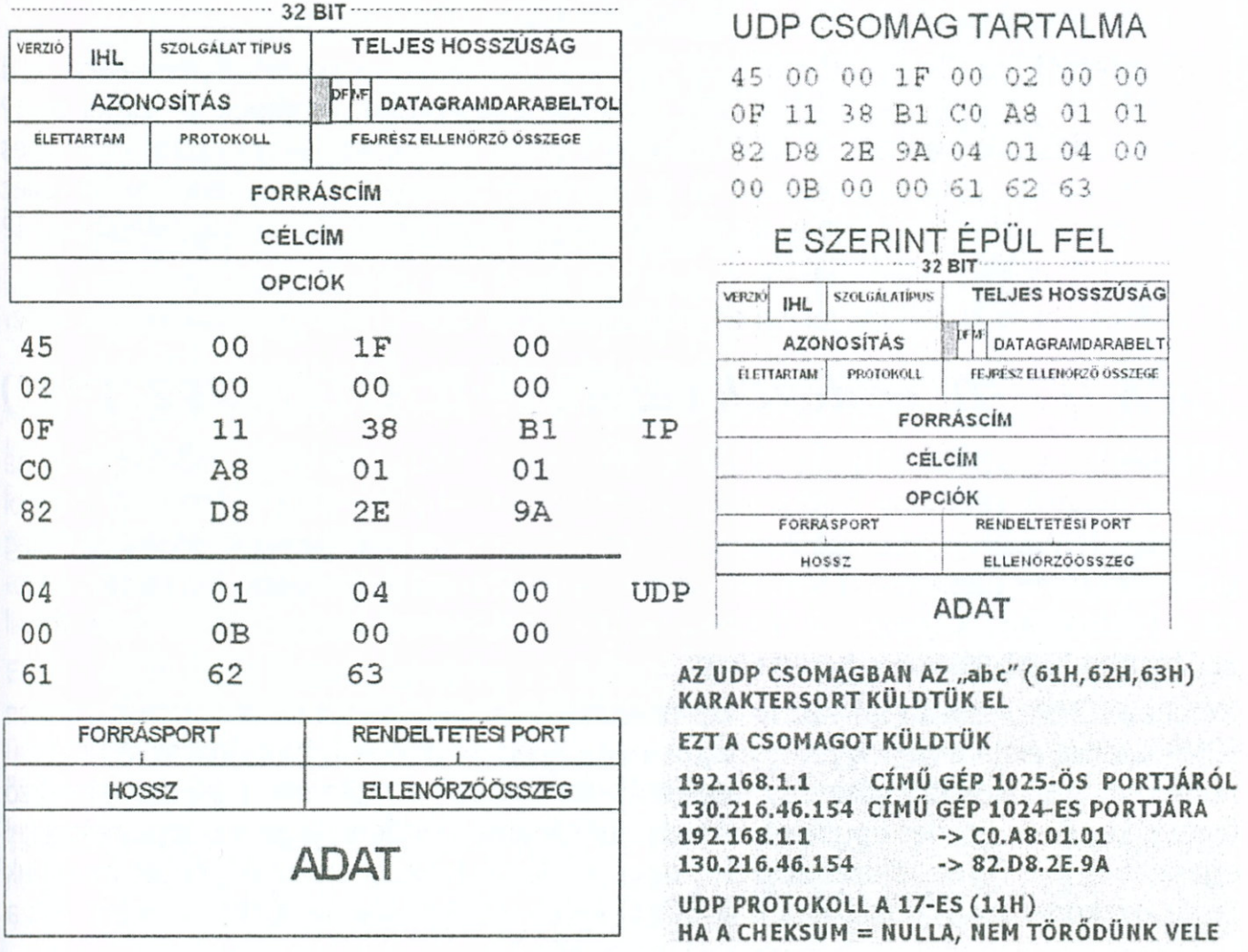
1. K küld egy TCP csomagot, amiben forrásportcímként a saját KP és célportcímként a fogadó jól ismert FP címét adja meg. A kapcsolat során a küldendő K_ISN kez-

deti sorszámot a sorszám mezőbe helyezve küldi el, és a csomagban SYN=1 bit jelzi a kapcsolatfelvétel kezdetét, és azt hogy a csomag kezdősorszámot tartalmaz.

2. Válaszolva a csomagra a K által megadott KP portcímre elküld egy csomagot, amiben nyugtázza a küldött K_ISN kezdeti sorszámot, és elküldi az általa a kapcsolatban használt csomagkezdő F_ISN sorszámot a sorszám mezőben, és ACK=1 bittel jelzi, hogy a kapcsolatfelvételt elfogadta, és az SYN=1 bittel pedig azt jelzi, hogy csomagkezdő sorszámot küld.
3. Harmadik lépésként nyugtázásul K egy olyan csomagot küld, amiben az ACK=1 (elfogadja a kezdősorszámot). Ez a megoldás megbízható kapcsolatfelvételt biztosít a további kommunikációhoz.

## 7.3.3 UDP (RFC 768/1980)

A másik szállítási protokoll az **UDP (User Datagram Protocol felhasználói datagram protokoll).** Hasonlóképpen illeszkedik a rendszerbe, mint a TCP. Az UDP összeköttetés-mentes protokoll. Az UDP információját egy IP csomagba helyezi, ellenőrző összeget számol hozzá és feladja. Így a kézbesítést nem garantálja, de a hibás kézbesítést észlelhetővé teszi. Olyan kérdés-válasz jellegű szolgáltatásokhoz használa-




*7-14. ábra IP/UDP csomag*

tos, ahol - ha a kérdés vagy a válasz elvész -, a hiba egyszerű újrakérdezéssel megoldható. Sok alkalmazás használ üzeneteket, amelyek elférnek egyetlen datagramban, és nem igényelnek nyugtázást. Példa erre a domén nevek kikeresése.

Amikor egy felhasználó egy másik rendszerrel kapcsolatba akar lépni, akkor általában az adott rendszer IP címe helyett a nevét fogja megadni. Ezt a nevet le kell fordítania IP címre. Ehhez a névszolgáltatóhoz kell a kérést tartalmazó üzenetet eljuttatni. A kérés annyira rövid, hogy biztosan elfér egyetlen datagramban, és a válasz ugyanilyen rövid. Ilyenkor nincs szükség a TCP teljes bonyolultságára. Ha egy pár másodpercen belül nem kapunk választ, akkor egyszerűen megismételjük a kérdést. Ezt a protokollt használja fel a hangtovábbító VoIP protokoll is.

A hálózati szoftver az adatok elejére ráilleszti az UDP fejlécet ugyanúgy, ahogy a TCP fejléc esetében teszi. Az UDP ezek után az IP-nek adja át a datagramot. Az IP hozzáteszi a saját fejlécét, amibe a TCP helyett az UDP protokollszámát helyezi el a Protokoll mezőben (lásd IP fejléc). Az UDP csak portszámokat biztosít, hogy egyszerre több program is használhassa a protokollt. Az UDP portszámok ugyanúgy használatosak, mint a TCP portszámok. Az UDP-t használó kiszolgálókhoz is léteznek jól ismert portszámok. Látható, hogy az UDP fejléc sokkal rövidebb, mint a TCP fejléce. Hogy az egész jól érthető legyen, a *7-14. ábrán* részletezett feladatot elemezzük röviden. A feladat: UDP protokollal átvinni az egyik gépről az „abc" karaktersorozatot a másik gépre. Az átvitel során egy IP csomagba illesztjük az UDP üzenetet.

Emlékeztetőül, az ábrán látható az IP csomag felépítése, a protokoll mezőjébe 11H (decimális 17) kerül, mert ez az UDP protokoll azonosítója. A gépek IP címei az IP csomag fejlécében vannak. Az UDP csomag - amely fejlécében tartalmazza a forrás és a célport címét, az üzenet hosszát és az ellenőrző összeget - közvetlenül az IP fejléc után kerül, majd utána vannak az adatok. Vagyis a teljes UDP csomag felépítése: IP fejléc – UDP fejléc – UDP adatok.

Az UDP protokoll is definiál jól ismert portcímeket az UDP-t használó alkalmazásokhoz.

## 7.4 A TCP/IP protokoll vezérlése:ICMP(RFC792/1981)

Az Internet működését az IMP-k és az útválasztók felügyelik olyan módon, hogyha valami gyanús esemény fordul elő, akkor az eseményt az ICMP (Internet Control Message Protocol — internet vezérlőüzenet protokoll) alapján jelentik. Megközelítőleg egy tucat ICMP üzenettípus létezik. Minden üzenettípus IP-csomagba burkolva vándorol a hálózatban. A felépítése hasonló az UDP protokollnál tárgyaltakhoz. A protokoll az Internet tesztelésére is használható.

Mivel az ICMP üzenetfajták száma korlátozott, ezért nem portcímeket használ. Minden ICMP üzenet elején van egy négybájtos nyitófejléc. Ennek első bájtjába az ICMP üzenettípusa, a második bájtba egy kiegészítő kód, illetve a maradék 2 bájtba ellenőrző összeg kerül. Az üzenet típusától és bonyolultságától függően újabb 4 bájtos részekkel egészülhet ki a fejléc. Illusztrációként egyetlen típus fejlécének a felépítését mutatjuk be: a gyakorlatban is sokat használt ECHO kérés – ECHO válasz ICMP üzenetét. (Ez a későbbiekben bemutatott Ping parancs használja.)

Az ICMP üzenetek összefoglalása:

**DESTINATION UNREACHABLE** (cél elérhetetlen) üzenet akkor keletkezik, amikor a hoszt, vagy egy átjáró nem tudja lokalizálni a címzettet, vagy amikor egy bebillentett DF bittel rendelkező csomagot egy közbenső "kis csomagú" hálózat miatt nem lehet kézbesíteni.

**TIME EXCEEDED** (időtúllépés) üzenet küldésére akkor kerül sor, ha egy csomagot a nullára csökkent számlálója miatt el kell dobni. Ez az esemény tünete lehet annak, hogy a csomag hurokban kering, hogy súlyos torlódás van, vagy hogy az időzítés értéke túl kicsire van beállítva.

**PARAMETER PROBLEM** (paraméterprobléma) üzenet azt jelzi, hogy illegális értéket vett észre valaki egy fejrészmezőben. Ez a probléma a küldő hoszt IP-szoftverének, vagy egy keresztezett átjáró szoftverének hibájára hívja fel a figyelmet.

**ECHO_REQUEST, és ECHO_REPLY csomag felépítése**

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| Típus | Kód | Cheksum | |
| Azonosító | | Sorszám | |
| Adatok... A hossza nem kötött, bárhol végződhet | | | |

Az ECHO_REQUEST egy kérés, hogy akinek küldjük, küldje vissza pontosan ugyanazt, amit kapott. Az ECHO_REPLY, pedig erre a válasz. Ezzel lehet megnézni, hogy van-e, és ha igen, akkor milyen gyors hálózati kapcsolat két gép között.
Az egyes mezők értékei:

- Típus: kérés esetén 8, válasz esetén 0
- Kód: mindig nulla
- Cheksum: Egy ellenőrző összeg. Jelzi ha meghibásodott az üzenet.
- Azonosító: Egy azonosító, hogy tudjuk, a válasz melyik kérésre jött. Lehet nulla is.
- Sequence number: Egy sorszám, hogy tudjuk, melyik kérésre érkezett a válasz. Lehet nulla is.

A válasz csomagnak ugyanazt az azonosítót, sorszámot, és adatot kell, hogy tartalmazza, mint amit a kérés. A típus persze változni fog, ezért az ellenőrző összeget újra kell számolni.

**SOURCE QUENCH** (forráslefojtás) üzenet a túl sok csomagot küldő hosztok megfékezésére használható. Amikor egy hoszt egy ilyen üzenetet vesz, akkor adási sebességét csökkentenie kell.

Az említett üzeneteken kívül, van még négy másik, amelyek az internet címzéssel foglalkoznak, és lehetőséget biztosítanak a hosztok számára, hogy azonosítsák saját hálózatszámukat, felfedezzék a címzési hibákat: például kezelni tudják azt az esetet, amikor egyetlen IP-címet több LAN birtokol.

**REDIRECT** (újrairányítás) üzenetet akkor küld egy átjáró, amikor észreveszi, hogy egy csomag valószínűleg rossz útvonalon halad. Ez segít a forgalomirányításnak a helyes út, megtalálásához.

**ECHO REQUEST** (visszhangkérés) és **ECHO REPLY** (visszhangválasz) üzenetekkel egy adott címzett elérhetőségét és működőképességét lehet megvizsgálni. Az ECHO üzenet kézhezvételét követően a címzettnek egy ECHO REPLY üzenettel kell válaszolnia. A megcímzett állomás meglétét ellenőrző **PING** parancs ezt használja.

**TIMESTAMP REQUEST** (időpontkérés) és TIMESTAMP REPLY (időpontválasz) üzenetek hasonlóak csak a válaszüzenetben a kérés megérkezésének és a válasz indulásának ideje is fel van jegyezve. Ez a szolgáltatás a hálózati teljesítmény mérésére nyújt lehetőséget.

# 7.5 IP címek kezelése

A most következő részben olyan kérdésekkel foglakozunk, amely a TCP/IP protokoll gyakorlati használata közben merültek fel. Először a fizikai (MAC) címek és IP címek összerendeléséről, utána az IP címek kiosztásáról, illetve közös használatáról írunk.

## 7.5.1 Hálózat elérési réteg — ARP,RARP (RFC826/1982)

Jelenleg a legtöbb hálózat fizikai és adatkapcsolati szinten Ethernet kártyákat használ. Mivel az Ethernet keretnek saját fejléce van, saját egyedi, 48 bites címzéssel rendelkezik, ezért az IP csomagokat ilyen hálózaton közvetlenül nem lehet átvinni, egy Ethernet keretbe kell csomagolni.

Az Ethernet kereteknek egy 14 bájtos fejléce van, amely a forrás- és a célgép Ethernet címét, valamint típuskódot tartalmaz. A hálózaton lévő gépek csak az olyan kereteket veszik, amelyek célmezőjében a saját Ethernet címük, vagy a mindenkinek szóló körözvénycím szerepel.

Minden számítógépnek van egy folyamatosan karbantartott adatbázisa, amely tartalmazza, hogy milyen Ethernet cím milyen Internet címnek felel meg. Ennek karbantartása egy protokoll, az ARP (**Address Resolution Protocol** — címfeloldási protokoll) segítségével történik. Így, egy hoszt megtalálhatja a megcímzendő másik hoszt fizikai címét, csupán annak IP címét ismerve.

### ARP/RARP protokoll formátum

Az ARP végrehajtásakor az Ethernet keret adatrészében átvitelre kerülő, 4*8=32 bites mezőkből felépített ARP/RARP csomag formátumát a *7-15. ábra* mutatja. Tartalmazza a Feladó, és a Címzett, hat bájtos (0...5) fizikai, és 4 bájtos (0...3) IP címét.

| 0 | 8 | 16 | 31 |
|---|---|---|---|
| Hardver típus | | Protokoll típus | |
| HLEN | PLEN | Művelet | |
| Feladó fizikai cím (0-3. bájt) | | | |
| Feladó fizikai cím (4-5. bájt) | | Feladó IP (0-1. bájt) | |
| Feladó IP (2-3. bájt) | | Címzett fizikai cím (0-1. bájt) | |
| Címzett fizikai cím (2-5. bájt) | | | |
| Címzett IP (0-3. bájt) | | | |

*7-15. ábra ARP/RARP csomag felépítése*

A **Hardver típus** mező értéke 1 az Ethernet esetében. A **Protokoll típus** mező értéke az IP-címekre: 0800h. A **Művelet** mező értékei 1 - ARP-kérés, 2 - ARP-válasz, 3 - RARP-kérés és 4 - RARP válasz. A **HLEN** (hardverhossz) és **PLEN** (protokollhossz) mezők hosszúságot tartalmaznak, és azért szükségesek, hogy az ARP tetszőleges hálózatra és tetszőleges protokollra is alkalmas legyen. A válaszoló hoszt a címeket felcseréli.

### Az IP-cím meghatározása induláskor (RARP)

Rendszerint a hosztok a saját begépelt vagy DHCP protokollal megkapott IP-címüket a háttértárolóban őrzik. De mi legyen a lemez nélküli (diskless) gépekkel, ha a TCP/IP protokollt akarják használni? Azért fontos, mert rendszerinduláskor le kell tölteni a működtető operációs rendszert tartalmazó **boot image** információt. Az ilyen eszközöknek az állománykiszolgálók biztosítják a háttértárolót.

A megoldás alapgondolata: feltételezzük, hogy a kiszolgáló ismeri a lemeznélküli gép fizikai címét, és ő párosítja hozzá az IP-címet. Mivel ezek a lemez nélküli gépek nem tudják a kiszolgáló fizikai címét emiatt csomagszórással (broadcast) jelentkeznek be, amire egy, vagy több kiszolgáló válaszol. A lemeznélküli gép önmagát a fizikai címével azonosítja. A megoldás neve: **Reverse Address Resolution Protocol (RARP)**, ami nagyon hasonló az ARP-hez és a formátuma is megegyezik vele. A RARP sikerességéhez a hálózaton legalább egy RARP-szervernek kell lenni.

### ARP protokoll: Példa

Ttegyük fel, hogy a **193.18.26.196 (C1.12.1A.C4)** IP című gépről a **193.18.26.75 (C1.12.1A.4B)** géppel szeretnénk kapcsolatba lépni. A kezdeményező **193.18.26.196** című hoszt megnézi, hogy szerepel-e a saját ARP táblázatában a **193.18.26.75** címhez tartozó Ethernet cím bejegyzés. Ha igen, akkor a datagramhoz egy Ethernet fejlécet csatol, és elküldi. Ha azonban ilyen bejegyzés az ARP táblázatban nincsen, akkor a csomagot nem lehet elküldeni, hiszen nincs meg az Ethernet cím.

```
Packet #1
  Packet Lenght:64
Ethernet Header
  Destination:  FF:FF:FF:FF:FF:FF Ether. Broadcast
  Source:       00:C0:26:80:0A:93
  Protocol Type:0x0806   IP ARP
ARP  - Address Resolution Protocol
  Hardware:                    1  Ethernet  (10Mb)
  Protocol:                    0x0800   IP
  Hardware Address Lenght:     6
  Protocol Address Lenght:     4
  Operation:                   1  ARP Request
  Sender Hardware Address:     00:C0:26:80:0A:93
  Sender Internet Address:     C1.12.1A.C4
  Target Hardware Address:     FF:FF:FF:FF:FF:FF
                               Ethernet Broadcast
  Target Internet Address:     C1.12.1A.4B

Packet #2
  Packet Lenght:64
Ethernet Header
  Destination:  00:C0:26:80:0A:93
  Source:       00:C0:F0:09:BD:C3
  Protocol Type:0x0806   IP ARP
ARP  - Address Resolution Protocol
  Hardware:                    1  Ethernet  (10Mb)
  Protocol:                    0x0800   IP
  Hardware Address Lenght:     6
  Protocol Address Lenght:     4
  Operation:                   2  ARP Response
  Sender Hardware Address:     00:C0:F0:09:BD:C3
  Sender Internet Address:     C1.12.1A.4B
  Target Hardware Address:     00:C0:26:80:0A:93
  Target Internet Address:     C1.12.1A.C4
```

***7-16. ábra ARP kérés és válasz***

Ekkor lép működésbe az ARP protokoll. A **193.18.26.196** hoszt egy „a **193.18.26.75** Ethernet cím kellene" tartalmú ARP kérést ad ki az Ethernet hálózatra. Az adott hálózaton minden hoszt figyeli az ARP kéréseket. Ha egy hoszt egy rá vonatkozó ARP kérést kap, akkor válaszol rá.

Ebben az esetben tehát a **193.18.26.75** hallja a kérést, és egy ARP üzenetet küld válaszul a kérdezőnek, amelyben megadja a **193.18.26.75** IP című gépben lévő kártya Ethernet címét: **00:C0:F0:09:BD:C3.** *(7-16 ábrán: Packet#2).* A kérést adó rendszer a kapott információt bejegyzi az ARP táblázatába.

Abban az esetben, ha a kért IP cím nincs a közös Ethernet hálózatba kapcsolt hosztok között, akkor a külvilág felé kapcsolatot biztosító átjáróban (routerben) lévő Ethernet kártyacímet felhasználva, oda kell küldeni az adott keretet.

A fentiekből nyilvánvaló, hogy az ARP-kéréseket tartalmazó kereteket üzenetszórás formájában kell a hálózatra kiadni. A kérés megfogalmazásához a csupa egyes bitből álló **FF:FF:FF:FF:FF:FF** Ethernet címet használják. Megállapodás szerint az Ethernet alapú hálózatok minden gépe figyeli az ilyen címre küldött kereteket. Ez azt jelenti, hogy az ARP kérést is látja mindegyikük. Minden egyes gép ellenőrzi, hogy a kérés rá vonatkozik-e. Ha igen, akkor választ küld. Ha nem, akkor egyszerűen nem veszi figyelembe. **Windows alatt ezt a működést az arp parancs valósítja meg**, amiről még a hasznos programok között írunk.

**Azért, hogy a dolgok tisztuljanak, foglaljuk össze mégegyszer a lényeget!**

Az információ datagramban terjed. A datagram (csomag) az üzenetben elküldött adatok összessége. Minden datagram a hálózatban egyedi módon terjed. Ezen csomagok továbbítására két protokoll, a TCP és az IP szolgál.

A TCP (Transmission Control Protocol) végzi az üzenetek datagramokra darabolását, míg a másik oldalon az összerakást. Kezeli az esetleges elvesző csomagok újrakérését és a sorrendváltozást. Az IP (Internet Prootocol) az egyedi datagramok továbbításáért felelős.

Pédául ha egy adathalmazt akarunk a hálózaton átvinni:

**xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx**

a TCP ezt datagramokká darabolja:

**xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx**

a TCP minden datagram elejére egy fejlécet rak (T=FEJ(TCP)) ami tartalmazza a forrás és a célprocessz port címét a sorozatszámot, és az ellenőrző összeget.

**Txxx Txxx Txxx Txxx Txxx Txxx Txxx**

ezt adja tovább az IP-nek a cél Internet címével együtt. Az IP ebből és a hely Internet címből újabb fejlécet képez (I=FEJ(IP)) :

**ITxxx ITxxx ITxxx ITxxx ITxxx ITxxx ITxxx**

A hálózat elérési szint, (amely lényegében a fizikai és adatkapcsolati szint) különböző lehet — pl. soros vonal, X25, vagy Ethernet — keretekkel dolgozik. Az Ethernet saját fejlécét (a két ETHERNET címmel) és C ellenőrző összegét illeszti EIT(FEJ(F)):

**EITxxxC EITxxxC EITxxxC EITxxxC EITxxxC EITxxxC**

Az ilyen módon „burkolt" (encapsulated) keretek megérkezése után az egyes fejléceket leszedi a megfelelő protokoll. Az Ethernet interfész az Ethernet fejlécet és az Ethernet ellenőrzőösszeget szedi le. Ezek után ellenőrzi a protokollra utaló típuskódot. Ha az IP-re mutat, akkor a datagramot átadja az IP-nek, amely a protokoll mező tartalmát megvizsgálja. Itt általában azt találja, hogy TCP, ezért a datagramot a TCP-nek adja át. A TCP a Sorszám mező tartalma és egyéb információk alapján állítja össze az eredeti állományt.

## 7.5.2 DHCP protokoll (RFC 2131/1997)

A TCP/IP alapú hálózatokban az egyik legfontosabb jellemző, a kommunikáló egység (a számítógép) IP címe. Ezt a címet két módon lehet megkapni:

- **statikus IP cím** - számítógép konfigurálásakor fixen rendeljük a számítógéphez,
- **dinamikus IP cím** - a bekapcsolás után egy adott címtartományból kerül kiosztásra, és egy, valamelyik hálózatba kapcsolt számítógépen futó program, a **DHCP szerver** szolgáltatja. A szerver konfigurálásakor adható meg a kioszthatő IP címtartomány.

**A dinamikus hoszt konfiguráló protokoll (=DHCP - Dynamic Host Configuration Protocol) alapján történő címkiosztás** egy bérleti (lease) rendszer szerint történik, azaz a címet kérőnek, időnként meg kell újítani a címfoglalását, különben az törlődik. Az a számítógép, amelyik kezdetben még nem rendelkezik IP címmel, egy mindenkinek szóló (broadcast üzenetet küld a hálózaton (az üzenet neve: *DHCPDISCOVER),* ezzel keresi a hálózatban lévő DHCP kiszolgáló(ka)t (szerver(eke)t). Az üzenetben a küldő IP címeként 0.0.0.0, (ez jelzi, hogy nincs IP címe) míg a címzett IP címe helyén a mindenkinek szóló broadcast 255.255.255.255 cím szerepel. A kérést küldő állomás az üzenetben elküldi a saját MAC (Ethernet kártya) címét is.

A szórt üzenetet megkapja valamelyik DHCP kiszolgáló, és a nyilvántartott szabad IP címei közül felkínál egyet, amit egy válaszüzenetben (*DHCPOFFER)* küld vissza a kérő felé (hiszen az Ethernet kártyacímét elküldte) a következő tartalommal:

- a DHCP kiszolgáló IP címe,
- a címzett IP címeként még a 255.255.255.255 szerepel (broadcast)
- az ügyfélnek felajánlott IP cím és a hálózati maszk,
- Az ügyfél MAC címe (ez alapján a szórt üzenetet csak az a számítógép veszi figyelembe, amelyiknek ez a hardver címe)
- Az időtartam, amennyi időre a kérő birtokolhatja (bérli) majd az IP címet (a bérleti idő általában 3 nap)

Ha *DHCPDISCOVER* üzenetre nem érkezik ajánlat (pl. nincs DHCP szerver), a kérő többször próbálkozik (9, 13, 16 mp-enként), majd utána sikertelenség esetén hibaüzenetet küld a felhasználónak. Utána 5 percenként ismét próbálkozik IP címhez jutással.

Az ügyfél elfogadja a részére felajánlott IP címet, és egy broadcast üzenetben a felajánlott címet lefoglaltatja a DHCP kiszolgálóval.

A DHCP kiszolgáló lefoglalja a kért IP címet, és erről egy nyugtázó broadcast üzenetet küld, amiben a további konfiogurációs beállításokat (útválasztó IP címe, a tartomány neve, DNS kiszolgáló IP címe, WINS kiszolgáló IP címe) is átadja a kérő kliensnek, persze az IP cím küldése előtt a DHCP kiszolgáló megbizonyosodik arról, hogy nincs a hálózatban használva ez az IP cím.

A *DHCPACK* üzenetet a kiszolgáló még mindig üzenetszórással küldi (hiszen csak most küldi a kérőnek az IP címet), de mi-

DHCP (Dynamic Host Configuration Protocol)

Működési vázlata:

1. DHCP kérdés: Ki tud adni egy IP címet?
2. A kérdés keretét üzenetszórásos küldéssel az alhálózat valamennyi csomópontja megkapja.
3. A DHCP szerverek feldolgozzák a kérdést: Ha a kezelt címtartományukban még van szabad IP cím, akkor azzal megválaszolják a DHCP kérdést.
4. A kliens a hozzá érkező DHCP válaszokból választ egyet, s visszajelzi a választását a megfelelő DHCP szervernek.
5. A DHCP szerver „könyveli" a címválasztást (foglalt lett a cím), s a könyvelésről megerősítést küld a kliensnek.

vel az üzenetben szerepel a kérelmező IP címe, ezért tudja, hogy az üzenet neki szól.

Amíg a bérlet fennáll, a bérlő nem kap minden gépindítás után IP címet, a DHCP kiszolgálótól, mert miután már kapott egy IP címet, indításkor ahhoz a kiszolgálóhoz fordul, akitől kapta az IP címét, és mintegy „megújítja a bérletét".

Ez a típusú címkiosztás nem megfelelő, ha egy számítógépnek mindig azonos IP címmel kellene rendelkeznie (mert pl. WEB-szerver fut rajta).

A DHCP szerverrel végzett címkiosztást azonban lehet úgy is konfigurálni, hogy egy számítógép mindig ugyanazt az IP címet kapja meg. Ehhez előtte a DHCP szerver által kezelt folytonos címtartományt két részre kell osztani: egyik részt kapják a statikus címmel felruházandó felhasználók, míg a másik címtartományt a dinamikusan konfigurált gépek. Ezek után két megoldás lehetséges:

- A statikus IP cím gépen történő begépeléssel történő megadása. Ilyenkor, a címeket az előzetesen a szerveren beállított statikus címtartományból választjuk ki.
- Rögzített IP címek adása a DHCP ügyfélnek a DHCP protokoll felhasználásával, vagy más néven címfoglalás. Ilyenkor a DHCP szervert konfiguráló program egy olyan táblázat megadását kéri, amelyben a gépeknek kiosztandó IP címek mellett szerepeltetjük az adott gép Ethernet kártyacímét az összerendelés megvalósítása érdekében.

A mindig állandó összerendelés használata azért előnyös, mert mindig meghatározható egy adott esemény bekövetkezése esetén, hogy melyik IP című gép okozta. Dinamikus IP címek használata esetén, ez csak kiosztást naplózó fájl vizsgálatával lenne kideríthető.

## 7.5.3 Tűzfal fogalma

A **tűzfalak (Firewall) olyan hálózati eszközök, melyek a rajtuk áthaladó forgalmat bizonyos általunk megadott szabályok alapján - elsődlegesen biztonsági okok miatt - szűrik.** Működésmódjukat illetően, mint átjárók működnek, de a továbbítást az IP csomag címétől is függővé teszik.

Általánosságban elmondható, hogy a hagyományos tűzfalak egy-egy IP csomagról pusztán az abban levő információ alapján döntik el, hogy az továbbítható-e vagy sem, illetve milyen egyéb intézkedéseket kell tenni. Azt is mondhatjuk, hogy ezek az eszközök a hálózati rétegben működnek.

## 7.5.4 NAT és Internet megosztás

Az internet eléréshez IP címre van szükség. Amennyiben többen szeretnének az internetre kapcsolódni, és csak egy vagy kevés IP cím áll rendelkezésre, akkor a problémát a címfordítás segítségével oldhatjuk meg. A címfordítás során (angol neve: NAT-Network Address Transforming) az IP csomagban lévő forrás-vagy célcímet valamint a portcímeket kicseréli egy másik IP címre és portcímre. Több megoldást használnak:

**Dinamikus címfordítás:** Ezt a belső hálózatról kifelé küldött csomagok esetén használjuk. Mikor dinamikus címfordítást végzünk, akkor a címfordító eszköz egyetlen kifelé érvényes IP címmel látja el a belső hálózat valamelyik gépéről érkező csomagot

úgy, hogy egyszerűen kicseréli a forrás IP címet (szükség esetén a forrás portot is), a csomagokban. Így természetesen a válaszok is erre a megváltozott címre fognak érkezni. Nyilvánvaló, hogy a címfordító hálózati eszköznek meg kell oldania azt is, hogy a csomagok visszajussanak a küldő géphez, vagyis a kapcsolat idejére meg kell jegyeznie, hogy milyen címet (és portot) mivel cserélt ki, és hová irányult az adott kapcsolat.

**Statikus célcím fordítás:** Mindig az adott címre érkező csomagokban levő célcímet cseréli ki, egy előre meghatározott címre. Mégpedig úgy, hogy a DNS-ben megadunk egy nyilvános IP címet a gépnek, de az erre a címre érkező kérések cél IP címét kicseréljük a gép belső, privát címére.

**Statikus forráscím fordítás:** Az adott belső címről érkező csomagok forrás IP címeit kcseréljük az előre meghatározott külső IP címre.

Például egy cég egyetlen Internet csatlakozásal rendelkezik. Legyen ennek a címe amit az Internet szolgáltatótól kapott: 152.66.213.80. Viszont 150 számítógéppel rendelkezünk, melyek mindegyikén Internet elérést akarunk lehetővé tenni. Ekkor 2 hálózati interfészt és egy NAT-ot lehetővé tevő szoftvert kell telepítenünk a közvetlen Internet kapcsolattal rendelkező eszközünkre (pl. egy UNIX/WINDOWS alapú PC).

Az egyik hálózati kártya megkapja az Internet eléréshez szükséges hálózati paramétereket, viszont a másik mint átjáró egy belső hálózati címet, a 192.168.0.1-es címet kapja. A belső hálózatban lévő gépek a 192.168.0.2-151 címet kapják, és átjárójuk pedig a 192.168.0.1 cím lesz. A belső hálózatban lévő számítógépeket nem lehet látni és elérni, ugyanis fizikai kapcsolat csak a NAT szerver és az Internet között létezik.

A NAT segítségével két feladatot is meg tudunk oldani: egyrészt lehetővé válik egy IP címen keresztül több gép Internet elérésnek a megvalósítása. Másrészt a hálózatot védhetjük a befelé jövő támadásoktól, mert internet kapcsolatot csak kifelé menő irányból tudunk kezdeményezni, hiszen minden gép egy IP cím mögött látszik.

Ma egy switch már tűzfal, router, NAT feladatokat is megvalósít!

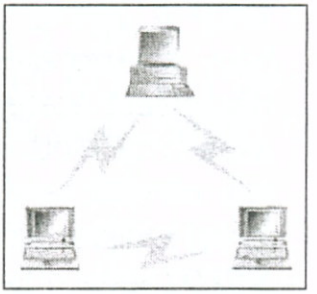
## 7.6 Vezeték nélküli (rádiós) hálózatok

Kétségtelen, hogy a számítógép hálózatok egyik legdinamikusabban fejlődő ága a vezeték nélküli hálózatok WLAN-ok (Wireless LAN-ok) robbanásszerű elterjedése. A rádiós csatornák nemcsak a hálózati összeköttetések kialakításában, hanem a mobiltelefon hálózatokon történő információtovábbításban is szerepet játszanak.

### 7.6.1 Topológiák

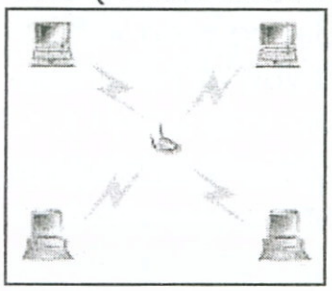
A rádiós hálózatok üzenetszórással kommunikálnak egymással. Alapvetően két toplógiát használnak

#### *AD-HOC üzemmód*



IBSS (Independent Basic Service Set – Független alap szolgáltatáskészlet) vagy ad-hoc mód teljes topológiát jelent, amelyben a résztvevő hostok mindegyike rendelkezik WLAN csatolóval, és közvetlenül kapcsolódhatnak ahhoz a hosthoz, amivel kommunikálni akarnak. Tehát ezt az üzemmódot akkor használjuk, ha csak vezeték nélküli hálózati kártyákkal szeretnénk egymás között kommunikálni.

### *Infrastruktúra üzemmód*



BSS ( Basic Service Set – Alap szolgáltatáskészlet ) vagy infrastruktúra mód esetén, legalább egy bázisállomás (Access Point - más néven hozzáférési pont, rövidítve AP) van a vezeték nélküli hálózatban, mely a hostokat egymáshoz és általában a vezetékes hálózathoz kapcsolja. Ha a hostok egymással akarnak kommunikálni, az is az Access Pointon keresztül történik. A bázisállomások újabb verziói négyféle üzemmódban képesek üzemelni:

**Access Point mód:** Ez a már említett infrastruktúra üzemmód. Klasszikus Master-Slave elven működik. Egy AP köré épülő vezeték nélküli hálózat, ahol az AP hálózati hídként üzemel.

**Access Point kliens mód:** Ezt az üzemmódot akkor használjuk, ha van egy fő vezetékes hálózatunk, és azt vezeték nélkül több kisebb vezetékes hálózattal kívánjuk összekötni. Ilyenkor csak AP-ra van szükség. Ilyenkor a fő vezetékes hálózatunkra kötött AP-t normál AP üzemmódba konfigurálva, míg a kisebb hálózatainkra kötött eszközöket AP kliens módba konfigurálva működtetjük hálózatunkat.

**Wireless Bridge funkció** Ez az üzemmód arra az esetre javasolt, ha két vezetékes hálózatot próbálunk vezeték nélkül áthidalni. Ilyenkor a két összekapcsolt AP úgy üzemel, mint egy vezeték nélküli híd (Bridge). A bázisállomásaink itt AD-HOC -nak megfelelő módban működnek. Kizárólag csak két eszköz kapcsolható össze ezzel az üzemmóddal.

**Wireless Bridge Point To MultiPoint funkció** Ez a topológia gyakorlatilag megfeleltethető az AP kliens módnak. Az elsődleges (Master) AP-t Wireless Bridge módba állítjuk, míg a hozzá kapcsolódó bázisállomásokat Wireless Bridge to MultiPoint módba állítva, vezeték nélküli hidat biztosítanak kettőnél több hálózat között. Azonban a gyártók figyelmezetése alapján háromnál több eszközt ne kössünk össze ezzel az üzemmóddal.

## *7.6.2 Vezeték nélküli hálózatok biztonsága*

A vezeték nélküli hálózatok biztonsága a vezetékes hálózatoknál is jelentősebb kérdés, hisz ebben az esetben már akkor is rákapcsolódhat valaki a hálózatra, ha az épület előtt megáll. Ekkor ugyanis egy kompatibilis hálózati kártyával a teljes hálózati forgalmat lehallgathatja. Ennek elkerülésére a wireless hálózatok nyújtanak néhány biztonsági megoldást.

- ***WEP*** A (Wires Equivalency Protocol – Vezetékessel megegyező protokoll) célja, hogy a vezetékes hálózatok biztonságának megfelelő szintű védelmet biztosítson. Használata esetén minden eszköz 40 vagy 128 bites RC4 titkosítást végez a hálózati adatátvitel előtt. Azonban az átvitel lehallgatásával a titkosítás visszafejthető, ezért a már elindult tendencia szerint, a WPA (Wi-Fi Protected Access – Wi-Fi védett hozzáférés) elnevezésű kódolás veszi át a helyét az eszközökben.
- ***SSID*** (Service Set ID – szolgáltatás azonosító) hasonló egy munkacsoport nevéhez, amelyet be kell állítani a kapcsolathoz, így elkerülhetőek a hálózatra való véletlen rácsatlakozások.

- ***MAC*** MAC cím szűrés segítségével meghatározhatók, hogy mely hálózati kártyákkal kerülhetnek kapcsolatba az AP-ok, így az illetéktelen behatolás megnehezíthető.

Természetesen az egyre újabb szabványok, mint a IEEE 802.11i már erőteljesebb biztonsági megoldásokkal rendelkezik. Ebben a szabványban megtalálható az AES protokoll, és a WPA fejlettebb felhasználó azonosítást és titkosítást alkalmaz a TKIP révén. **A 9. fejezetet teljes egészében a biztonsági kérdéseknek szenteltük.**

## 7.7 Hasznos programok

A következőkben néhány, a hálózatok vizsgálatánál hasznos programot mutatunk be, amelyek a ma legjobban elterjedt **Windows XP** operációs rendszer alatt futnak. Természetesen a programoknak Linux alatt is megvannak a hasonló párjai. Mivel az XP operációs rendszerben lévő magyar nyelvű segítség igen részletes, ezért a programokat csak röviden mutatjuk be.

Használatuk előtt érdemes a hozzájuk tartozó részletes leírást is elolvasni, amiben a szakmai fogalmak, és kifejezések megértése már többnyire nem okoz gondot, mert hiszen a könyvben ezt már megtanultuk...

Javasolt, hogy a parancsok átnézése közben próbáljuk ki a parancsokat, amihez először a START menü Futtatás... ablakában a CMD parancs begépelésével az operációs rendszer szöveges üzemmódjába kerülünk.

### ARP

Az IP címek – MAC címek összerendelését nyilvántartó Address Resolution Protocol (ARP) az összerendeléseket tároló gyorsítótár bejegyzéseit jeleníti meg és módosítja. Ez a gyorsítótár, egy vagy több olyan táblázatot tartalmaz, amelyek IP-címek és azok feloldott Ethernet fizikai címeinek (MAC) tárolására szolgálnak. A számítógépre telepített minden egyes Ethernet hálózati csatolóhoz külön táblázat tartozik. Paraméterek nélkül az **arp** parancs a súgót jeleníti meg. *(7-17 ábra)*

```
C:\Documents and Settings>arp -a

Kapcsolat: 192.168.1.2 --- 0x2
  Internetcím          Fizikai cím           Típus
  192.168.1.254        00-e0-98-c8-a8-d3     dinamikus
```

*7-17. ábra Arp parancs*

```
>arp /h      - megjelenik a help
>arp /a      - megjeleníti az összes bejegyzést
>arp /s 172.16.1.7 00-60-97-33-90-a3
```

A hozzárendelés lehet statikus (állandó), vagy dinamikus (ideiglenes). Az `arp/s` statikus beállítás fixen marad, a dinamikus 10 perc múlva törlődik, ha nincs közben rá hivatkozás. (a paramétereknél használhatjuk a / és – jeleket is.

### Ipconfig

Megjeleníti a TCP/IP-hálózat összes aktuális konfigurációs értékét, és frissíti a DHCP- és DNS-beállításokat. Paraméter nélkül használva a parancs segítségével meg-

jeleníthető az összes adapter IP-címe, alhálózati maszkja és alapértelmezett átjárója.

```
>ipconfig /h    - megjelenik a help
>ipconfig /all - részletes információ megjelenítése
```

**A gépben lévő DNS címfeloldó gyorsítótár kezelése is ezzel a paranccsal történik.** Ideiglenesen - esetleg ismételt felhasználásra – ide kerülnek a DNS szervertől kapott címfeloldási információk.

**`/flushdns`** DNS-ügyfél névhozzárendelő gyorsítótárának ürítése és alaphelyzetbe állítása. A DNS hibaelhárítása során ezzel az eljárással törölheti a negatív (hibás) és a dinamikusan hozzáadott egyéb bejegyzéseket a gyorsítótárból.

**`/displaydns`** Megjeleníti a DNS-ügyfél névhozzárendelési gyorsítótárát, amely tartalmazza a helyi Hosts fájlból előzetesen betöltött erőforrásrekordokat és az új, névkeresések feloldásából származó erőforrásrekordokat. Ezt az információt a DNS-ügyfélszolgáltatás a gyakran lekérdezett nevek gyors feloldásához használja, mielőtt lekérdezné a beállított DNS-kiszolgálókat.

**`/registerdns`** Kezdeményezi a számítógépen beállított DNS-nevek és IP-címek dinamikus manuális regisztrációját. A paraméter a sikertelen DNS-névregisztráció vagy az ügyfél és a DNS-kiszolgáló közötti dinamikus frissítéssel kapcsolatos hiba javításához használható az ügyfélszámítógép újraindítása nélkül. A TCP/IP protokoll speciális tulajdonságainak DNS-beállításai határozzák meg, hogy a DNS-kiszolgáló mely neveket regisztrálja.

## Ping

Az Internet Control Message Protocol (ICMP) visszhangkérő (ECHO) üzeneteket küldve, ellenőrzi az IP-szintű kapcsolatot egy másik TCP/IP protokollos számítógéppel. A megfelelő visszhangválasz üzenetek fogadását megjeleníti az oda-vissza út idejével együtt. A ping a kapcsolat, az elérhetőség és a névhozzárendelés hibaelhárításához használható elsődleges TCP/IP parancs. Paraméterek nélkül a ping a súgót jeleníti meg. Vigyázat! A tüzfal beállításai általában lehetővé teszik, hogy az ICMP kéréseket (ilyen a PING is) blokkoljuk. Ilyen gépre kiadva a Ping parancsot a kérés eredménytelen lesz. (Mintha azon az IP címen gép nem lenne.)

```
C:\Documents and Settings\drk>ping www.gdf-ri.hu /l 1000

opala.gdf-ri.hu [193.224.141.245] pingelése 1000 bájt méretű adatokkal

Válasz 193.224.141.245: bájt=1000 idő=96 ms TTL=123
Válasz 193.224.141.245: bájt=1000 idő=93 ms TTL=123
Válasz 193.224.141.245: bájt=1000 idő=96 ms TTL=123
Válasz 193.224.141.245: bájt=1000 idő=96 ms TTL=123

193.224.141.245 ping-statisztikája:
    Csomagok: küldött = 4, fogadott = 4, elveszett = 0 (0% veszteség),
Oda-vissza út ideje közelítőlegesen, milliszekundumban:
    minimum = 93ms, maximum = 96ms, átlag = 95ms
```

*7-18. ábra Ping parancs használata*

## Tracert

ICMP Visszhangkérő (ECHO) üzenetek küldésével meghatározza az adott célállomáshoz vezető elérési utat. A célállomáshoz küldött sorozatüzenetek során az élettartam (Time to Live, TTL) mezők értéke apránként növekszik. A megjelenített elérési út valójában a forrásállomás és a célállomás közötti útvonalon található útválasztók forrásol-

dali kapcsolatainak listája. A forrásoldali kapcsolatok az elérési úton a forrásállomáshoz legközelebb lévő útválasztási kapcsolatok. Paraméterek nélkül a **tracert** parancs a súgót jeleníti meg.

## Pathping

Információt szolgáltat a hálózat okozta késésről és a hálózati veszteségről a forrás és a cél közti ugrások esetében. A Pathping parancs több visszhangkérő üzenetet küld egy bizonyos idő alatt a forrás és a cél között lévő összes útválasztónak, majd az eredményt, az egyes útválasztók által visszaküldött csomagok alapján számítja ki. Mivel a **pathping** parancs megmutatja az egyes útválasztóknál és csatolásoknál bekövetkező csomagvesztés mértékét, meghatározható, hogy mely útválasztóknak, illetve alhálózatoknak vannak hálózati problémái. A **pathping** a **tracert** paranccsal megegyezően azonosítja, mely útválasztók találhatók az elérési úton. Paraméterek nélkül a **pathping** parancs a súgót jeleníti meg.

## Netstat

Megjeleníti az aktív TCP-kapcsolatokat és a portokat, amelyek hívásaira a számítógép figyel, az Ethernet statisztikát, az IP-útválasztási táblát, az IPv4-statisztikát (az IP, ICMP, TCP és UDP protokollhoz) és az IPv6-statisztikát (az IPv6, ICMPv6, TCP over IPv6 és UDP over IPv6 protokollhoz). Paraméterek nélkül a **netstat** parancs az aktív TCP-kapcsolatokat jeleníti meg.

```
>netstat /h - megjelenik a help
>netstat /a - Megjeleníti az összes aktív TCP-kapcsolatot
és azokat a TCP és UDP portokat, amelyek hívására a szá-
mítógép figyel.
```

## Nslookup

A DNS-infrastruktúra hibakereséséhez használható adatokat jelenít meg. Az eszköz használata előtt meg kell ismerni a DNS működését.

## Route

Megjeleníti és módosítja a helyi IP útválasztási tábla bejegyzéseit. Paraméterek nélkül a **route** parancs a súgót jeleníti meg.

| Parancs | Leírás |
|---|---|
| Add | Útvonalat hozzáad. |
| Change | Meglévő útvonalat módosít. |
| Delete | Útvonalat, vagy útvonalakat töröl. |
| Print | Útvonalat, vagy útvonalakat nyomtat. |

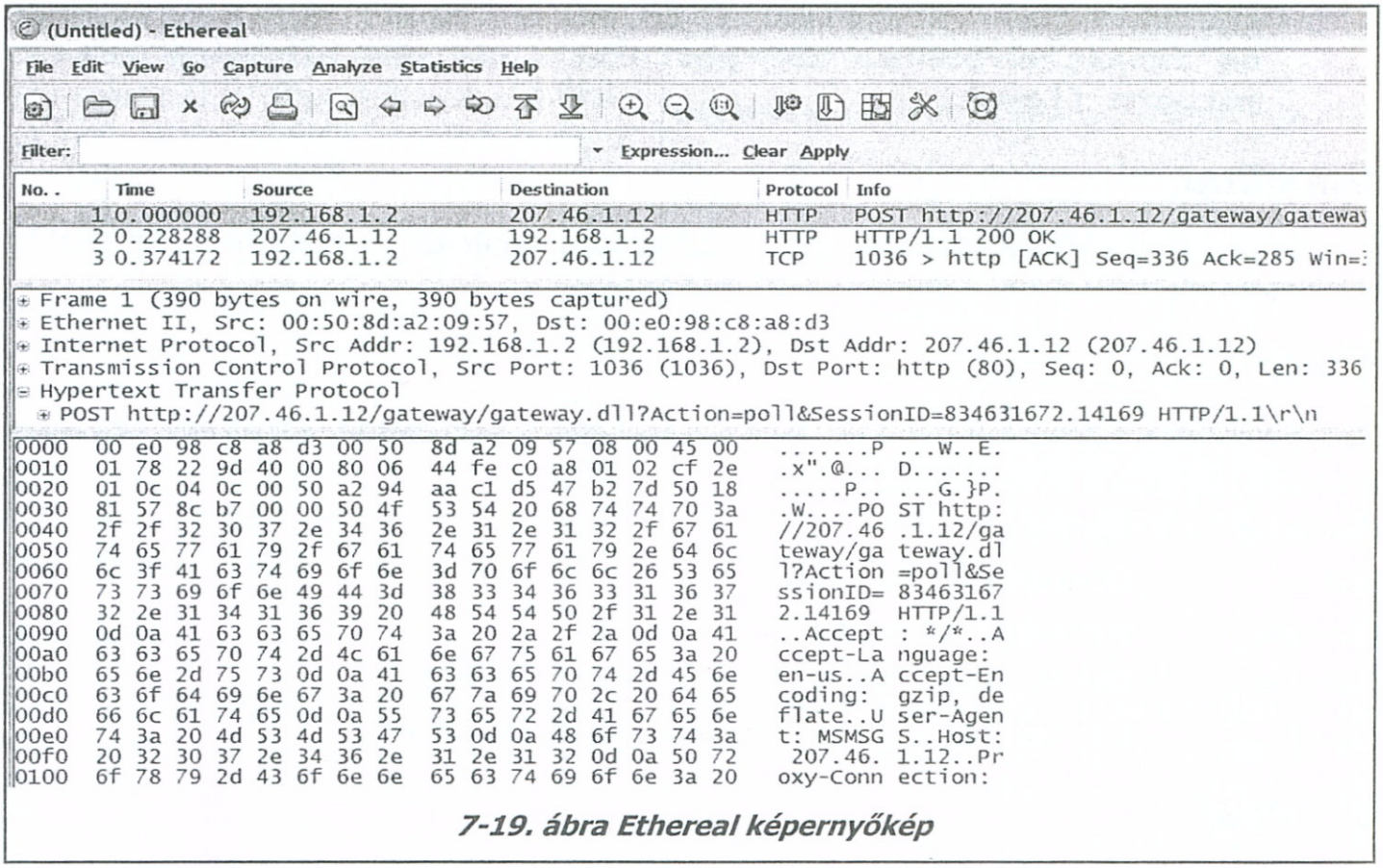
## Ethereal

Ez a szabadszoftver képes a hálózati forgalomban, a különféle protokollok által készí-

tett csomagok tartalmát megmutatni. Mivel ezek a csomagok legtöbbje Ethernet keretbe csomagolva utazik, ezért a hálózatban áramló – és nem csak a gépünk címére érkező kereteket úgy tudjuk a kártyánkkal fogadni, hogy a program átkapcsolja a hálózati kártyánkat „mindentfigyelő" – promiszkuitív üzemmódra. A programnak működéséhez szüksége van egy hálózati eszközmeghajtóra, ami a hálózat közvetlen elérését biztosítja számára. Ehhez le kell töltenünk – az ugyancsak ingyenes – **WinPCap** telepítőkészletét. Foglaljuk össze a program tulajdonságait:

- UNIX és Windows alatt is működik.
- A hálózaton megjelenő – nem csak a gépünknek szóló - kereteket veszi, és azok tartalmát nagyon részletesen megmutatja.
- A gyűjtött információkat képes fájlba elmenetei, és később, elemzésre visszatölteni.
- A sok csomagból, előre megadott szűrőfeltételek alapján képes csak a számunkra érdekeseket gyűjteni, illetve az elmentett csomagokból csak a megadott feltételeknek megfelelőket megkeresni.
- A könyebb láthatóság kedvéért az összetartozó csomagokat színekkel kódolva jeleníti meg.
- Az adatok alapján különféle statisztikákat készíthetünk, és ezeket grafikonként megjeleníthetjük.

Illusztrációként egy képernyőkép látható a programról a *7-19 ábrán.*




***7-19. ábra Ethereal képernyőkép***

# 7.8 Példák az alkalmazási rétegre: Internet szolgáltatások

A következőkben az alkalmazási réteg funkcióinak jobb megértésére röviden az Internet alkalmazási rétegét mutatjuk be.

## 7.8.1 Kapcsolódás az Internetre

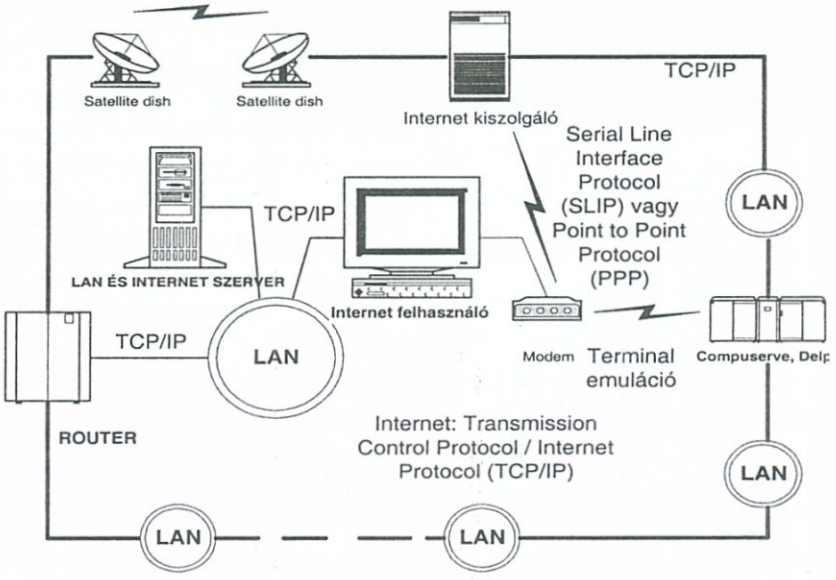
Az Internet logikai felépítést a *7-20. ábrán* láthatjuk. A legfontosabb része a nagy adatátviteli sebességű, általában optikai kábelekből, és műholdas kapcsolatokból álló gerinchálózat (bone), amely az ide kapcsolódó hálózatok információit szállítja.

A csomagokat routerek irányítják a különféle útvonalakon. Azonban kevés felhasználónak adatik meg a gerincre csatlakozás közeli lehetősége, általában a „főúttól messze", mellékutak mentén, vagy csak egy kis ösvény végén laknak.

Ez a hasonlat itt azért is találó, mert valóban tükrözi az adatátviteli sebesség csökkenését, amit például a telefonos kapcsolat jelenthet.

A felhasználó által elérhető adatátviteli sebességet a gerincig vezető alhálózatok adatátviteli sebessége közül a legkisebb fogja meghatározni.

A megfelelő hálózati teljesítmény eléréséhez csak nagyteljesítményű gépekkel lehet a gerincvonalakra csatlakozni.




*7-20. ábra Az Internet felépítése*

Az átlagos felhasználók ezért a helyi hálózati kapcsolataikat használhatják fel, míg egyéni felhasználók számára az Internet szolgáltatók (providerek) által üzemeltetett nagyteljesítményű gépeken keresztül való csatlakozás a megoldás. Ennek megfelelően a következő kapcsolódási megoldások lehetségesek:

- **Hálózati kapcsolódás.** Feltétel: a helyi hálózaton a TCP/IP protokoll használata. Egy routeren keresztül az Internetre küldött csomagok eljuthatnak a célokig.
- **SLIP/PPP kapcsolat**. Telefonvonalon keresztüli kapcsolódás. Ilyenkor egy modem és a telefonvonalon TCP/IP szerű kapcsolatot megvalósító SLIP/PPP (SLIP — Serial Line Interface Protocol, PPP — Point to Point Protocol) protokoll szükséges. Ma már a széles sávú internet ADSL vagy kábelmodemes kapcsolatot használ. Számítógépünk a vonal másik végén egy Internetre kapcsolódó kiszolgáló számítógépen keresztül egy IP címet hordozó hálózatra kapcsolt géppé válik.
- **On-line szolgáltatón keresztül** (terminál emulációval) az Internetre kap-

csolódó gépen fut az a program, amelyet a telefonvonalon keresztül a számítógépet terminálként használva kezelünk. Ma már nem használt.

### *Internet elérés hagyományos modem segítségével*

Az egyszerű felhasználók Internetre való kapcsolódásának legegyszerűbb módja: modemmel, telefonvonalon keresztül valósítható meg. Ez természetesen feltételez a másik oldalon is egy modemes kapcsolatot. Ez általában az Internet szolgáltató (provider), amely már nagy sávszélességű vonalon kapcsolódik az internetre. Mivel a kapcsolatban IP csomagok átvitele történik, ezeket a telefonvonalon átvitt keretekbe kell elhelyezni.

Két protokollt használnak az IP csomagok soros vonalon történő átvitelére:

- **(Serial Line Internet Protocol — SLIP)** de facto szabványnak tekinthető, a másik
- **(Point-to-Point Protocol — PPP)** de jure Internet szabvárny.

### Serial Line Internet Protocol — SLIP

Az RFC 1055 írja le a SLIP-et IP csomagok soros (aszinkron, fullduplex) vonalon történő átvitelére. Ez az ún. packet framing protokoll karakterek sorozatát definiálja, amely IP csomagokat visz soros vonalon át. Az alábbi vázlat a SLIP csomag formáját mutatja:

| *határoló (delimiter)* *0xC0* | *csomagadatok (packet data)* | *határoló (delimiter)* *0xC0* |
|---|---|---|

A SLIP a hexa 0xC0-t használja az Internet csomagok keretezésére (framing), ezt a szekvenciát helyettesíti szükség esetén (az adatsorban 0xC0 szerepel) a 0xDB és a 0xDC kétbájtos szekvenciával.

A SLIP nem tesz lehetővé címzést, csomagtípus- (packet type) azonosítást, hibaellenőrzést és -javítást, és az alap SLIP protokoll még tömörítést sem.

### Point-to-Point Protocol — PPP

A SLIP hiányosságainak kiküszöbölésére hozták létre a PPP protokollt. Aszinkron (8 adatbit, paritás nélkül) és bitorientált szinkron üzemmódban egyaránt használható, fullduplex kapcsolatot igényel.

A PPP módosított HDLC keretformát használ protokollcsomagok átvitelére, mindegyik keretben hibaellenőrző és -javító kóddal (HDLC Frame Check Sequence — FCS). A teljes PPP keret nyolc bájtot használ az egyes keretek becsomagolására, de lehetőséget ad ennek két bájtra történő rövidítésére is (nagy sebességű hardver esetén mód van a 32 bites korlátozások betartására), továbbá escape mechanizmust biztosít kontrolladatok (pl. XON/XOFF) átvitelére.

A teljes PPP csomag keretformája:

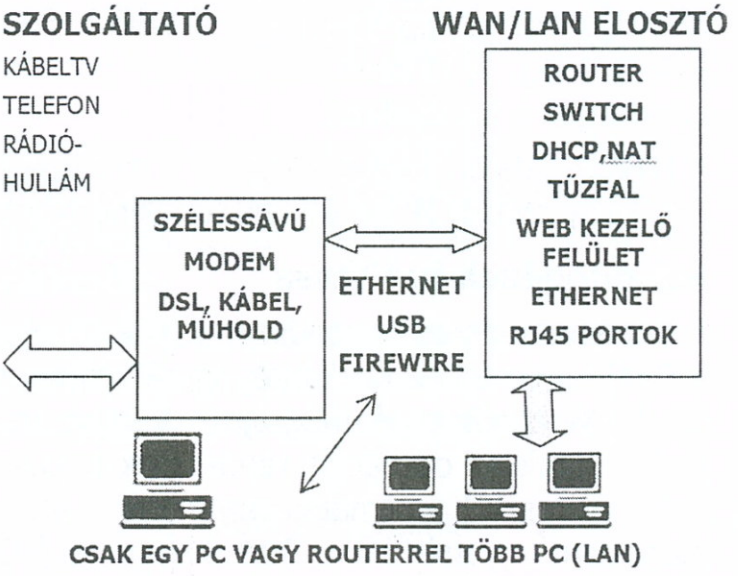
| *delimiter* *0x7E* | *cím* *0xFF* | *Ctrl* *0x03* | *protocol* *0xXXXX* | *adatcsomag* | *CS* | *delimiter* *0x7E* |
|---|---|---|---|---|---|---|

A PPP a datagramok becsomagolásának leírásán kívül tartalmaz egy ún. **Link Control Protocolt (LCP)** a pont-pont közti kapcsolatok felépítésére, konfigurálására, a kap-

csolat tesztelésére, valamint egy ún. **Network Control Protocol (NCP)** családot a különböző hálózati protokollok (szimultán) átvitelére, továbbá hibák és problémák kezelésére. Az LCP automatikusan kioszthatja és menedzselheti az IP címeket, egyezteti a két pont között a keretezési formát (nem szabványos keretezésre is lehetőséget ad), a csomagok hosszát, és számos egyéb szolgáltatást nyújthat. A PPP protokoll a kapcsolat azonosítására, biztonsági funkciókra, a forgalom és a hibák tesztelésére tartalmaz ajánlásokat.

### *Szélessávú internet kapcsolat ADSL vagy kábelmodemmel, illetve routerrel.*

Ahogy azt már leírtuk, a szélessávú Internet szolgáltatást a szolgáltató vagy a telefonvonalon vagy kábelTV hálózaton keresztül szolgáltatja. Amenyiben egy ilyen csatlakozásra több gépet szeretnénk rákapcsolni, a szolgáltató engedélyével a DSL/Kábel routerek felhasználásával tehetjük meg. Ezek a **broadband routerek** a szolgáltató felől érkező WAN hálózati végpontot kapcsolják össze egy szabványos, általában 10/100-as Ethernet helyi hálózattal.




*7-21. ábra Szélessávú Internet hozzáférés*

#### WAN interface

A WAN (Wide Area Network) végpont rendszerint egy telefonvonalon kapcsolódó ADSL modem, vagy egy kábeltévé hálózaton üzemelő kábelmodem. A végpont WAN oldala Ethernet, USB, vagy Firewire illesztéssel csatlakozik az egy, vagy több számítógép rákapcsolását lehetővé tevő útválasztóhoz.

A routerek képesek automatikusan bejelentkezni PPP over Ethernet (PPPoE) protokollon (főleg ADSL megoldásoknál jellemző), kapcsolatba lépni a szolgáltató DHCP szerverével (jellemzően kábelTV), de statikusan is beállíthatjuk a szükséges beállításokat. (IP cím, útválasztó IP címe, DNS címe.)

A PPPoE protokoll a pont-pont jellegű, már bemutatott PPP kapcsolatot a nagy sebességű Ethernet csatlakozással valósítja meg.

Egyes szolgáltatók csak az általuk biztosított Ethernet kártyán keresztül engedélyezik az internetezést, a kártyát annak egyedi MAC-címe alapján azonosítják. A speciális beállítások között tetszőleges MAC-címet definiálhatunk a routerünk számára.

#### LAN interfész

A helyi hálózat irányában a routeren egy vagy több RJ-45 típusú portcsatlakozó található. Gyakoriak a 4 portos, 10/100 Mb/sec-os switch-el összekombinált eszközök, léteznek USB porttal is felszerelt termékek, és kaphatók Access Pointként működő routerek is (vezeték nélküli Ethernethez). A switch-el rendelkezők eszközöknél csak néhány UTP kábelre és megfelelő hálózati kártyákra (amelyek már az számítógép alaplapjára vannak integrálva) van szükségünk ahhoz, hogy felépítsük a saját helyi

hálózatunkat. Az USB porttal ellátott routerekre 1 db számítógépet USB porton is csatlakoztathatunk, ehhez elegendő egy USB kábel.

### Beállítás

Mivel ezek minden gyártónál hasonlóak, ezért csak általánosan ismertetjük. A beállításokat webes felület könnyíti meg, aminek eléréshez elegendő egy web böngészőbe beírni az eszköz IP címét, majd megadni a jelszót. A **SETUP** oldalon beállíthatjuk az router IP címét, az alhálózati maszkot, és a WAN felé történő bejelentkezés jellemzőit (statikus IP, DHCP, PPPoE, PPTP).

A **STATUS** oldalon megnézhetjük, hogy minden rendben történt, megtudhatjuk a külső IP címünket, a DNS szerverek címét, és a LAN beállításait. Itt értesülünk arról, hogy sikerült-e kapcsolódni az Internet szolgáltatóhoz, az esetleges hibaüzenet is itt jelenik meg. A Connect/Disconnect gombbal értelemszerűen felépíthetjük, vagy lebonthatjuk a kapcsolatot.

A **DHCP** oldalon engedélyezhetjük az eszközbe épített DHCP szervert, és megnézhetjük, hogy milyen IP címek kerültek kiosztásra a számítógépeknek.

### Számítógépek beállítása

A Windows operációs rendszerek TCP/IP protokoll használata esetén alapbeállításként a DHCP szerverhez fordulnak, így általában semmilyen beállítást nem kell végrehajtani, WinXP rendszernél még újraindítás se szükséges megfelelő konfiguráció esetén. Az újabb Windows operációs rendszerek a routert UPnP (Universal Plug and Play) protokollon keresztül automatikusan internet átjáróként felismerik. Az engedélyezés/letiltás menüpont egyenértékű a Status lapon lévő Connect/Disconnect gombbal.

Amennyiben valamilyen szervert kívánunk üzemeltetni, akkor a router beállításaival összhangban válasszunk egy fix IP címét, és állítsuk be a router Advanced szekciójában a megfelelő portokra érkező kérések továbbítását a szerver felé (Forwarding).

### Tűzfal, NAT, biztonság

A router beépített tűzfallal és hálózati címfordítóval (NAT = Network Adress Translator) rendelkezik, mely hatékony védelmet nyújt az Internetről történő behatolások ellen. Alapbeállításként a router semmilyen kívülről érkező TCP/UDP csatlakozási kísérletre nem válaszol, a ping parancsokat is beleértve.

Természetesen az alkalmazások szintjén jelentkező internetes veszélyektől (férgek, vírusok) nem tud megvédeni a tűzfal, de azok kártékonyságát sok esetben képes csökkenteni. Beállítható, hogy csak konkrét vírusellenőrző futtatása mellett engedélyezze az internet elérést a számítógépek számára. Megfelelő naplózó program segítségével folyamatosan nyomon követhetjük a hálózatunkon zajló forgalmat, észlelhetjük a külső behatolási kísérleteket, vagy egyes számítógépeken megtelepedett vírusok aktivitását. (A hálózati biztonság a 9. fejezet témája.)

## *7.8.2 Internet szolgáltatások*

A felhasználót általában az elérhető szolgáltatások érdeklik. A szolgáltatások alapvetően két csoportba sorolhatók: közvetlen hálózati kapcsolatot nem igénylő (off-line)

szolgáltatás, — ilyen a levelezés — és a folyamatos kapcsolatot igénylő (on-line) szolgáltatások. Ennek megfelelően is több megoldás lehetséges:

A legegyszerűbb szolgáltatás a **levelezés**: ez lényegében nem igényel hálózati kapcsolatot. Általában egy Internet szolgáltató számítógépén elhelyezett postaládát használunk: ennek tartalmát modemes kapcsolaton keresztül kezelhetjük;

**UUCP- (Unix to Unix Copy)** protokoll segítségével, Unix-ot futtató géppel, modemen keresztül kapcsolódunk a szolgáltató gépére és a leveleket egy menetben fel-, illetve letöltjük;

Ha megvan a lehetőség rá, **beköthetjük helyi hálózatunkat az Internetbe egy szolgáltatón keresztül.** TCP/IP-t és az Internet segédprogramokat telepítünk a hálózaton, majd a LAN-t valamilyen hálózati kapcsolattal (hagyományos- ADSL- kábelmodem, X.25, ISDN-, nagy sebességű bérelt telefonvonal) rácsatlakoztatjuk az Internetre.

Az Interneten, mivel eltérő felépítésű hálózatokat kötnek össze, szükséges az Interneten folyó alkalmazási szintű kommunikáció közös szabványainak kidolgozása, amelyeket szintén az RFC (Request for Comments) dokumentumok tartalmazzák. Az Internet lényegesebb alkalmazási protokolljai a következők:

- **SMTP Simple Mail Transfer Protocol** egy alkalmazási protokoll, amely a hálózati felhasználók egymással való kommunikációját teszi lehetővé. Leveleket tud küldeni és fogadni. A **POP3** és **IMAP** protokollok leveleink fogadását, illetve, távolról való letöltését teszik lehetővé.
- **TELNET** Terminál emuláció segítségével a saját gépét terminálnak használva egy távoli hosztra felhasználóként lehet bejelentkezni.
- **FTP File Transfer Protocol** A fájl átviteli eljárás a felhasználónak lehetővé teszi az általános könyvtár és fájlműveletek végrehajtását a saját gépe és egy távoli hoszt lemezegysége között. Pl.: fájlokat vihet át, törölhet, átnevezhet fájlokat.
- **GOPHER** Hierarchikusan felépített információban kereső protokoll
- **HTTP** HyperText Transport Protocol
- **NEWS** Hálózaton keresztüli hírszolgáltatást megvalósító protokoll (Network News Transport Protocol)

Ezek segítségével az Internet által jelenleg biztosított lényegesebb szolgáltatások ABC sorrendben:

| | |
|---|---|
| ARCHIE | SZOFTVERKERESŐ SZOLGÁLTATÁS |
| EJOURNALS | HÁLÓZATON KERESZTÜL TERJESZTETT ÚJSÁGOK |
| E-MAIL | ELEKTRONIKUS LEVELEZÉS |
| FINGER | HÁLÓZATI FELHASZNÁLÓK ADATSZOLGÁLTATÁSA |
| FTP | TÁVOLI GÉPEK KÖZÖTTI ÁLLOMÁNYCSERE |
| GOPHER | MENÜRENDSZERŰ ADATFORRÁS-TALLÓZÓ |
| IRC | TÖBBCSATORNÁS, TÖBBIRÁNYÚ PÁRBESZÉDES KAPCSOLAT |
| JUGHEAD | KORLÁTOZOTT KÉPESSÉGŰ GOPHER MENÜTALLÓZÓ |
| LISTSERVER | LEVELEZÉSI CSOPORTOK KISZOLGÁLÁSA |
| USENET | HÍRCSOPORTOK KISZOLGÁLÁSA |
| TALK | KÉTIRÁNYÚ PÁRBESZÉDES KAPCSOLAT |
| TELNET | TÁVOLI GÉPEKEN TÖRTÉNŐ MUNKA |

| | |
|---|---|
| VERONICA | VILÁGMÉRETŰ GOPHER MENÜTALLÓZÓ |
| WAIS | ADATBÁZIS ALAPÚ INFORMÁCIÓ SZOLGÁLTATÓ |
| WWW | HYPERTEXT ÉS MULTIMÉDIA ALAPÚ GOPHER |
| WHITE PAGES | HÁLÓZATI-FELHASZNÁLÓ KERESŐ SZOLGÁLTATÁS |
| WHOIS | HÁLÓZATI-FELHASZNÁLÓ KERESŐ SZOLGÁLTATÁS |
| ZINES | HÁLÓZATON KERESZTÜL TERJESZTETT MAGAZINOK |

A következőkben a fenti protokollok és szolgáltatások közül a legfontosabbakat mutatjuk be. A most következő részekben felhasználjuk a TCP/IP protokollal foglalkozó részben szereplő ismereteket.

### *E-mail (Electronic mail) (RFC821, 822, 2822)*

A legalapvetőbb szolgáltatás, a legelső, amit az Interneten használtak, az elektronikus levelezés. Egy levelezőprogram (mailer) segítségével szöveges állományt küldhetünk az Internet bármelyik felhasználójának

Az RFC 822 szabvány az ilyen leveleknek csak a fejlécét definiálta, magát a tartalom összeállítását a felhasználóra bízta.

A levél – a hagyományoshoz hasonlóan – egy, a címzettet, a feladót és néhány kiegészítő információt tartalmazó **fejléc**-ből, és a tartalmat (a levél szövegét tartalmazó) **levéltörzs**-ből áll.

A levelezés megvalósításához az kell, hogy minden levelezőnek egyedi címe legyen, és a címzés is szabványos legyen. Egy felhasználó Email címe általánosan a következőképpen épül fel:

**felhasználói_név @ gépnév.subdomain_név.domain_név.ország(intézmény)azonosító**

Általánosan fogalmazva a cím egy felhasználói névből (**username**) és egy cím (domén) részből áll, a kettő között a @ jel található. A cím rész a DNS-nél megismert domén struktúrát követi. Ez a "kukac" az angol "at" szót jelenti, vagyis arra utal, hogy ez a felhasználó HOL (melyik gépen) található meg.

Az elektronikus levelezés a legalapvetőbb a szolgáltatások között, és szerencsére van két nagyon kedvező tulajdonsága:

- Ha csak levelezni szeretnénk, a vonal adatátviteli sebességének nem kell nagynak lenni, mert egy átlagos levél néhány kilobájt hosszúságú, és ennek átvitele rövid idő alatt történik meg.
- Ez a szolgáltatás off-line jellegű: azaz a levél írása és olvasása és a tényleges elküldése időben szétválik. Nem kell folyamatos hálózati kapcsolatának lenni a levelezőnek, hogy a levélkézbesítés megtörténjen.

A levelek küldését és fogadását ténylegesen egy, folyamatos hálózati kapcsolattal rendelkező számítógépen futó program, a Mail-szerver (levelezés kiszolgáló) végzi. A felhasználók ténylegesen ennek a programnak küldik leveleiket, illetve ettől kapják meg a leveleket. Az elküldött és kapott leveleket ez a program tárolja, és a címek alapján végzi a hálózaton keresztüli kézbesítést.

**Lényeges megkülönböztetni a hálózati internet címeket a levélcímektől. A levelek címrésze határozza meg annak a gépnek az internet címét, amelyen a levelezés kiszolgáló program fut, és a címrész alapján a gépre küldött le-**

**veleket egy olyan lista segítségével kézbesíti, amely a gépen a levelezésbe bevont felhasználókat azonosítja.**

Levelezni valamilyen levelező programmal lehet. Mindegyik megvalósítja az alábbi funkciókat:

- Levél küldése közvetlenül, vagy egy listán szereplő címzetteknek (send),
- kapott levelek a levél témája (subject) szerinti listázása,
- válasz adott levélre (reply),
- levél továbbküldése (forward),
- levél tárolása különböző irattartókba (folderekbe),
- levél törlése (delete).

### POP3 és IMAP

A levelező rendszerek képesek leveleket küldeni, és fogadni. Felmerült az igény, hogy a levelek fogadása önmagában is lehetséges legyen. Vagyis távolról, az internet felhasználásával le tudjuk kérdezni, és el tudjuk olvasni a beérkezett, az SNMP protokollal továbbított leveleinket. Erre két megoldás is rendelkezésre áll: a POP3 és az IMAP.

**POP3** (Post Office Protocol 3) Postahivatal protokoll 3. verzió

Levelek letöltésére használt protokoll a felhasználó (MUA-Mail User Agent=Levelező ügynök) kezdeményezi a POP3 kapcsolatot a szerver felé, ahová a felhasználó levelei megérkeznek, és tárolódnak letöltésükig.

**IMAP** (Interactive Mail Access Protocoll) Interaktív levél hozzáférési protokoll

A POP3-hoz hasonlóan lehetővé teszi, hogy leveleinket a fenntartott postafiókunkban letöltés előtt megnézhessük, és távolról kezelhessük. Ellentétben a POP3-al, mely minden levelet teljes terjedelmében átküld a gépünkre, az IMAP először csak azok fejléceit küldi el. Lehetővé teszi, hogy a címeiket elolvassuk, a leveleket ott csoportosítsuk, és az egyes levelek letöltéséről külön dönthessünk. Postafiókunk tárhely-igénye és a letöltési idő természetesen megnő. Az IMAP használata tágabb lehetőségei ellenére is csak lassan terjed; nem minden szolgáltató támogatja a nagyobb társszükséglet miatt.

Ha egy levél érkezik, akkor általában nemcsak maga az üzenet jön meg, hanem egy pár soros kiegészítés is. Ez a legtöbb esetben, közvetlenül nem tartalmaz az átlagos felhasználó számára lényeges információt. Vegyünk egy példát:

A „**Received**": sorokból azoknak az állomásoknak a neveit, és a használt protokollokat tudhatjuk meg, amelyeken a levél keresztül ment. Minden levél esetében van legalább egy ilyen sor. Ha a levél nagyon "kavargott" a világban, akkor 10 fölé is mehet a sorok a száma. Egy levélnek általában 4-5 ilyen sora van.

A „**Date**:" a levél elküldésének dátuma.

A „**From**": sor azt a címet tartalmazza, ahonnan a levél jött.

A „**To**": a címzett email címét tartalmazza. Lehet, hogy egy levelet több helyre küldtek, ilyenkor vesszővel vannak elválasztva a címek.

„**Subject**": a levél tárgya

A „**Message-Id**": egy egyedi azonosító. Általában a levelek útjának követésére használják.

"**Reply-To**": ha a feladó nem szeretné, hogy arra a címre válaszoljunk, ahonnan a levél jött (ezt a Return-path: sor tartalmazza), akkor megad egy másikat.

A „**Mime-Version**" és a „**Content-Type**" sorok jelentése rögtön tárgyalásra kerül.

```
Received: from MAIL by KKMF_MSZI (Mercury 1.21); 8 Mar 96 08:28:19 GMT+1
Return-path: <drdani@mazsola.iit.uni-miskolc.hu>
Received: from gold.uni-miskolc.hu by novserv.obuda.kando.hu (Mercury 1.21); 8
Mar 96 08:28:13 GMT+1
Received: from [193.6.4.39] by gold.uni-miskolc.hu (AIX 3.2/UCB 5.64/4.03) id
AA27796; Fri, 8 Mar 1996 08:11:41 GMT
Received: by mazsola.iit.uni-miskolc.hu (SMI-8.6/SMI-SVR4) id IAA04556; Fri, 8
Mar 1996 08:15:15 +0100
Date: Fri, 8 Mar 1996 08:15:14 +0100 (MET)
From: Drotos <drdani@pigmy.iit.uni-miskolc.hu>
To: Konya Laszlo <KONYA@novserv.obuda.kando.hu>
Subject: Re: RS422/RS232 keres/kerdes
Message-Id: <Pine.LNX.3.91.960307202228.508A-100000@pigmy>
Reply-To:
Mime-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII
```

Fontos tudni, hogy Email-en keresztül közvetlenül csak 0-127-es kódú ASCII karakterek küldhetők át. Ha olyan karaktert küldünk, aminek a 8. bitje 1, azt a rendszer levágja, elvész. Így közvetlenül bináris fájlok átvitele nem lehetséges. Több módszer van annak megoldására, hogyan küldjünk át a hálózaton levelezés segítségével bináris fájlokat. A következőkben ezeket foglaljuk össze:

## UUENCODE/UUDECODE (rövidítve: UUE)

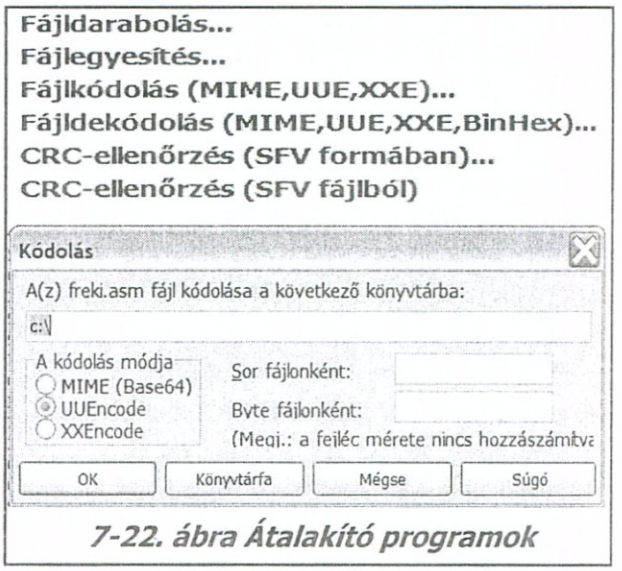
Az UUENCODE a fájlt alkotó bináris bájtsorozatot konvertál 7 bites szöveggé oly módon, hogy a fájl elejéről kezdve sorban vesz 3 db 8 bites bájtot, és azt szétbontja 4 db 6 bites darabra. Például, legyen 3 bájtunk hexadecimális formában: AFH, 01H, 65H.

Ezt bontsuk fel négy 6 bites csoportra:

```
AFH, 01H, 65H = 10101111 00000001
01100101
= 101011  110000  000101  100101
```

Ezek a 6 bites csoportok számként 0...63 tartományba tartoznak. Mivel a speciális ill. vezérlőkarakterek ASCII kódjai 0...31-ig terjednek, a normál szöveg kialakítása érdekében mind a négy így kapott bitcsoporthoz hozzáadunk 32-t (00100000).

```
101011  110000  000105  100101  =>
1001011 1010000 0100101 1000101 =
4BH 50H 25H 45H = KP%E
```




*7-22. ábra Átalakító programok*

Vagyis az AFH,01H,65H bájthármasból a **KP%E** karakternégyes lesz, ami már levélben elküldhető.

Az UUDECODE program az így keletkezett fájlt kódolja vissza a fogadó oldalon. A kódolt fájl természetesen hosszabb lesz, mint az eredeti, mert a módszer semmilyen más változtatást (pl. tömörítés) nem végez.

### Base64

24 bites csoportokat tördel fel 6 bites egységekbe, és minden ilyen egység egy legális ASCII karakterként továbbítódik. A 0-nak az "A", 1-nek a "B" felel meg, stb. Ez így folytatódik, majd következik 26 db kisbetű, 10 számjegy 0-tól 9-ig, és végül + és / a 62-nek és 63-nak felel meg.

A Total Commander fájl menüjében megtalálható mind a kódolásra, mind a dekódolásra alkalmas művelet. *(7-22. ábra)*

### MIME (Multi-purpose Internet Mail Extensions)

Egy másik megoldás esetén már lehetőség van levélben nem ASCII karakterek, képek, hangok küldésére is. Ezt az eljárást MIME-nek (Multi-purpose Internet Mail Extensions) nevezik. Amelyik levelezőprogram ismeri ezt, azzal írható, illetve olvasható akár magyar ékezeteket tartalmazó levél is. A megoldást legelőször az RFC 1341-ben definiálták, amely frissítését az RFC 1521-es tartalmazza. Az alapötlet az volt, hogy a MIME folytatja az RFC 822-ben definiált levélformátumot, de az üzenetek, levelek törzsének is létrehoz egy struktúrát, valamint definiálja a nem ASCII tartalmú (bináris) üzenetek kódolását elvégző eljárásokat is.

| Fejléc | Jelentés |
|---|---|
| MIME-Version: | azonosítja a MIME verzióját |
| Content-Description: | az üzenet tartalma |
| Content-Id: | egyedi azonsító |
| Content-Transfer-Encoding: | a törzs átviteli kódolásának formátuma |
| Content-Type: | az üzenet jellege |

Bármely levél, amely nem tartalmazza a MIME fejlécet, alapértelmezés szerint egy sima, angol nyelvű üzenetként dolgozódik fel. A MIME öt új üzenet fejlécet definiál (ezek láthatók a táblázatban). Ezek közül az első csak egy jelölés ahhoz, hogy a fogadó program felismerje, hogy milyen MIME verziójú üzenettel van dolga.

- **Content-Description:** ez a fejléc egy ASCII szöveg, amely megadja az üzenet tartalmát.
- **Content-Id:** a standard üzenet azonosítójának formátumával megegyező egyedi azonosító.

| Típus | Altípus | Leírás |
|---|---|---|
| ***Text*** | Plain | Formázatlan szöveg |
| | Richtext | Egyszerű formázási utasításokat tartalmaz |
| ***Image*** | Gif | GIF formátumú állókép |
| | Jpeg | JPEG formátumú állókép |
| ***Audio*** | Basic | hallható hang |
| ***Video*** | Mpeg | MPEG formátumú mozi |
| ***Application*** | Octet-stream | interpretálatlan byte sorozat |
| | Postscript | PostScript formátumú nyomtatható dokumen- |
| ***Message*** | Rfc822 | RFC 822 üzenet |
| | Partial | az üzenet részekre lett bontva az átvitel idejére |
| | External-body | magát az üzenetet a hálózatról kell letölteni |
| ***Multipart*** | Mixed | egymástól független részekből álló üzenet |
| | Alternative | ugyanaz az üzenet különböző formátumokban |
| | Parallel | párhuzamosan felhasznált részekből álló üzenet |
| | Digest | minden rész egy teljes RFC 822 üzenet, de egybe lett csomagolva |

- **Content-Transfer-Encoding**: azt mondja meg ez a fejléc, hogy milyen formában kódolták az üzenet törzsét az átvitelhez. Több lehetőség létezik:
  - Sima ASCII szöveg, a sorok nem hosszabbak 1000 karakternél.
  - A karakterek 8 biten tárolódnak (255 db karakter). Ez a kódolási séma már megsérti az eredeti Internet e-mail protokollt, de néhány területen használják az eredeti protokoll kiegészítésével. A standard, maximum 1000 karakter hosszú soroknak itt is meg kell felelni.
  - Bináris kódolást használó üzenetek. Ezek tetszőleges bináris állományok, amelyek nemcsak 8 bitet használnak, hanem az 1000 karakteres határnak sem felelnek meg. (pl. futtatható programok). Bináris adatokat a következő két kódolást felhasználva kell elküldeni
    - A == és = jelek azt fogják jelenteni, hogy az utolsó egység csak 8 vagy 16 bitet tartalmaz. A kocsi vissza és soremelés karakterek figyelmen kívül maradnak és szabadon beszúrhatók a sorok megfelelő rövidségének érdekében. Ezzel a módszerrel tetszőleges bináris szöveg biztonságosan elküldhető.
    - Azokra az üzenetekre, amelyek néhány kivétellel csak ASCII karaktereket tartalmaznak, nem hatékony a base64 kódolás. Helyette a **quoted-printable kódolást** alkalmazzák. Ez megegyezik a 7 bites ASCII-vel, azzal a különbséggel, hogy a 127 feletti karaktereket egy egyenlőségjel kódol, amit az eredeti karakter értéke követ két hexadecimális szám formájában.

Ha szükséges, akkor lehetőség van saját, felhasználó által definiált kódolás specifikálására is a Content-Transfer-Encoding: fejlécben,

- **Content-Type**: Ez az utolsó fejléc eléggé összetett, felépítését a táblázat mutatja, amely az egyes igényeknek megfelelően folyamatosan bővül. Lényegében az üzenet típusát ill. altípusát tartalmazza, amelyeket egy / jel választja el egymástól.

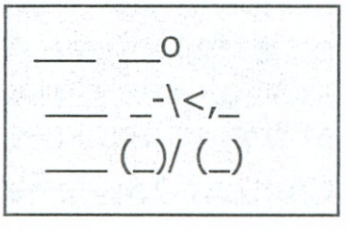
Mivel nincs eltérés az RFC 822-ben leírtakhoz képest, a MIME levelek minden probléma nélkül továbbíthatók a már létező levelező programokkal és protokollokkal. Csak a feldolgozásuk igényel néhány — a levelező programokban jelentkező — változtatást. Ez utóbbi okozhat a levelező programok között inkompatibilitást.

### Smileys (mosolygók)

Élő beszélgetéseknél a beszéd közben az érzelmeket a szöveg előadási módja mellett a testbeszéd (arcjáték, vállrándítás, stb.) is közvetíti. Email esetén ez a metakommunikáció hiányzik. Ennek pótlására kitalálták a mosolygókat. Ha a fejünket bal oldalra döntjük, és úgy nézzük:

Még néhány:

| | |
|---|---|
| ;-) | kacsintás; |
| :-( | helytelenítés; |
| :-O | meglepődés; |
| 8-) | szemüveg viselés; |
| =\|:-)= | Lincoln Ábrahám (Jó rendben,ez egy kicsit erős volt : -) ) |




A levél karakteres jellege egy új „művészeti" ágat is megteremtett: a karakterekkel kialakítható rajzok készítését. Egy ilyen „ASCII-Art" látható az oldalt lévő keretben.

### Levelezési listák és a Usenet

Az olyan levelezési fórumokat, amelyek hasonló témájú információcserére alakultak **levelezési listák**-nak nevezzük. A csoport tagjai levelezésen keresztül állnak kapcsolatban egymással, a tagok egy központi helyre küldik a leveleiket, majd onnan kerülnek az egyes csoporttagoknak elküldésre, vagy levelenként, vagy időszakonkénti, pl. naponkénti gyűjtésben. Ez utóbbi esetben, egy levélben kapja meg a lista résztvevője az összegyűjtött napi levelezést, ezt szokták **digest**-nek hívni.

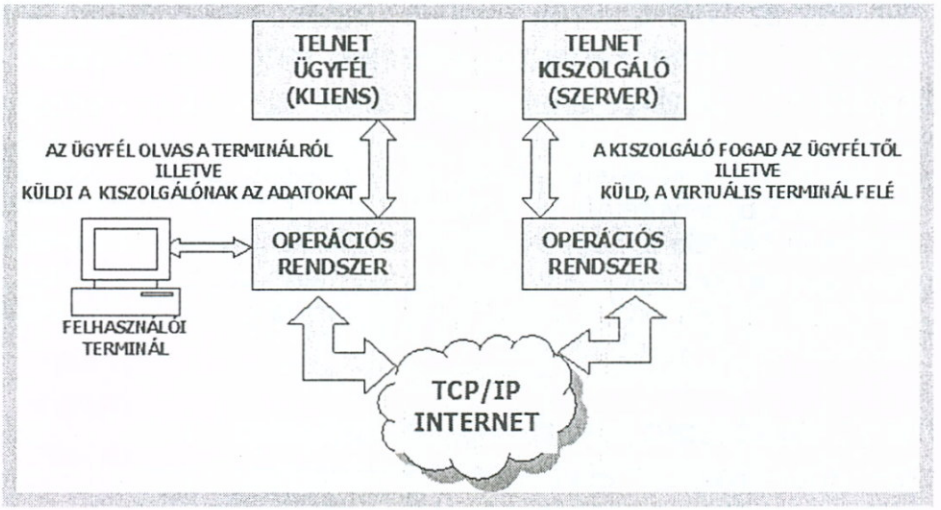
A USENET több mint 6000 témával foglalkozik. Ezek a **hírcsoport**-ok. A levelezési listáktól eltérően a hírcsoportba küldött leveleket nem kézbesítik, hanem anyagaikat szervereken tárolják, amit az adott géphez hozzáférési jogot kapott személyek elolvashatnak. Az összes hírcsoport anyagát csak néhány nagy hírszerver tárolja, a többieken csak egy-egy kiválasztott részük található.

A kezdők bekapcsolódását, kérdéseket és rá a válaszokat tartalmazó dokumentumok, az ún. FAQ-ok (Frequently Asked Questions, magyarul: GYIK = Gyakorta Ismétlődő Kérdések) segítik.

### Telnet

Egy távoli gépre úgy lehet belépni, mintha egy a vele való munkát lehetővé tevő terminálja előtt ülnénk.

(Emlékeztetőül: Terminál egy billentyűzettel és képernyővel ellátott általában soros vonalon kommunikálni tudó eszköz.)




*7-23. ábra TELNET működése*

**Telnet a gépek közti távoli bejelentkezést lehetővé tevő protokoll neve.** Ez is folyamatos (on-line) hálózati kapcsolatot igénye. Telnettel csak akkor tudunk egy másik gépre belépni, ha azon a gépen is van accountunk. (Felhasználói nevünk és jelszavunk).

Az ilyen típusú bejutás módot ad arra is, hogy a távoli gépet felhasználva (a Telnet programot ott elindítva) lépjünk be egy „kényesebb" gépre.

A belépés felderítésének a megakadályozása érdekében a távoli gépen naplózott fájlt a távoli gépről töröljük, amiben benne volt a bejutást kezdeményező gép (a behatoló) címe.

Bejelentkezés után a rendszer úgy viselkedik, mintha ott ülnénk a távoli gép előtt, azaz a távoli gép operációs rendszerének konvenciói érvényesek, parancsainkat a telnet protokoll adja át a távoli gép operációs rendszerének, és a távoli operációs rendszer hajtja végre. Így a távoli gépen programokat futtathatunk, megnézhetjük az odaérkezett leveleinket, stb.

## A Telnet protokoll

[15] A TELNET egy általános, kétirányú kommunikációt lehetővé tevő megoldás. Mivel sok program a felhasználóval való számítógéptípus független kapcsolattartást terminálok segítségével oldják meg, ezért ez az eszköz lehetővé teszi távoli programokkal történő kommunikációt. A fizikailag használt számítógép (általában a PC) a **telnet kliens**, a **telnet szerver** pedig a távoli számítógép, amelyhez a kliens a hálózaton keresztül csatlakozik. A telnet egyik fontos tulajdonsága, hogy képes az adatfolyamok átvitelét egyeztetni a telnet kliens és a szerver között. Ez az egyeztetés szimmetrikus, azaz a kliens és a szerver is kezdeményezhet és elfogadhat kéréseket.

A Telnet összeköttetés a TCP protokollt használja az adatok és a vezérlő információk átviteléhez. A protokoll három összetevője:

1. A hálózati virtuális terminál,
2. az egyeztetett opciók,
3. az egyeztetések (negotiations) szimmetriája.

Mivel a valóságban sokfajta terminál típust gyártottak, ezért a Telnet összeköttetés mindkét oldala **hálózati virtuális terminált** (NVT Network Virtual Terminal) tételez fel. Ezt az NVT-t kezeli mindkét oldal, a tényleges, fizikailag létező terminálok egy illesztő felületen keresztül jelennek meg. Így minden kommunikációt ezen NVT-k együttműködésével lehet megoldani.

Az NVT kétirányú karakteres eszköz, amely a bejövő adatokat megjelenítő nyomtatóból és a kimenő adatokat előállító billentyűzetből áll. Ez utóbbit az NVT szükség esetén echózza. A hétbites ASCII kód a 8 bites adatmezőt úgy tölti ki, hogy a legfelső bit nulla. Az NVT nyomtatójának nincs fizikai sor és laphossza, hanem csak a következő ASCII karaktereket értelmezi: NUL-üres művelet, LF-soremelés, CR-soremelés, BEL-hangjelzés, BS-egy pozíció vissza, HT-vízszintes tabulátor, VT-függőleges tabulátor, FF-lapdobás. A sorvég jelzése a "CR LF" kóddal történik.

2. Az egyeztetett opciók segítségével, a kapcsolatfelvétel során, a tényleges terminálok által nyújtott esetleges extra további szolgáltatásokba is meg tudnak egyezni. Az opciókat bármelyik oldal kezdeményezheti, a másik oldal pedig vagy elfogadja, vagy visszautasítja. Az opciókkal a két oldal a lehetséges legjobb szolgáltatást kísérli megvalósítani.

3. Az egyeztetések szintakszisának szimmetriája azt jelenti, hogy a két fél egyenrangú a kapcsolatban.

```
Üdvözli a Microsoft Telnet ügyfélprogram

Az escape-karakter 'CTRL+ú'

Microsoft Telnet> help

A parancsokat lehet rövidíteni. A támogatott parancsok:

c     - close                    a jelenlegi kapcsolat bezárása
d     - display                  a működési paraméterek megjelenítése
o     - open szgépnév [port]     csatlakozás számítógéphez
                                 (alapértelmezett port: 23)
q     - quit                     kilépés a telnet programból
set   - set                      beállítások bekapcsolása (lista: 'set ?')
sen   - send                     karakterláncok küldése a kiszolgálónak
st    - status                   állapotinformáció megjelenítése
u     - unset                    beállítások kikapcsolása (lista: 'unset ?')
?/h   - help                     súgó megjelenítése
Microsoft Telnet> set ?
&bsasdel              Backspace küldése delete-ként
crlf          Új sor mód - a return billentyű CR, LF kódot küld
delasbs       Delete küldése backspace-ként
escape x      Az x escape-karakter a telnet-ügyfél parancssorába lépéshez
localecho     A helyi echo bekapcsolása.
```

*7-24. ábra Windows Telnet kliens*

A Telnet parancsok – hogy elválasztódjanak a szöveges információktól – legalább kétbájtos szekvenciák: az első bájt (értéke:255) jelzi hogy parancs jön: "Értelmezd parancsként" (Interpret as Command IAC), a második bájt a parancs kódja, egyeztető parancsoknál van egy harmadik bájt, ez az opció kódja. Ha az IAC kódot adatként küldjük, akkor meg kell ismételni.

A Telnet TCP összeköttetéssel kapcsolatot teremt a felhasználó portja és a kiszolgáló portja között. A kiszolgáló várja saját jól ismert portján (értéke 23) az összeköttetést, majd ha igény érkezik, létrehozza. A Windows XP Telnet kliens programjának parancsai a *7.24. ábrán* láthatók. Részletes leírás az XP helpjében.

A hagyományos telnet TCP/IP kapcsolatban a kommunikáció a kliens és a szerver között titkosítás nélkül zajlik. A TCP/IP protokollnak titkosítással folytatott SSL kommunikációja alapjait a *9.4.3 fejezetben* foglaljuk össze. Létezik egy szabadszoftver, neve: **Putty** ami a biztonság kezelését is megoldja.

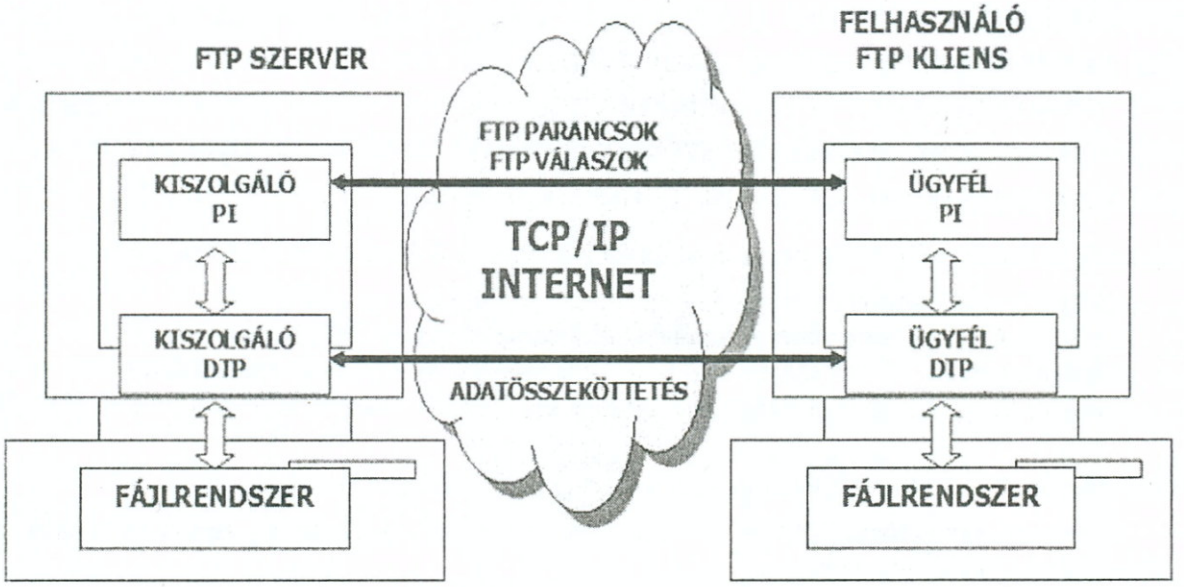
### *FTP (File Transfer Protocol)*

Az ftp protokoll a hálózatban lévő gépeken megtalálható fájlok átvitelére használható. Adatátviteli sebesség igénye is jelentősebb, hiszen elfogadható időn belül kell átvinnünk esetleg több száz kilobájtnyi adatot. Néhány kbájt/s-os átviteli sebesség már elfogadható.

Az ftp protokoll két átviteli módban működhet: ascii és binary. Az előbbi, mivel 7 bites kódokat használ, szövegállományok átvitelére alkalmas, az utóbbi bármilyen általános fájlra. Fontos továbbá, hogy egyes rendszerek (pl. Unix) különbséget tesznek kis és nagybetűk közt, azaz a fájl nevében tetszőlegesen lehetnek kis és nagybetűk.

A kapcsolat egy ftp programmal lehetséges, ott kell megadni a célgép nevét, ami egy internet cím. Ha a kapcsolat létrejött, a rendszer kéri az azonosítót és a jelszót. Ha a belépés sikeres, akkor a következő legalapvetőbb parancsokat használhatja:

A felhasználói általában akkor tud egy távoli gépről/gépre másolni, ha a távoli gépen is rendelkezik felhasználói jogosultsággal (account-tal).




*7-25. ábra FTP hatásvázlata*

- **dir** paranccsal listázhatja a célgép könyvtárszerkezetét,
- **cd** paranccsal válthat a könyvtárak között,
- **get** paranccsal hozhat le fájlokat a távoli gépről,
- **mget**-tel egyszerre többet
- **put** paranccsal tölthet fel fájlt a távoli gépre,
- **mput**-tal egyszerre többet.

## FTP kapcsolat

[15] Az FTP feladata a megbízható és hatékony adatátvitel egy ügyfél és egy kiszolgáló között. Két összeköttetést használ: egy adat és egy parancs összekötetést. Az adat összeköttetés kétirányú, és nem áll fenn az egész idő alatt. Az FTP protokoll az ügyfél és a kiszolgáló két folyamatának (processzének) az együttműködésén alapul. Az egyik folyamat a protokollértelmező (PI-Protokoll Interpreter), a másik az adatátviteli protokoll (DTP-Data Transfer Protocol).

Az ügyfél protokollértelmező (ügyfél-PI) kezdeményezi a vezérlő összeköttetést. A vezérlő összeköttetés a Telnet protokoll szerinti. Kezdetben az ügyfél-PI generálja a szabvány FTP parancsokat és továbbítja a vezérlő összeköttetésen keresztül. A szabvány válaszokat a kiszolgáló-PI küldi az ügyfél-PI-nek a vezérlő összeköttetésen keresztül a parancsok válaszaképpen.

Az FTP parancsok határozzák meg az adat összeköttetés paramétereit (data port, átviteli mód, stb.) a fájlrendszerben végrehajtott műveletek mibenlétét (store, retrieve, append, delete, stb.).

Mivel az FTP a Telnet protokollt használja az összeköttetetés vezérlésére, a legtöbbször úgy, hogy az ügyfél-PI illetve a kiszolgáló-PI a rendszer már létező Telnet modulját használja.

## FTP adatkezelése

Az adatátvitel során az adatok a küldő hoszt háttértárolójából a fogadó hoszt háttértárolójába jutnak. Gyakran szükség van adattranszformációra, mert a két rendszer különbözik. Kívánatos, hogy a karakterkonverzió az NVT-ASCII szerint valósuljon meg, különböző rendszerek esetén, ha szövegállományt viszünk át. Ilyenkor mindkét oldal végrehajtja a saját belső ábrázolási módjának megfelelő transzformációt.

**Adat típusok:** A felhasználó állítja be az adatábrázolási módot. A lehetséges adat típusok:

- **ASCII** Ez az alaptípus és minden egyes implementációnak ismernie kell.
- **EBCDIC** Azon gépek számára, amelyeknél a belső ábrázolás EBCDIC-et használ.
- **Kép(image)** Az adatok folyamatos bitenként mennek át 8-bites átviteli bájtban.
- **Lokális** Az adatok logikai bájtokban mennek át. Ilyenkor szükség van a bájtméret megadására.

**Adat struktúrák:** Az FTP-ben definiált állományszerkezetek a következők:

- **állomány-struktúra**, ilyenkor valójában strukturálatlan az állomány, vagyis folyamatos bájtsorozat. Ez az alapértelmezés.
- **rekord-struktúra**, az állomány szekvenciális rekordokból épül fel és
- **lap-struktúra**, amikor az állomány független indexelt lapokból áll.

**Átviteli módok:** Az FTP-ben definiált átviteli módok:

- **STREAM MODE** Az adatok bájtfolyamként mennek át.
- **BLOCK MODE** Az adatok blokkok sorozataként mennek át, amelyek a blokkot leíró fejrésszel rendelkeznek.
- **COMPRESSED MODE** Az adatismétlődéseket az előfordulásukkal együtt küldi, ami bizonyos tömörítést jelent.

**FTP parancsok:** Az FTP parancsok kategóriái:

- **hozzáférést vezérlő parancsok**- a távoli fájlrendszer jogosult elérését biztosítja
- **átviteli paraméterek parancsai** – átviteli mód beállítása
- **szerviz parancsok** – a fájlrendszer használatát, benne a mozgást biztosítja.

```
C:\Documents and Settings\drk>ftp
ftp> help
A parancsok rövidíthetőek. A parancsok a következők:

!           delete      literal     prompt      send
?           debug       ls          put         status
append      dir         mdelete     pwd         trace
ascii       disconnect  mdir        quit        type
bell        get         mget        quote       user
binary      glob        mkdir       recv        verbose
bye         hash        mls         remotehelp
cd          help        mput        rename
close       lcd         open        rmdir
ftp>
```

*7-26. ábra Windows XP FTP parancsok*

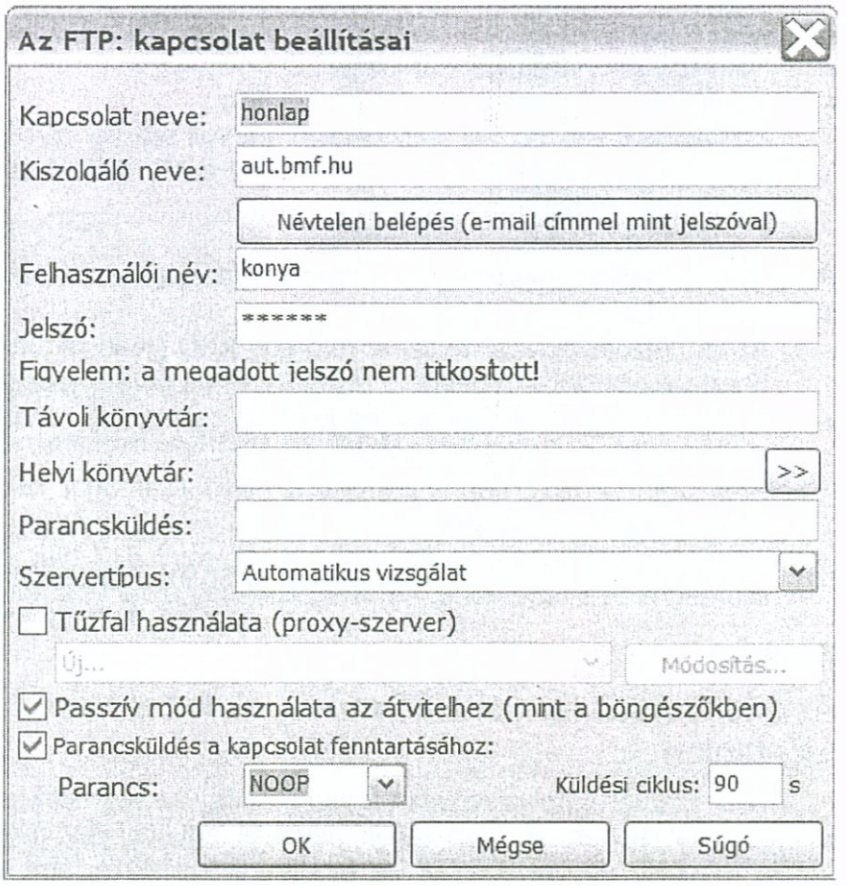
A kliensprogram a szerver 21-es TCP Portjához csatlakozik, és itt végrehajt egy bejelentkezést. Erre USER és PASS parancsokat használják.

A kliens beléphet alkönyvtárakba pl. a CWD paranccsal, vagy kezdeményezhet olyan parancsot is, amihez adatátvitel is szükséges. Ilyen parancsok például az NLST ahol a könyvtárlistázás az átvitt adat, illetve a RETR és a STOR fájl le- illetve feltöltésére.

Az adatkapcsolat kezdeményezése attól függ, hogy **passzív FTP** kapcsolatról, vagy **aktív FTP** kapcsolatról beszélünk. Aktív FTP kapcsolat esetén a PORT parancsot kell kiadnunk, melynek paramétereit vesszővel választjuk el. Paraméterei: az IP Cím, a Kliens Adatcsatornájának TCP Portja a hálózaton használt bytesorrend szerint.

Passzív FTP kapcsolat esetén a PASV parancsot kell kiadnunk paraméterek nélkül, és a szerver a válaszban fogja megmondani, hogy hova kapcsolódjon a kliensünk. A válasz formátuma hasonló a PORT parancsnál használatos formátumhoz.

Amennyiben az előző lépésben adatforgalommal járó kapcsolatot szerettünk volna kezdeményezni, akkor kiadhatjuk az ennek megfelelő (NLST, RETR, STOR, stb.) parancsot. Innen az előző lépés elejétől folytathatjuk a működésünket (vagyis, újabb adatforgalommal járó tevékenységhez új adatcsatornát kell kérjünk), vagy kiléphetünk a QUIT paranccsal.




*7-27. ábra Total Commander FTP kliens*

A Windows XP ftp kliens programjának parancsai a *7-26.. ábrán* láthatók. Részletes leírás az XP helpjében. A Total Commander fájlkezelőben is van FTP kliens *(7-27. ábra).*

### Anonymous FTP

Vannak mindenki számára elérhető, nyilvános elérésű gépek, amelyekre természetesen nem kell account-tal rendelkezni, ez az ún. anonymous ftp. Az ilyen gépekre bejelentkezve bejelentkező (login) névként az "anonymous" szót kell begépelni.

A rendszer ekkor arra kér, hogy jelszóként a saját email címünket adjuk meg, ez sokszor gyakorlatilag nem kötelező, kizárólag statisztikai célt szolgál. Ezek után a távoli gépet, pontosabban annak nyilvánosan elérhető könyvtárait láthatjuk, és az összes fenti ftp parancs használható.

Ez egy nem túl kényelmes, de jól használható módszer fájlok letöltésére, ha nincs más mód. Természetesen ehhez ismerni kell a letöltendő fájl pontos útvonalát is.

Azok részére, akik csak Email kapcsolattal rendelkeznek, létezik a levéllel történő off-line ftp, az **ftpmail**. Ennek az a lényege, hogy vannak olyan hálózatra kötött számítógépek, amelyek az ftpmail server programot futtatják. Ez fogadja a leveleket, és feldolgozza a bennük ftp-vel elérni kívánt gép címét, és az ftp parancsokat tartalmazó utasításokat. Az ftpmail program végrehajtja a kijelölt ftp kapcsolatot, letölti a megadott fájlt, uuencode-olja, majd elküldi levélben a feladónak.

### TFTP (Trivial FTP)

Használják még az FTP egyszerűbb változatát, a Trivial FTP-t kisebb fájlok átvitelére, mint például mikor egy kliens a hálózaton keresztül bootol.

A működéséhez a 69-es UDP portot használja (az FTP a TCP 21-es portját).

- Nem tud könyvtártartalmat listázni
- Nincs azonosító, vagy titkosító megoldása
- Csupán fájlokat tud írni vagy olvasni a távoli szerver-re/ről
- Csak három átviteli módot támogat: "netascii", "octet" és "mail". Az első kettő megfelelaz FTP "ASCII" és "image" (bináris) módjának.

Mivel UDP protokollt használ, ahol nincs a kapcsolatfelvételnél háromutas kézfogás, ezért röviden leírjuk az adatátvitel protokollját:

- Az "A" kezdeményező hoszt A küld egy RRQ (read request) vagy WRQ (write request) csomagot a "B" hosztnak, elhelyezve benne a fájlnevet és az átviteli módot.
- B válaszol egy ACK (acknowledgement) csomaggal, amiben "A"-tájékoztatja, hogy "B" melyik portján várja a küldendő többi csomagot.
- Ezek után "A" 512 bájtos számozott ADAT csomagot küld "B"-nek, amely mindegyiket számozott ACK csomagokkal nyugtázza.
- Az utolsó ADAT csomag kevesebb lehet, mint 512 bájt, ez egyben jelzi az átvitel végét. Ha az utolsó csomag is 512 bájt lenne, "A" még küld egy nulla bájt hosszúságú csomagot.

### Archie

Az anonymous ftp-vel elérhető fájlok keresésére használható adott név, vagy névrészlet alapján.

Az archie szerverek folyamatosan figyelik az ftp-vel elérhető szervereket egy adott régióban, és az elérhető könyvtárakat a bennük lévő fájlok neveivel együtt, egy folyamatosan frissített adatbázisba helyezik. Az archie kezelése egyszerű, csak be kell írni a kulcsszót, ami alapján keresünk, és egy listát kapunk arról, milyen néven mit talált az archie a saját adatbázisában. Telnet protokollal lehet archie serverekkel kommunikálni.

## 7.8.3 Információk szervezése a hálózaton.

Az információk összegyűjtése, rendezése és megkeresése, a dinamikusan változó hálózaton nem egyszerű dolog. Hogyan szervezhetjük meg az információk közötti kapcsolatokat? Alapvetően két megoldás kínálkozik:

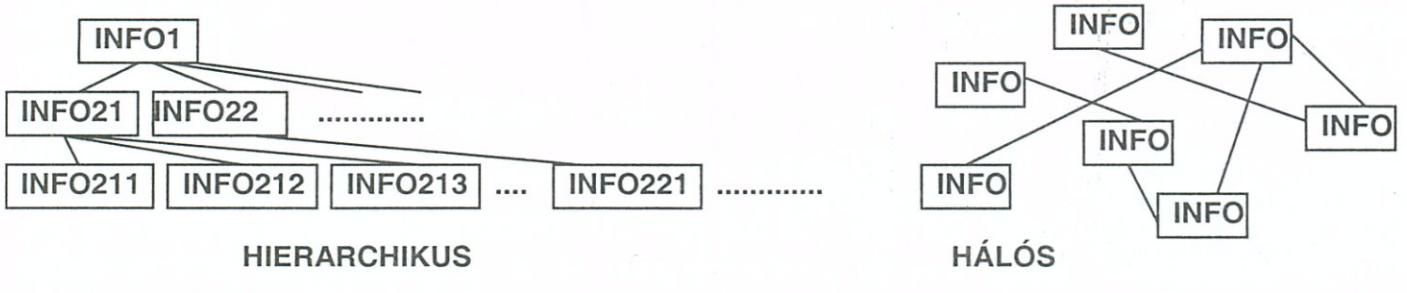
Az egyik a **hierarchikus**: ez azt jelenti, hogy az információk közötti összefüggéseket egy szintekből álló rendszerbe szervezzük. Elindulunk a legmagasabb szintről, és minden szinten kijelöljük, amihez tartozó alatta lévő szinten folytatjuk a keresést. Ez hasonló a számítástechnikában megszokott menürendszerhez: egy menüpontot kiválasztva megnyílik egy almenü, majd abból kiválasztva egy újabb, és így tovább.

Jó példa erre a könyvtárakban használt Egyetemes Tizes Osztályozás (ETO) rendszer, amely az emberiség tudásanyagát ilyen módon próbálta rendszerezni: tíz főcsoportot választott, és ezeken belül újabb alcsoportokat definiált. Például az elektrotechnika témakör: a 6. főcsoport (Alkalmazott tudományok, technika) 62 alcsoportjának (Technika) 621 jelölésű (Gépek, elektrotechnika) témaköre.

Ennek a rendszernek számos előnye mellett hátrányai is vannak:

- Nagyon kötött, a struktúra felsőbb szintjei már nem változtathatók meg.
- Minden információt valahova be kell sorolni, ez felveti a határterületek besorolásánál a nem egyértelműség kérdését.
- Nem veszi figyelembe azt, hogy az egyik terület robbanásszerű fejlődése miatt aránytalanná válik az egyes alszintek egymáshoz viszonyított súlya.

A másik lehetőség az információk olyan szervezése, amely az egymásra való utalásokon, hivatkozásokon alapul. Erre is van példa az írott médiákban: a lexikonokban szereplő ún. kereszthivatkozások (jelölése: ->, jelentése: lásd).




**INFORMÁCIÓK KÖZÖTTI KAPCSOLATOK KIALAKÍTÁSA**

*7-28. ábra: Kapcsolatok az információk között*

Ez a hálós, egymásra hivatkozó (mutató) módszer csak a számítógépes dokumentumkezelés megjelenésével vált a gyakorlatban kényelmesen használhatóvá, és az ilyen módon kialakított szövegeket **hipertext**-nek hívjuk.

### *Gopher*

A hálózaton való hierarchikus keresés végezhető a programmal. Erőforrásigény nem nagy: Mikor elindítjuk a programot egy számozott menüpontokból álló menűt gördít le. A menüpont kiválasztása a megfelelő számbillentyű megnyomásával történik, és a kiválasztott menüponthoz tartozó almenü jelenik meg, szintén számozott menüpontokkal, és így tovább. Elavult. A mobiltelefon hálózatokban él tovább WAP néven.

## WWW (World Wide Web)

A WWW általános ügyfél-kiszolgáló hálózati koncepcióra épül. Az információszolgáltató gépeken egy WWW kiszolgálóprogram (Web szerver) program fut, amely a felhasználók gépein futó böngészőprogramok (Netscape, Explorer, Mozilla, Opera, stb.) által küldött kérésnek megfelelően elküldi a kért információt az adott gépre, amely ebben az esetben az ügyfél (kliens).

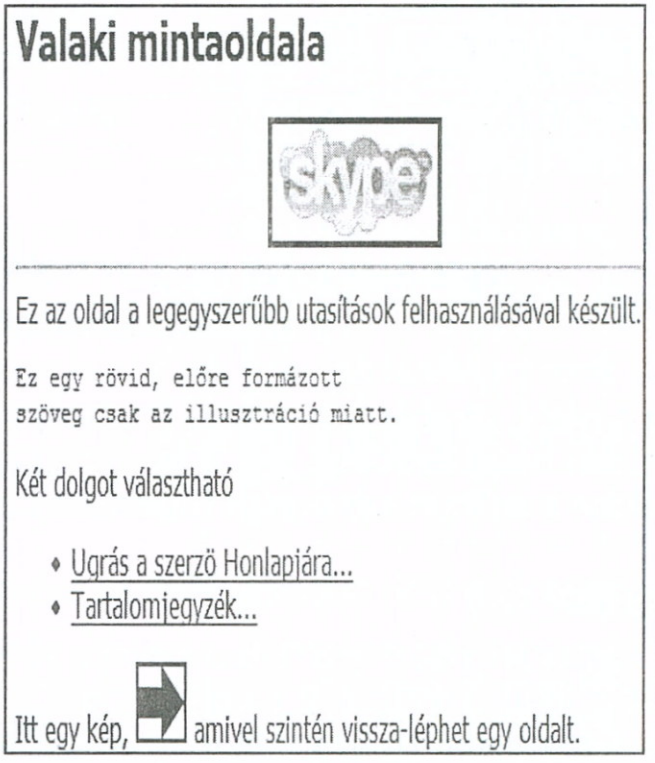
Minden információkérés és az arra adott válasz független a többitől, vagyis a kapcsolat csak az átvitel idejére jön létre. A kiszolgáló nem figyeli külön az egymás után beérkező igényeket, mindet új kérésként kezeli, még akkor is, ha az esetleg azonos helyről érkezett. A WWW működését a gyakorlatban több tényező biztosítja:

Minden információs egység — kép, grafika, animáció, szöveg — forrásként jelenik meg a hálózaton.

A forrásokra olyan módon lehet hivatkozni a kapcsolatok felépítése során, hogy meg kell adni a forrás helyét, és annak módját, hogy a használt program hogyan tudja megjeleníteni, használni ezt a forrást.

Az alkalmazott megjelenítési módot a később ismertetendő **URL (Uniform Resource Locator — egységes forrásazonosító)** adja meg.

```
<HTML>
<HEAD>
<TITLE>Mintaoldal</TITLE>
</HEAD>
<BODY BACKGROUND="bg_hatter.jpg">
<H2>Valaki mintaoldala</H2>
<CENTER>
<IMG SRC = "skype.gif" BORDER=3><BR>
</CENTER>
<HR>  Ez az oldal a legegyszerűbb uta-
sítások  felhasználásával készült.
<PRE>
Ez egy rövid, előre formázott
szöveg csak az illusztráció miatt.
</PRE>
Két dolgot választható <BR>
<UL>
<LI><A HREF =
"http://www.aut.bmf.hu/konya"> Ugrás a
szerzö  Honlapjára...</A>
<LI><A HREF =
"start.html">Tartalomjegyzék...</A>
</UL>
Itt egy kép, <A HREF = "start.html">
<IMG SRC = "up.gif"></A> amivel  szin-
tén vissza-léphet egy oldalt.
</BODY>
</HTML>
```




*7-29. ábra HTML forrás és megjelenése*

### HTML

A dokumentum fogalmát itt általánosabban kell értelmeznünk: ezek objektumok, amelyek lehetnek: szöveg, kép(grafika), hang (zene), de akár mozgókép (film) is, illetve futtatható alkalmazások (flash, java, javascript, stb).

A dokumentumok logikai struktúráját a HTML (Hyper Text Markup Language) jelölései segítségével lehet szabályozni [9,17]. A HTML arra készült, hogy segítségével a dokumentumok szokásos, sorban egymás utáni olvasása helyett, a szövegben elhelyezett kapcsolatok alap-

ján az egész dokumentum könnyebben legyen áttekinthető és elolvasható. Segítségével logikusan szervezett és felépített dokumentumokat lehet készíteni, olyan módon hogy a nyelv alkalmas logikai kapcsolatok létrehozására a dokumentumon belül és dokumentumok között, amit a dokumentum olvasója kezelhet.

Ahogy ezt már az előzőekben megírtuk az ilyen módon szervezett szöveget **hypertext**-nek hívjuk. A folyamatos, sorokba rendezett szöveg végigolvasása helyett a kereszthivatkozásokat követve könnyen el lehet menni a szöveg egy más részére, megnézni más információkat, azután visszatérni, folytatni az olvasást, azután megint egy másik bekezdésre ugrani. Ilyen szerkezetűek a weboldalak, valamint majdnem minden korszerű program súgója. Amennyiben a szöveg mellett más médiaobjektum is megjelenik, akkor hipermédiáról beszélünk.

Lényeges, hogy a kapcsolt objektum is tartalmazhat további kapcsolásokat különböző objektumokhoz, amelyek elvileg a hálózaton bárhol lehetnek. A WWW úgy is tekinthető, mint egy dinamikus információ tömeg, amelyben a hypertext segítségével kapcsolatok (linkek) vannak.

Az előbbiekben leírtak illusztrálására egy mintaoldal forrásnyelvi alakját, és böngészővel történő megjelenítését a *7-29. ábrán* mutatjuk be.

A HTML formátumú fájl valójában egy szöveges fájl, szintén szöveges (olvasható) vezérlőkódokkal. Ezek a vezérlőkódok < és > jelek között szerepelnek, és a szöveg megjelenését, formátumát, például a betűk nagyságát, formáját, stb. jelölik.

A szöveg egyéb dokumentumokra vagy a dokumentum más részeire való hivatkozásokat is tartalmazhat, amit a vezérlőkódok segítségével adhatunk meg linkek formájában. Ezek a linkek — amelyek a megjelenítéskor általában kék színű, aláhúzott szövegekként, vagy kék keretes ikonokként jelennek meg — hypertext alakúvá teszik a dokumentumot.

### URL specifikációk

URL (Uniform Resource Locator) egységes forrásazonosító: megadja a megjelenítő program számára, hogy az adott szövegrészhez, képhez, grafikához kapcsolt dokumentumot milyen módszerrel lehet megjeleníteni, milyen típusú kapcsolatot kell felépíteni, illetve hogy ez a forrás hol, az Internetre kapcsolt gépek közül melyiken található [10].

Példaként egy URL:

```
http://www.aut.bmf.hu:8080/WEB/ai/leiras.html
```

A kapcsolt (a kapcsolatban hivatkozott) állomány a leiras.html nevet viseli, a WEB/ai könyvtárban található a www.aut.bmf.hu című gépen, amely a Web-szolgáltatásokhoz az alapértelmezésként szereplő 80-as port helyett a 8080-at használja. A kiszolgáló a HTTP-protokollal érhető el. A kiszolgáló domén neve helyett IP-címe is használható:

```
http://193.224.41.151:8080/
```

Az URL a következő információkat tartalmazza:

- **a protokollt, amelyet az adott forrás eléréséhez használunk** (ftp, http, gopher stb.); Az URL első tagja az adott forrás eléréséhez használandó protokollt

adja meg. Az URL segítségével az Interneten használt legtöbb információforrás elérhető.

- **annak a kiszolgálónak az domén-nevét, amelyen az adott forrás található.** Nem anonymous kapcsolat esetén, ha szükséges, itt kell megadni a felhasználó nevét és a jelszót is. Ez az információ két perjellel (//) kezdődik és egy (/) zárja le.
- **a kiszolgáló portjának a számát.** Ha ez nem szerepel, akkor a megjelenítőprogram az általánosan használt alapértelmezést feltételezi. Ha a kapcsolódáshoz nem a WWW-hez javasolt 80-as portcímet használják, akkor ezt az URL-ben a kiszolgáló nevéhez vagy címéhez kettősponttat (:) kapcsolva kell megadni.
- **a forrás helyét a kiszolgáló lemezegységének hierarchikus állományrendszerében (könyvtár/fájlnév).** Ez közvetlenül a kiszolgáló nevét lezáró perjel (/) után áll. A keresési útvonal megadásának formája attól függ, hogy milyen fajta szolgáltatáshoz kapcsolódtunk. Gyakran egészen az állomány szintjéig meg kell adni az elérési utat.

Azonos könyvtárban lévő dokumentumok esetén elég csak először a teljes URL-t megadni, utána már elég a többi fájlnak csak a nevét megadni.

Egy adott HTML-kapcsolaton belül az azonos könyvtárban lévő állományok eléréséhez nem kell a teljes keresési útvonalat megadni. Ha egy dokumentumot elértünk a rendszeren, ez már bizonyos információkat szolgáltat a következő kapcsolat felépítéséhez. Így a szomszédos állományok eléréséhez elegendő egy rész-URL alkalmazása, ami az aktuális dokumentumhoz viszonyítva relatív kapcsolódást biztosít. A

```
http://www.aut.bmf.hu/
```

URL esetén a megjelenítő program a megadott kiszolgáló főkönyvtárát keresi. A WWW-szerver konfigurálásakor megadható, hogy ilyen esetben melyik legyen az a HTML-dokumentum, amelyet a kiszolgáló elküld a felhasználónak. Ez lehet pl. üdvözlés, vagy információ a szolgáltatásokról, más URL megadása, tartalomjegyzék, hibaüzenet.

A WWW kiszolgálót futtató gépen a felhasználók a saját könyvtárukban lévő, a rendszer konfigurálásakor definiált speciális nevű alkönyvtárban mindenki számára hozzáférhető, személyes HTML-dokumentumokat hozhatnak létre. A könyvtári hivatkozás a felhasználó neve. Ez azt jelzi a kiszolgáló számára, hogy az adott felhasználó alkönyvtárában kell az állományokat keresni. Például a konya felhasználói névhez tartozó személyes dokumentumok a

```
http://www.aut.bmf.hu/konya
```

URL segítségével érhetők el. A kiszolgáló konfigurálásakor meg kell adni annak az alkönyvtárnak nevét, amelyben a felhasználók létrehozhatják az ilyen személyes dokumentumaikat (home page, honlap).

Ez a könyvtárnév a kiszolgáló konfigurációs állományában (a UNIX-rendszereknél általában a /etc/httpd.conf) megtalálható (pl. public_html, wwwhomepage).

Még egy példa. Egy FTP-URL legáltalánosabb formája:

```
ftp://[USER[:PASS]@]SZERVER[:PORT] [;type=<TYPECODE>] [/DIR]
```

Ahol:

| | |
|---|---|
| **USER** | **a felhasználó neve** |
| **PASS** | **a jelszó (Password)** |
| **SZERVER** | **IP vagy DNS - cím** |
| **PORT** | **Portszám (default érték: 21)** |
| **TYPECODE** | **a letöltés módja: a ... ASCII I ... Image (binary)** |
| **DIR ...** | **directory** |

Ha valakit pl. 'kovacs'-nak hívnak, es 'golem' a jelszava, akkor egy nem anonymous FTP-URL :

**ftp://kovacs:golem@ftp.aut.bmf.hu type=a /pub/systems**

A böngésző erre bejelent ezzel a névvel és jelszóval, ASCII-módot állít be és a /pub/systems könyvtárat hívja le. A parancsátvitel az alapértelmezett 21-es porton történik.

Ugyancsak a rendszer létrehozása során definiálható annak az állománynak a neve, amely a rendszerbe való belépéskor, illetve a saját könyvtárak címzésekor megjelenik a felhasználók képernyőjén. Ezt a HTML dokumentumot általában a következő nevek valamelyikével látják el: **welcome.html index.html index.htm.**

## CGI

Ahogy azt már leírtuk, a HTTP-protokollt a WWW ügyfél (a böngésző) a HTTP-kiszolgálókkal való kommunikációra használja. Ennek segítségével az ügyfélprogram adatokat kérhet a kiszolgálótól, és információkat küldhet a kiszolgálóra. Más esetekben az ügyfélprogram akar valamit küldeni a kiszolgálónak feldolgozásra. Általában ezeket a kapott adatokat a kiszolgáló nem maga kezeli, hanem továbbítja őket az ún. gateway programoknak, amelyek nem a HTTP-rendszer részei.

A CGI-specifikációk (Common Gateway Interface) írják le, hogy a HTTP kiszolgálók hogyan kommunikálnak a böngészők által küldött információkat ténylegesen feldolgozó programokkal [10]. Amikor a megjelenítő egy olyan kapcsolathoz ér, amely egy programra hivatkozik, a kiszolgáló elindítja ezt a programot és a CGI-leírást használva, átadja az ügyféltől érkező adatokat (ha vannak). A külső program a kapott információt felhasználva elvégzi a feldolgozást vagy lekérdezést, és a választ (ugyancsak a CGI-leírást használva) visszaküldi a kiszolgálónak. A kiszolgáló ezt azután HTML dokumentum formájában továbbítja a kérést küldő megjelenítő programnak.

Példaként gondoljunk arra, hogy egy weblapon lévő anyagok között keresünk. Amikor beírjuk a keresendő szót a böngészővel megjelenített weboldalra, és rákattintunk a jóváhagyásra, akkor a böngésző a szerverrel kialkított kétiráányú kapcsolatot felhasználva, elküldi a keresendő szót, a CGI specifikáció szerint. A szerver fogdja, majd kifejezés feldolgozása után meghívja a gépen lévő keresőprogramot, átadja neki a keresendő szót. A kereső ez alapján magtalálja (vagy nem) a keresett anyagot, és jellemzőit visszaadja a web szerver programnak. A szerver program HTMl kódba ágyazza, és visszaküldi a bögészőnék, és az megjeleníti.

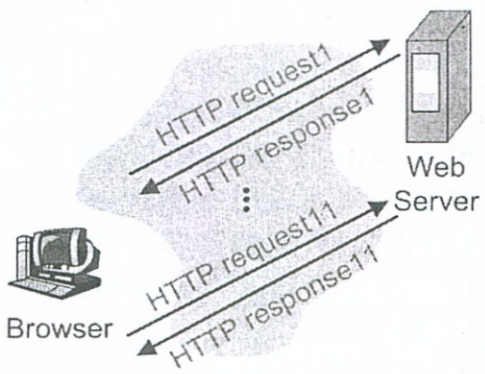
## A HTTP protokoll

A HTTP ügyfél kiszolgáló protokollt hypertext dokumentumok gyors és hatékony megjelenítésére tervezték. A protokoll **állapotmentes**, vagyis az ügyfélprogram több kérést is küldhet a kiszolgálónak, amely a kéréseket egymástól teljesen függetlenül kezeli, és minden dokumentum elküldése után le is zárja a kapcsolatot.

A HTTP-kapcsolat négy lépése:

1. **A kapcsolat megnyitása.** Az ügyfél meghívja a kiszolgálót az Interneten keresztül az adott cím és port azonosító alapján (alapértelmezésben a 80-as porton keresztül).
2. **A kérés elküldése.** Az ügyfélprogram üzenetet küld a kiszolgálónak, amelyben valamilyen kiszolgálást kér. A kérés HTTP-fejlécbőI és a kiszolgálónak küldött adatokból áll (ha van ilyen). A fejléc információkat tartalmaz a kiszolgáló számára arról, hogy milyen típusú a kérés, és megadja, hogy az ügyfélprogramnak milyen lehetőségei vannak.
3. **A válasz.** A kiszolgáló a választ visszaküldi az ügyfélprogramnak. Ennek része a fejléc, amely leírja a válasz állapotát (sikeres vagy sikertelen, a küldött adatok típusát), és ezt követik maguk az adatok.
4. **A kapcsolat lezárása.** A kiszolgáló a válasz elküldése után lezárja a kapcsolatot, így az erőforrások megint felszabadulnak a következő kérésekhez.

Az állapotmentesség miatt a kapcsolatok semmit nem tudnak az előző kérésekről, mivel a kiszolgáló minden dokumentum elküldése után lezárja a kapcsolódást, és minden kérést egyenként, külön-külön kezel.




*7-30. ábra HTTP protokoll*

Ha egy dokumentum több képet vagy grafikát tartalmaz, akkor ezek megjelenítéséhez az ügyfél annyiszor építi fel a kapcsolatot, ahány hivatkozást talál: egyet magának a dokumentumnak, és a többit egyenként a grafikáknak, illetve képeknek. (A protokoll 1.1-es verziója, már szakít ezzel a tiszta állapotmentességgel, és egy kapcsolatban több részt is letölthetünk. A HTTP protokoll nagyon egyszerű: a kérések és a válaszok egy vagy többsoros szövegek, soronként kocsi vissza - soremelés (CR-LF) karakterekkel befejezve. Ha a kérés sikeres, akkor a dokumentum ugyanezen a kapcsolaton keresztül kerül átvitelre, majd a kapcsolat lezárásra kerül.

A felhasználók felől érkező kérésekről azonosításra a következő információkat tárolja a program:

- A kérést küldő gép Internet-címe, ahonnan a kérés érkezett; Ez lehet a gép Internet-neve vagy IP-címe
- a dátum és a helyi idő;
- **a kérés módja (GET, POST);** Jelzi, hogy a megjelenítő-program milyen kérést küldött a kiszolgálónak.
    - GET elküldi a kért dokumentumot.
    - HEAD elküldi a dokumentum HTTP-fejlécében lévő információkat.
    - LINK egy meglévő objektumot (képet, programot, állományt stb.) egy másikhoz kapcsol. Ez pl. egy HTML dokumentum számára azt jelentheti, hogy módosítja a dokumentumot, és a fejlécbe beírja a kapcsolás (LINK) információit.
    - POST elküldi az adatokat a megjelölt URL-nek. Ennek már léteznie kell.
    - PUT elhelyezi az ügyfél által küldött adatokat a megjelölt URL-ben, felülírva a régebbi tartalmat. Az URL-nek már léteznie kell.

- UNLINK eltávolítja a meglévő kapcsolási információt, amelyet pl. előzőleg egy LINK parancs helyezett el a dokumentumban.
- TEXTSEARCH megkeresi a kért URL-t és elvégzi a keresést. ehhez a GET módszert, és azt az URL-t használja, amely tartalmazza a kéréskor elküldött adatokat.

- a kért dokumentum neve;
- a kiszolgáló által használt HTTP protokoll verziószáma;
- a kapcsolatkérés eredményére utaló kód;
- az elküldött dokumentum hossza.

### Cookie (Süti)

A cookie egy olyan azonosító, amelyet a kliens-oldali böngésző program (pl. Netscape Navigator, Internet Explorer, stb.) eltárol, és minden egyes, a megfelelő szerver felől érkező kérés esetén a szervernek elküld. Lényegében a felhasználó azonosíthatóvá válik.

Például, ez teszi lehetővé a TV-műsorok számát, és sorrendjét a szerver megjegyezze, és a műsor oldal linkjére kattintva a felhasználó által beállított módon jelenítse meg az oldalt.

A cookie-k nem biztosítanak semmiféle információszerzési vagy visszaélésre alkalmas lehetőséget. Nem lehet velük a kliensrôl vagy a futtató gépről információkat szerezni. Egyszerűen csak a kommunikáció egyszerűsítésére szolgálnak. Felmerül az az igény is, hogy a szerver a kérés küldése mellett egyéb információt is kapjon a klienstől.

A cookie üzembe állítását a szerver kezdeményezi. A kliensnek joga illetve lehetősége van ezt megtagadni, de ekkor — értelemszerűen — a szerveren a cookie használatához kötött dolgok nem működnek (jól). Egy szerver által beállított cookie-t a browser (kliens) csak az adott szervernek küldi vissza, esetleg azon belül is csak egyes dokumentumok elérésekor; tehát az egyes szerverek az egymás által beállított cookie-król nem tudnak.

Egy cookie-nak lehet lejárati határideje, ekkor a kliens csak a megadott ideig használja azt, annak lejárta után "elfelejti". Ha nincs megadva lejárati határidő, akkor a cookie csak a böngészőből való kilépésig él.

**Minden alkalmazás kialakítható cookie használata nélkül is, legfeljebb több szerver oldali erőforrást köt le.**

### Portálok

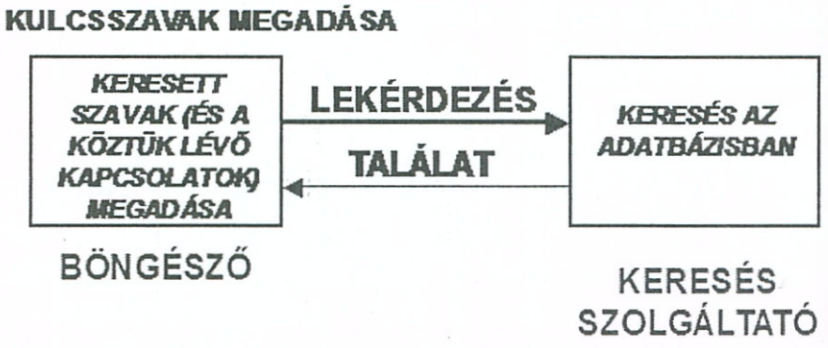
Egyre jobban terjednek az olyan WEB helyek, amelyek számos szolgáltatással „csábítják" oda az internetezőket. Ezeket hívjuk **portál**-oknak. Ezekben több szolgáltatást gyűjtöttek egybe:

- Hírszolgáltatás
- Kereső szolgáltatás
- Csevegő csatorna (chat)
- Sportrovat,
- Internetes vásárlóhely,
- Ingyenes e-mail

- Szaknévsor,
- Időjárás,
- Térkép

## Keresés az Interneten

Az interneten folyamatosan változó, hatalmas információtömegben az eligazodás csak a keresőprogramok használatával lehetséges. A keresőeszközök a következőképp csoportosíthatók:




A KERESÉS FOLYAMATA

**keresőmotorok:** a weboldalakon található szavakat indexelt adatbázisba gyűjtik.

**témakatalógusok:** - emberek végzik a weblapok katalogizálását

**metakeresők:** a keresési feladatot egyszerre több keresőeszközzel végzik

Az automatikusan működő keresőrobot programok a weboldalakon található szavakat indexelt adatbázisba gyűjtik. (Kereső robot: egy olyan program, amely a keresőmotor adatbázisát automatikusan frissíti a különféle bejárt weblapok elemzése, vizsgálata alapján. Az indexelés: logikai sorbarendezés). A keresést az adatbázisban végző keresőmotorok jellemzői:

- **pontosság (precision)** – a találat mennyire fedi le a lekérdezést
- **találati arány (recall)** – a katalogizált és a keresés szempontjából fontos dokumentumok hány százaléka van a találatok között
- **lefedettség (coverage)** – a világon lévő összes vonatkozó dokumentumból mennyi van a keresőmotor adatbázisában.

Például, a világon 50 dokumentum foglalkozik ceruzahegyezővel. A kereső motor adatbázisában összesen 10 dokumentum van, ebből 5 foglalkozik a ceruzahegyezővel. Beírva a „ceruzahegyező" szót, 6 találatot kapunk, de csak 4-ben szerepel ténylegesen a szó. Ezért: pontosság: 4/6=0.66, találati arány: 4/5=0.8, lefedettség: 5/50=0.1

A keresőmotorok a weblapon szereplő következő tartalmi elemeket gyűjtik ki:

title (cím) - description (leírás) - keywords (kulcsszavak) - body (törzs) – maga a weblap szövege.

Az első három elem a weblap megjelenítésekor nem látható, az adatbázist frissítő kereső robotok ezeket is vizsgálják. *Saját weblap esetén célszerű ezen elemek átgondolt megadása!*

## A hálózat emberi tényezői

Az eddigiekben a hálózatról, mint a technikai lehetőségről beszéltünk, amivel egymástól távoli emberek képesek kommunikálni. A távolság, és a bizonyos mértékű „anyagtalan személytelenség" miatt, etikai kérdésekről is érdemes szót ejteni. Stílszerűen

ezek egy szabályrendszerben, az. úgynevezett Netiquette-ben (hálózati etikett) szerepelnek. Igaz, hogy az Internet a nagy szabadság egyik megnyilvánulása, de itt is szükség van bizonyos íratlan szabályok betartására. Alapgondolata: ***Ne éljünk vissza a hálózat nyújtotta lehetőségekkel!***

Ennek több összetevője van, vegyük közülük néhányat sorjában:

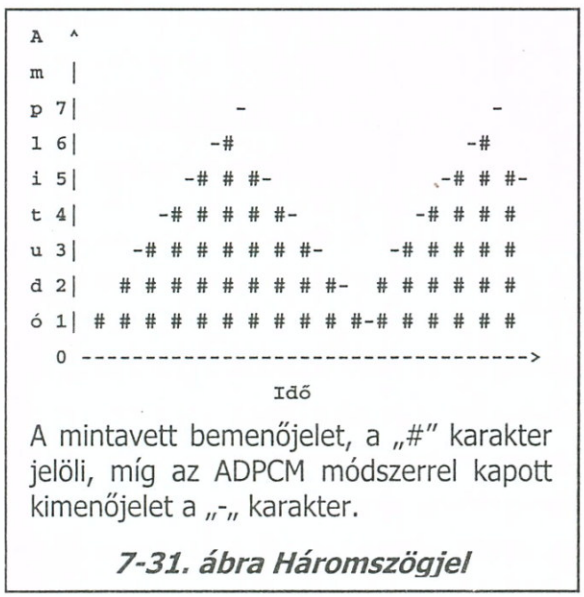
Fontos a vitafórumokon való kultúrált hangvétel megválasztása, továbbá fontos tudni, hogy aki a levelünket olvassa, (legtöbbször) nem ismer minket élőben, nem ismeri hátterünket, kizárólag a leírt soraink alapján ítél meg minket. Gyakran okoz félreértést, hogy sokan elfelejtik, egy leírt mondatot sokféle hangsúllyal fel lehet olvasni, gyakran különböző jelentéstartalommal. Ezért fokozottan ügyelni kell arra, hogy a csatorna csak a száraz szavakat küldi át, nem képes a személyes beszédet kísérő és az értelmezését segítő metakommunikációra. Ezen segítenek a levelezésnél már leírt smiley-k.

Ne éljünk vissza azzal, hogy a vitapartner fizikailag úgysem tud visszavágni. Mert amit élőben nem mernénk szemébe mondani, leírhatjuk levélben, hiszen nem kell senkivel szembe néznünk. Az ilyen magatartás gyakran vezet kilátástalan vitákhoz az egyes fórumokon.

A netikett a hálózat használatával kapcsolatban mondja azt, hogy mivel a hálózat közös terület, az adatátviteli sebesség korlátozott, ne terheljük le feleslegesen, mert mindenki munkáját megnehezítheti. Mindig az adott rendszer teljesítőképességein belül kell maradni, például ne fűzzünk a leveleinkhez szép nagy ASCII grafikát aláírásunkként, mert sok felhasználó olyan vonalat használ amelynek kicsi az adatátviteli sebessége.

Ez konkrétan általában nincs megtiltva, a felhasználó józan belátásában célszerű reménykedni.

Az Internet nagy sikerének egyik záloga az volt, hogy nonprofit alapon működik. Ez alól persze kivételek a helyi szolgáltatók, de nemzetközi viszonylatban el lehet mondani. Reméljük ez így is marad, és a technika fejlődésével mindenki hozzáférhet előbb-utóbb, aki akar. Az persze kérdéses, hogy amikor már boldog boldogtalan üzleti alapon Internetet szolgáltat, mennyire marad nonprofit a Hálózat, de reméljük, a gerinchálózatok, az Internet túlnyomó része legalább megmarad nonprofit szervezésűnek.




A mintavett bemenőjelet, a „#" karakter jelöli, míg az ADPCM módszerrel kapott kimenőjelet a „-„ karakter.

***7-31. ábra Háromszögjel***

## *7.8.4 Valós idejű hangtovábbítás (VoIP)=Voice over IP)*

A bevezetőben említett távközlés és számítástechnika közeledésének példája az interneten keresztül történő telefonálás. A VoIP (Voice over IP = Hangátvitel IP felett) egy módszer, amelynek segítségével telefonbeszélgetéseket bonyolíthatunk egy IP hálózaton, jóval olcsóbban. A megvalósítása nem könnyű feladat, mert az Internet Protokollt nem erre tervezték. A legnagyobb probléma a hallható hang jó minőségének a biztosítása.

### *Digitális hangátvitel*

Az emberi hang analóg természetű. Ahogy ezt már megtanultuk, digitalizáláskor az analóg hangot digitálisan továbbítható PCM formára (pulzus kód moduláció) alakítjuk. A PCM másodpercenként 8000-szer mintavételez és minden mintát egy binárisan kódolt számértékké alakít. Ez az érték a hanghullámnak a mintavételezés pillanatában mért amplitudóját reprezentálja. A szabványos telefon PCM kód 8 bites, így beszélgetésenként 64000 bit/sec (bps) sávszélességet igényel. Az a viszonylag nagy sávszélesség igény, kódolási és tömörítési módszerekkel csökkenthető.

Ha csak az egymást követő minták közötti változásokat kódoljuk, akkor az mintánként kevesebb bittel írhatjuk le. A **különbségi PCM, (rövidítve: DPCM (Differenciális PCM))** kódoláskor az előző, és az aktuális jel digitális értékének a különbségét visszük át. A változásoknak az átvitele már kisebb sávszélességet igényel.

### *ADPCM*

Az **Adaptív (=alkalmazkodó) DPCM esetén (rövidítve: ADPCM** Adaptive Differential Pulse Code Modulation**)**, az ADÓ és a VEVŐ oldalon is azonos algoritmust használva, megjósoljuk az előző minta és a pillanatnyi minta különbségét. Amit ténylegesen átküldünk a csatornán, az a jósolt és a tényleges különbség eltérése. Ezzel a módszerrel még hatékonyabb, kisebb sávszélességet igénylő átvitel lehetséges.

Példaként nézzük egy háromszög alakú jel esetén a módszereket. Az adaptív algoritmus, amit a jóslásra használunk, a következő: az előző különbség ugyanaz mint a következő.

A példából is látható a DPCM és az ADPCM hatékonysága: a három bittel kódolt PCM jelfolyam helyett, a DPCM és az ADPCM jelfolyam is mindössze két bittel kódolható. Az ADPCM szabvány: CCITT, ITU-T G.726

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bemenőjel | 1 | 2 | 3 | 4 | 5 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | 4 | PCM JELFOLYAM |
| Előző kim. | - | 1 | 2 | 3 | 4 | 5 | 6 | 5 | 4 | 3 | 2 | 1 | 2 | 3 | |
| Különbség | - | 1 | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | DPCM JELFOLYAM |
| Jósolt | - | - | 3 | 4 | 5 | 6 | 7 | 4 | 3 | 2 | 1 | 0 | 3 | 4 | |
| Hiba | - | - | 0 | 0 | 0 | 0 | -2 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | ADPCM JELFOLYAM |

Csak az első és a második minta elküldése után lehetséges a következő minta értékének a becslése. Az első és a következő jósolt mintaértékek pontosak, kivéve, amikor a jel növekedési iránya megfordul. A fenti táblázatban láthatók a leírtak.

A **hagyományos telefonrendszerekben** a PCM vagy ADPCM révén a kommunikáció idejére egy-egy szinkron digitális beszédcsatornát, azaz állandó sebességű bitfolyamot generálnak, akár beszélgetnek, akár éppen nem. Ennek azért van jelentősége, mert egy átlagos telefonbeszélgetésben számos rövid időtartamú szünet (csönd) van, aminek a hálózaton való továbbítása sávszélesség-, és pénzveszteség.

**VoIp hálózatokban** a beszélgetés adatként kerül továbbításra, azaz csak a tényleges beszédintervallumok idején generálódik adatforgalom, ami körülbelül egyharmadával csökkenti a hangkommunikációhoz szükséges sávszélességet. Természetesen, minél jobban csökkentjük a leképezéshez használt bitek számát, annál rosszabb a hangátvitel minősége. Egy másik tényező, a késleltetés is fontos szerepet játszik a hangtömörítéssel kapcsolatban. Bár a késleltetés önmagában nem rontja a hangminőséget, a nem kívánatos visszhanghatás elkerülése érdekében szükség lehet visszhangelnyomásra, valamint a változó késleltetésből (jitter) származó zavarok kiszűrésére.

Összegezve: a VoIP alkalmazása két módon is hatékonyabbá teszi a hálózat használatát: egyrészt **csökkenti a hangátvitelhez szükséges sávszélességet**, másrészt a **beszéd közbeni szüneteket** (a hálózat terhelése szempontjából) **kiszűri**. Az előnyöknek a kihasználásához azonban a továbbító hálózatnak képesnek kell lennie az alacsony sávszélességű kommunikációs kapcsolatok kiszolgálására, valamint arra, hogy a szünetek idejére megkímélt sávszélességet a vele egyidőben zajló más hangkommunikáció rendelkezésére tudja bocsátani.

### *VoIP megvalósítása*

Az IP telefon két megvalósítása létezhet: egy multimédiás PC-n futó szoftver, vagy egy különálló készülék formájában. Az összeköttetések (vagyis a telefonhasználat) megvalósításhoz az IP telefon gateway szükséges. Az alapfunkcióját könnyű definiálni: az eszköz átalakítja a „tradicionális" analóg hang adatot digitális adattá és az IP hálózaton keresztül, mint pl. az Internet, elküldi egy másik gatewaynek, és az visszaalakítja analóg adattá, és tovább küldi egy telefonra.

A VoIP jellegéből adódóan ez a fajta telefon már nem a pont-pont kapcsolaton alapuló klasszikus megoldás.

Mivel a rendszer nyitott szabványokra épül, kész új alkalmazások integrálására. Ezek után az is elképzelhető, hogy a jövőben az IP telefonunkról szöveg-beszéd átalakító segítségével lehallgassuk az e-mail-jeinket. Továbbá lehetővé válik olyan központok kialakítása, ahol az e-mail, telefon hang és fax üzeneteinket egységesen kezelik.

A már működő megoldások: hívásátirányítás, a call-center-ekre jellemző hívások sorbaállítása, és priorítás szerinti kiszolgálása, több résztvevős konferenciabeszélgetések mind a rendszer alapszolgáltatásait fogják képezni.

### *H.323 és a SIP*

Ha IP fölötti telefóniáról beszélünk, akkor napjainkban két szabvány harcol egymással.

- Az egyik a távközlési iparból jövő, és az ITU által jóváhagyott H.323 keretajánlás, amely egyéb szabványokkal kiegészítve határozza meg, hogy milyen rendszerelemekből, milyen vezérlési üzenetekkel és milyen folyamatokkal építhető ki multimédia kommunikáció nem garantált minőségű csomagkapcsolt hálózatokon.
- A másik az Internetes közösség által kidolgozott, és IETF, az Internet specifikációs szervezet által kidolgozott ajánlás multimédia kommunikáció kiépítésére, és lebontására. Az ajánlás egy kapcsolatkezdeményező protokoll, a neve: **Session Initiation Protocol**, röviden SIP. Ez egy egyszerű, szöveg alapú kérés-felelet protokoll. SIP esetében az ügyfelek híváskezdeményezést, a kiszolgálók pedig a hívások fogadását végzik (ügyfél-kiszolgáló felépítés). Hasonlóan, mint a GSM mobil hálózatokban a SIP-telefon bejelentkezik a hálózatba, és az Internet-telefonhálózat regisztrálja azt az Internet csatlakozást, amelyről a telefon bejelentkezett. Az előzetesen kiválasztott SIP- telefonszolgáltatónál a regisztrálás teljesen automatikusan történik.

Mind a SIP és mind H.323 helymeghatározási és híváskezelési szolgáltatást nyújt a felhasználók részére. A kapcsolat kezelését, és multimédia adatfolyamok továbbítását különválasztva kezelik: az utóbbi megvalósítását az IETF **specifikálta valósidejű**

**átviteli protokoll (Real-Time Protocol = RTP)** valósítja meg mindkét megoldásban.

**Melyik a jobb?** H.323 erőssége a távközlési világ szabványosítási múltjában rejlik: a következetes, pontos ITU szabványok, és jól kidolgozott átjárhatóság a hagyományos telefonhálózatok irányában. A SIP az internetes környezetével, a gyors, változásokra képes alkalmazkodó képességével, a jelen állás szerint ki fogja szorítani a H.323 megoldást.

*(A szerző dilemmája: Lehet hogy erről a részről a megoldás robbanásszerű elterjedése miatt többet kellett volna írni???)*

## Ellenőrző kérdések: 7. Fejezet

?

1. Milyen rétegei vannak az Internet hálózatnak, és ez hogyan viszonyul az OSI modellhez? Mi az egyes rétegek feladata?
2. Mi az RFC?
3. Mi az IP protokoll feladata? Milyen információkat tartalmaz a IP csomag és a fejléce?
4. Ismertesse az Internet címzési rendszerét! Hogyan épülnek fel a címek? Mik azok az A, B, C osztályú címek? Hogyan szokták ábrázolni?
5. Mi a cím maszkok feladata? Adjon egy konkrét példát is a használatára!
6. Hogyan működik az információátvitel a hálózatok között? Mi az a átjáró (gateway)?
7. Mi a belső hálózati címek szerepe?
8. Mik azok az alhálózatok, és hogyan hozzuk ezeket létre? Mik a legfőbb szabályok?
9. Mi a DNS? Magyarázza el a domén nevek rendszerét! Miért monjuk, hogy hierarchikus felépítésű? Mi a hierarchia tetején lévő csúcsszintű domének jelentősége?
10. Mik azok a zónák? Mi a delegálás? Mi a FQDN? Hogy működik a névfeloldás?
11. Hogyan működik az IP útválasztása? Mi az alapértelmezett átjáró?
12. Hogyan határozzuk meg az útvonalat, a routing tábla segítségével?
13. Foglalja össze az új IP protokollal, az IPV6-al kapcsolatos legfontosabb ismereteket!
14. Mit jelent a mobil kommunikáció? Foglalja össze a mobil kommunikációval kapcsolatos legfontosabb ismereteket!
15. Mik azok a portok? Mit jelent az, hogy egy port „jól ismert"? Hogyan kezelik ezt a TCP/IP protokollban?
16. Mi a TCP protokoll feladata? Milyen információkat tartalmaz a TCP csomag?
17. Mi az a háromutas kézfogás?
18. Mi az UDP protokoll feladata? Hasonlítsa össze a TCP-vel!
19. Milyen célt szolgál az ICMP protokoll?
20. Milyen célt szolgál az ARP protokoll? És a RARP? Magyarázza el működésüket! Mit jelent egy csomag „beburkolása"?
21. Foglalja össze egy üzenetváltás fő lépéseit két gép között, kitérve a keretekre és csomagokra!
22. Hogyan működik a DHCP protokoll?
23. Mi a tűzfal? Mi az a NAT és hogyan lehet az Internet hozzáférést megosztani?

24. Mutassa be a rádiós hálózatokat! Mi az AD-HOC, illetve infrastruktúra üzemmód? Mi az Access Point? Hogyan oldják meg a biztonsági problémákat?
25. Pár mondatban foglalja össze a következő hálózati programok feladatát: ARP, IPCONFIG, PING, TRACERT, PATHPING, NETSTAT, NSLOOKUP, ROUTE. Szerezzen bővebb információt az XP helpje segítségével!
26. Mi az Ethereal program feladata?
27. Hogyan tudunk az internetre kapcsolódni? Hogyan működik egy internet szolgáltató?
28. Ismertesse a SLIP és PPP protokollok feladatát!
29. Mire szolgál, és milyen részeket tartalmaz egy broadband router?
30. Milyen internet szolgáltatásokat használhat egy felhasználó? Melyik nem igényel állandó hálózati kapcsolatot?
31. Mutassa be az Email legfontosabb jellemzőit! Mi az SMTP, POP3 és IMAP? Milyen funkciókat kell megvalósítani egy levelező programnak?
32. Mi az UUENCODE/UUDECODE eljárás? Miért használjuk? És mi az a MIME?
33. Mi az a mail szerver? Mik azok a smiley-ik?
34. Mik azok a levelezési listák? Mi a digest és a FAQ (GYIK) ?
35. Mi az a TELNET? Hogyan működik? Milyen biztonsági problémákat okozhat?
36. Mutassa be az FTP alkalmazás legfontosabb jellemzőit! Hogyan működik? Mi az anonymous FTP? Melyik programmal tudok létrehozni FTP kapcsolatot? Mi az Archie?
37. Hogyan lehet az információkat szervezni a hálózaton? Mi a GOPHER?
38. Mi a WWW? Mi a HTML, URL?
39. Mi a hipertext és mi a hipermédia?
40. Mit jelent, és milyen célt szolgál a CGI?
41. Mi az a HTTP protokoll? Hogyan működik?
42. Mi az a cookie?
43. Mi a proxi szerver?
44. Sorolja fel a hálózati etikett (netikett) néhány alapszabályát!
45. Ismertesse a digitális hangátvitel alapjait! Hogyan lehet az átvieendő hanginformáció igényelte sávszélességet csökkenteni?
46. Mi az ADPCM tömörítés lényege?
47. Mi az a VoIP?

# 8. HELYI HÁLÓZATOK, INTRANET

*Te nagyon jól játszottál, én milyen voltam? (Moldova György)*

A számítógép hálózatok egyik legdinamikusabban fejlődő területe a kisebb helyi hálózatok, azaz a **lokális hálózatok. (Local Area Network – LAN).** Képes megoldani egy szervezet egységes, gyors, szervezett, ügyviteli, irányítási, raktározási, vagy akár munkaszervezési feladatait. Milyen előnyei vannak a számítógépek ilyen módon történő összekapcsolásának?

- Hatékonyabban lehet felhasználni a rendszer erőforrásait; nem kell minden programot és adatot egy gépen tartani a munkához, az adatokhoz, amennyiben ez szükséges mások is hozzáférnek.
- A perifériák száma is csökkenthető: közös nyomtatók, közös CD meghajtók is használhatók.
- A fentiek mellett ma már a hálózat a munkatársak közötti hatékony kommunikáció eszköze is, levelezésre, közös adatbázisok és egyéb információk kezelésére is felhasználható.

Az Internet megjelenésével a LAN-ok összekapcsolása is megvalósult, kitágítva a felhasználható lehetőségek körét.

A lokális hálózatok egyre jobban egy működő szervezet hatékony működésének a meghatározójává váltak. Ezért a megfelelő működés biztosítására három igen fontos problémára kellett megoldást találni.

1. A szervezet egységeit, illetve kapcsolatukat hagyományosan a fájlrendszer is leképezte. (minden főosztályhoz egy, a gyökérben lévő alkönyvtár tartozott, illetve az ez alatti könyvtárakban voltak a főosztályhoz tartozó osztályok adatai.
2. Meg kellett oldani a rendszerben lévő gépek távolról történő karbantarthatóságát
3. A bonyolultabb hálózatkialkítás miatt nem voltak kötelezhetők a felhasználók arra, hogy ismerjék a rendszer felépítését: hol milyen szerver, nyomtató stb. áll rendelkezésre.

Jelenleg a három leggyakoribb hálózati operációs rendszer: **Windows, Unix/Linux, és a Novell Netware.** Mivel hasonló feladatokra tervezték a rendszereket, ezért nagyon sok hasonlóság található az operációs rendszerekben, és a megoldásokat az előbbiekben felsoroltakra mindegyik rendszerben megtaláljuk.

A könyv előző fejezeteiben már a LAN-ok több összetevőjét bemutattuk, ezért csak a további, kiegészítő ismereteket foglaljuk össze, elsődlegesen a szoftver összetevőkre kitérve.

## 8.1 LAN az OSI modell alapján: MAP ÉS TOP

Az IEEE 802 szabvány csak a hálózati rétegig szabványosítja a LAN hálózatokat. Azért, mert a három szabványban a közeghozzáférési módszerek eltérnek, nem célszerű a teljes felépítést különállóan kezelni. Ez vezetett két, a szabványokon alakuló protokoll:

- A valós idejű működést követelő **MAP (Manufacturing Automation Protocol** — gyártásautomatizálási protokoll),

- Az ilyen igényt nem támasztó **TOP (Technical and Office Protocol** — technikai és hivatali protokoll) irodaautomatizálásra szánt megoldás

kifejlesztéséhez. [1] Bár a MAP és a TOP az alsó rétegekben különböző, a felső rétegekben teljesen kompatibilesek, azonos protokollokat használnak.

| | MAP | | | TOP | | |
|---|---|---|---|---|---|---|
| 7 | FTAM | DS | MNS | FTAM | DS | MNS |
| 6 | OSI megjelenítési protokoll (8823) | | | OSI megjelenítési protokoll (8823) | | |
| 5 | OSI viszonyprotokoll (8327) | | | OSI viszonyprotokoll (8327) | | |
| 4 | 4-es osztályú összeköttetés alapú szállítás (8073) | | | 4-es osztályú összeköttetés alapú szállítás (8073) | | |
| 3 | Összeköttetésmentes kapcsolat (8473) | | | Összeköttetésmentes kapcsolat (8473) | | |
| 2 | Logikai kapcsolat-vezérés (LLC) (8802/2) | | | Logikai kapcsolat-vezérés (LLC) (8802/2) | | |
| 1 | Vezérjeles sin (8802/4) | | | Ethernet (8802/3) | Vezérjeles gyűrű (8802/5) | |

*8-1. ábra: A MAP és a TOP felépítése*

Az első két szintről már írtunk, nézzük a felsőbb rétegeket. Az összeköttetés-mentes hálózati szintű protokolljuk az ISO 8473-as. Ez nagyon hasonló az IP protokollhoz, de nagymértékben eltér az X.25-től, azaz a datagramos megközelítést választották.

A szállítási rétegnek az ISO 8072/8073 protokollt használják. Ez a réteg saját maga kezeli a forgalomszabályozást és a hibavédelmet.

A 4-es osztály azt jelzi, hogy a megbízhatatlan szállítási réteg esetén is megfelelően fog működni. (Az X.25-ös szállítási réteg esetén fölötte alacsonyabb osztályba tartozó szállítási protokollt lehetne használni.)

A viszony, a megjelenítési és az alkalmazási rétegeiben szintén ISO szabványú megoldások találhatók (állománytovábbítás, virtuális terminál).

Az olyan protokollgyűjteményt, amely minden rétegben csak egyetlen protokollt tartalmaz, **protokollkészletnek (protocol suite)** vagy más néven **protokoll-veremnek** (protocol stack) nevezünk. Ilyen például a MAP, TOP, illetve az Internet TCP/IP protokoll készlete.

## 8.1.1 LAN-ok összekapcsolása

A TOP hálózatok ötféle fizikai eszközt használnak az összekötetések megvalósítására: a hosztokat, jelismétlőket, hidakat, útválasztókat (routereket) és átjárókat (gateway).

A **hosztok** lényegében az információ forrásai és céljai.

**Jelismétlőkről (repeater), jelelosztókról (HUB)** már az Ethernet hálózat kapcsán írtunk, feladatuk a jelregenárálás, bitek továbbítása az egyik hálózatból a másikba.


| Több hálózat összekapcsolásánál alkalmazott eszközök | | | |
|---|---|---|---|
| | Összekötő OSI réteg | | |
| | Fizikai | Adatkapcsolati | Hálózati |
| Jelismétlő (repeater) | azonos | azonos | azonos |
| Hálózati híd (bridge) | eltérő is lehet | azonos | azonos |
| Forgalomirány ító (router) | eltérő is lehet | eltérő is lehet | azonos |
| Hálózati átjáró (gateway) | eltérő is lehet | eltérő is lehet | eltérő is lehet |

*8-1. ábra LAN-ok összekapcsolása*

A **híd (bridge)** két hálózat adatkapcsolati szintű összekapcsolását is végezheti. Egy Ethernet és egy vezérjeles sínű hálózat között a híd teremti meg a kapcsolatot. Lényegében egymásba átalakítja az eltérő keretformátumokat.

**Útválasztókat (router)** akkor kell alkalmazni, ha az összekötendő hálózatok különböző hálózati, de azonos szállítási réteggel rendelkeznek. Pl. Ethernet és X.25-ös hálózat között útválasztó alakítja az Ethernet kereteket X.25-ös keretekké.

**Átjárókat (gateway)** akkor használnak, ha olyan hálózathoz csatlakoznak, amely felépítése nem követi az OSI modellt.

### *FDDI*

A ma már elavuló FDDI alapgondolata: üvegszál alapú lokális hálózat, ha drágán is, de kialakítható. Az FDDI multimódusú üvegszálakat használ olcsóbb volta és kisebb veszélyessége (nem lézer-fény, csak LED) miatt.

Az FDDI két optikai szálas gyűrűből áll, amelyekben az adatforgalom ellentétes irányú. Ha az egyik meghibásodik, a másikon az adatforgalom tovább folyik. Ha mindkettő ugyanazon a ponton szakad meg, akkor a két gyűrű egyetlen dupla hosszú gyűrűvé alakítható. Minden állomás olyan relékkel van felszerelve, amelyek a gyűrűk összekapcsolására, és a meghibásodott állomások kiiktatására használhatók. Az alap FDDI protokoll modellje a 802.5 protokollon alapszik.

## *8.2 Lokális hálózatok fizikai egységei*

A lokális hálózati kommunikáció feladatait az erre a célra tervezett hardver és firmware (ROM-ba égetett, a kártyán lévő program) végzi. Személyi számítógépekből álló hálózatokban használt fizikai eszközök a következők:

### Adapterkártya

Ezt a speciális perifériakártyaként kapható eszközt, az adapterkártyát, a hálózat állomásaként használni kívánt valamennyi személyi számítógépbe beépítik. Az adapterkártya tartalmazza a logikai kapcsolatvezérlést, és a közeghozzáférés vezérlő funkciókat megvalósító áramköri részt. Mivel a működtető áramkörét egy tokba integrálták, az ma már minden alaplapon megtalálható, általában szüségtelenné téve a különálló adapterkártya használatát.

### Kábelrendszer

A kábelrendszer azt a kábelt, ill. vezetéket jelenti, amelyet a hálózatban lévő eszközök összekapcsolására használnak. Általában idetartoznak még azok a csatlakozószerelvények is, amelyek lehetővé teszik, hogy az eszközök a kábelre csatlakozzanak. A legtöbb lokális hálózatnál használt alapvető vezetékválaszték a következő: sodrott érpár kötegből álló kábel, koaxiális kábel és a fénykábel.

### Elosztók, erősítők

Egyes lokális hálózati kialakítások koncentrátorokat, ill. hozzáférési egységeket használnak, hogy a hálózati jelek erősítése és elosztása megoldására, illetve a hálózatban levő eszközök egy központi helyen kerüljenek összeköttetésre. Ezeket szokták HUB-oknak, jelismétlőknek is nevezni.

**Média konverterek**

Ezek az eszközök különféle fizikai hálózatok összekapcsolását teszik lehetővé, az adatátviteli közegek összekapcsolásával. Elsődlegesen a jelek elektromos illesztését biztosítják a közegek között.

Például egy **10Base-T(RJ-45)/10Base-2(BNC)** átalakító egy 10 Mbit/sec-os csavart érpárat használó hálózatrész és egy BNC csatlakozós vékony koax hálózat összekapcsolását teszi lehetővé. Léteznek még 10Base-T(RJ-45)/10Base-FL(SC) illetve 10/100Base-TX(RJ-45)/100Base(FX) stb. eszközök is.

## 8.3 Gépek logikai összekapcsolása

A lokális hálózatokban használt számítógépeket - a hálózati funkciójukat tekintve - két csoportba sorolhatjuk:

- az információt felhasználó **munkaállomások**,
- információt szolgáltató számítógépek az ún. **szerverek.**

Természetesen ez a két kategória — ahogy ezt a későbbiekben is látni fogjuk — fizikailag nem válik élesen ketté.

A hálózatba kötött számítógépek kapcsolatának kialakítása után a gépek képesek egymással kommunikálni, erőforrásaikat egymással megosztani. Amennyiben a központi erőforrások megosztása az elsődleges cél, akkor **ügyfél-kiszolgáló kapcsolatot** valósítunk meg, ha a kölcsönös kommunikáció is fontos szerepet játszik, akkor törekszünk az **egyenrangú kapcsolat** kialakítására.

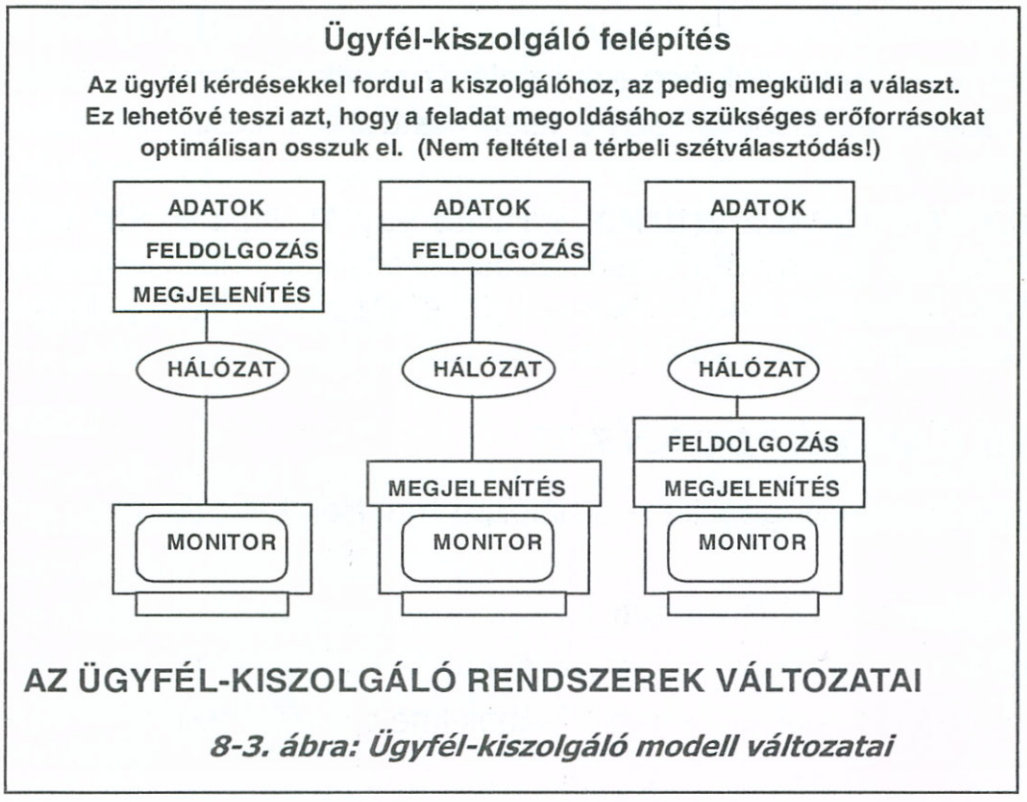
### 8.3.1 Ügyfél-kiszolgáló kapcsolat

Az első esetben van egy kitüntetett, általában a hálózatba kapcsolt gépeknél nagyobb teljesítményű gép (a szerver) amelynek feladata a többi gépről (kliensektől) érkező kérések kiszolgálása. Ezt a kialakítást **kliens-szerver**, magyarul **ügyfél-kiszolgáló** modellnek nevezik.

**Valójában, mind a szerver- mind a ügyfél (kliens) funkciók a gépeken futó programok formájában jelennek meg, amelyek a gépek közötti összeköttetést kihasználva végzik a munkájukat.** Természetesen az ügyfél-kiszolgáló modellnek több, minőségileg más kialakítása lehetséges, attól függően, hogy egy adott feladat mekkora, és milyen részét hajtja végre a kliens-, illetve a szerver program, hogyan osztják el a közös munkát.

Példaként gondoljunk egy szerveren elhelyezett adatbázisban történő keresésre! A legegyszerűbb esetben a kliens gép egy „buta" (dumb) terminál, amely egy együttes adatbeviteli és megjelenítő egység: a billentyűzetén begépelt adatokat átküldi pl. soros vonalon a szerver gépre, az ott futó program a begépelt parancsok alapján a keresést végrehajtja, és a keresett rekordokat visszaküldve a soros vonalon, a terminál azokat megjeleníti. Szokták ezt a felépítést elkülönítve is kezelni, és **hoszt – terminal** kapcsolatnak hívják.

A klasszikus nagyszámítógépes környezetben a felhasználók ilyen módon tudták a gépet használni, példaként a kapcsolódás módját az IBM BISYNC protokoll ismerteté-




*8-3. ábra: Ügyfél-kiszolgáló modell változatai*

sén keresztül már bemutattuk. Nagy előnye, hogy terminálóknak nem kell korszerű, nagyteljesítményű gépeknek lenniük, ezért a Windows is támogatja a terminál szerveres kialakítást.

Egy másik lehetőség lehet az, hogy a kikeresett adatokat a szerver csak „ömlesztve", nyers formában küldi vissza a kliensnek, ahol a futó program megfelelő formában megjeleníti.

A harmadik esetben a kereső program a kliens gépen fut: a keresés végrehajtásához szükséges adatbázis rekordokat a vonalon a szerver elküldi a kliensnek. Az, a leküldött részen végrehajtja a keresést, majd ezt a továbbiakban leküldött részekkel is folytatja. Egy rendszerben természetesen egynél több szerver is elképzelhető.

**Dedikált szerver**-en kizárólag a szerverfunkciókat megvalósító program fut, és felhasználói munkára nem alkalmas, ellentétben a **nem dedikált szerver**-rel, ahol a kiszolgáló alkalmas a szerverfunkciók ellátására is. (Ez utóbbi megoldást ma már ritkán, esetleg fejlesztésre használják.)

A kiszolgálókat a végzett tevékenységük alapján a következőképpen csoportosíthajuk:

- **Beléptető, vagy hitelesítő kiszolgáló:** Adatbázisban tárolja a hálózati erőforrások és felhasználók jogosultságait, és kérésre ezt szolgáltatja.
- **Fájl vagy nyomtató kiszolgáló:** A fájlszerver tárolja, és jogosultságtól függően felkínálja a tárolt állományait a felhasználóknak. A nyomtató szerver kezeli a hálózatban lévő nyomtatók használatát, és az alkalmazások nyomtatókra való várakozása miatt kialakuló nyomtatási sorokat.
- **Alkalmazás kiszolgáló:** A szerveren található különféle alkalmazások felhasználók által történő alkalmazását kezeli.

*Ismételten felhívjuk a figyelmet arra, hogy a szerverfunkciókat programok valósítják meg, és ezek a programok akár egy közös számítógépen is futhatnak!*

Például a Windowst futtató rendszerekben lévő IIS (Internet Information Services) tartalmazza a szerverprogramként használható Network News Transfer protokollt (NNTP), a File Transfer protokollt (FTP) és a Simple Mail Transfer protokollt (SMTP).

Mivel a kliens – és szerver szerepek ilyen módon változtathatók, a szélső esetben eljuthatunk oda, hogy a két szerep nem válik szét: a gépek egyenrangú szerepet játszanak a kapcsolatban.

## 8.3.2 Egyenrangú kapcsolat

Az eszközök összekapcsolhatók a demokrácia szabályai alapján: minden gép egyenrangú, és erőforrásainak egy részét bocsátja a hálózaton keresztül a többi gép számára. Ezek az **egyenrangú**, vagy más néven **peer-to-peer** (röviden: P2P) hálózatok.

A „peer-to-peer" szó jelentése tehát „egyenrangútól egyenrangúhoz", vagyis egy olyan hálózat, amelyben az összes csatlakozó - peer - teljesen egyenrangú, vagyis mindegyik ugyanazokkal a jogokkal és felelősséggel rendelkezhetnek, és az erőforrásokat, adatokat, programokat stb., egymás számára kölcsönösen elérhetővé tehetik, közösen használják, kicserélhetik, és ehhez nincs szükség központi szerverre.

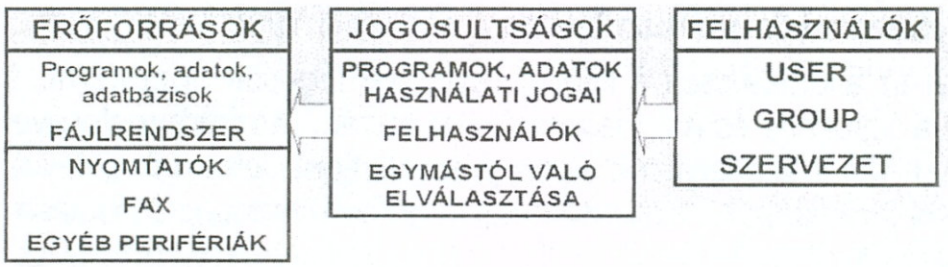
- Minden peer egyenrangú és a hálózatban egyszerre működhet szerverként és kliensként, azaz egyszerre bocsáthat erőforrásokat a többi rendelkezésére és használhatja a többi peer erőforrásait, mindegyik adatokat küldhet, tárolhat és fogadhat.
- Mindegyik peer megtartja az ellenőrzést a saját gépe felett, vagyis önállóak maradnak, maguk határozzák meg, hogy mikor, kivel és mit cserélnek ki, illetve, hogy mikor csatlakoznak a hálózathoz.
- A peereknek nem kell állandóan csatlakoztatva lenniük.
- A peerek közti kommunikáció közvetlen, nem halad át valamilyen szerveren, amely lassítaná azt, minden tevékenység a hálózat végpontjaiban történik.
- Az eddigiekből következik, hogy a hálózat dinamikus felépítésű, így különböző időpontokban eltérő lehet a felépítése.

Windows környezetben az egyenrangú kapcsolat a **megosztás.**

## 8.4 Lokális hálózati operációs rendszerek

Szokásos ezt **NOS = (Network Operating System)** betűszóval is rövidíteni.

Az OSI-modell két legalacsonyabb szintjének funkcióit már szabványosították, és a legtöbb LAN összhangban van a három IEEE-szabvány valamelyikével.




LÉNYEGÉBEN EGY MULTITASZKINGOS (TÖBB FELADATOT PÁRHUZAMOSAN FUTTATÓ) RENDSZER AHOL A FOLYAMATOK SZÉTVÁLASZTÁSA A FELHASZNÁLÓK SZERINT IS MEGTÖRTÉNIK.

A HÁROM MEGHATÁROZÓ HÁLÓZATI OPERÁCIÓS RENDSZER:

UNIX(LINUX), NOVELL-NETWARE ÉS A WINDOWS.

*8-4. ábra LAN operációs rendszerek*

Bár az adatkapcsolati szint feletti funkciók közös jellemzőit már kidolgozták, de a hálózati és az e fölötti rétegek esetében a szabványok még nem állnak azon a szinten, mint az adatkapcsolati és fizikai rétegek esetén.

A hálózatba kapcsolás az információk közös kezelését biztosítja, azonban alapvető

kérdés a nem közös információk védelme. Ezeket a hálózat részeként működő, beépített védelmi rendszer valósítja meg.

LAN eszközökkel megvalósított fizikai hálózat az állomásai számára általános adatcserére alkalmas kommunikációs szolgáltatást nyújt. Ahhoz azonban, hogy megkapjuk a lokális hálózat valamennyi előnyét, az általános kommunikációs szolgáltatáson túl további funkciókat is meg kell valósítanunk. Ezek a funkciók a hálózat magasabb szintű rétegeihez tartoznak, és ezeket a hálózati operációs rendszerként ismert programrendszer szolgáltatja.

**A hálózati operációs rendszer egy szoftver, amely a hálózatba kapcsolt eszközökön fut, és feladata az eszközök közötti kommunikációs szolgáltatások biztosítása.**

Először összefoglaljuk a lokális operációs rendszerek funkcióit, amit ma már majdnem kizárólag az internetből átemelt technológiákkal oldanak meg.

Ezután röviden áttekintjük a három meghatározó hálózati operációs rendszert. A nagyfokú hasonlóság miatt, mindegyik esetén csak egy-egy fontosnak tartott részletét emeljük ki: A Windows esetén a címtár bemutatására és a hálózat felépítésére összpontosítunk, Novell esetén a felhasználói oldalt mutatjuk be, majd röviden írunk a UNIX/LINUX hálózatkezeléséről.

## *8.4.1 Lokális hálózati operációs rendszerek funkciói*

Foglaljuk össze, hogy melyek a hálózati operációs rendszerek által kínált közismertebb funkciók:

**Fájlkiszolgáló** A nagy kapacitású lemez a legfontosabb erőforrás, amelyet a hálózat megoszthat. A fájlkiszolgáló egy olyan számítógép, amely a hálózati kapcsolatán túlmenően a nagykapacitású merevlemezes meghajtókat is kezel. A fájlkiszolgáló teszi lehetővé a munkaállomások számára a tárolt fájlokhoz való hozzáférést. Ez a fájlmegosztás, különböző módon valósítható meg.

- Megtehető könyvtárak alapján, amikor az állomás hozzáférhet egy adott könyvtárhoz és használhatja az ebben a könyvtárban található bármelyik fájlt.
- Fájl szintű megosztásnál az állomás csak a kijelölt fájlokhoz jogosult hozzáférni. A jogosultságoknak külön a fájlokhoz és a felhasználókhoz történő hozzárendelésével a hozzáféréseket finoman lehet szabályozni. Ilyen jogosultságok lehetnek a fájlok megnyitásának, módosíthatóságának, írhatóságának, létrehozásának, másolásának, törlésének engedélyezése.

**Nyomtatókiszolgáló** A lokális hálózat egyik előnye, hogy képes megosztani a perifériákat, különösen az olyan drága készülékeket, mint a színes lézernyomtató és a fényszedő. A nyomtatószerver a hálózat valamennyi állomása számára lehetővé teszi, hogy másik állomáshoz tartozó nyomtatót használjon. A nyomtatási anyag pontosan úgy rendelhető a nyomtatóhoz, mintha a nyomtató közvetlenül a felhasználó saját állomására lenne kötve, valamint rendelkezésre áll egy általános sorban állási szolgáltatás, amivel a nyomtatási anyag akkor is a nyomtatóhoz rendelhető, amikor a nyomtató foglalt. Erre a megoldásra a nyomtató viszonylagos lassúsága miatt is szükség van.

**Elektronikus levelezés** Ma ez már természetes szolgáltatás.

**Hálózati névszolgáltatás** A hálózat felhasználói és az alkalmazói programok a hálózati nevek alapján kérik a hálózati operációs rendszerrel kapcsolatos szolgáltatásokat. A hálózati neveket mind a hálózati felhasználók, mind a megosztott erőforrások megelölésére használják. A hálózati névszolgáltatás a hálózati neveket hálózati címekre fordítja le, hogy a szolgáltatáskérés megvalósításához szükséges üzenetek helyesen legyenek címezhetők. Ilyen módon a hálózat erőforrásaira szimbolikus nevekkel (alias) hivatkozhatunk.

**Összekapcsolhatóság** Az összekapcsolhatóság egy általános fogalom, ami a lokális hálózaton kívüli kommunikációra utal. A hálózati operációs rendszer az összekapcsolás különböző típusait valósíthatja meg. Például a lokális hálózathoz nem tartozó személyi számítógéppel egy nagy távolságú távközlési berendezésen, például telefonvonalon keresztül, hozzáférhet a lokális hálózathoz. Ez a **távoli hozzáférés (remote access).**

**Hálózatszervezés** Bár az alacsonyabb rétegek szabványai, mint pl. az IEEE 802-es szabványok, bizonyos mértékig irányítják a hálózatszervezést, nem definiálják azonban azt részletesen, és nem foglalkoznak a magasabb rétegek komplex hálózatszervezésének követelményeivel. A hálózati operációs rendszer általában lehetőséget ad a hálózat elérésének megbízhatóságára vonatkozó szervezési szolgáltatásokkal, de a szolgáltatások pontos természete hálózatról hálózatra változhat.

## 8.4.2 Windows hálózatok

A Microsoft által kifejlesztett, összefoglalóan Windows néven ismert operációs rendszereinek a legfontosabb ismérve: a **skálázhatóság**: ez azt jelenti, hogy különféle teljesítményt igénylő feladatokra ugyanazon operációs rendszert használó, különféle megoldásokat kínál:

- Kisebb teljesítményű klienseket (vagy terminálokat).
- Nagy teljesítményű, önálló munkára is alkalmas munkaállomásokat, (Windows XP, Windows Vista).
- Sok felhasználó számára elérhető nagy teljesítményű kiszolgálókat. Windows Server 2003).

A tervezés során egy olyan megoldás mellett döntöttek, ami lehetővé teszi mind a kliens-szerver, mind az egyenrangú felépítés nyújtotta előnyök kihasználását. A gépek az erőforrások megosztásának segítségével képesek egymással is kommunikálni, erőforrásaikat kölcsönösen használni, de konfigurálhatók a gépek a hálózatban lévő nagyteljesítményű kiszolgálókkal való kapcsolat tartására. A lokális hálózatba szervezett Windows alapú gépek a hálózatban látják egymást, és csupán konfigurálás kérdése a kapcsolat mértékének a beállítása. Nagy előnyt jelent az egységesen kezelhető ablakos felhasználói felület használata.

### *NetBEUI protokoll - (NetBIOS Extended User Interface)*

Bár jelentősége csökken, de említést kell tenni a Windows-os belső hálózatok protokolljáról. Kicsi, 20-200 állomásból álló helyi hálózatok számára tervezték. Úgy gondol-

ták, hogy ezek a hálózatok átjárókon keresztül kapcsolódnának más LAN szegmensekhez és nagygépekhez.

Előnyei:
- Gyors protokoll kis helyi hálózatokon;
- Jó teljesítményt nyújt lassú hálózatban;
- Teljes mértékben önhangoló protokoll, és jó hibavédelemmel rendelkezik;
- Kevés memóriát használ;
- Nem igényel konfigurálást.

Hátrányai:
- nem működik hálózatok között (nem jut át útválasztón-routeren);
- nagy kiterjedésű hálózatokon (wan) gyenge teljesítményt nyújt.

A NETBEUI egy szállítási protokoll, amelyet a **NETBIOS** programozási felületen keresztül lehet programozni.

### *Active Directory*

A Windows szerverek központi összetevője az **Active Directory** más néven: a címtár (a továbbiakban: AD). A hálózatokban az erőforrások, felhasználók kezelhetőségét bitosító adatbázis. Címszolgáltatás, amely a hálózaton található objektumok adatait tárolja, és hozzáférhetővé teszi ezt az információt a felhasználók és a rendszergazdák számára. Lehetővé teszi, hogy a felhasználók egyszerű bejelentkezési eljárás után, hozzáférhessenek a hálózat bármely részén található, részükre engedélyezett erőforrásokhoz. A rendszergazdák részére szemléletes hierarchikus képet ad a hálózat felépítéséről, és lehetővé teszi, hogy egyetlen helyről felügyeljék az összes hálózati objektumot.

Az lényegében egy hierarchikus felépítésű adattömeg, tárolja a felhasználók, számítógépek, hálózati elemek és a rendszer felügyeletét leíró szabályok – ún. **Házirend**-ek adatait.

Két fontos összetevője: az **objektum**-ok – ezekben tárolódnak a kapcsolódó számítógépek és a felhasználói sokaság adatai, valamint az objektumokat tartalmazó **tárolók**. A tárolók és objektumok rendszere egy hierarchikus fa struktúrában jelenik meg. Az AD objektumainak elérését és azok tulajdonságainak módosítását **hozzáférési listák** (Access Control List = ACL) védik.

Az AD hitelesíti a hozzáféréseket, az AD adatbázisainak integritását (egységességét) **replikáció**-val védi. Ez azt jelenti, hogy az adatbázisokat szétosztva, több példányban és helyen tárolja.

Segítségével a számítógépek, és más, a hálózaton elérhető eszközök, erőforrások, felhasználók, azok csoportjai tartományokba szervezhetők. Ez egységként kezelt adminisztratív és biztonsági egység, amelynek adatait a **tartományvezérlő** (domain controller) tárolja. Egy munkaállomás csak egy tartomány tagja lehet, belépésekor a címtárban létrejön a tulajdonságait tartalmazó **számítógépfiók** (computer account).

Az AD-ba beléptetett számítógép hozzáfér a címtár elemeihez. Megnyitása után tallózhat a címtárban, és a szereplő erőforrásokat használhatja. Ha például egy tartomány valamelyik számítógépéhez nyomtatót telepítünk – és ezt meg is osztjuk! – az ott, automatikusan a címtárban is megjelenik.

A gép indulásakor és bejelentkezéskor önmüködően is lefuthatnak a programok, alkalmazások települhetnek. Mindezt a **csoport házirend**-be írjuk le. Induláskor egy gép letölti az őt tartalmazó tartományra vonatkozó házirendet, és az abban lévőket végrehajtja.

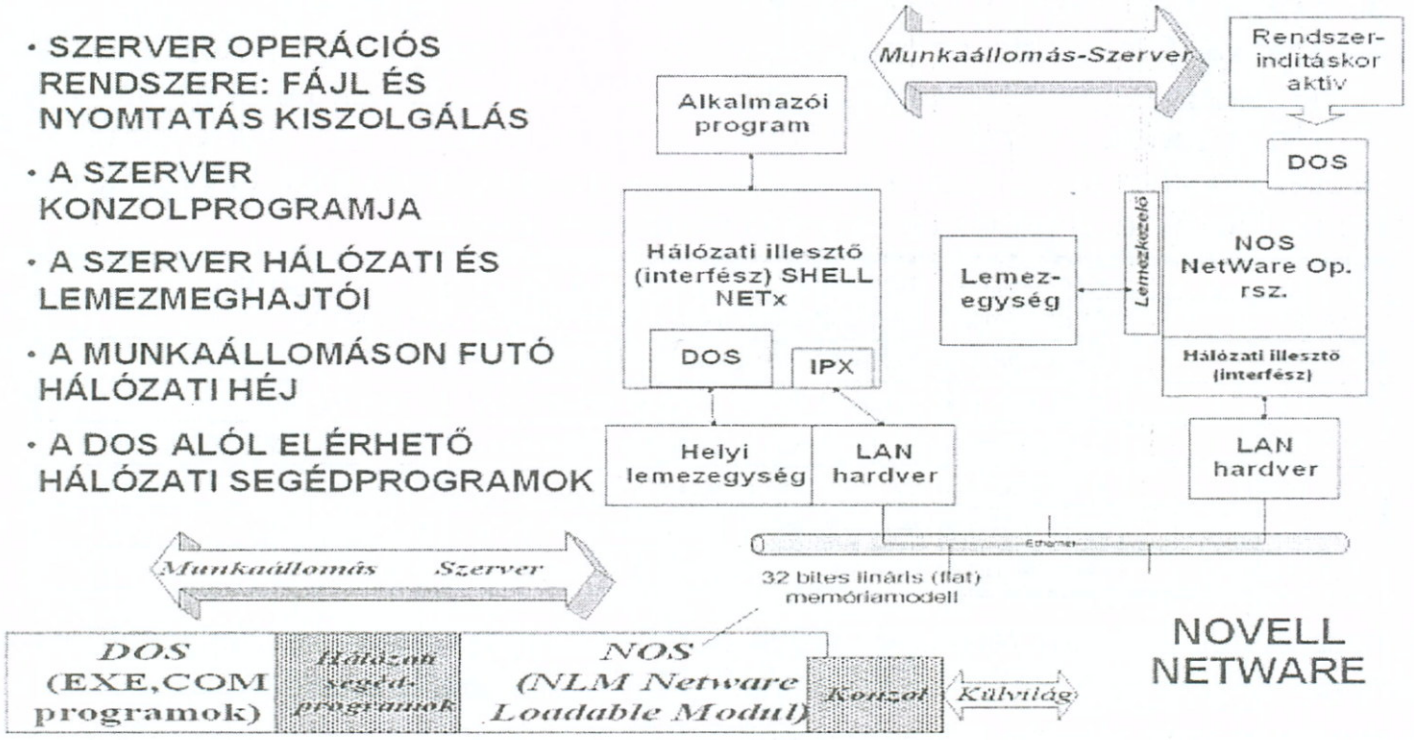
Csoportházirend tartalmazza:

- Helyi biztonsági házirendet – gép biztonsági beállításait.
- A számítógép beálításait meghatározó rendet.
- Indítás, leállás be- és kijelentkezéssel kapcsolatos szabályokat
- Alkalmazások telepítését előíró csomagokat.

A rendszergazdáknak az AD sok lehetőséget ad: Fiókok, erőforrások egységes kezelése – erőforrások közzététele – általános, más alkalmazások számára is adatokat biztosít pl. Levelezés – csoportházirend: a helyi gépek beállításai távolról változtathatók – rendszergazda munkája távolról megosztható – távtelepítési lehetőségek.

## 8.4.3 Novell Netware

A hálózati operációs rendszerek illusztrálására, egy rövid összefoglalót adunk az egyik legnépszerűbb ilyen rendszer a Novell cég PC-ken futó **Netware** operációs rendszeréről. A Novell hálózat felépítését tekintve alapvetően csillagpontos hálózat melynek központjában egy Novell Netware szerver található, a hálózatban megtalálható munkaállomások kapcsolódnak a központi kiszolgáló számítógéphez. Alapvetően fájl-, és




*8-5. ábra Novell Netware NOS felépítése*

nyomtatószerver szolgáltatást nyújt. A munkaállomások általában önmagukban is használhatók, és rajtuk a DOS (illetve WINDOWS) operációs rendszer fut.

A szerver operációs rendszere a nagyobb teljesítményt nyújtó, többfeladatúságot biz-

tosító **Netware Operációs Rendszer (NOS)**, a hálózathoz nagyteljesítményű hálózati kártyával csatlakozik.

A munkaállomás hálózati működését a gépbe helyezett és installált hálózati kártya valamint a DOS alatt futó két program (a hálózati kártyától függő IPX.COM és a DOS-ba beépülő NETX.EXE) biztosítja. A kapcsolati rendszert megvalósító programok rezidensként 40-60 kbájt helyet foglalnak el a munkaállomás memóriájában. Windows-os környezetben egy Netware ügyfél programot kell telepíteni.

E programok lefuttatása után a szerverrel való kapcsolat már kiépült. Már létezik a fájlszervert jelentő (általában F:) meghajtó, ahol a hálózatba történő be- és kilépést segítő **LOGIN, LOGOUT** és **SLIST** programok vannak és ezek már használhatók.

A szerverek merevlemezegységei kötetekre (**VOLUME**) vannak osztva. A köteteken belül az alkönyvtárak ugyanúgy helyezkednek el mint a DOS rendszerben.

Ezért az útvonal megadása a DOS-ban megszokotthoz hasonló, de kiegészül a fájlszerver és a rajta lévő kötetek nevével:

**SZERVERNÉV/KÖTET: alkönyvtárak\...\ fájlnév.kit**

A DOS szempontjából vizsgálva a dolgokat, a kötetekben lévő tetszőleges alkönyvtáraknak adhatunk logikai meghajtó nevet, és attól kezdve az alkönyvtár, mint a kinevezett meghajtó gyökérkönyvtára címezhető (hasonlóan a DOS SUBST parancsához).

Ez a hozzárendelés a MAP hálózati paranccsal tehető meg, azaz tetszőleges alkönyvtárhoz egy meghajtó nevet rendelhetünk hozzá, és a továbbiakban a teljes útvonal megadása helyett csupán e meghajtó alatti útvonalat kell megadnunk.

Mivel számos felhasználó osztozik a rendszer erőforrásain, ezért biztosítani kell a rendszer egyes részei eléréseinek a védelmét is. Ezen védelmek közül csupán egyik lehetőség a **jogok** használata. A könyvtárak és az azokban lévő fájlok elérését és kezelését jogokhoz kötik. Ezek a jogok egy 8 elemű kétállapotú [RWCEMFAS] jelölésű vektorral írhatók le. A jelölésben szereplő rövidítések jelentése:

| | |
|---|---|
| R(ead) | A felhasználó megnyithatja és olvashat a könyvtárban lévő fájlokból. |
| W(rite) | A felhasználó megnyithat és írhat a könyvtárban lévő fájlokba. |
| C(reate) | Joga van a könyvtárban fájlokat létrehozni. Lezárás után W jog kell újraírni! |
| E(rase) | Joga van a könyvtárat illetve a könyvtárban lévő fájlokból törölni. |
| M(odify) | A felhasználónak joga van a könyvtár fájljainak attribútumát változtatni. |
| F(ile Scan) | A felhasználónak joga van keresni a könyvtár fájljai között. |
| A(ccess Control) | Joga van a kezelői jogokat a könyvtár alkönyvtáraira átörökíteni. |
| S(upervisory) | Összes jog biztosított, és átadható a felhasználónak. |

A használatot biztosító tényleges jogok a felhasználónak adott kezelői jogok, és a könyvtár örökölt jogmaszkjának eredőjéből (ÉS kapcsolatából) adódnak.

A rendszergazda minden felhasználónak ad kezelői jogokat és a felhasználók által elérhető könyvtárak jogait is meghatározza.

A könyvtárak alatt létrehozott új alkönyvtárak a felettük lévő könyvtár jogait öröklik (örökölt jogmaszk).

**Tényleges jogok = kezelői jogok ÉS könyvtár örökölt jogmaszk**

**Effective rights = Trustee rights & inherited right mask**

A hálózatba történő bejelentkezést és kijelentkezést végző programok:

**SLIST** Listázza az elérhető szervereket.

**LOGIN [szervernév/] [felhasználónév]** Bejelentkezik a megadott szerverre a megadott felhasználói névvel.

**LOGOUT[szervernév]** A felhasználó kijelentkezését biztosítja a megadott szerverről.

Természetesen ez a rendszer nagyon rövid összefoglalása, bemutatása, részletes ismeretek a [6,7,8] irodalmakban találhatók.

A fejlesztések során a kommunikáció a Netware-ben TCP/IP alapúvá vált és megjelent az **NDS (Netware Directory Service)**.

Ezt megelőzően, a **bindery elv** szerint minden szerver kizárólag a saját erőforrásait, és az azokhoz tartozó hozzáférési jogokat tartotta nyilván. Így minden egyes szerveren külön kellett tárolni a felhasználói adatokat. Az egyes felhasználóknak minden egyes szerveren külön kell létrehozni a jogokat adó jelszavas belépési lehetőséget, ami több szervert tartalmazó hálózatban nehézkes. Adminisztrációs nehézségeket jelent nagyobb hálózatok esetén az is, hogy az egyes felhasználóknak több neve is lehet, számos szerveren szétosztva. A felhasználók számára az igazi nehézség az egységes, könnyen áttekinthető hálózati környezet kialakítása.

Az NDS nagyon hasonlit az Windows AD-hez, mert hálózat- és objektumorientált: a hálózatban a konkrét felépítést nevek használatával eltakaró elérhető erőforrásokat, objektumokat kezelünk, a hálózat fizikai felépítésnek ismeretét feltételező különálló szerverekre való hivatkozások megszűnnek. Ez lehetővé teszi a fájlrendszer és a hálózatot használó szervezet logikai felépítésének különválasztását.

Nincs külön rendszergazda, csak rendszerfelügyelő, akinek a munkáját szintén lehet ellenőrizni, ez az **Átvilágítás (security audit)**: minden hálózati adminisztratív tevékenység naplózódik, ezt csak lekérdezni lehet, törléséhez speciális jogosultság kell.

## 8.4.4 UNIX/LINUX hálózatkezelése

Az először itt megjelenő TCP/IP protokollt egy primitívhalmazon keresztül lehet elérni, amelyeket rendszerhívásokként valósítottak meg (implementáltak). A hívásokkal érheti el a felhasználó a szállítási szolgálatokat. A főbb rendszerhívásokat a következő táblázatban soroltuk fel [1]:

| NÉV | FUNKCIÓ |
|---|---|
| Socket | Létrehoz egy adott típusú TSAP-ot |
| Bind | ASCII nevet rendel egy korábban létrehozott sockethez |
| Listen | Létrehoz sort, amely a bejövő összeköttetés-kéréseket tárolja |
| Accept | Eltávolít a sorból, vagy vár egy összeköttetés-kérést |
| Connect | Összeköttetést kezdeményez egy távoli sockettel |
| Shutdown | Lezárja az összeköttetést a socketen |
| Send | Üzenetet küld el egy socketen keresztül |
| Recv | Üzenetet vesz egy adott végponton |
| Select | Megvizsgál egy sockethalmazt, hogy készek-e olvasásra vagy írásra |

A szolgálatinterfész központi jelentőségű fogalma a **socket (foglalat)**, amely hasonló az OSI TSAP-jához. A socketek végpontok, amelyekhez alulról (az operációs rendszer felől) az összeköttetések, míg felülről (a felhasználó felől) a folyamatok kapcsolódnak.

A socket rendszerhívás létrehoz egy socketet (egy operációs rendszeren belüli adatstruktúrát): a hívások

paraméterei kijelölik a címformátumot (pl. egy Internet nevet), a socket típust (pl. összeköttetés-alapú vagy összeköttetés-mentes), valamint a protokollt (pl. TCP/IP).

Miután egy socket már létrejött, a bejövő összeköttetés kérések tárolásához puffer allokálható. Ezt a **listen** hívással lehet végrehajtani. Egy listen hívásban megadott socket passzív végponttá válik, amely a kívülről hozzá érkező összeköttetés kérésekre várakozik.

Azért, hogy egy távoli felhasználó összeköttetéskérést küldhessen egy socketnek, a socketeknek névvel (TSAP címmel) kell rendelkezniük. A socketekhez neveket a **bind** hívással rendelhetünk. Ezután a neveket valamilyen módon ismertté kell tenni, és a távoli felhasználók máris megcímezhetik azokat.

Az **accept** hívással lehet egy felhasználói folyamatot egy sockethez hozzárendelni, és passzív módon összeköttetés kérésekre várakoztatni. Ha egy kérés érkezik, akkor a hívás kiveszi azt a sorából; egyébként a folyamat blokkolódni fog addig, amíg egy kérés be nem érkezik (kivéve, ha a socketet nemblokkolósnak specifikálták).

Amikor egy kérés beérkezik, egy új socket jön létre és válik az összeköttetés végpontjává. Így egyetlen port több összeköttetés létesítésére is használható.

Távoli sockethez való összeköttetés létesítéshez, a folyamatoknak **connect** hívást kell kiadniuk, amelyben paraméterként a helyi és a távoli socketet kell kijelölniük. Ez a hívás összeköttetést létesít a két socket között.

Ha a socketek összeköttetés-mentes típusúak, akkor az operációs rendszer e hívás hatására feljegyzi a kettő közti kapcsolatot, és így a lokális socketen a későbbiekben kiadott **send** hívások távoli socket felé tartó üzeneteket eredményeznek.

Egy összeköttetés lebontását, vagy egy socket-socket pár közötti összerendelés megszüntetését a **shutdown** hívás használatával lehet elérni. Egy duplex összeköttetés két irányát külön-külön is le lehet zárni.

A **send** és **recv** hívások üzenetek küldésére és vételére használhatók. Ezen alaphívásoknak több változata is létezik.

Végül a **select** rendszerhívás olyan folyamatok számára hasznos, amelyeknek több létesített összeköttetésük van. Sokszor előfordul, hogy egy ilyen folyamatnak minden egyes olyan socketre **recv** hívást kell kiadnia, amelyen üzenet érkezett számára. Sajnos azonban nem tudja, hogy melyek ezek a socketek.

Ha véletlenszerűen választja ki azokat, akkor előfordulhat, hogy blokkolódik egy olyan végponton, ahol nincs is üzenet, míg más socketeken üzenetek várják. A **select** hívás lehetővé teszi a folyamatnak, hogy addig blokkolódjon, amíg a paraméterként magadott socket halmazon sikeres olvasási vagy írási kísérlet végrehajtható nem lesz.

## 8.5 Intranet, Extranet

Az Intranet rendszereket általában egy már az Internetre kapcsolódott cégen belül hozzák létre. Ilyenkor a meglévő infrastruktúra mellett csupán egy kiszolgáló gépre és szoftverre van szükség, és a felhasználók (kliensek) gépein lévő internetes böngészővel a kiszolgálóhoz való hozzáférést már biztosított. Ha a vállalat még nem kapcsolódik az Internetre, akkor ez a rendszer kiépítésének költségeit ugyan megnöveli, de nem jelentős mértékben.

Az információt szolgáltató belső szerver már biztosítja a felhasználók számára a mind a belső vállalati információs anyagokhoz történő hozzáférést, mind a már meglévő Internet hozzáférést. Biztonsági szempontból ezért szükség van egy tűzfalra, amely biztosítja, hogy a belső rendszerben lévő anyagokhoz kívülről illetéktelenek ne férjenek hozzá a belső vállalati anyagokhoz.

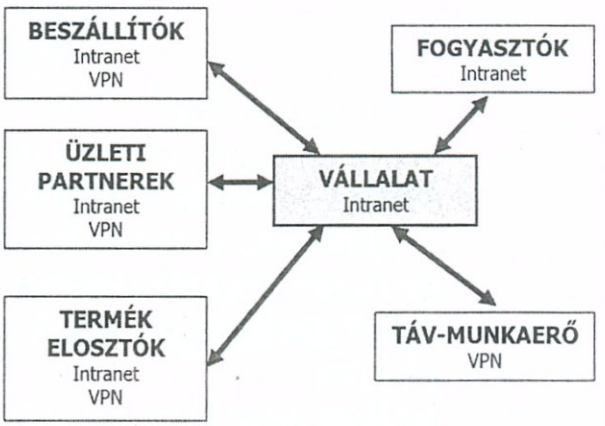
Előnyei:

- Az Intranet legnagyobb előnye a platformfüggetlenség, amit az internetes technológia biztosít.

- Megtanulni sem bonyolult egy ilyen rendszert, mert a már ismert internetes böngésző felhasználói felületét lehet használni.
- További előnye az, hogy helyileg a cég részei egymástól távol is lehetnek, mivel egyszerűen csak rá kell csatlakozni a meglévő, az egész világot átfogó internetre.

**Az Intranet egy olyan belső szervezeti információs rendszer, amely az Interneten használt protokollokra és a Web technológiájára épülve, ezek képességeit kihasználva, a szervezet tagjai számára a vállalat belső anyagaihoz való hozzáférést biztosítja, a belső kommunikáció kiterjesztésével a szervezet működésének hatékonyságát növeli.**

A Web technológia sokoldalúságának köszönhetően többfajta információ kerülhet fel a belső hálózatra: szövegek, képek, hangok, videók. Ennek köszönhetően a rendszer számos alkalmazási területtel rendelkezik. Az Intranet rendszerekben használatos alkalmazásoknak három alaptípusát lehet megkülönböztetni:




*8-6. ábra Extranet*

- **publikációs alkalmazások**: ezek segítségével személyek, csoportok, osztályok vagy a vállalat egésze tehet közzé különböző információkat a vállalat többi osztálya és alkalmazottja számára;
- **tranzakciós alkalmazások**: olyan kétirányú kapcsolatot tesznek lehetővé, melynek segítségével az információ fogadója meghatározhatja, hogy milyen információra van szüksége, illetve kiválogathatja, hogy csak a számára szükséges információ jelenjen meg képernyőn. Az információt nem csak fogadni, hanem adni is képes a felhasználó. Ilyen tranzakciós alkalmazásnak számítanak például a következők: szoftverek letöltése és feltöltése, személyre szóló jelentés, vagy levél elkészítése, megrendelés, vagy kérvény feladása.
- **közösségi alkalmazások**: több személy közti kapcsolat fenntartását, gondolatok, vélemények közzétételét, problémák közös megvitatását teszik lehetővé. Ezek a közösségi alkalmazások általában olyan hírcsoportokat jelentenek, melyeket egy-egy közös téma, érdeklődési terület köt össze.

Az internet technológia, amit a belső intranetként használunk, a szervezet környezetével történő kommunikációt is lehetővé tesz az interneteten keresztül, a kommunikációs csatornákat VPN technikával titkosítva. Az így felépített rendszer az **Extranet**. (*8-6 ábra*)

## Ellenőrző kérdések: 8. Fejezet

1. Miért előnyös a számítógépek helyi hálózatba kapcsolása?
2. Mi az a MAP és TOP?
3. Milyen megoldásokat ismer a különböző helyi hálózatok összeköttetéseinek a megvalósítására? Hogyan illeszthetők ezek be a rétegmodellbe?

4. Mi az FDDI?
5. Milyen fizikai egységeket használunk a lokális hálózatokban a gépek összekötésére?
6. Mi a munkaállomás és mi a szerver? Mi a kliens-szerver felépítés lényege? Mi az a hoszt-terminál kapcsolat?
7. Mi az a dedikált szerver? Hogyan csoportosíthatjuk a kiszolgálókat a végzett tevékenységük alapján? Ismertesse a fájl szerver fogalmát és funkcióit!
8. Mi az egyenrangú (peer to peer) hálózat?
9. Adja meg a hálózati operációs rendszer meghatározását!
10. Ismertesse a hálózati operációs rendszer funkcióit! Mit ért az összekapcsolhatóság fogalmán?
11. Röviden foglalja össze a Windows hálózatok legfontosabb tulajdonságait! Mi jelent a skálázhatóság?
12. Mi a NETBEUI protokoll?
13. Mi az Active Directory, vagyis a címtár? Mi a tartomány?
14. Mi az a házirend? Mi az hozzáférési lista?
15. Röviden jellemezze a Novell Netware operációs rendszert! Hogyan épül fel?
16. Pár mondatban foglalja össze a Netware használatához szükséges felhasználói ismereteket! (Belépés, kezelői, könyvtár és fájl jogok).
17. Mi az a socket?
18. Mi az Intranet és az Extranet? Alkalmazásuk milyen előnyökkel jár?
19. Milyen alkalmazásokat használunk az Extranet rendszerekben?

# 9. HÁLÓZATOK BIZTONSÁGI KÉRDÉSEI

*Ha egy vírust meg lehet írni, akkor meg is írják. Ha egy vírust nem lehet megírni, akkor is megírják, csak egy kicsit tovább tart. (Murphy)*

Az Internet és a munkahelyi/intézményi, sőt az otthoni számítógép hálózatok elterjedése miatt, meg kell oldani az információk bizalmas, másokra nem tartozó kezelését, amit összefoglaló kifejezéssel **hálózati biztonságnak** nevezünk. Alkalmazásával elkerüljük az illetéktelen hozzáféréseket. A biztonság megvalósításának vannak fizikai (hardveres), illetve szoftveres megoldásai, és döntően emberi tényezői.

A hálózatok biztonságnak három alapon kell nyugodnia:

1. **A fizikai biztonságot garantáló védelmek:** vagyonvédelem, tűzvédelem, megfigyelő rendszerek (biztonsági kamerák), a hálózati táplálás zavarait kiszűrő szünetmentes áramforrások.
2. **Logikai védelemek:** azonosítás (autentikáció) biztosítása (jelszavak), jogosultsági rendszer, beahatolások megakadályozása (tűzfalak) és érzékelése, mentési, archiválási rendszer, titkosítás, vírusvédelem.
3. **Adminisztratív védelem:** informatikai biztonsági politika, biztonsági szabályzat, katasztrófa elhárítási terv.

**Fontos mindig észben tartanunk, ha titkosítatlan csatornán kommunikálunk az Interneten, akkor bármely küldött vagy fogadott információ (személyes adatok, jelszavak, bankkártya információk, stb.) elfogható, lehallgatható, és egy esetleges rosszindulatú harmadik fél, tudtunk nélkül felhasználhatja és/vagy módosíthathatja.**

A fizikai világban a lopás egyértelműen kimutatható: ha valakinek ellopják az autóját, az viszonylag hamar kiderül – legkésőbb akkor, ha az eredeti tulajdonos nem találja ott, ahol legutóbb hagyta. Az információ, jellegéből fakadóan, máshogy is eltulajdonítható. Az eredeti gazda értékei nem tűnnek el, ám másokhoz is eljuthat az információ. Az értéket itt nem a fizikai objektum, hanem az információ jelenti, amely igen könnyen sokszorosítható, akár a tulajdonos tudta nélkül is. Ez pedig annak értékét azonnal csökkenti, sőt, akár értéktelenné is teheti. Például mit érnek a legújabb fejlesztés részletei, ha azokat a konkurencia megszerezheti, és kutatás, invesztíció nélkül is felhasználhatja?

A hálózati biztonsággal kapcsolatos kérdések nagyjából négy, szorosan összefüggő területre oszthatók:

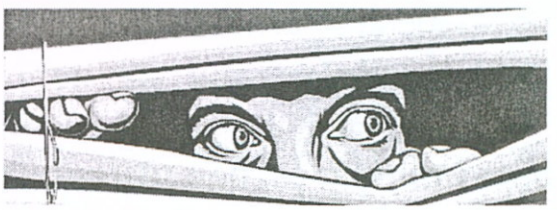
- **Titkosság** (secrecy vagy confidentiality): Adataink titkosítása segít abban, hogy az információ ne juthasson illetéktelenek tulajdonába.
- **Hitelesség** (authentication): lényege, hogy megbizonyosodjunk a velünk kapcsolatban álló személy kilétéről, mielőtt üzleti tranzakció(ka)t hajtanánk végre vele, vagy fontos adatokat fednénk fel előtte.
- **Letagadhatatlanság** (nonrepudiation): az elektronikus aláírások bizonyíthatóságával foglalkozik.

- **Sértetlenség** (integrity): annak kérdése, hogy hogyan tudjuk garantálni a fogadott üzenetről, hogy valóban azt kapjuk, amit vártunk és nem manipulálta senki a tartalmát.

## 9.1 Emberi tényező

Az informatikai vezetők szerint a vállalatokat fenyegető reális veszélyek listájának élén a vírusok állnak, a második leggyakoribb kockázati tényező, amely veszélyeztetni tudná az adataink biztonságát: az ember, a maga tudati, érzelmi, függőségi viszonyaival. Kevin Mitnick a „Megtévesztés művészete" című könyv szerzője technikai és egyéb módszerekkel a világ legbiztonságosabb rendszereit sikerült kijátszania. Véleménye szerint a nem technikai jellegű módszerek különösen alattomosak, hiszen ilyenkor a hacker gyakran szakembernek adja ki magát a behatoláshoz szükséges bizalom és információ megszerzéséhez. Ha már a humán oldal vizsgálatánál tartunk, akkor melyek az adatbiztonságot legjobban befolyásoló adatlopások fő kiváltó okai? Erre a következő hét emberi tulajdonság és cselekedet a leggyakoribb válasz (a „7E"):

- Hiúság **Ego**
- Sikkasztás **Embezzlement**
- Lehallgatás **Eavesdropping**
- Ellenségeskedés **Enmity**
- Kémkedés **Espionage**
- Zsarolás **Extortion**
- Hibás döntés **Error**



## 9.2 Biztonsági szintek

Hálózati környezetben a TCSEC (Trusted Computer System Evaluation Criteria) biztonsági szintek szerint osztályozzák a rendszereket. Az egyes szintek egymásra épülnek oly módon, hogy a következő szintnek tartalmaznia kell az alatta lévő szintek tulajdonságait is.

**D - Minimal Protection (Minimális védelem).** Például a DOS operációs rendszer

**C1 - Discetionary Security Protection (Megkülönböztetéses védelem):** A felhasználók azonosításához nevet és jelszót használnak. A saját erőforrásaikkal maguk rendelkeznek, mások elől elrejthetik, jogokat (olvasható, írható, futtatható) rendelhetnek hozzá. A felhasználók csoportokba sorolhatók, és ezekre is hasonló jogrendszer adható meg.

**C2 - Controlled Access Protection (Ellenőrzött hozzáférés):** Az előző szintet kibővítve megjelenik a naplózás, vagyis a rendszer biztonságával kapcsolatos történések feljegyzése.

**B1 - Labelled Security Protection (Címkézett biztonság):** Felhasználó által nem módosítható biztonsági, titkossági fokozatokat használnak a védett rendszerben lévő adatok és állományok egy részénél.

**B2 - Structured Protection (Strukturált védelem):** Minden állományt biztonsági szintet jelölő azonosítóval látnak el. A készített biztonsági mentések is tartalmazzák

az azonosítókat, de a bevezetett titkosítások miatt, a másolathoz hozzájutva sem olvasható el a titkosított anyag. Szükséges a megoldáshoz a rendszer hardveres támogatása.

**B3 - Security Domains (Biztosított egységek)** Az egységeket fizikailag is védik. Két egység közötti kommunikációs utat teljes mértékben ellenőrizni kell (ellenőrzött út). Pl. csak adott terminálról lehet egy adott programba bejelentkezni.

**A - Verified Design (Ellenőrzött Tervezés)** Azonos az előzővel, de itt megkövetelik a működés bizonyítását matematikai módszerekkel.

Természetesen az egyes szinteken használt módszereket a gyakorlatban keverni is lehet a nagyobb biztonság elérése érdekében.

Gondoljuk csak meg, milyen biztonsági fenyegetést jelent a nagykapacitású USB memória kulcsok megjelenése. A jelszóval való visszaélés ellen, például a banki átutalásoknál a következő módon védekeznek: tranzakció elindítását a banki program a számlatulajdonos mobiltelefonjára küldött, egy óráig aktív jelszó bekérése után hajtja csak végre.

## 9.3 Fizikai biztonság

| FIZIKAI BEHATÁS | % |
|---|---|
| Áramkimaradás vagy feszültségtüskék | 45,3 |
| Vihar okozta károsodás | 9,4 |
| Hardver vagy szoftverhiba | 8,2 |
| Tűz vagy robbanás | 8,2 |
| Árvíz, vagy egyéb víz okozta kár | 6,7 |
| Földrengés | 5,5 |
| Hálózat kiesése | 4,5 |
| Emberi tényező | 3,2 |
| Hütőtechnika kiesése | 2,3 |
| Egyéb | 6,7 |

Elsődleges veszélyforrás a biztonságra, maga a hálózat fizikai környezete. Ezen tényezők közül is az egyik legfontosabb az energiellátás minősége. A szerverek, valamint egyéb hálózatfentartást szolgáló aktív eszközök (pl. routerek) védelmében gondoskodni kell a szünetmentes tápellátásról, amelyek a pillanatnyi feszültség kimaradásoktól, és feszültség ingadozásoktól (túl kicsi feszültség, vagy éppen túlfeszültség) óvják a rendszert. Ezek ellen megfelelő biztonságtechnikai berendezésekkel védekezhetünk. Meg kell említeni a hűtés jelentőségét, mivel sok esetben fordul elő, hogy a kényesebb alaktrészek túlmelegednek, mert rosszul vannak méretezve, vagy a hűtőventillátor visszajelzés nélkül áll le. A táblázat, a fizikai behatások okozta káresetek százalékos eloszlását mutatja.

## 9.4 Adat titkosítás

Mint már bemutattuk, a titkosítás egy olyan matematikai eljárás, melynek során egy adatot felismerhetetlenre változtatunk úgy, hogy abból az eredeti adat csak valamilyen, kizárólag a küldő és a fogadó által ismert eljárás segítségével fejthető vissza.

Az eljárás során a titkosító algoritmus bemenete a titkosítandó adat és a kulcs, kimenete pedig a titkosított információ. A visszafejtésnél ez utóbbi és a kulcs lesz a bemenet, a visszafejtett szöveg pedig a kimenet. Ha a kulcsok tárolása saját gépünkön történik, a kulcsok csak annyira biztonságosak, amennyire maga a számítógép.

Minél hosszabb a titkosításhoz használt kulcs, illetve bonyolultabb a titkosítási algorit-

mus, annál hosszabb az esetleges visszafejtéshez szükséges idő, és annál erősebb a titkosítás. Az algoritmus/kulcs jellege alapján két, alapvetően eltérő titkosító megoldást különböztetünk meg, ezek: a titkos, szimmetrikus kulcsú (pl. DES), illetve az aszimmetrikus, vagy másnéven nyilvános kulcsú (public key) eljárások (pl. PGP).

## 9.4.1 Digitális aláírás

A nyilvános kulcsú titkosítás teszi lehetővé azt, hogy az információk titkosítása mellett elektronikus aláírásokat is használjunk. Az elektronikus aláírás az üzeneteket nem rejtjelezi, célja mindössze az, hogy a címzett meggyőződhessen arról, hogy a neki küldött információ valóban a feladótól származik, és azt más nem módosította.

**Az elektronikus aláírás létrehozásához a küldő a saját privát kulcsát használja, hitelességét pedig a címzett ellenőrzi le a küldő nyilvános kulcsával.**

1. A DOKUMENTUM (PL.SZERZŐDÉS) ELEKTRONIKUS FORMÁBAN TÖRTÉNŐ ELKÉSZÍTÉSE (PL. WORD FÁJL)

2. LENYOMAT (FINGERPRINT) KÉSZÍTÉSE MD5 MÓDSZERREL. (A BEMENET A TETSZŐLEGES HOSSZÚSÁGÚ DOKUMENTUM A KIMENET EGY PL. 1024 BITES „SŰRITMÉNY"). A DOKUMENTUM AKÁRMILYEN MEGVÁLTOZTATÁSA A LENYOMATOT VÁLTOZTATJA. AZ EREDMÉNY EGY KARAKTERSOROZATKÉNT ÁBRÁZOLT BITHALMAZ.

3. A LENYOMAT TITKOSÍTÁSA AZ ALÁÍRÓ TITKOS KULCSÁVAL. EZ A DIGITÁLIS ALÁÍRÁS (AZ EREDMÉNY SZINTÉN EGY KARAKTERSOROZAT), ÉS A DOKUMENTUMHOZ FŰZÉSE ELEKTRONIKUS FORMÁBAN. AZ ALÁÍRÓ NYILVÁNOS KULCSÁT EGY SZOLGÁLTATÓ HITELESÍTI: BIZONYÍTJA A NYILVÁNOS KULCS ÉS A (JOGI) SZEMÉLY AZONOSSÁGÁT.

ELLENŐRZÉS: A DOKUMENTUMBÓL LENYOMAT KÉSZÍTÉSE, ÉS AZ ALÁÍRÓ NYILVÁNOS KULCSÁVAL A DIGITÁLIS ALÁÍRÁSBÓL VISSZANYERT LENYOMAT ÖSSZEHASONLÍTÁSA.

TÖBB ALÁÍRÓ ESETÉN A LENYOMAT TITKOSÍTÁSA AZ ALÁÍRÓK NYILVÁNOS KULCSAIVAL EGYMÁS UTÁN.

*9-1. ábra Digitális aláírás módszere*

### Tanúsítványok

A nyilvános kulcsú titkosítási eljárások használatakor fontos, hogy a titkosított üzenet küldése előtt, megbizonyosodjunk arról: valóban a címzett nyilvános kulcsát használjuk-e. Ha közvetlenül tőle kaptuk meg, ez nem okoz problémát, ha azonban nyilvános helyről jutunk hozzá, valamilyen módon meg kell győződnünk az eredetiségéről.

Anna és Béla titkosított levelezést folytat, és ezt Marci szeretné lehallgatni. Tegyük fel, hogy el tudja csípni a kapcsolatfelvétel kezedetét, amikot Anna elküldi Bélának saját nyilvános kulcsát, Béla Annának a nyilvános sajátját. Az üzeneteket megszerezve Marci két kulcspárt generál. Anna felé elhiteti, hogy ő Béla, elküldi az egyik nyilvános kulcsát Béla nevében, Béla felé elhiteti, hogy ő Anna és elküldi Bélának a másik generált nyilvános kulcsát Anna nevében. Természetesen, bár Anna és Béla azt hiszi, hogy egymással vált tikosított üzenetet, pedig Marci az, aki mindkét fél felé megszemélyesíti a másikat.

Ez a **„man-in-the-middle attack"** (=ember a középen), vagy más néven a „közbülső harmadik" típusú támadás.

A „közbülső harmadik" típusú támadások megakadályozhatók a bizalmi láncon alapuló **elektronikus tanúsítványok** (**certificate**) alkalmazásával.

Tanusítvány alapja: a bizalom. Anna és Béla hisz közös barátjuknak, Ceciliának. Béla megbízható módon (pl. személyesen) az Annának küldendő nyilvános kulcsát Cecilia saját titkos kulcsával aláíratja. Cecilia Annának megbízható módon (pl. személyesen) odaadja saját nyilvános kulcsát. Béla, mikor Annának ír, az üzenetben az Annának szóló nyilvános kulcsot tartalmazó tanusítványt is küldi. Anna birtokában van Cecilia nyilvános kulcsa, ezért meg tudja nézni a tanusítványt, és benne a hiteles Bélától származó nyilvános kulcsot fogja használni.

**A tanusítvány egy megbízható harmadik személy (Certification Authority = CA) által kiadott dokumentum.** Tartalmazza a tanusítványt kérő tulajdonos azonosítóját és a nyilvános kulcsát. A CA digitálisan aláírta a saját kulcsával. CA tanúsítja, hogy a nyilvános kulcs az adott azonosító birtokosához tartozik.

A formátumát szabványok definiálják, neve: X.509. Ez egy olyan kommunikációs szabvány, mely az elektronikus tanúsítványok szerkezetére, felépítésére, tartalmára ad előírásokat. Az elektronikus tanúsítványnak a következő információkat kell tartalmaznia:

- A tanusítványt kérő személy vagy szervezet nyilvános kulcsát.
- A személy vagy szervezet azonosítására alkalmas, vagy személyes adatait.
- Egy, vagy több digitális aláírást: azoknak a szervezeteknek vagy személyeknek az aláírását, akik igazolják a fentiek valódiságát.

### *Hitelesítéstípusok*

- **Személyes hitelesítés** azt szavatolja, hogy azok vagyunk, akiknek mondjuk magunkat. Ez a hitelesítés olyan személyes adatokat igényel, mint például felhasználónevünk és jelszavunk. Az adatokra akkor van szükség, amikor az Interneten keresztül személyes információt küldünk egy olyan Web-helyre, amely kilétünket bizonyító hitelesítést igényel.
- **Helyhitelesítés**, más néven **szervertanúsítvány** azt igazolja, hogy az adott Web-hely biztonságos és valódi. Garantálja, hogy más webhely nem élhet vissza az eredeti webhely azonosítóival. A helyhitelesítéseket kiállításukkor dátummal is ellátják. A böngészőben a hiteles Web-helyeket nem az egyszerű HTTP protokollon keresztül, hanem az SSL titkosítást használó **HTTPS** protokollon nyitunk meg. Pl. Internetes banki ügyintézés, tőzsdei tranzakciók, stb.
- **Programkészítői hitelesítés,** más néven **szoftvertanúsítvány** azt igazolja, hogy az éppen gépünkre telepítendő programhoz gyártója a nevét adja, és az ő személyazonosságát, a program eredetiségét és sértetlenségét egy megbízható harmadik fél tanúsítja.

### Tanúsítvány osztályok (class)

Tanúsítvány és tanúsítvány között nagy különbségek lehetnek: míg vannak tanúsítványok, amelyek gyakorlatilag semmit sem érnek, mert a kibocsátó egyáltalán nem kezeskedik azért, hogy a tulajdonos valóban az, aki fel van benne tüntetve, vannak olyanok is, amelyekhez nagyon komoly garanciák kapcsolódnak.

A Netlock magyar hitelesítő szolgáltató három elnevezett osztályt használ: Expressz (C, leggyengébb), Üzleti (B), illetve Közjegyzői (A) tanúsítványok.

### *Visszavonási lista (CRL)*

Bár a tanúsítványoknak létezik érvényességi ideje, szükség lehet arra, hogy a lejárat előtt visszavonásra, felfüggesztésre kerüljenek a kibocsátó által (akár egy bankkártya).

A visszavont tanúsítványok egy tanúsítvány visszavonási listára kerülnek, melyet a kibocsátó nyilvánosan elérhetővé tesz.

## 9.4.2 Virtuális magánhálózat (VPN - Virtual Privat Network)

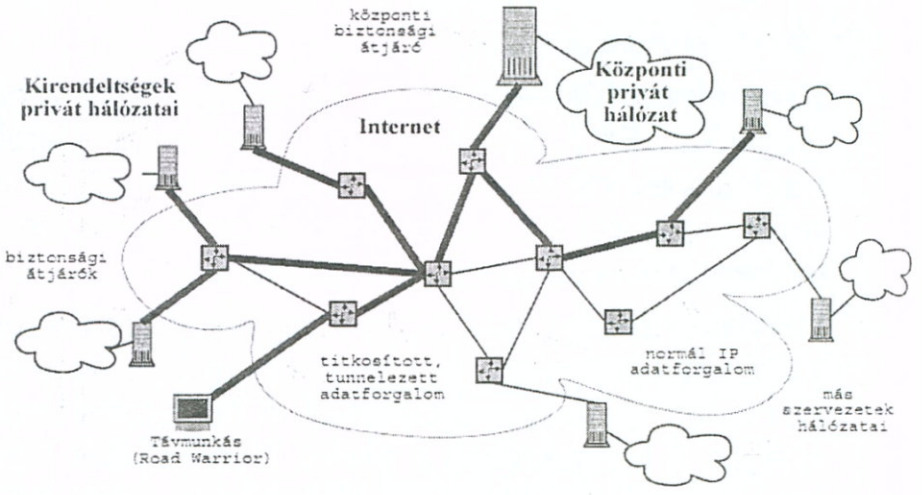
**A virtuális magánhálózat nem más, mint egy - az Internetet, mint továbbító közeget felhasználó - titkosított csatorna, amely a távoli felhasználóknak is lehetővé teszi elszigetelt hálózat használatát.**

Segítségével például, egy több helyen telephellyel rendelkező cég alhálózatait kapcsolhatjuk össze. Ekkor a központi hálózat és a telephelyeken lévő alhálózatok folyamatosan kölcsönösen elérhetők, a VPN kiszolgálók állandó kapcsolattal csatlakoznak az Internetre. Egyedi felhasználók, távmunkások (Road warrior) is kapcsolódhatnak ilyen módon a VPN kiszolgálókhoz.

A VPN kapcsolatnak két alapvető összetevője van:

- A virtuális hálózati protokoll,
- A titkosítási protokoll

Vagyis a VPN csak megfelelő, és titkosítást alkalmazó protokollok segítségével valósítható meg, vagyis a nyilvános hálózaton keresztüli biztonságos kommunikáció érdekében a hálózati forgalmat titkosítani kell.




*9-2. ábra VPN hálózat*

A gyakorlatban két protokoll terjedt el:

- **Point-to-Point Tunneling Protocol (PPTP)**: Az RFC 2637-ben definiált alagútprotokoll, amely MPPE (Microsoft Point-to-Point Encryption) titkosítással működik.
- **Layer 2 Tunneling Protocol (L2TP):** Az L2TP protokoll specifikációja az RFC 2661-ben található. Maga az L2TP protokoll saját titkosítást nem tartalmaz, ezért a virtuális magánhálózatot az „L2TP over IPSec", azaz az IPSec (IP security) titkosítással segített L2TP kapcsolat valósítja meg.

Az IPSec egy olyan megoldás, amely az eredetileg titkosítatlan IP csomagokat kulcs segítségével titkosítja és az esetleges módosítás ellen védve, hitelesítéssel is ellátja, és így juttatja át a csomagot a hálózaton.

Egy VPN kialakítása leegyszerűsítve úgy néz ki, hogy minden egyes összekapcsolni kívánt hálózatrész és a publikus hálózat közé biztonsági átjárókat (security gateway) helyezünk.

Az átjárók titkosítják a csomagokat, melyek elhagyják a privát hálózatot és dekódolják a publikus hálózatból érkező csomagokat, ezzel titkosított csatornát (tunnelt=alagútat) kialakítva a publikus hálózaton.

A VPN gyakorlati megvalósítására három megoldás kínálkozik:

**Hardver alapú megoldások.** A legtöbb hardware alapú megoldás tulajdonképpen olyan átjárók alkalmazását jelenti, amelyek képesek az adatforgalom titkosítására. A legmagasabb fokú **hálózati áteresztőképességet (throughput)** nyújtják az összes többi megoldással szemben, hiszen nem emésztenek fel fölösleges erőforrásokat a megvalósításhoz. Sajnos nem rugalmasak, mint például a software alapú megoldások.

**Tűzfal alapú megoldások.** A tűzfal alapú megoldások kihasználják a tűzfal biztonsági mechanizmus előnyeit, mint például a hozzáférés korlátozását és a címfordítást (NAT). Az ilyen elven működő VPN szerverek lényegében tűzfal feladatokat is ellátó kiszolgálók. A teljesítményre kihatással van az, hogy maga a titkosítás is nagy számítási teljesítményt igényel a tűzfalfunkciók megvalósításán túlmenően.

**Szoftver alapú megoldások.** Manapság a VPN menedzselésére a legrugalmasabb megoldásokat a szoftver alapú termékek nyújtják. A legtöbb ilyen programcsomag lehetőséget ad a csomagok címzés, vagy protokoll szerinti tunneling-jére, és így nem kell minden forgalmat titkosítottan kezelni. Kétféle adat lehet: olyanok, amelyeket a VPN-en keresztül kell küldeni (pl. adatbázisok), és olyanok, amelyeket nem (web böngészés). A megoldáshoz az operációs rendszerek túlnyomó része ma már önmagában is ad valamilyen szintű támogatást. Természetesen ebben az esetben az IPsec beállításait a két oldalon, rendszergazdai szinteken kell megvalósítani.

A VPN megoldások két módban képesek működni:

- **Tunnel módban** a teljes eredeti csomag újból becsomagolásra és titkosításra kerül, valamint új fejlécet kap. Ez lehetőséget ad arra, hogy az olyan eszközök, mint például a routerek IPSec proxy-ként funkcionáljanak, azaz a hostok helyett ezek végezzék el a titkosítást, és a dekódolást. A **tunnel** módban az IPSec-et alkalmazó gépek átjáróként funkcionálnak és akárhány logikailag az átjáró mögött elhelyezkedő gép forgalmát képesek továbbítani a tunnelben. A kliens gépeken semmilyen IPSec-hez kapcsolódó feldolgozást nem szükséges végezni, az egyetlen kritérium, hogy az útválasztó táblájukban szerepeljen a megfelelő IPSec átjáró címe.
- **Transport módban** pont-pont kapcsolat épül fel a résztvevő két gép között. Ebben a módban, csak az IP csomagtörzs kerül titkosításra, az eredeti IP fejlécek változatlanul maradnak. Ennek a módnak az az előnye, hogy csak néhány byte-ot ad hozzá minden csomaghoz.

A csomagok titkosítása mellett azok hitelesítése is megtörténik. A hitelesítés során felhasznált algoritmusok az MD5 (Message Digest Algorithm) és az ehhez hasonló SHA (Secure Hash Algorithm). Az algoritmusok célja, hogy közel lehetetlenné tegyék az illetéktelenek számára a hitelesített adatok módosítását anélkül, hogy a fogadó fél észre ne venné.

A küldő kiszámol az algoritmussal egy tömörítvény értéket az adatból és belefoglalja a tényleges fejlécet bővítő hitelesítő fejlécbe. A fogadó fél is elvégzi az előbbi számítást és összehasonlítja a kapott értéket a csomag fejlécben érkezővel. Módosítatlan csomag esetén a két értéknek egyeznie kell. Természetesen a számítások felhasználják a csomagot tikosító kulcs értékét, így a támadó mesterkedése reménytelen.

## 9.4.3 Biztonsági Alréteg (SSL - Secure Socket Layer)

A nyílt szövegen alapuló HTTP forgalmat titkosíthatjuk a HTTP+SSL vagy HTTPS protokollok segítségével. A HTTPS protokoll a 443-as portot használja.

| Alkalmazási réteg |
|---|
| HTTP FTP POP3 IMAP stb |
| SSL Biztonsági alréteg |
| Hálózati réteg |

*9-3. ábra SSL helye*

Az SSL (Secure Socket Layer; Biztonsági Alréteg) egy protokoll réteg, amely a hálózati (Network layer) és az alkalmazási rétegek (Application layer) között van. Ez a biztonságot biztosító alréteg mindenféle forgalom titkosítására használható - POP3, IMAP és legfőképp HTTP.

Az SSL háromféle titkosítási módszert használ: nyilvános-titkos kulcs (Public-Private Key), szimmetrikus kulcs (Symmetric Key), és a "digitális aláírás" (Digital Signature) módszerét.

**Nyilvános-titkos kulcsú titkosítás:** Ebben az algoritmusban a titkosítás és a visszafejtés nyilvános-titkos kulcspárral történik. A webszerveré a titkos kulcs, a nyilvános kulcsot pedig a tanúsítványban küldi el a kliensnek.

1. A kliens kéri a HTTPS-t használó Web szervertől a tartalmat.
2. A web szerver válaszol egy Digitális Tanúsítvánnyal (Digital Certificate), amiben benne van a szerver nyilvános kulcsa.
3. A kliens ellenőrzi, hogy lejárt-e a tanúsítvány.
4. Ezután, a kliens ellenőrzi, hogy a tanúsítványhatóság (Certificate Authority, továbbiakban CA), amely aláírta a tanúsítványt, megbízott hatóság-e a böngésző listáján. Ez a magyarázata annak, miért van szükségünk egy megbízott CA-tól kapott tanúsítványra.
5. A kliens ellenőrzi, hogy a webszerver teljes domén neve (Fully Qualified Domain Name) megegyezik-e a tanúsítványon lévő közönséges névvel (Common Name).
6. Ha minden megfelelő, létrejön az SSL kapcsolat.

**Szimmetrikus titkosítás - az adatok tulajdonképpeni átvitele**: Az SSL kapcsolat létrejötte után, szimmetrikus titkosítást használ az adatok titkosítására, amely kevésbé számításigényes. (Szimmetrikus titkosításkor az adat ugyanazzal a kulccsal titkosítható és visszafejthető.) **A szimmetrikus titkosítás kulcsa a kapcsolat indításakor kerül átadásra, a nyilvános-titkos kulcspárral történő titkosítás alatt.**

**Üzenet ellenőrzés** A szerver kivonatot készít az üzenetről valamilyen algoritmus szerint, mint például MD5, majd ezek alapján ellenőrzi az adatok sértetlenségét.

## 9.5 Védekezési eszközök

A hálózati védelem egy szoftverréteg a hálózat és a felhasználói programok között. Olyan szolgáltatásokat biztosít, mint a felhasználóazonosítás, a jogosultság megállapítása, adatvédelem, naplózás, stb.

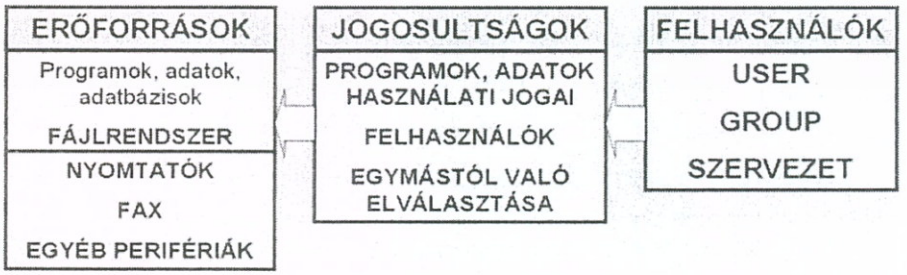
Védekezéskor első lépésként azzal kell tisztában lenni, hogy kicsoda a felhasználó. Egy regisztráció és egy beléptetőrendszer segítségével, a felhasználói név és jelszó alapján, nyomon lehet követni a felhasználókat.

A digitális dokumentumok eredetiségének megőrzése sokszor fontos elvárás. Az anya-

gok (szándékos vagy véletlen) módosítása, tolvajlása és illegális másolása elleni intézkedések között szerepelhet a titkosítás, digitális lenyomatok elhelyezése, tűzfalak használata és vírusirtó szoftverek alkalmazása.

## 9.5.1 Jogosultságok

A számítógép hálózatokban található erőforrások és információk használatát a hálózati felhasználóknak kiosztott **jogosultságokkal** szabályozhatjuk. Mivel a jogosultságok kezelése, és kiosztása, vagyis az adminisztrálása bonyolult feladat, ezért a jogosultságok kezelésénél két elvet követtek:




*9-4. ábra Jogosultságok*

1. A jogosultságok kezelése lehetséges nem csak felhasználói szinten, hanem felhasználókból csoportokat képezve, csoportszinten.
2. A jogosultságok kifinomultabb, és differenciáltabb kiosztását teszi lehetővé, ha a jogosultsági rendszert nem csupán a felhasználói, hanem az erőforrások oldalán is kialakítjuk. (Gondoljuk csak meg: egy adott információt tartalmazó fájl törlésének a megakadályozását a hozzátartozó törlési jog kikapcsolásával valósíthatjuk csak meg, a felhasználók oldaláról ezt nem is lehetne megtenni.)

Mikor a hálózatokban a felhasználók használják az erőforrásokat, a felhasználói és erőforráshasználati jogokat a rendszer folyamatosan kiértékeli, és ennek eredménye alapján valósul meg az eredő használati jog, biztosítva a rendszer védelmét.

## 9.5.2 Jelszavak

**Informatikai értelemben a jelszó a hálózatok vagy adattárak hozzáféréséhez szükséges azonosító kulcsszó.** Általában a jelszó egy felhasználóhoz tartozik, és felhasználói név/jelszó párosként használják. Tanácsos ezeket fejben (vagy írásban), de mindenképpen a számítógépen kívül megjegyezni. Mivel szorosan egy emberhez és annak memóriájához kötődik, ezért a jelszavak megszerzése, vagy kitalálása a rendszervédelem gyenge pontja. A jövő mindenképpen a biometrikus azonosításé lesz: ujjlenyomat, szem- írisz, DNS azonosítás alapján. (DNS itt NEM Domain Name System !!!)

### Jelszavak kezelése

| A jelszó kezelésekor erre ügyeljünk | A jelszó kezelésekor ezt ne tegyük |
|---|---|
| Tartsuk titokban a jelszót. | Ne írjuk le. |
| Minden webhelyhez más jelszót használjunk. | Ne használjunk a webhelyeken ajánlott jelszó emlékeztető szolgáltatásokat. |
| Legalább hathavonta változtassuk meg a jelszavunkat. | Ne ismételjük ciklikusan, időről időre a jelszavunkat. |

Meglepődnénk, ha tudnánk, hogy hányan írják le a titkos jelszavukat, és ragasztják a

monitorukra, vagy berakják a számítógép melletti asztalfiókba. A következő táblázatban összefoglaltuk azokat az irányelveket, amelyek segítségével biztonságosabban tudjuk kezelni a jelszavakat.

Milyen jó dolog, ha bankkártyát tartalmazó tasakra ráirják a pin kódot? Pedig naponta megtörténik...

### *Jelszavak készítése*

Az egész jelszókészítés ellentmondásosságát a következőkben foglalhatjuk össze: Ha bonyolult, és hosszú a jellszó, akkor nehéz feltörni, de nehéz megtanulni is. Ha leírjuk, akkor elveszthetjük, és más megszerezheti. Ha meg rövid, vagy velem kapcsolatos, akkor egyszerűen feltörhető.

| A jelszó készítéskor erre ügyeljünk | A jelszókészítéskor ezt ne tegyük |
|---|---|
| Legyen hosszú (legalább 7 karakter). | Ne használjunk a felhasználó nevünket vagy annak egy részét jelszóként. |
| Legyenek benne kis- és nagybetűk, számok és szimbólumok. | Ne használjunk egyetlen nyelv szavait sem jelszóként. |
| A második és a hatodik karakter közt használjunk legalább egy szimbólumot is. | Ne használjunk a szavak kialakítására a betűk helyett azokhoz hasonlító számokat. |
| A jelszóban legalább négy különböző karakter legyen (ne ugyanazok a karakterek ismétlődjenek). | Ne használjunk az ábécében egymás után álló karaktereket vagy egymás utáni számokat (pl.: „abcdefg" vagy „234567"). |
| Használjunk véletlenszerűen kiválasztott számokat és betűket. | Ne használjunk a billentyűzeten egymás után álló karaktereket (például „qwerty"). |

**Egy módszer a jelszókészítésre:** Először találjunk ki egy könnyen megjegyezhető mondatsort (pl.: „Hol, mikor születtem? Tatabánya, 1949!"). Emeljük ki a mondatból a szavak első betűjét és az írásjeleket, majd ezek összevonásából alkossunk egyéni jelszót. („Hms?T1!") Elég csúnyán néz ki, ugye? De, ez a jelszó semmilyen szótármódszerrel nem fejthető meg, és semmilyen lexikonban sem szerepel. Eredményt kizárólag csakis a nyers erő módszerrel érhetünk el, ez azonban sokáig tart.

### *Jelszavak titkosítása*

A jelszavakat általában a következő módszerrel titkosítják: **A jelszóból létrehoznak egy fix hosszúságú jelsorozatot, bizonyos hash függvény segítségével.** Minden különböző jelszóhoz különböző jelsorozat generálódik, a jelszó legkisebb változtatásánál is az egész jelsorozat teljesen más lesz. Az algoritmust egyirányú, nem lehet megfordítani, vagyis a jelszó titkosított alakját ismerve, sem lehet belőle a jelszót visszaállítani.

A kódtörő programok működése a következő: Előállítják a kipróbálandó jelszóból a hash függvény segítségével a fix hosszúságú jelsorozatot, majd ezt hasonlítják össze a jelszó fájlban lévő, hash algoritmussal átalakított jelszóval. Ha megegyezik a két hash jelsorozat, akkor megtalálta a jelszót. Ha nem, akkor jöhet a következő lehetőség, de ez idő, idő...

### *Jelszavak megfejtése*

A jelszavak megfejtésére három módszer terjedt el:

- **„Nyers erő":** Végig próbáljuk az összes lehetséges megoldást. Először kezdjük a legrövidebb jelszavakkal : kétbetűs számok és betűk összes lehetősége, aztán a hárombetűsek, és így tovább.

Előnye hogy így biztos hogy megtaláljuk a jelszót, hátránya az időigényessége. Ha pedig a jelszavakat gyakran változtatják akkor, ha megtaláljuk, már idejét múlta.

- **Szótár használata**: A módszer az emberi lustaságot használja ki, mert az emberek többsége hétköznapi szavakat használ jelszóként (vagy éppen a keresztnevét, vagy más hasonló személyes és ismert adatokat magáról). Azért választja ezeket, hogy el ne felejtse. Tehát kell egy minél nagyobb szótárfájl, ami az adott nyelv szavaiból tartalmaz minél többet. Ezek után jelszótöréskor a szótárfájl szavaiból egyesével előállítják a kódolt jelszót, majd ezt összehasonlítják a jelszó fájlban található szavakkal.
- **Kevert módszer (hybrid attack)**: Az előbbi két módszer kombinálása, ami arra a tényre épül, hogy a felhasználók sokszor választanak olyan jelszót, ami egy sima szóból és egy-két egyéb karakterből áll (pl.: laci46, lakat12). Ehhez is kell egy szótár-fájl kell, és egy megírt, vagy interneten beszerzett jelszótörő program. Ez a program egy rendes szótárfájlban található minden egyes szóból elkészíti a kevert módszer szerinti bővített szavakat, majd ezeket beleírja egy másik fájlba. Ezek után a jelszótörőt már csak ezzel a szótárral kell futtatni.

## 9.5.3 Biztonsági másolatok

A biztonsági mentések (másolatok) készítése az otthoni és az irodai számítógépeknél is egyre fontosabb. A merevlemezes meghajtók megnövekedett kapacitása lehetővé teszi egyre több fájl és adat tárolását. Sajnos az adatok nagyon gyorsan el is veszhetnek. Meghibásodhat a merevlemez, a különböző vírusok és kártékony programok fájlokat törölhetnek le, de lehet, akár véletlen illetve szándékos felhasználói fájltörlés valamint fájl felülírás is. Az otthoni számítógépek esetében is sok olyan információ lehet, amelyek elvesztése sok bosszúsággal, illetve felesleges munkával jár. Ezért célszerű a dokumentumok, adatbázisok illetve konfigurációs fájlok, az e-mail-ek, stb. rendszeres mentése. Ha megoldható, akkor **érdemes a rendszer egészének mentését is megoldani**, mert vírusfertőzés, vagy egy hardver hiba miatt összeomlott operációs rendszer helyreállítása is gyorsabb.

Ezek az adatvesztések könnyen megelőzhetők, és sokszor nagyon jól használhatók a legegyszerűbb megoldások. Sokszor az is elég, ha a fontos állományainkat átmentjük egy másik merevlemezre (ma már nem drága, vagy a fiókban is van...), vagy kiírjuk CD-re, illetve DVD-re. Ha hálózaton más számítógépekhez is kapcsolódhatunk, akkor a megosztás segítségével a más gépen lévő könyvtárakba is célszerű mentést készíteni.

Sajnos, bár a rendszeres adatmentés nem bonyolult folyamat, mégis az emberi lustaség miatt sokszor elmarad. Pedig ez a feladat nagyon jól automatizálható kötegelt fájlok segítségével, és olyan munkastílussal, hogy a változó, illetve keletkező fájljainkat csak néhány kitüntetett könyvtárban helyezzük el. Az ütemezett, könnyen konfigurálható biztonsági mentések készítéséhez rendelkezésre áll a **Cobian Backup 6** program, amely ingyenesen tölthető le, illetve használható.

## 9.5.4 Naplózás

A számítógépeken és a hálózaton történő eseményeket az operációs rendszerek naplófájlokban rögzítik. Az általánosságok helyett nézzünk meg egy valódi példát. Windows XP esetén az eseménynapló előhívása: *Start menü » Vezérlőpult » Felügyeleti eszközök » Számítógép-kezelés* (vagy jobb gombbal kattintás a Sajátgépen, majd Kezelés, ott: Eseménynapló).

Háromféle naplót látunk: Alkalmazás, Biztonság és Rendszer.

- **Alkalmazás napló**: a számítógépre telepített alkalmazásokkal kapcsolatos információkat nézhetjük meg. A programoktól származó üzenetek és hibák itt tárolódnak.
- **Biztonsági napló**: ez a napló alapbeállításként nincs engedélyezve. A *Vezérlőpult » Felügyeleti eszközök » Helyi biztonsági házirendnél* tudjuk bekapcsolni. Az engedélyezés után, a sikeres és sikertelen események kerülnek itt rögzítésre.
- **Rendszer napló**: a rendszerrel kapcsolatos vagy a rendszer által generált eseményeket találhatjuk. Pl. valamelyik szolgáltatás nem indult el vagy beléptetési hiba történt.

### *Bejegyzések fajtái*

A naplófájl lényegében a naplózott esemény jellemzőit tartalmazó szövegsorokból áll, ezek:

- **Információ:** egy esemény sikeres végrehajtása, például egy szolgáltatás elindítása.
- **Figyelmeztetés**: nem túl jelentős probléma megnevezése, de esetleg hibához vezethet.
- **Hiba:** valami nem jó, és ez veszélyes lehet a rendszerre.
- **Sikeres események:** bekapcsolt biztonsági naplózás esetén, a naplóba bejegyzésre kerülnek a sikeres események.
- **Sikertelen események:** bekapcsolt biztonsági naplózás esetén, a naplóba bejegyzésre kerülnek a sikertelen események.

### *Bejegyzések szűrése*

Csak kiválasztott információk megjelenítéséhez lehetőség van a bejegyzések szűrésére. Ehhez a megfelelő napló (alkalmazás, biztonsági, rendszer) nevére állva és az egér jobb gombjával kattintva, kiválasztjuk a *Tulajdonságok/Szűrő* fület.

Az öt lehetséges bajegyzésfajta naplózását a neve melletti jelölő dobozban adhatjuk meg. A *Tulajdonságok/Általános* fülre kattintva, hasonló módon tudjuk a megjelenő ablakban a naplófájl méretét, nevét, és elérési útvonalát, ha szükséges, megváltoztatni. Windows a naplókat egy EVT kiterjesztésű bináris fájlban tárolja, és csak az Eseménynaplóval olvasható (jobb klikk, Naplófájl megnyitása...).

A fájl maximális mérete alapbeállításként 512KB, ez csökkenthető ill. növelhető. Ha megtelik a napló, az komoly problémákat okozhat, ezért a felkínált lehetőségek küzül döntsük el, mit is tegyen a rendszer ilyen esetben.

## 9.6 Külső hálózati támadások, és védekezés ellenük

Ezek a támadások a gépünkhöz kapcsolt hálózat felől érkeznek, természetesen tudatos rosszindulat eredményeként.

Ezt egy **hacker**-nek elnevezett személy valósítja meg, aki egy megszállott szoftverszakértő, aki az Internetre kapcsolt rendszerek szisztematikus *letapogatásával* megkeresi azok gyenge pontjait és behatolási lehetőségeit.

## 9.6.1 Hackertámadások

### *Szolgáltatásmegtagadási támadások*

A **szolgáltatásmegtagadási (DoS = Denial of Service)** támadások során egy rendszert vagy hálózatot **kezelhetetlen mennyiségű adattal árasztanak el**. Ekkor a rendszer összeomlása, illetve a hálózati sávszélesség „eltömődése" megakadályozza a szabályos kommunikáció folytatását.

Az **elosztott DoS (DDoS = Distributed DoS)** támadások még kifinomultabbak. Ebben az esetben a támadók az interneten keresztül számos számítógép felett veszik át az ellenőrzést úgy, hogy titokban a távirányítást lehetővé tevő szoftvert telepítenek a gépekre. Ezután ezeket a *szolga* vagy *zombi* elnevezéssel illetett számítógépeket használják más rendszerek vagy hálózatok elleni DoS támadás megindítására, ezáltal a támadás nem követhető vissza a valódi támadó személyéhez. A védtelen rendszerek nem csak a DoS támadások célpontjaként vannak veszélynek kitéve, hanem azért is, mert DDoS támadás közvetítésére használhatók fel.

A szolgáltatásmegtagadás előidézésére számos különböző technikai módszer létezik. Gyakori DoS/DDoS támadás közé tartozik a puffertúlcsordulási, a „SYN flood", a „teardrop" és a „Smurf" támadás. Sajnálatos módon a hackereknek nem kell túlságosan képzettnek lenniük az ilyen támadások kivitelezéséhez, mert DoS/DDoS eszközök tucatjai állnak rendelkezésükre az interneten.

### *Portletapogatás („port scanning")*

Amint azt már a 7. fejezetben leírtuk, a port a hálózati alkalmazások által két számítógép közötti kommunikációra használt logikai kapcsolattartási pont. A portok számozottak, a különböző alkalmazások pedig különböző portokat használnak. A POP3 (Post Office Protocol) protokoll például, amely az internetszolgáltató kiszolgálójáról az e-mail üzenetek letöltéséhez használható, a 110-es portot alkalmazza. Egy szokásos számítógépes rendszeren 65 536 port áll rendelkezésre.

A portletapogatás technikai értelemben nem támadás, de gyakran annak előfutára lehet. A támadók pásztázószoftver segítségével meghatározzák, hogy a rendszer mely portjai nyitottak, majd az egyik nyitott porton keresztül próbálnak bejutni a rendszerbe. Tűzfal segítségével a szükségtelen portok blokkolhatók (és érdemes is blokkolni őket). A portletapogatás hasonló ahhoz, mint amikor egy betörő minden házba bekopogtat, hogy ellenőrizze, hol nincsenek otthon, azaz hová lehet betörni.

### *Hamisítás („spoofing")*

A hamisítás sem támadást jelent, hanem a támadók által alkalmazott mechanizmust a támadás forrásának álcázására. Az IP-hamisítás a hálózaton át küldött adatok forrás IP-címének meghamisítását jelenti, ezáltal az adatok más számítógépről vagy hálózatról érkezőnek tűnnek. Az e-mail üzenetek hamisítása az e-mail üzenetek fejrészinformációinak módosítását jelenti, ekkor az e-mail a valódi küldő helyett más nevével jelenik meg. A webes hamisítás során a támadók egy webhelyről hamis másolatokat, illetve egy teljes webhelyet készítenek — melyet maguk felügyelnek —, ezáltal az áldozatok valójában a támadó webkiszolgálóját keresik fel, miközben azt hiszik, hogy egy másik kiszolgálón tárolt helyes webhelyre léptek.

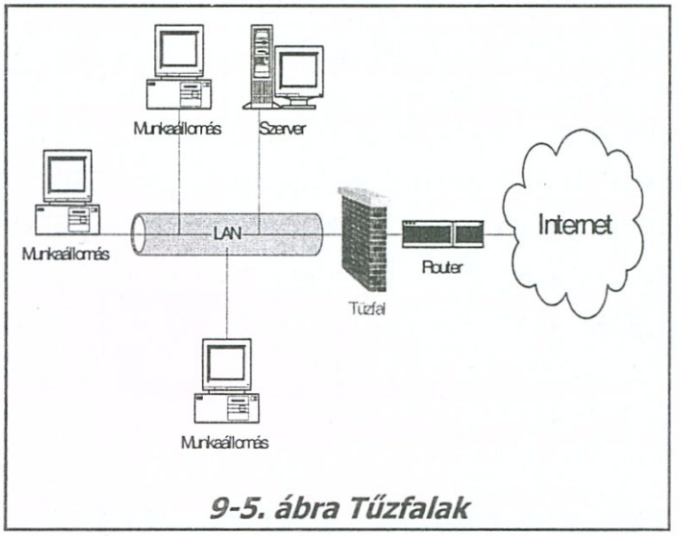
## 9.6.2 Tűzfalak, proxy szerverek

A **tűzfalak (Firewall)** és a proxy-k alapvetően olyan hálózati eszközök, melyek a rajtuk áthaladó forgalmat bizonyos általunk megadott szabályok alapján szűrik.

Általánosságban elmondható, hogy a hagyományos tűzfalak egy-egy IP csomagról pusztán az abban levő információ alapján döntik el, hogy az továbbítható-e vagy sem, illetve milyen egyéb intézkedéseket kell tenni. Azt is mondhatjuk, hogy ezek az eszközök a hálózati rétegben működnek

Egy tipikus csomagszűrő tűzfal a következő információk alapján dönti el, hogy mit kell tenni egy adott IP csomaggal:

- forrás IP cím
- forrás portszám (TCP vagy UDP csomagok esetén) vagy típus (ICMP esetén)
- cél IP cím
- cél portszám vagy típus
- a csomag melyik hálózati interfészen érkezett ill. melyiken szeretne távozni
- egyéb protokollspecifikus információk: pl. TCP SYN, ACK, RST jezőbitek, IP töredékek (fragments). Például a TCP SYN bitek figyelésével, lehetővé válik kapcsolatfelépítési kérések blokkolására, így bár az IP csomagokat mindkét irányban átengedjük, kapcsolatot csak az egyik irányból lehet kezdeményezni.




*9-5. ábra Tűzfalak*

Ha nem csupán az egyedi csomagokat, hanem az egy kapcsolathoz tartozó csomagokat együttesen figyeljük, jutunk el a **Stateful Packet Filter Firewall**-okhoz, melyek **a vizsgálandó csomagok által tartalmazott információn túl képesek figyelembe venni az addig érkezett csomagokban lévő információt is**. Tulajdonképpen állandó kapcsolatokat tartanak nyilván, mégsem mondhatjuk, hogy 4. rétegben dolgozó eszközök, mivel a kezelt elemek továbbra is IP csomagok. Fő előnyük a sebességük, mivel egy létrejött kapcsolathoz tartozó csomagokról jóval könnyebben hoznak döntést. Biztonsági szempontból azonban hátrányuk az, hogy a csomagszintű manipulációknak nem képesek jól ellenállni.

A tűzfal a következő lépéseket teheti, miután megvizsgálta a csomagokat:

- feltétel nélkül továbbítja (ekkor úgy viselkedik, mint egy hagyományos útvonalválasztó) (accept vagy forward néven emlegetik)
- a csomagot eldobja (drop vagy deny)
- a csomagot eldobja, és a feladónak egy ICMP csomagot küld, melyben a csomageldobásról tájékoztatja (reject),
- a csomagot egy helyi portra irányítja át, aminek segítségével átlátszó alkalmázásszintű átjárók (transparent proxy) hozhatók létre
- egyéb kiegészítő lépést tesz: naplóbejegyzést készít, riasztja az adminisztrátort, esetleg megváltoztatja a szűrési szabályokat, vagy külső programot indít (ezeket dinamikus tűzfalaknak hívják)

## Proxy szerverek

**A proxy-k olyan eszközök, ahol a szűrést magasabb rétegben valósították meg, tipikusan az alkalmazási szinten (application layer).** Így lehetőség nyílik a kapcsolatok és a bennük lévő tartalmak finomabb szűrésére, ill. felhasználóra szabott jogosultságok kialakítására. A proxy szerverek szükségszerűen megvárják, amíg az összes olyan IP csomag megérkezik, melyek az alkalmazói szinten összetartozó protokollelemet hordoznak, majd a szűrés után továbbítják azokat.

A proxy-k a finomabb szűrés mellett bizonyos ideiglenes tároló (cache) funkciót is elláthatnak, amiről később még részletesebben lesz szó. El kell azonban mondani, hogy a proxy-k számos előnyük mellett néhány kedvezőtlen tulajdonsággal rendelkeznek. Az egyik – még könnyebben leküzdhető — probléma a nagyobb erőforrásigény (ami elsősorban a finomabb vizsgálatokból ill. az erősen store-and-forward működésből adódik, hiszen nyilván kell tartania az összes aktív kapcsolatot, ill. tárolnia a megérkezett IP csomagokat). A másik probléma elsősorban a felhasználók életét teszi nehezebbé, mivel a legtöbb proxy működése nem átlátszó (transzparens), így az alkalmazásokat fel kell készíteni mind a kód mind a konfiguráció szintjén a proxy-k használatára. A proxy-kat szokás még alkalmazás szintű átjáróknak (application gateway) nevezni.

Maga a proxy szó angolul megbízottat, szolgát jelent. Alapvetően a proxy szervereknek két - részben átfedő típusa van.

1. **Proxy szerver.** A kifejezés abból adódik, hogy van pl. egy hálózat, amiből az összes gép csak egy gépen keresztül látja a külső hálózatokat és a külvilág is csak azon az egyen keresztül látja a benti gépeket. Kifelé induló kérelemkor az összes gépnek a nevében ez az egyetlen gép jár el, aminek többek között az előnye, hogy elég azt az egyetlen gépet konfigurálni, felügyelni. (**tűzfal**)
2. **Cache proxy**: az előzőhöz hasonló, de a gép ki van egészítve egy nagykapacitású merevlemezegységgel, és sok memóriával. Ebben az esetben a szerver valamilyen algoritmus alapjan menti a leggyakrabban lehívott oldalakat (vagy ftp cache proxy esetén a fájlokat) és amikor egy rajta keresztül kapcsolódó gép kér egy oldalt, és azt már lehívta valaki, akkor a távolról történő letöltés helyett a helyi merevlemezén eltárolt példányt tölti le (nagyon gyorsan) a kliens számára.

A tűzfalnál a biztonsági beállítások alapelveként az a legmegfelelőbb módszer, ha alapértelmezetten minden csatornát (TCP és UDP portok) zárva tartunk, és csak akkor és azokat nyitjuk meg, amelyekre okvetlenül szüksége van a rendszer működéséhez. Így jobban követhető, és áttekinthető a tűzfal beállítása, esetleges módosítása.

Fontossága miatt, ma már minden hálózati eszköz, amely csomag továbbítást végez, (router és switch) rendelkezik tűzfal funkcióval.

## A tűzfal konfigurálása

Amikor begépeljük a böngésző címsorába: http://www.ezazahely.hu, akkor a begépelt domén név IP címét a DNS címfeloldással kapjuk meg, és a http protokoll azt jelzi, hogy a 80-as jól ismert port címre kell a kapcsolatfelvételi csomagot küldeni.

Vagyis a böngésző a következő fejlécű csomagot küldi (a címünk adott, a portszám egy kapcsolatkor kiválasztott):

Küldő IP cím és port: **195.56.44.12:4057** címzett: **195.232.34.44:80**

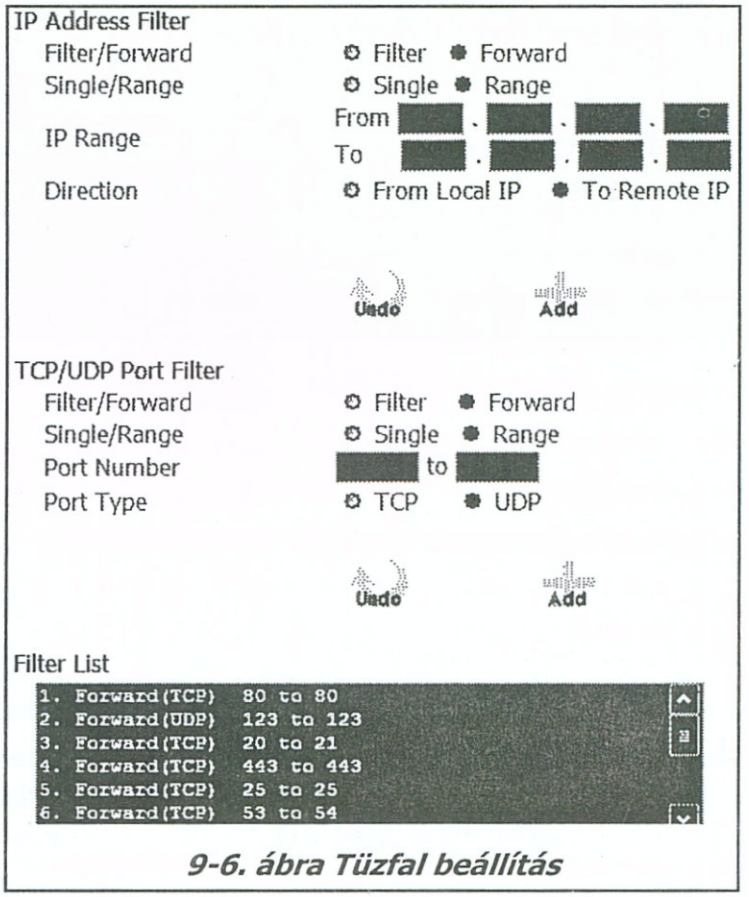
A web szerver által küldött válaszcsomag fejléce:

Küldő IP cím és port: **195.232.34.44:80** címzett: **195.56.44.12:4057**

Vagyis mivel mi a 80-as portcímű csomagokat fogadjuk, akkor csak a webszerverek küldhetnek nekünk csomagokat.

A tűzfal programban, konfiguráláskor a következő adatokat adjuk meg: portok, protokoll-típus (TCP, UDP, ICMP), bejövő-kimenő).

Ha csak böngészünk, akkor az előbbi példánál maradva: az összes kimenő portot engedélyezzük (mert ennek kiválasztása véletlenszerű), és csak a 80-as portra irányított, TCP protokoll által kezelt IP csomagokat engedjük be. A ki- és bemenő forgalmat célszerű a lehető legnagyobb mértékben korlátozni. A legtöbb alkalmazás csak bizonyos távoli portokon keresztül folytat kommunikációt, a böngészők például legtöbbször a 80-as távoli portra csatlakoznak, bár bizonyos szolgáltatások igénybe vételekor a 443-as vagy a 21-es távoli portot is használhatják.




*9-6. ábra Tüzfal beállítás*

**Minimális biztonsági szintet** választva minden szolgáltatás és alkalmazás számára engedélyezi az adatok fogadását vagy küldését, hacsak nincs ezzel ellentétesen rendelkező, tiltó jellegű szűrési szabály. A minimális biztonsági szint azon felhasználók számára a legjobb választás, akik saját szabályaikat szeretnék létrehozni.

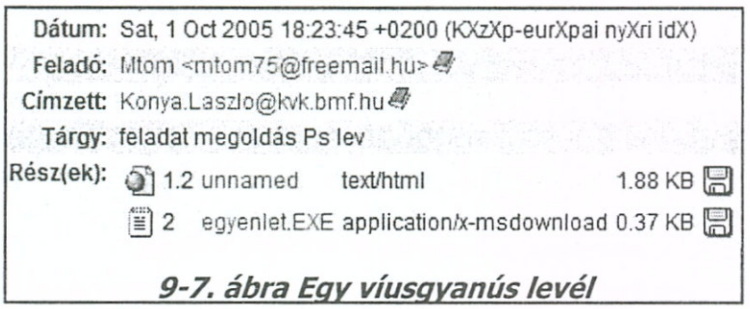
**Közepes szintű biztonság** az alapértelmezett beállítás, és a legtöbb felhasználó számára megfelelő. A minimális biztonsággal ellentétben közepes szintű biztonságnál a tűzfal minden IP forgalmat letilt, és a felhasználótól kérdezi meg a megfelelő beállításokat. A könnyebb használat érdekében a tűzfalban már telepítés után is található néhány szűrési szabály, melyeket a felhasználók saját belátásuk szerint törölhetnek, vagy meghagyhatnak. Ha egy kapcsolatra nincs szabály, akkor a tűzfal rákérdve megtanulja a használat közben a szabályokat. Vagyis minden alkalommal, amikor ismeretlen csomagot talál, a tűzfal megkérdezi a felhasználótól, hogy engedélyezi, vagy letiltja azt.

**Maximális biztonságot** választva az összes szűrési szabályt és az alkalmazásokhoz már megadott engedélyeket alkalmazza, de a tűzfal új szabályt létrehozó kérdése nem jelenik meg. Ha egy csomag nem felel meg egyik engedélyező jellegű szabálynak

sem, a tűzfal eldobja. A maximális biztonság működése éppen ellentétes a minimális biztonságéval. Minden forgalom le van tiltva, kivéve azt, ami megfelel valamelyik engedélyező jellegű szabály előírásainak.

## 9.6.3 Vírusok elleni védekezés

Jelenleg a számítógépet használók mindennapos gondja a vírusprogramok elleni védekezés. A számítógép legértékesebb részét az adatok jelentik. Ezek lehetnek táblázatok, szerkesztett szövegek, saját programok, nagyobb adatbázisok, vagy esetleg ott lehet a cég teljes könyvelése. Az adatok értéke mellett eltörpülhet a hardver és a szoftver értéke. Egyes emberek szántszándékkal, az adatoknak és programoknak a megsemmisítésére készítenek (erkölcsileg nagyon alacsony, programozói tudásban elég magas színvonalon állva) sajátos programokat.




*9-7. ábra Egy víusgyanús levél*

A vírusfertőzés három forrása:

1. Cserélhető adathordozók (floppi, hordozható merevlemez, USB-pen, CD/DVD),
2. E-mail-ek mellékletei,
3. Internetes letöltések, hálózat használat

### *Mi is a vírus?*

Olyan program, amelyik képes önmagát reprodukálni. Más szoftverek megfertőzésével, esetleg formázott floppyval vagy magával a géppel terjednek. (Ezeket nevezik hardvervírusoknak.) Terjedési forrás lehet ma már a számítógépes adatátviteli hálózat is. Röviden: **a vírus automatikusan terjedő, kár okozására alkalmas program.**

### *A vírus felépítése*

Minden vírus általában négy programrészből áll:

- **Az első rész egyfajta ismertetőjegy, amelyről a vírus felismeri önmagát.** Ennek a segítségével tudja bármikor ellenőrizni, hogy meg van-e már fertőzve egy fájl.
- **A második rész tartalmazza a tulajdonképpeni fertőző programot.** Először egy, még nem fertőzött, végrehajtható fájlt keres. Ha talál ilyet, a vírus bemásolja a programkódját a fájlba. Ebben a részben található a programkód is, amely szükség esetén úgy alakítja át a fájlt, hogy a vírus a program indításakor azonnal aktiválódni tudjon. Az esetleges álcázási eljárás szubrutinja is itt található.
- **A harmadik rész dönti el, hogy ártalmatlan vírusról van-e szó**, „aki" csak tréfálkozik, vagy egy romboló kártevőről. Ártalmatlan esetben itt található az utasítás, hogy a vírus, mondjuk egy adott napon rajzoljon egy képet a monitorra, vagy írjon ki egy meghatározott szöveget. A vírusnak ezt részét szokták objektív rutinnak nevezni.
- **A negyedik részben található az a parancs**, amellyel a program a víruskód végrehajtása után visszatér oda, ahol a vírus megszakította a program futását.

### *Hogyan fertőz meg egy vírus egy fájlt?*

Sok vírus egy futtatható fájl végéhez fűzi a saját programkódját, az elejére pedig egy hivatkozást helyez el erre a kódra. Ha a programot elindítják, az a saját feladatainak a futtatása előtt először a vírusprogramra ugrik. Ha ezt végrehajtotta, megint visszaugrik arra a helyre, ahol eredetileg megszakította a folyamatot. Így tehát minden alkalommal, mikor a programot elindítják, először a vírus indul el, és ettől a perctől kezdve még meg nem fertőzött fájlokat keres, hogy azokhoz is hozzáférkőzzön, és megfertőzze őket.

A víruskódnak a fertőzött fájlhoz fűzése nem minden esetben okoz maradandó károsodást, a vírusokat el lehet távolítani. Egyes vírusok azonban sokkal veszélyesebben viselkednek, és egyszerűen átírnak a fájlból annyit, amennyire a programkódjukhoz szükség van.

### *Legfontosabb vírustípusok:*

A vírusfigyelmeztetések során általában azt is megadják, hogy milyen típusú vírusról van szó. Ez fontos, mert a „fertőzésveszély" a típustól függően nagyobb vagy kisebb lehet. Létezik a katagolizált vírusoknak egy rendszeresen frissített hivatkozási listája: a Vírus Bulletin, ami az interneten a http://www.virusbtn.com címen érhető el. Az alábbiakban egy rövid áttekintést adunk a legfontosabb vírustípusokról és sajátosságairól.

Több szempont szerint csopotosíthatók:

- **Fertőzési célpont alapján** (boot/partíció-, programvírusok)
- **Szaporodási szempont alapján:** -nem felülíró (a fertőzött fájl programkódját nem írja felül),-felülíró
- **Memóriával való viszony alapján** -Rezidens (=memóriában maradó,) –Nem rezidens
- **Az objektív rutin alapján:** -nem romboló (jópofa), szoftver, vagy hardver romboló.
- **Generációs típus szerint:** -első generációs, -lopakodó (stealth) (a rendszer általa megváltoztatott részeit változatlannak mutatja) , -polimorf (mindig változik), makróvírusok.
- **Szaporodási sebesség szerint**: -lassan fertőző,-gyorsan fertözö
- **Visszafejthetőség szerint:** -könnyen megfejthető, nyomkövető program billentyűit letiltó, víruskód titkosító vírusok.

**Bootszektor/partíciós tábla vírusok:** Az ilyen vírusok a hajlékonylemezek bootszektorát és/vagy a master boot record-ot (MBR), illetve a master boot partíciós szektort vagy a DBR-t, a DOS boot recordot, illetve DOS-bootszektorát fertőzik meg a merevlemezen. A különböző boot-szektor vírusok működési módjai a részleteikben ugyan különböznek egymástól, de az alapelv mindegyiknél ugyanaz. Semmi jelentősége nincs, hogy milyen operációs rendszert használunk, vagy hogy milyen víruskereső programot telepítettünk rá, mert abban a pillanatban, mikor a bootszektor vírus telepíti magát, az operációs rendszer vagy a védőprogram még egyáltalán nincsen betöltve.

**Companion vírusok:** Ha egy *.com és egy *.exe fájlnak ugyanaz a neve, és ezt a nevet begépeljük, a DOS először mindig a COM fájlt hajtja végre. A companion vírusok is ezt a körülményt használják ki. Az EXE fájlhoz készítenek egy azonos nevű COM fájlt, amelyben benne van a víruskód. Az EXE fájl elindítása helyett, a COM program indul el. Majd a vírus készít egy újabb companion vírust egy újabb fájlhoz.

**Polimorf vírusok:** A polimorf vírus olyan kártevő, amelyből egy helyen nem fordul elő két másolat, amelyek ugyanazt a bájtsorozatot tartalmazzák. Ezért egy ilyen vírust nem lehet egyszerűen egy meghatározott bájtsorozatról felismerni, ennél sokkal összetettebb és nehezebb feladatot kell megoldania a víruskereső programozóknak.

**Rejtőzködő vagy lopakodó vírusok:** Ha egy vírus tartóssan a memóriába tud maradni (memóriarezidenssé tud válni), ami a legtöbb vírusra igaz, akkor a renszermegszakításokon keresztül fertőz. A lopakodási

technikát egyes makróvírusok is felhasználhatják: eltüntetik az adott alkalmazás azon menüpontjait, amelyen keresztül meg lehet nézni az aktív makrókat.

**Számítógép férgek – modern kori vírusok:** Ezek a vírusok önállóan tudnak egy hálózaton terjedni. Ezeknél nem klasszikus vírusokról van szó, hanem azzal rokon zavaróprogramokról, amelyek azonban vírust is tartalmazhatnak. A férgek önálló programok, amelyeknek nincs szükségük arra, hogy gazdaprogramokhoz fűzzék magukat. Többnyire több, egymáshoz kapcsolódó programszegmensből állnak. A komputerférgek saját magukat tudják reprodukálni, és hálózati funkciók segítségével más számítógépekre másolódnak.

**Makrovírusok**: Olyan vírusok, amelyek adatfájlokat fertőznek meg. Amint megnyitunk egy fertőzött Word dokumentumot, az megfertőzi a *Normal.dot* fájlt. Ha ezután egy dokumentumot mentünk vagy nyitunk, az is megfertőződik a vírussal.

### *Honnan ismerjük fel a vírusfertőzést?*

Egy vagy több jelről: a programok hossza, létrehozási dátuma megváltozik, megváltoznak könyvtárbejegyzések, rejtélyes állományok jelennek meg, lassabban futnak a programok, gyakran lefagynak, a képernyőn szokatlan ábrák jelennek meg, eltűnnek végrehatható állományok, gyakran nyúl a rendszer a merevlemezhez, elvesznek fájl vagy lemeztartalmak, stb.

Összefoglalva: valami furcsa lesz a gépen. **Azonban vigyázzunk: az emberek hajlamosak a közel egyszerre bekövetkező független eseményeket összekapcsolni!**

### *Hogyan működnek a víruskereső programok?*

A víruskereső egy adatbázist használ (ez általában egy *.dat fájl), amely tartalmazza a vírus nevét és lenyomatát. Ezt a letöltött vírusadatbázist telepítve, vagy online interneten kersztül frissíthetjük naprakészre víruskereső programunkat.

Néhány kapcsolatos fogalom:

**Szkennermodul** (szignatúra, szekvencia vagy bájt-minta keresés). A szkenner ellenőrzi as a vírus adatbázisban szerplő bájt mintákkal hasonlítja össze egy fájl vagy program kódját. Ha a szkenner megváltoztatott kódot vagy valamilyen egyéb eltérést észlel, akkor riasztást ad.

**Víruspajzs**, amely automatikusan elindul minden egyes rendszerindításnál. Ez egy memóriarezidens, a háttérben működő szkennermodul. Valós időben, folyamatosan követi a felhasználók minden fájlelérését.

Ha egy fájlt elindítanak, kijelölnek, átnevezik, vagy letöltik a hálózatról, a víruspajzs azonnal megvizsgálja, és ha szükséges, jelentést tesz.

**Vírus eltávolítás.** Ha a szkenner vírust talál, és ha a „vírus megtisztítása az adott fájlból" opciót jelöltük be, akkor eltávolítja. Ha ez nem lehetséges, felajánl további opciókat, amellyel egy külön helyre (karanténba) tehetjük, vagy törölhetjük a fájlt.

### *Vírusirtás*

A vírussal megfertőzött, Windows XP operációs rendszerű gépet, a következő lépések végrehajtásával kell megtisztitani egy meglévő vírusirtó program segítségével.

1. Először a **vírus definíciós adatbázisát kell frissíteni**. Ez a legtöbb korszerű virusírtó program a hálózati Internet kapcsolaton keresztül automatikusan megtörténik.

2. **Kikapcsoljuk** a Windows alapértelmezett **rendszervisszaállítás** opcióját. (*Vezérlőpult>Rendszer>A rendszer visszaállítása fül)*
3. A Windows **indítása csökkentett módban** (Újraindításkor F8 gomb nyomvatartása.)
4. **Vírusirtó program futtatása** kézi, vagy automatikus opcióval.
5. **A gép újraindítása.** Sikeres vírusirtás után a számítógép merevlemezei már nem tartalmaznak vírust, vagyis újraindításkor már nem indul el kártékony programkódok futása.

## *9.6.4 Trójai programok (nevét a trójai faló-ról kapta)*

Látszólag hasznos programok, amelyek azonban tartalmaznak (és ezt álcázzák) olyan rejtett képességeket, amelyek veszélyeztetik a rendszerünk biztonságát. A kutakodó változatait szokták ezeket más néven **spyware**-eknek (kémprogramnak), míg a képernyőt kéretlen hírdetéssel megtöltő programokat **adware**-nek nevezni. Az általuk nyújtott "szolgáltatások" egy rövid összefoglalása:

**FTP szerver** A trójai "FTP szerver" szolgáltatást telepít a számítógépen, ami lehetővé teszi a gép merevlemezén történő kutatást, fájlok letöltését (ellopását) és elhelyezését gépen.

**Billentyűzetleütés figyelő(keylogger):** A trójai program működése során feljegyzi a billentyűzetleütéseket, és azokat összegyűjti egy fájlba, majd a hálózaton keresztül elemzésre saját magának letölti. Ha jelszavakat, számlaszámokat gépelünk, akkor...

**Ál dialógusablak**: A képernyőn szabályosnak tűnő dialógusablak jelenik meg, különböző információkat kérve a felhasználótól. A kitöltött ürlap a megszerzett információkkal egy fájlba tárolódik, majd a trójai elhelyzője magának letölti.

**Távoli parancsvégrehajtás (telnet):** A trójai program lehetővé teszi, hogy a támadó a távolból parancsokat adjon a számítógépünknek.

**Automata trójai:** Aktivizálódás után a program az előre beállított információkat gyűjti össze (felhasználói név, jelszó, IP cím, stb.), majd a böngészőnket a támadó által létrehozott weblapra irányítja, és a weboldalba épített kérdőívet (form-ot), ezekkel az adatokkal automatikusan kitölti és elküldi magának.

**E-mail küldő trójai**: Hasonló az előzőhöz Gyakorlatilag a fent említett automata trójai azon változata, amely az összegyűjtött információk "hazajuttatását" nem egy web-lapon keresztül, hanem egy beépített e-mail küldő program segítségével oldja meg.

Sok esetben a faló úgy kerül a rendszerünkre, hogy a gyanútlan felhasználó letölt az Internetről egy programfrissítést.

Érdekes, hogy varez oldalakon kutakodva, az esetleges megtalált „érték" letöltését ahhoz kötik, hogy valami kiegészítő programot töltsük le. Még installálni sem kell, mert azt magától teszi. Ugye kitaláljuk, hogy mi fog történni?

A trójai falovak ellen vírusirtóval védekezhetünk, de vigyáznunk kell, mert vannak köztük olyanok, amelyek vírusirtónak adják ki magukat, ezért a legjobb megoldás a rendszer megtisztítása és biztonságos forrásból való újratelepítése.

**FONTOS!** Ha internetre kapcsolódunk, akkor feltétlenül, ha nem, akkor is ajánlatos folyamatosan, rendszerindításkor elinduló víruskereső programot telepíteni a számítógépekre!

### 9.6.5 Programférgek

Az operációs rendszerekben megmaradó biztonsági réseket, és a hálózati kapcsolatokat kihasználva terjednek gépről-gépre. Itt nincs külön hordozóprogram, maga a program rendelkezik önmagát reprodukáló képességgel. A féregtelenítések mellett nagyon fontos a megtalált biztonsági rések befoltozása is.

## Ellenőrző kérdések: 9. Fejezet

1. Mi hálózati biztonság három alappillére?
2. Mit jelent az autentikáció magyarul?
3. Fogalmazza meg néhány mondatban, hogy mit jelentenek a hálózati biztonsággal kapcsolatosan a következő területek: Titkosság, Hitelesség, Letagadhatatlanság, Sértetlenség?
4. Foglalja össze a biztonsággal kapcsolatos emberi tényezők szerepét!
5. Milyen biztonsági szinteket lehet egy hálózatban kialakítani?
6. Mit értünk fizikai biztonság alatt?
7. Foglalja össze a digitális aláírással kapcsolatos ismereteket!
8. Mi az a „közbülső harmadik probléma"?
9. Mik azok a tanúsítványok, és mire használhatók?
10. Mik azok a virtuális magánhálózatok (VPN)? Milyen összetevői vannak? Hogyan valósítják meg a gyakorlatban? Mi az a throughput?
11. Mi az a biztonsági alréteg (SSL)? Hogyan működik?
12. Mik azok a jogosultságok? Kire, vagy mire vonatkoznak? Hogyan kell kezelni ezeket? Mi a csoportszintű jogosultság?
13. Foglalja össze a jelszavakkal kapcsolatos ismereteket! Hogyan kezeljük, milyenek legyenek? Hogyan fejthetők meg?
14. Foglalja össze a biztonsági másolatokkal kapcsolatos ismereteket!
15. Mi az a naplózás, és milyen naplók vannak? Ezeket hogyan kezeljük? Mit lehet naplózni?
16. Hogyan lehet a kivülről jövő támadások ellen védekezni? Kik azok a hackerek?
17. Milyen támadástípusok vannak? Mi az a DoS, illetve DDoS? Mi az a portletapogatás és a hamisítás?
18. Mik azok a tűzfalak, hogyan működnek és milyen típusai vannak?
19. Mik azok a proxy szerverek? Milyen két típusa van?
20. Hogyan kell egy tűzfalat konfigurálni? Milyen biztonsági szinteket tudunk beállítani?
21. Mik a vírusok, hogyan épülnek fel, és hogyan védekezünk ellenük?
22. Hogyan fertőznek a vírusok? Hogyan ismerjük fel ezt?
23. Hogyan működnek a víruskereső programok? Hogyan tudjuk a vírust kiírtani?
24. Mik azok a trójai programok? Milyen „szolgáltatásokat" nyújtanak?
25. Mik azok a programférgek?

# 10. HÁLÓZATI FELÜGYELET

*Ha már minden kisérlet csődöt mondott, olvasd el a használati utasítást (Murphy)*

A számítógépes hálózatok nagysága és összetettsége (komplexitása) folyamatosan növekszik. Ezért fontosak azok a megoldások, amelyek az informatika felhasználásával tudják a rendszereket kezelni.

## 10.1 Hálózati felügyelet típusai (menedzsment)

Erre együtt nincs magyar szó, azért használjuk **hálózati menedzsment** kifejezést.

Amikor a hetvenes években az első számítógépes hálózatok megjelentek, ezek kicsi, egymással össze nem kötött hálózatok voltak. Ezeket könnyű volt átlátni, karbantartani és javítani. Ám amikor növekedni kezdtek, összekapcsolódtak, szövevényessé váltak, megjelentek a **hálózatfelügyelő** és **hálózatfigyelő** tevékenységek.

A hálózati menedzsment több tevékenységből tevődik össze.

- **Hiba menedzsment** alatt a hálózat működési zavarainak érzékelését, behatárolását és kijavítását értjük.
- A **teljesítmény menedzsment** feladata a kiszámítható és hatékony szolgáltatás biztosítása, a jelenlegi kapacitások tervezése és tesztelése, valamint jövőbeli bővítése.
- A **konfigurációs menedzsment** célja a rendszeresen jelentkező változások (ahogy a rendszer növekszik, gépek cserélődnek, stb.) és az alkotóelemek nevének és címeinek helyi adminisztrációja.
- A **biztonsági menedzsment** biztosítja a rendszer alkotóelemeinek fizikai védelmét, valamint a fontos információk védelmét a különböző akaratlagos és a szándék nélküli sérülésektől. Jelenleg, a kommunikációs rendszerek szerepének és fontosságának a növekedésével a biztonságkezelés szerepe egyre fontosabb.

A tevékenységeket egységesítették, szabványosították, példaként egy jellemző, a TCP/IP hálózatokban elterjedt szabványos protokollt mutatunk be.

## 10.2 SNMP (RFC1155-1158, RFC1213)

A kidolgozott protokoll neve: **Simple Network Management Protocol – SNMP**, magyarul: egyszerű hálózatfelügyeleti protokoll. Az SNMP lényegében egy a hálózatfelügyelet alapjait magába foglaló hálózati kommunikációs meghatározásgyűjtemény. A hálózati környezet karbantartását, megfigyelését, módosítását (egyszóval: menedzselését) írja le. Olyan módon tervezték, hogy a meglévő hálózatot a használatához ne kelljen nagyon megváltoztatni. Ezt úgy érik el, hogy a cserélt üzenetekben hálózati adatokat küldünk. Az IP réteget használja a csomagjainak elküldésére, és a 161-162 porton kommunikál, az UDP protokollt használva.

**A felügyelt objektumok** egy gép (vagy hálózati eszköz) azon erőforrásai, amelyeket a hálózat egy másik helyéről lehet lekérdezni, vezérelni. Ezen adatok egy **felügyeleti**

**adatbázis**-ban vannak, és tartalmukat a **felügyeleti konzol**-on keresztül lehet lekérdezni.

**Az SNMP protokol az SNMP menedzser és SNMP ügynök fogalmán illetve együttműködésén alapul.**

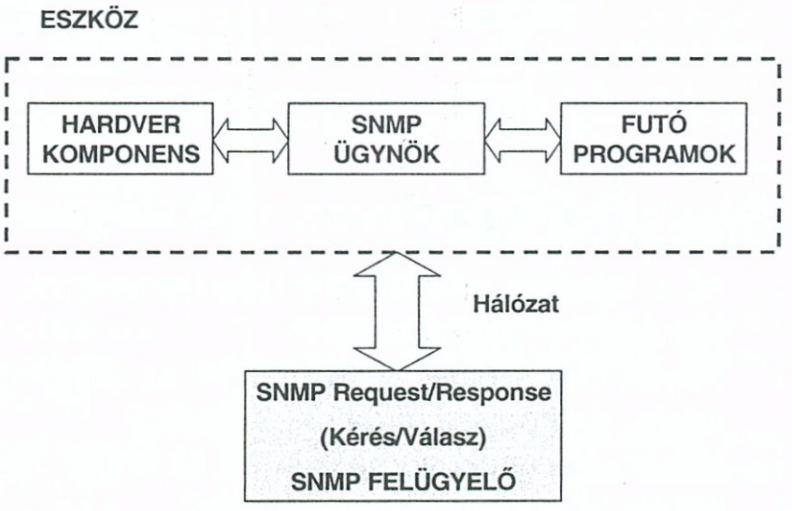
Az **SNMP ügynök** (angolul: **agent)** olyan szoftver, amely gyűjti a hálózat adott helyén az adatokat, és válaszol az **SNMP menedzser** kérdéseire. Az ügynök program futhat munkaállomáson, HUB-okon, routereken, kapcsolókon (switch). Az SNMP ügynököt két alapvető szerepben használják: ez a figyelés és a vezérlés, valamint a beavatkozás.

A hálózati menedzsment protokoll architektúrálisan két részből áll. Az első a kommunikációt kezeli, míg a második a vezérelhető adatokkal foglalkozik. Két szabvány van használatban: a **Simple Network Management Protocol** (**SNMP**) és a **Common Management Information Services**(**CMIP**) a TCP felett.

Mindkettő a **Management Information Base (MIB)** adatbázist használja

- **Figyelés**-kor, amikor egy ügynök fut valamelyik gépen, akkor figyelő szerepben a hardvert és a futó programokat figyelheti. Ha az SNMP menedzser lekérdez egy figyelt információt, akkor az ügynök vagy egy előzőleg gyűjtött és tárolt információt ad vissza, vagy a parancs hatására lekérdezi az eszközt vagy a futó programját. Ez egy passzív folyamat, az ügynök nem avatkozik be a figyelt egység működésébe. Az ügynök önmagától csak ún. **csapda** esemény esetén küld automatikusan információt. Ez lehet az eszköz indulása, leállása, túlterhelése (pl. egy lemezegység megtelt...).
- **Vezérlés** esetén az ügynök közvetlen tudja irányítani az eszköz hardverét, vagy szoftverét. Ez adott esetben veszélyes is lehet.

A hálózati részek szétválasztása érdekében az SNMP konfigurálásakor a hálózatot ún. közösségekre osztják fel. Egy közösségbe több felügyelő és ügynök tartozhat.




*10-1. ábra: SNMP vázlat*

Amikor egy ügynökhöz egy kérés érkezik, először mindig azt ellenőrzi, hogy a kérés a saját közösségébe tartozó felügyelőtől jött-e. Ha igen akkor válaszol, ha nem akkor esetleg egy úgynevezett „csapda esemény" küldésével jelzi saját felügyelőjének a kommunikációs kísérletet. Egy ügynök három összetevőből áll:

- A protokollt kezelő és megvalósító részből,
- a figyeléshez szükséges komponensekből,
- valamint a gyüjtött adatokat tartalmazó **menedzsment információs adatbázisból (Management Information Base=MIB).**

Minden információ elérés az ügynökön keresztül az információs adatbázis elérését jelenti. A MIB írásán olvasásán keresztül valósul meg az ügynök-menedzser együttműködés.

A MIB-ek a következő információt tárolják a figyelt objektumokról:

- Az eszköz adateleméhez rendelt név,
- az adatelem típusának leírása,
- szöveges leírása,
- sorszám,
- az elem elérhetősége.

Az SNMP menedzser feladata az adott ügynök információinak az elérése, módosítása, amelyet a MIB alapján végez. A menedzser egy munkaállomáson futó program, amely a hálózati kapcsolaton keresztül valósítja meg a kapcsolat tartását az ügynökökkel.

| MIB kategória | Miről tartalmaz információt |
|---|---|
| system | hoszt és az útválasztó operációs rendszere |
| Interfaces | az egyes, egyedi hálózati interfészekről |
| addr.trans | cím-transzformációról(pl. ARP leképezés) |
| ip | az internet protokoll szoftverről |
| Icmp | az internet ICMP szoftverről |
| tcp | az ineternet TCP szoftverről |
| udp | UDP szoftverről |
| egp | EGP szoftverről |

HÁLÓZATI MANAGE-MENT- MIB

A kategóriákhoz tartozó változókra néhány példája:

| MIB változó | Kategória | Jelentés |
|---|---|---|
| sysUpTime | system | A legutolsó boot óta eltelt idő |
| ifMtu | interfaces | Az adott interfész MTU-ja |
| ipDefaultTTL | ip | Az IP által használt TTL érték |
| ipInReceives | ip | A vett datagramok száma |
| ipForwDatagrams | ip | A továbbított datagramok száma |
| ipOutNoRoutes | ip | Az útválasztási hibák száma |
| ipFragOKs | ip | A fragmentált datagramok száma |
| ipRoutingTable | ip | IP útválasztási tábla |
| icmpInEchos | icmp | A vett ICMP echo kérések száma |
| tcpMaxConn | tcp | A TCP maximális újraküldési ideje |

Az adatok ábrázolását a Structure of Management Information (SMI) specifikálja az ún. Abstract Syntax Notation (ASN.1) segítségével. Az ASN.1 hierarchikus felépítésű jelölésrendszer segítségével.

*10-2. ábra: MIB adatok*

A MIB a rendszer információk leírásához nyolc kategóriát definiált *(10-2.ábra)*, és a kategóriákhoz szabványos elnevezésű változókat.

Ma már egyre több, még olcsóbb árkategóriájú hálózati eszközbe építik be ezt a menedzselési lehetőséget.

## Ellenőrző kérdések: 10. Fejezet

1. Mi a feladata a hálózati menedzsmentnek, és milyen részekből áll? Miért kell alkalmazni a hálózatokban?
2. Milyen célt szolgál az SNMP? Mi az SNMP ügynök, és mi az SNMP menedzser feladata, és hogyan működnek együtt?
3. Milyen szerepekben használják az SNMP ügynököt? Mi az a csapda esemény?
4. Milyen részekből áll egy ügynök?
5. Mit jelent, és hogyan épül fel a MIB?
6. Milyen információkat tárolunk a megfigyelt objektumról a MIB-ben?

***Vége a könyvnek...***

*Hiszek abban, hogy az életünk nem a kifürkészhetetlen véletlenek szövevénye,*
*hanem a következménye öröklött génjeinknek, valamint mindannak, amit létrehívunk gondolatainkkal,*
*cselekedeteinkkel, megformálunk jellemünkkel, szokásainkkal és neveltetésünkkel,*
*nem kizárva a véletlen és a váratlan történések irányformáló erejét.*

# IRODALOMJEGYZÉK

Az irodalomjegyzék csak egy rövid válogatás a kapcsolódó számos könyvből, a teljesség igénye nélkül.


| # | IRÓ ÉS CÍM | KIADÓ | MEGJEGYZÉS |
|---|---|---|---|
| [1] | Tanenbaum A. S.: Számítógép - hálózatok | Panem-Prentice Hall, Budapest, 1999. ISBN 963 545 213 6 | A legteljesebb összefoglaló könyv új kiadása a számítógép-hálózatokról. A könyvet az OSI rétegmodell alapján építi fel. Egy-egy témakör részletesebb megismerését célszerű e könyv vonatkozó részével kezdeni. Igen részletes irodalomjegyzék is található a könyvben. |
| [2] | Davies D.W.-Barber D.L. A-Price W.L.-Solomonides C.M.: Számítógép - hálózatok és protokollok | Műszaki Könyvkiadó, Budapest, 1982. ISBN 963 10 4328 2 | Egy kissé régi, de szintén alapmű. A legértékesebb és mai is időszerű része a HDLC és az X.25 protokollok igen részletes leírása. Szakkifejezés gyűjtemény és a Függelékében a különféle szabványok felsorolása is megtalálható. |
| [3] | Davies D.W.-Barber D.L. A: Számítógép - hálózatok | Műszaki Könyvkiadó, Budapest, 1978. ISBN 963.10.1813.X. | Ez még régebbi mű az előző könyv szerzőpárosától, de sok megállapítása ma is igaz. Tanulságos, mert az áramköri működési leírásoknál TTL áramköröket használ illusztrációknak. Minden lényeges kérdést a kor akkori technikai színvonalán tárgyal. |
| [4] | Martin J. - Chapman, K. : Lokális hálózatok | Novotrade, Budapest, 1992. ISBN 963 585 163 4 | Az összefoglaló könyv a lokális számítógép-hálózatokkal foglalkozik. A könyvet az OSI modell segítségével építi fel. |
| [5] | Cseh Kálmán: IBM PC alapú helyi hálózatok | Számalk, Budapest, 1989. ISBN 963 553 217 2 | Bár nem egy új kiadás, de nagyon jól foglalja össze a ma is aktuális alapismereteket. Elég részletesen foglakozik a hálózati hardver kialakítással is. |
| [6] | Móricz Attila: NOVELL hálózati alapismeretek I. Felhasználóknak. | LSI Oktatóközpont. Budapest, 1994. ISBN 963 084 7 | A címben szereplő operációs rendszer bemutatása példákon keresztül, mintha a gép előtt ülnénk. |
| [7] | Móricz Attila: NOVELL hálózati alapismeretek II. Rendszergazdáknak. | LSI Oktatóközpont. Budapest, 1994. ISBN 963 083 9 | Talán az első magyar nyelvű könyv, amely a rendszergazdáknak szóló ismereteket tanítja meg. |
| [8] | Hakala, David: Modemek | Panem-McGraw-Hill, Budapest, 1993. ISBN 963 545 013 3 | A modemekről szóló ismeretek kissé terjengős formában, felhasználóknak. |
| [9] | Füstös János: World Wide Web - Bevezetés a hálózati információszolgáltató rendszer tervezésébe és használatába | Szak Kiadó Kft, Bicske, 1996. ISBN 963 855 6404 | Egy nagyon figyelemreméltó könyv, amely a WWW-vel kapcsolatos praktikus és elméleti ismeretekkel foglalkozik. |

| | | | |
|---|---|---|---|
| [10] | Rét András-Svéd János: Távadatfeldolgozó rendszerek | Műszaki Könyvkiadó, Budapest, 1981. ISBN 963 10 3601 4 | Egy könyv az ESZR korszakból. Számos része elavult, de az információ-csere eljárásait, a BYSINC és HDLC protokoll lényegét jól leírja. |
| [11] | Rendszergazda alapismeretek | LSI Oktatóközpont. Budapest, 2003. ISBN 963 577 334 X | A három operációs rendszert: Unix, Windows, Novell Netware írja le, elsődlegesen a rendszergazdáknak szánva. Jó könyv. |
| [12] | Kovács Péter: Számítógép-hálózatok | Computerbooks, Budapest, 2003 ISBN 963 618 313 9 | Praktikus utmutató hálózatépítéshez, Internetes kapcsolatok beállításához, vezetékes és vezeték nélküli technológiákhoz. Jó! |
| [13] | Kis Balázs: Windows XP haladó könyv | Szak Kiadó, 2003 ISBN 963 9131 58 X | Amit a Windows XP hálózatokról tudni kell, az itt megtalálható |
| [14] | Comer, D. E.: Internetworking With TCP/IP | Prentice Hall, Englewood Cliffs 1995. ISBN 0-13-216987-8 | Alapmű a TCP/IP-ről angol nyelven Nagyon jó könyv a windows alapú hálózati programokról, elméleti és gyakorlati példákkal |
| [15] | Dr. Farnady László: Számítógépes hálózatok | Széchenyi István Főiskola, Győr, 1995 | A konkurencia... Nagyon jó könyv, bár a tematika egy kissé más. A legjobb rész (számomra) a TCP/IP-ről szól |
| [16] | Szász Gábor-Kun István-Zsigmond Gyula: Kommunikációs rendszerek | LSI Oktatóközpont, Budapest, 1999. ISBN 963 577 250 5 | A kommunikációról szóló, azt sok gyakorlati példával magyarázó, jól tanulható könyv. |
| [17] | Móricz Attila: Webdesign a gyakorlatban | Computerbooks, Budapest 2003 ISBN 963 618 302 3 | Praktikus, gyakorlati útmutató weboldalak szerkesztéséhez. |
| [18] | Lamár, K.-Antal, G.: Modern Solutions to Integrated Building Automation Systems | Proceedings of the International Konference „Kandó 2002" Bp. Hungary | p5.2002.- ISBN 963 7158 03 0 A csak megemlített mikrovzélő részhez |
| [19] | Lamár Krisztián. A világ leggyorsabb mikrovezélője | ChipCAD Kft, 1999 | Az SX mikrokontroller alkalmazástechnikája A csak megemlített mikrovzélő részhez |
| [20] | McMahon,R.A,: PC hálózatok a gyakorlatban | Panem, Budapest, 2004 ISBN 963 545 401 5 | Az elméleti bevezető után, a Novell Netware, és a Windows XP rendszereket tárgyalja. |

ISBN 963962516-7

9 789639 625167