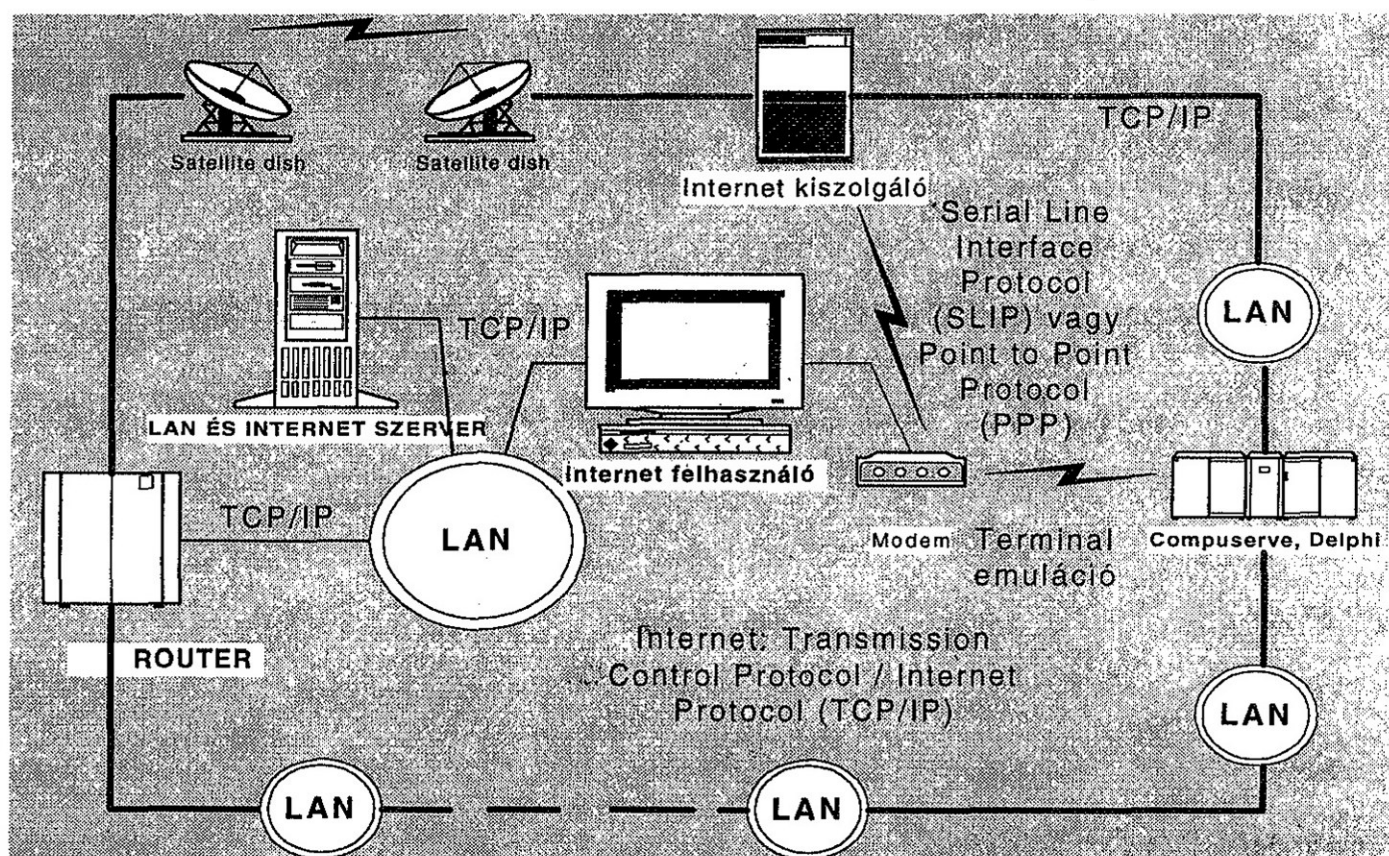


Kónya László

Számítógép-hálózatok



Nyitott rendszerű képzés - Távoktatás –
Oktatási segédlete

Felsőoktatási tankönyv

LSI Informatikai Oktatóközpont
A Mikroelektronika Alkalmazásának
Kultúrájáért Alapítvány

Budapest, 2002

Lektorálta: Dr. Hosszú Gábor
a műszaki tudomány kandidátusa,
egyetemi docens
Dr. Németh István
informatikai szakértő
Kőrössi Ádám
Kántor Zoltán

**A könyv megrendelhető illetve megvásárolható az
LSI Informatikai Oktatóközpontban
1037 Budapest, Bécsi út 324.
Tel/fax: 436-6521**

ISBN 963 577 222 X

Kiadó: LSI Oktatóközpont
Felelős vezető: Dr. Kovács Magda
Témafelelős: Flier István

LIGATURA KFT. – NASZÁLY PRINT KFT.

TARTALOMJEGYZÉK

TARTALOMJEGYZÉK	3
BÉVEZETÉS	8
1. A HÁLÓZATOK CÉLJA, ALKALMAZÁSA, ALAPFOGALMAK	13
HÁLÓZATI STRUKTÚRÁK	14
Két pont közötti csatornával rendelkező alhálózat (pont-pont összeköttetés).....	15
Üzenetszórásos csatornával rendelkező alhálózatok (multipont összeköttetés).....	16
HÁLÓZATI ARCHITEKTÚRÁK	17
HÁLÓZATSZABVÁNYOSÍTÁS	19
AZ OSI MODELL	21
SZOLGÁLATOK A RÉTEGEK KÖZÖTT	25
Összeköttetés alapú szolgálat.....	27
Összeköttetés-mentes szolgálat.....	27
ELLENŐRZŐ KÉRDÉSEK	28
2. FIZIKAI ÁTVITELI JELLEMZŐK ÉS MÓDSZEREK	30
ÁLTALÁNOS ELMÉLETI ALAPOK	30
CSILLAPÍTÁS, SÁVKORLÁTOZÁS ÉS A ZAJ HATÁSA A JELEK ÁTVITELÉRE	30
VONALAK MEGOSZTÁSA	32
Multiplexelés frekvenciaosztással.....	33
Multiplexelés szinkron időosztással.....	34
Vonalkapcsolás.....	35
Üzenet és csomagkapcsolás.....	36
VEZETÉKES ÁTVITELI KÖZEGEK	38
Csavart érpár (UTP,STP).....	38
Koaxiális kábelek.....	40
Alapsávú koaxiális kábelek.....	41
Szélessávú koaxiális kábelek.....	41
Üvegszál kábel.....	42
VEZETÉK NÉLKÜLI ÁTVITELI KÖZEGEK	45
Infravörös, lézer átvitel.....	46
Rádióhullám.....	46
Szórt spektrumú sugárzás.....	46
Műholdas átvitel.....	46
VSAT rendszerek.....	47
ANALÓG ÁTVITEL	48
Telefónia.....	48
Cellás mobil rádiótelefonok.....	51
Előfizető- és készülékazonosítás.....	53
Hálózat alrendszer.....	54
Bázis állomás alrendszer.....	54
Üzemeltetést támogató alrendszer.....	54
Gsm — hívásfelépítés.....	55
Bolyongás (roaming).....	56
Aktuális mobilos fogalmak: SMS, GPRS, WAP.....	56

<u>Modemek</u>	58
<u>Modulációs protokollok</u>	59
<u>Modem parancsok</u>	61
<u>Modem Regiszterek</u>	62
<u>Szabványos hibajavító protokollok:</u>	63
<u>Szabványos adattömörítő protokollok:</u>	63
<u>MNP (Microcom Networking Protocol)</u>	63
<u>Fájltviteli protokollok</u>	64
<u>Modemek fejlődése</u>	65
<u>Modem szabványok</u>	66
<u>DIGITÁLIS ÁTVITEL</u>	67
<u>Digitális jelek kódolása</u>	69
<u>Karakterek ábrázolása — az ASCII kódrendszer</u>	72
<u>Unicode</u>	77
<u>Párhuzamos és soros adatátvitel</u>	78
<u>Aszinkron soros adatátvitel</u>	79
<u>RS-232C szabvány</u>	80
<u>Áraminterfész</u>	84
<u>Az RS-449, -422, -423, és az RS-485-ös szabványok</u>	85
<u>X.21 interfész</u>	86
<u>ISDN — integrált szolgáltatású digitális hálózat</u>	87
<u>Az ISDN szolgáltatásai</u>	88
<u>Az ISDN rendszerarchitektúrája</u>	89
<u>A digitális alközpont (PBX)</u>	91
<u>Az ISDN interfész</u>	92
<u>Az ISDN jelzémód</u>	94
<u>ATM — Asynchronous Transfer Mode</u>	94
<u>ADSL rendszer</u>	97
<u>Beágyazott rendszerek kommunikációja</u>	98
<u>Bájt-soros bitpárhuzamos adatátvitel (GPIB)</u>	98
<u>Az I²C busz</u>	101
<u>Bit átvitel</u>	103
<u>Bájt átvitel</u>	103
<u>Adatforgalom a buszon</u>	104
<u>Egy vezetékes adatátvitel</u>	106
<u>ELLENŐRZŐ KÉRDÉSEK</u>	107
3. ADATKAPCSOLATI PROTOKOLLOK	109
<u>KÖZEG-HOZZÁFÉRÉSI MÓDSZEREK</u>	109
<u>VÉLETLEN ÁTVITEL-VEZÉRLÉS</u>	110
<u>Ütközést jelző vivőérzékeléses többszörös hozzáférés (CSMA/CD)</u>	111
<u>Réselt gyűrű (slotted ring)</u>	112
<u>Regiszter beszúrásos gyűrű (register insertion ring)</u>	113
<u>OSZTOTT ÁTVITEL-VEZÉRLÉS</u>	114
<u>Vezérjeles gyűrű (Token Ring)</u>	114
<u>Vezérjeles sín (Token bus — Vezérjel busz)</u>	115
<u>Ütközést elkerülő, vivőérzékeléses többszörös hozzáférés (CSMA/CA)</u>	116
<u>KÖZPONTOSÍTOTT ÁTVITELVEZÉRLÉS</u>	116
<u>Lekérdezéses (polling) eljárás</u>	116
<u>Vonalkapcsolásos eljárás</u>	117
<u>Időosztásos többszörös hozzáférésű eljárás (TDMA)</u>	117
<u>Adatátviteli protokollok</u>	117

<u>KERETEK KÉPZÉSE</u>	118
<u>HIBAKEZELÉS</u>	120
<u>Hamming távolság</u>	120
<u>CRC — Cyclic Redundancy Check</u>	121
<u>Adatkapcsolati protokollok</u>	122
<u>Korlátozás nélküli, egyirányú (szimplex) protokoll</u>	123
<u>Egyirányú „megáll és vár” protokoll</u>	123
<u>Egyirányú összetett protokoll</u>	124
<u>Kétirányú protokollok</u>	125
<u>Visszalépés n-el technikájú protokoll</u>	127
<u>Szelektív ismétlő protokoll</u>	127
<u>CCITT V.41-es ajánlása</u>	128
<u>IBM BISYNC (Binary SYNchronous Communication)</u>	129
<u>HDLC (High level Data Link Control)</u>	131
<u>ELLENŐRZŐ KÉRDÉSEK</u>	135
4. HÁLÓZATI RÉTEG	137
<u>FORGALOMIRÁNYÍTÁS</u>	139
<u>A legrövidebb út meghatározása</u>	140
<u>Determinisztikus forgalomirányítás</u>	141
<u>Adaptív algoritmusok</u>	141
<u>Központi adaptív forgalomirányítás</u>	143
<u>Elszigetelt forgalomirányítás</u>	143
<u>Elosztott adaptív forgalomirányítás</u>	144
<u>TORLÓDÁSVEZÉRLÉS</u>	145
<u>X.25 HÁLÓZAT</u>	148
<u>KERET-RELÉZÉS (FRAME RELAY)</u>	152
<u>ELLENŐRZŐ KÉRDÉSEK</u>	153
5. A FELSŐBB RÉTEGEK	154
<u>SZÁLLÍTÁSI RÉTEG</u>	154
<u>VISZONYRÉTEG</u>	157
<u>MEGJELENÍTÉSI RÉTEG</u>	157
<u>Adatábrázolás</u>	158
<u>Adattömörítés</u>	158
<u>Hálózati biztonság és védelem (titkosítás)</u>	160
<u>Helyettesítéssel rejtjelezés:</u>	161
<u>Felcseréléssel rejtjelezés:</u>	161
<u>DES (Data Encryption Standard) — Adattitkosítási szabvány</u>	162
<u>Nyilvános kulcsú titkosítás</u>	163
<u>Hogyan működik PGP?</u>	164
<u>ALKALMAZÁSI RÉTEG</u>	166
<u>ELLENŐRZŐ KÉRDÉSEK</u>	166
6. LOKÁLIS HÁLÓZATOK	168
<u>IEEE 802-ES SZABVÁNYOK</u>	168
<u>A fizikai réteg</u>	169
<u>Közeghozzáférés-vezérlési (MAC) alréteg</u>	169
<u>Logikai kapcsolatvezérlési (LLC) alréteg</u>	170
<u>Rétegek közötti szolgálatok</u>	171
<u>Hálózati/LLC felületek közötti szolgáltatási előírások</u>	171
<u>LLC/MAC felületek közötti szolgáltatási előírások</u>	172

A lokális hálózatok fizikai egységei	172
<u>AZ IEEE 802.3 SZABVÁNY ÉS AZ ETHERNET.</u>	173
Struktúrált kábelezés.....	178
A struktúrált kábelezés felépítése:.....	178
Hub, bridge (híd), switch (kapcsoló).....	179
Fast Ethernet.....	180
A 802.3 MAC-protokollja	181
<u>VEZÉRJELES SÍN (VEZÉRJEL-BUSZ).</u>	183
A sín MAC protokollja.....	184
<u>VEZÉRJELES GYŰRŰ</u>	186
A vezérjeles gyűrű MAC protokollja.....	189
A vezérjeles gyűrű karbantartása	191
<u>FDDI.</u>	193
<u>A MAP ÉS TOP.</u>	194
<u>LOKÁLIS HÁLÓZATI OPERÁCIÓS RENDSZEREK</u>	195
Lokális hálózati operációs rendszerek funkciói	197
Novell Netware felhasználói ismeretek — Összefoglaló.....	199
<u>ELLENŐRZŐ KÉRDÉSEK</u>	201
7. A TCP/IP PROTOKOLL ÉS AZ INTERNET.	203
<u>TCP/IP PROTOKOLL</u>	204
Az Internet szállítási rétege: a TCP	204
Az Internet hálózati rétege: az IP.....	207
IP csomagok tördelése (fregmentációja)	208
Címzési rendszer.....	209
IP útválasztás (routing).....	214
IP útválasztás (routing).....	215
Az útvonal kiszámítás algoritmus.....	217
Statikus és dinamikus útválasztás.....	218
Forgalomirányítás.....	218
Összeköttetés-mentes szállítási protokoll: az UDP.....	219
Az Internet vezérlése: az ICMP protokoll	219
Hálózat elérési réteg — ARP.....	220
Röviden a UNIX hálózatkezeléséről.....	223
<u>PÉLDA AZ ALKALMAZÁSI RÉTEGRE: AZ INTERNET SZOLGÁLTATÁSOK</u>	224
Kapcsolódás az Internetre.....	225
Internet-elérés telefonvonalon keresztül	226
Serial Line Internet Protocol — SLIP.....	226
Point-to-Point Protocol — PPP	226
Internet szolgáltatások	227
E-mail (Electronic mail)	229
UUENCODE/UUDECODE.....	231
MIME (Multi-purpose Internet Mail Extensions).....	232
Smileys (mosolygók).....	234
FTP (File Transfer Protocol)	234
Telnet.....	235
Archie.....	236
Finger	236
Levelezési listák és a Usenet.....	237
Információk szervezése a hálózaton.....	237
Gopher.....	238

TARTALOMJEGYZÉK

<u>WWW (World Wide Web)</u>	238
<u>HTML</u>	239
<u>URL-specifikációk</u>	240
<u>A CGI</u>	242
<u>A HTTP-protokoll</u>	243
<u>Cookie</u>	245
<u>Portálok</u>	245
<u>A hálózat emberi tényezői</u>	245
<u>TÚZFALAK ÉS Proxy szerverek</u>	247
<u>Valós idejű hangtovábbítás interneten (VoIP=Voice over IP)</u>	249
<u>A legfontosabb RFC dokumentumok</u>	250
<u>ELLENŐRZŐ KÉRDÉSEK</u>	252
8. HÁLÓZATI FELÜGYELET	254
<u>Hálózati felügyelet (menedzsment)</u>	254
<u>SNMP</u>	254
<u>ELLENŐRZŐ KÉRDÉSEK</u>	257
<u>IRODALOMJEGYZÉK</u>	258
<u>TÁRGYMUTATÓ</u>	260

BEVEZETÉS

Ami működik, az csodálatos...

Tisztelt Olvasó!

A könyv, amely már az eredeti kiadás harmadik átdolgozása, a Gábor Dénes Műszaki Informatikai Főiskolán oktatott Számítógép-hálózatok című tárgy tanulásához nyújt segítséget.

Jól tudjuk, hogy ilyen témával foglalkozó könyvek (ld. Irodalomjegyzék) magyar nyelven is hozzáférhetők, de azok témaválasztása sokkal átfogóbb, hogy azt kötött óraszámban, és az oktatás korlátjait figyelembe véve, oktatni lehessen.

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

- 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 a cél az eredetiség: sokszor használtunk fel ábrákat és magyarázatokat az adott témák tárgyalásánál. Ilyenkor utaltunk az eredeti irodalomra is.

A könyv célja a számítógépes 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 befolyásoló megoldási módokról,
- az egyes kommunikációs részcsoportok működésének alapjairól.

Az olvasó számára megteremti az alapokat a konkrét megvalósítási igények megbízható meghatározásához; a párhuzamosan elsajátított hardver, szoftver és eszközismeretek birtokában, a követelményeket kielégítő számítógépes hálózati rendszerek összeállításához, azok üzembehelyezéséhez és későbbi működtetéséhez, esetleges karbantartásához.

A könyv nyolc fejezetből áll, amelyek tárgya a következő:

Az **1. Fejezet** a számítógép-hálózatok elméleti alapjait foglalja össze. Bemutatja a hálózatok kialakításának a célját, a rétegszemléletet, és az ehhez kapcsolódó alapismereteket.

A talán leghosszabb **2. Fejezet** a hálózatok fizikai jellemzőit mutatja be, foglalkozik a vonalmegosztási kérdésekkel, az információk átvitelének különféle megoldásaival, az analóg és digitális átvittel, a soros adatátviteli szabványokkal, kódolási, karakterábrázolási kérdésekkel, az ISDN-el. Egy külön rész tárgya a beágyazott rendszerek kommunikációja. Ilyen rendszerek magja a mikrokontroller, amely egy teljes, komplett mikroszámítógép memóriával, perifériával.

A **3. Fejezet** első fele a hálózaton közös vonalakon elhelyezkedő eszközök vonalmegszerzési stratégiáit, a közeg-hozzáférési módszereket mutatja be. Két eszköz közötti adatcsere megvalósításához adatkapcsolati protokollok kidolgozása szükséges. A fejezet további része az ehhez kapcsolódó ismereteket tárgyalja.

Az információcsere több csomópontot tartalmazó hálózatban, az információ útjának kijelölését igényli. Ezt a feladatot hálózati réteg látja el, kapcsolódó ismeretek az **4. Fejezetben** találhatóak.

Mikor az információ eljut a távoli csomópontba, még számos feladatot kell megoldani, amely feladat a felsőbb rétegekre hárul. A **5. Fejezet** ezeket a feladatokat mutatja be.

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. A **6. Fejezet** ezeket a helyi hálózatokat a LAN-okat mutatja be.

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 előbbieken szereplő ismereteket.

A **8. 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 magyar nyelvű könyvet, folyóiratcikket.

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

A legnagyobbak a nagy kiterjedésű 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.

Mi a könyv célkitűzése?

A számítógép-hálózatok felépítésének, működésének általános megismertetése, az egyes főbb hálózattípusok (kiemelten az ETHERNET) struktúrájának, működésmódjának, részletesebb bemutatása, ismeretük elsajátíttatása. A világméretű hálózatok (kiemelten az INTERNET) használatának a megismertetése.

Kapcsolódó ismeretek:

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

Hogyan kell a tantárgyat megtanulni ?

A tananyag elsajátítását a minden fejezet végén található kérdés-csoportra adott válaszokkal ellenőrizhetjük. Természetesen az elsajátításban sokat segít a kapcsolódó előadásokon való részvétel, vagy az azokról készült videofelvétel meghallgatása, mert az ott elhangzottak áttekintést adnak a tananyag összefüggéseiről, segítséget nyújtanak a bonyolultabb részek megértéséhez. A szerző honlapja (<http://alpha1.obuda.kando.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 tananyagrészt megértéséhez, és a számonkéréshez

Előadásaimon — különösen kisebb hallgatóság előtt — mindig fel szoktam tenni előadásról, előadásra a kérdést: Van-e valakinek kérdése az elhangzottakkal kapcsolatban. Ha nem volt kérdés, akkor ebből sajnos nem a számomra hízelgő „szuperelőadó, aki mindent csodálatosan magyaráz” következtetést vonhattam le, hanem az anyaggal való foglalkozás, a tanulás hiánya. A kérdések megjelenése és száma számonkérések előtt hirtelen megnövekedett. Aki tanul, annak kérdései is vannak. Ezeket a kérdésfelvetéseket segítik a minden fejezet végén található átfogó kérdéscsoportok.

A tárgy elsősorban elméleti jellegű ismeretekkel foglalkozik és emiatt nincsenek gyakorlatok a megtanulandók mélyebb megértéséhez, elsajátításához. Ez azonban nem jelenti azt, hogy egyszerűbb problémák és feladatok megoldására ne legyen szükség. A feladatok megoldását az előadásokon elhangzottak és a könyvben leírtak segítik.

Az előadás és a könyv mellett, általános tájékozódásul, ismereteik bővítésére a hallgatók igénybe vehetik a könyvtárak szolgáltatásait, az ott megtalálható szakkönyveket és folyóiratokat. mint például ez utóbbiak közül a "CW Számítástechnika" című hetilapot, a "Chip Magazin", a "PC World", a "Byte" című folyóiratokat, stb. Ezek olvasása által, a legkorszerűbb eszközök, technikák megjelenéséről szerezhetnek információkat a hallgatók. A könyv anyaga, és a témakörhöz kapcsolódó kiegészítő folyamatosan bővülő információt az olvasó a következő WEB-lapon találhatja meg:

<http://alpha1.obuda.kando.hu/~konya>

Befejezésül fontosnak tartom megjegyezni azt a tényt, amit a számítógépes hálózatok elterjedése tett világossá: 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 utánajárással 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: ezeket a dokumentumokat el kell olvasni, és meg kell tanulni, ami természetesen 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 kívánok felidézni, amely egy információközlési módszernek is tekinthető: a **lényegkiemelés**. Ez azt jelenti, hogy bármilyen téma bemutatásakor nagyon egyszerűen, lényegre törően, akár bizonyos gondolati pongyolaság árán próbáljuk a lényeget kiemelni, felhíva a figyelmet az egyszerűsítés korlátaira is.

Jól tudom, hogy a túlzott egyszerűsítés nagyon veszélyes, felületességre készítet, de pszichológiailag azt az előnyt adja, hogy az adott téma elsajátítója önbizalmat kap, és hitet arra, hogy a részleteket is meg tudja érteni, illetve megtanulni.

„Az ördög (értsd: a bonyolultság) a részletekben lakozik...”

Immár a kedves olvasó a könyv átdolgozott, harmadik kiadását tartja a kezében. Az új kiadásban kijavítottam a felfedezett (szerencsére nem annyira sok) hibát, amelyek feltárásához ezúton is szeretném megköszönni az olvasók segítségét. A könyvhöz kapcsolódó előadások megtartása során gyűjtött tapasztalatok alapján a tematikát és a tárgyalásmódot is kismértékben átalakítottam, alkalmazkodva a gyors fejlődéshez. Néhány idejétmúlt dolgot elhagytam, vagy rövidítettem (pl. FDDI, MAP, TOP, stb.), és új dolgok is a könyvbe kerültek:

- Mobil telefonos részbe: mobil hálózatok.
- A lokális hálózattal kapcsolatos részbe: a korszerű Ethernet megoldások: csavart érpár, HUB, bridge, switch, 100Mbit/sec-os hálózatok.
- A TCP/IP alapú hálózati ismeretek bővítése.
- Hálózatfelügyelet.

A könyv belső formátuma: a lapok szélén egy figyelemfelhívó és megjegyzéseket, lényegkiemelő vázlatpontokat tartalmazó hasáb van. A könyv törzsszövege ezzel szerves egységet alkot, a jobb megértés érdekében elolvasása fontos!

A lapszéleken még két jelölést helyeztünk el: ? a kérdéseket, ! a nagyon fontos részeket jelöli.

A lapok szélét a kedves olvasó is bátran használja fel a jegyzeteléshez!

Néhány szó az idegen szavak használatáról. Aki egy ilyen könyvet ír, annak nagyon nagy a felelőssége, hiszen a tanulók a könyv alapján jegyeznek meg szavakat és kifejezéseket, és használják ezeket a továbbiakban. Ezért, ahol lehetett, ott már a megszokott magyar elnevezéseket használtam, sokszor leírtam a fogalom angol megfelelőjét, illetve, ha nem volt jól értelmezhető magyar megfelelője, változatlanul hagytam (pl. HUB, switch, stb.)

A könyv — bár gondosan készült, valószínűleg nem hibátlan — mert emberi munkával készült. Ezért itt kérem a könyv olvasóit, a könyvből tanuló hallgatókat, hogy a könyvvel kapcsolatos észrevételeiket, az esetleges hibákat részemre eljuttatni szíveskedjenek. Ezeket az esetleges újabb kiadásba kívánom beépíteni.

Hiszen az ember nem alkot tökéleteset, de mindig törekedni kell rá...

Budapest, 2000. december

a szerző

E-mail: konya@novserv.obuda.kando.hu

<http://alpha1.obuda.kando.hu/~konya>

1. A HÁLÓZATOK CÉLJA, ALKALMAZÁSA, ALAPFOGALMAK

A számítógépek megjelenésekor mindegyik egymástól elkülönülve, önállóan dolgozott. Még a személyi számítógép, a „personal computer” nevében is hordozza az elkülönültségre utaló „személyi” jelzót. 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 használatát, 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 működést eredményez. Például egy nyomtató hibája nem jelenti azonnal a nyomtatási lehetőségek megszűnését, mivel szerepét a rendszerben lévő másik nyomtató is átveheti. A fontosabb programok, adatok a rendszer több számítógépének lemezegegyiségén is tárolódhatnak és az egyik példány megsemmisülésével sem történik helyrehozhatatlan károsodás.
- A fenti előnyök anyagi oldalról tekintve költségmegtakarítással járnak. Az eszközöket (pl. nyomtatókat, háttértárakat) kevesebb példányban kell megvásárolni.
- A hálózatok további célja a skálázhatóság, ami azt jelenti, hogy a teljesítmény növelése fokozatosan úgy lehetséges, hogy újabb processzorokat adunk a rendszerhez. (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.



Számítógép-hálózatok alatt az egymással kapcsolatban lévő önálló számítógépek rendszerét értjük.

Milyen előnyökkel jár a gépek hálózatba kapcsolása?

- erőforrások megosztása
- nagyobb megbízhatóságú működés
- költségmegtakarítás
- adatbázisok elérése
- kommunikációs közeg

Ezen előnyök mellett a hálózatba kapcsolás a számítógépek használati körének kibővülését sőt kiterjesztését is lehetővé teszi.

- 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 csökkenti egy számítógépben elhelyezett egységek számát, és ezért nagyon olcsó. Érdekes kérdés ennek, a két betűvel NC-nek nevezett (Network Computer) hálózati számítógépnek a jövője.
- Interaktív szórakoztatás. Mivel a számítógépek és a hálózatok az otthonokban is megjelentek rövid idő múlva lehetővé válik a hálózati videózás, amely a hálózaton eljuttatott videojelek segítségével valósul meg.

Ezeket a dolgokat már tényleg napról napra éljük meg. Tessék csak az Internet robbanásszerűen gyors elterjedésére gondolni.

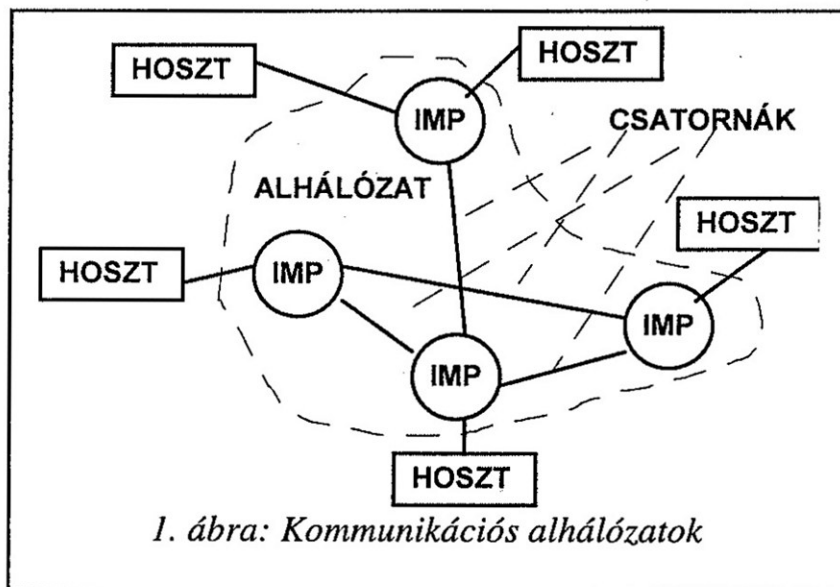
A három legfontosabb társadalmi vetülete:

- Hozzáférés távoli információkhoz,
- Személyek közötti kommunikáció,
- Interaktív szórakoztatás

HÁLÓZATI STRUKTÚRÁK

Azokat a számítógépeket, amelyeket egy számítógépes rendszerben összekötünk hosztoknak (host) nevezünk.

Ezt magyarul gazdagépnek is hívjuk, itt futnak a felhasználói programok, helyezkednek el az adatbázisok. Ezeket a gépeket kommunikációs alhálózatok kötik össze, amelyek feladata a hosztok közötti kommunikáció megvalósítása, azaz üzenetek továbbítása.



Általában ezek az alhálózatok két jól szétválasztható részből: az átvitelt biztosító **vonalakból** más néven csatornákból (ahol a bitek „áramlanak”, szokták még vonalnak, áramkörnek, vagy trónknak nevezni) és a **kapcsolóelemekből** állnak.



FONTOS ALAPFOGALMAK!

Azokat a számítógépeket, amelyeket egy számítógépes rendszerben összekötünk hosztoknak (host) nevezük.

az alhálózatok csatornákból és kapcsolóelemekből állnak

Valójában az IMP a kommunikációt megvalósító hardver és szoftver együttes.

A valóságban az IMP-eket sokszor a hosztokban alakítják ki (pl. egy hálózati kártya + a szoftver)

Ez utóbbi 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 vonalak kapcsolását végzik, a bemenetükre jutó adatot valamelyik meghatározott kimenetre kapcsolják (pl. routerek, hálózati átjárók).

További szokásos nevük: hálózati kapcsoló pontok (internal network switching node).

Az alhálózatokat alapvetően két nagy csoportra oszthatjuk: **két pont közötti**, illetve közös csatornát használó, más néven **adatszóró** (broadcasting) alhálózatok.

Az alhálózatokat alapvetően két nagy csoportra oszthatjuk: két pont közötti, illetve közös csatornát használó alhálózatok.

KÉT PONT KÖZÖTTI CSATORNÁVAL RENDELKEZŐ ALHÁLÓZAT (PONT-PONT ÖSSZEKÖTTETÉS).

Ebben az esetben a két kommunikációs végpontot pl. egy kábellel kötik össze, és az üzenetek (vagy annak a szétdarabolt részei a csomagok (=packet)) ezen a kábelen keresztül haladnak.

Az üzenet hosszú is lehet, azért kell részekre bontani.

Amikor egy vevő megkapja a csomagot és az nem neki szól, 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.

Abban az esetben, ha a csomagok rövidek és azonos méretűek gyakran **celláknak** (cells) nevezik őket.

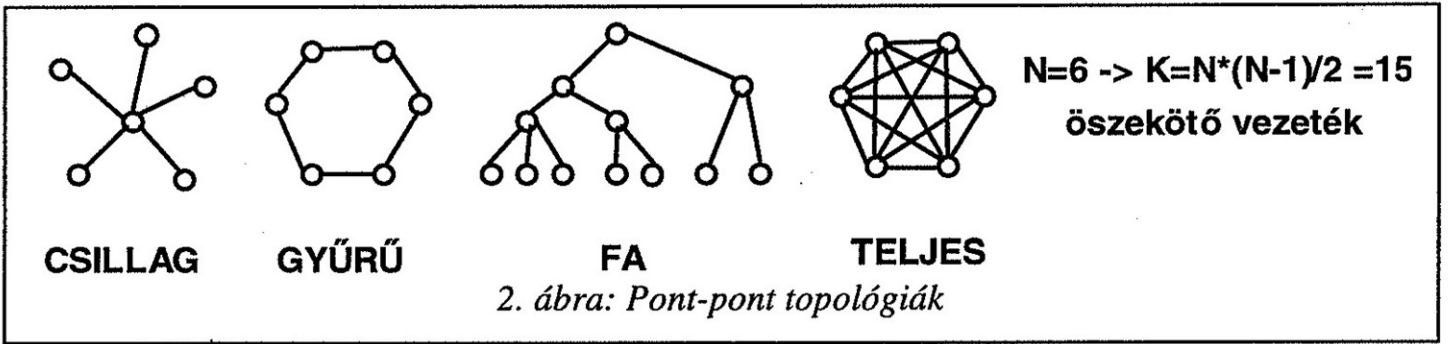
Az ilyen kialakításnak lényeges előnye az, hogy a két pont közötti kapcsolatból adódóan a kommunikációs problémákat elsődlegesen ezek pontok közötti csatorna hordozza, és hibák behatárolásánál is előnyös ez a kialakítás.

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

A kommunikációban résztvevők csak egymást látják, nem szükséges a másik fél azonosítása (nem kell címzés!)

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 5 pont összekötéséhez $5*4/2=10$ pont-pont összeköttetés szükséges.

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



ÜZENETSZÓRÁSOS CSATORNÁVAL RENDELKEZŐ ALHÁLÓZATOK (MULTIPONT ÖSSZEKÖTTETÉS)

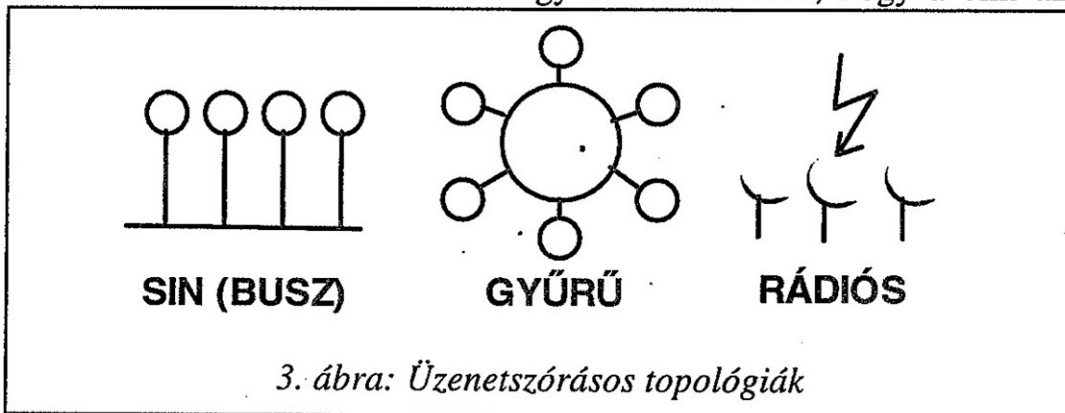
Ilyen típusú hálózatoknál ténylegesen egy kommunikációs csatorna van, és ezen az egy csatornán osztozik az összes hálózatba kapcsolt számítógép. A források által küldött üzeneteket a hálózat minden állomása veszi, (ami nehezíti az adatvédelmet) és azt hogy az üzenet kinek szól, az üzenetben elhelyezett egyedi — gépet címző — címinformáció hordozza. A forráscím minden esetben egy egyedi (unicast, vagy single-node) cím. A célcím pedig lehet:

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 azon 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. Az ilyen módon működő hálózatok esetén a jellegzetes topológiák:

Természetesen ehhez a címeket úgy kell kialakítani, hogy a cím az



A közös csatornahasználat már felveti azt a problémát, hogy ki és mikor használhatja a csatornát, a felek azonosításához címzés szükséges.

Szokták a kommunikációs végpontokat node (e.: nód) -nak is hívni

A gyakorlatban a címek bináris számok. Bizonyos címkombinációkat (pl. a cím egy részének minden bite 1-es) fenntartjuk a **csoportcím**-nek. Ha ezt a címet küldjük ki, akkor a csoporthoz tartozó hosztok mindegyike a magáénak tekinti a hozzá kapcsolódó üzenetet.

A pont-pont jellegű kapcsolatban lényegében csak egyedi címzés valósul meg.

egyedi gépcímek mellett a csoportcímezésre is tartalmazzon egy a csoportra utaló bits csoportot.

Egy további bit elegendő a mindenkinek szóló üzenet (broadcast) megvalósításához. Vagyis a használt cím általánosan:

broadcast cím (1bit) — csoportcím (n bit) — egyedi cím (m bit)

felépítésű.

Az adatszórásos hálózatoknál a közös 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 eljárást, amely ezt a versenyhelyzetet feloldja. Ezt a versenyhelyzet feloldást hívják **arbitráció** - nak

A közeghozzáféréssel kapcsolatos kérdésekre egy későbbi fejezetben 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 struktúráját jelenti.

A hálózatok más típusú osztályozása is lehetséges: vannak nyilvános hálózatok, mint a telefon, telexhálózat és vannak magánhálózatok amelyeket egy intézmény, vagy szervezet tart fenn pl. MÁV, HM, BM, stb.



HÁLÓZATI ARCHITEKTÚRÁK

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

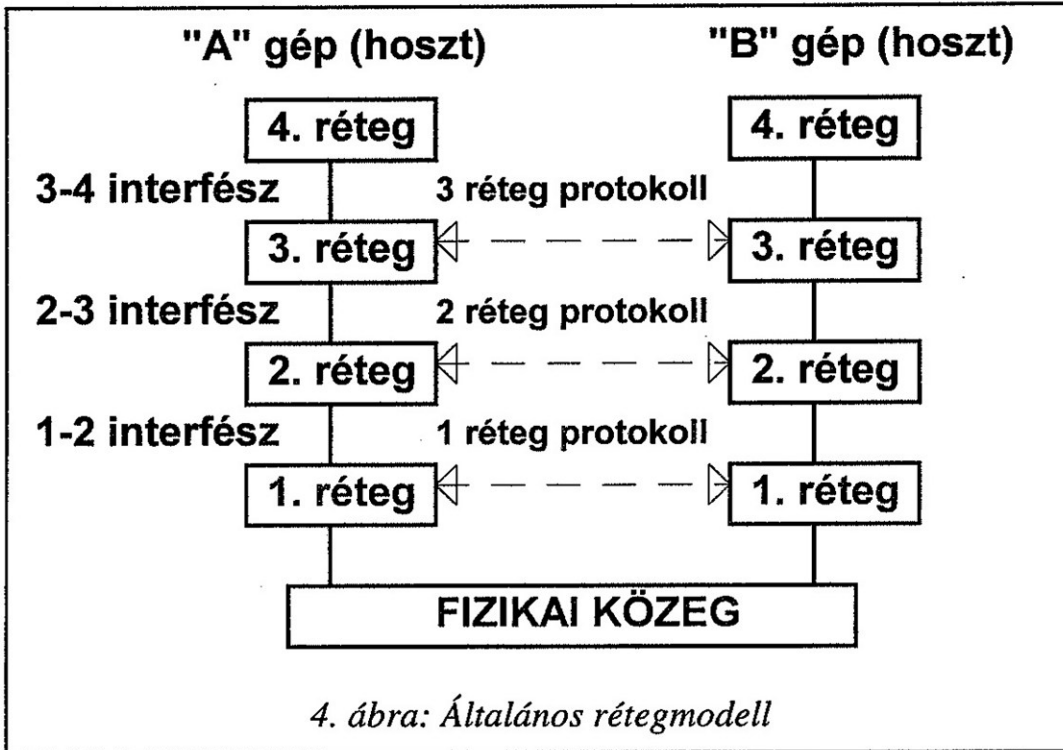
Ez a rétegszemlélet lehetővé teszi az összetett bonyolult rendszerek egyszerűbb leírását, a réteghez tartozó belső kérdések elszigetelt kezelését.

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

A kommunikáció megvalósítása bonyolult feladat, nehéz teljes egységben tárgyalni, ezért célszerű egymásra épülő részekre bontani. Lényegében ezek a rétegek.

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

A szomszédos rétegek között egy **réteginterfész** húzódik, amely az alsóbb réteg által a felsőnek nyújtott elemi műveleteket, és szolgáltatásokat határozza meg



Az egymásra épülő rétegek elfedik az alattuk lévő rétegeket a felettük lévőktől: azaz nem kell az alsó rétegek működését „tudni” a felsőbb rétegek tervezésekor.

FONTOS FOGALOM!

Mivel a rétegek is egymással kommunikálnak, rétegprotokollok biztosítják a köztük lévő kommunikáció egzaktságát.

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

A legfontosabb az, hogy ez az interfész minden réteg között tiszta legyen olyan értelemben, hogy az egyes rétegek egyértelműen definiált funkcióhalmazból álljanak. Ez egyszerűvé teszi az adott réteg különböző megoldásainak a cseréjét, hiszen a megoldások az előbbieket alapján ugyanazt a szolgáltatást nyújtják a felettük levő rétegnek, segítve a nyílt rendszerek kialakítását.

A rétegek és rétegprotokollok halmazát 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,
- döntést kell hozni az adatátvitel szabályairól: az átvitel egyirányú (szimplex), váltakozóan két irányú (fél duplex) vagy egyszerre két irányú (duplex) legyen.
 - 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)



Számos szakirodalomban szokásos a fentiekre a

- szimplex
 - duplex
 - full duplex
- elnevezéseket használni, ami elég zavaró az ugyanazon duplex szó eltérő értelmezése miatt.

- 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 = flow control),
- ha bizonyos okok miatt az üzenetek hossza korlátozott, és ezért a küldés előtt szét kell darabolni, felmerül a kérdés, hogy hogyan biztosítható a helyes összerakásuk,
- az előbbi esetben biztosított-e az üzenetek sorrendjének a helyessége,
- nagyon sokszor ugyanazon a fizikai csatornán több párbeszéd zajlik. (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.

Gyors vevő-lassú adó esetén nincs gond, mert amit a lassú adó ad, azt a gyors vevő gond nélkül veszi.

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

HÁLÓZATSZABVÁNYOSÍTÁS

Már az eddigiek alapján is nyilvánvaló, hogy a hálózatok kialakításában (de ez igaz minden műszaki tudományra) alapvető szerepet játszik a szabványosítás.

Minden új dolog kialakulását megelőzi a kutatás, az ehhez kapcsolódó írásos és szóbeli információcserék (cikkek, konferenciák), majd az új dolgot gyártó rendszerek kialakítása.

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 az esetleg a még nem alapos ismeretek miatt a szabvány nem lesz megfelelő, mivel az új, későbbi kutatási eredményeket már nem lehet beilleszteni, kedvező megoldásokat kell elhagyni.

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

A szabványok központi szerepet játszanak a fejlődésben, ez teszi a rendszereket nyíltakká, egységeit cserélhetővé.

Sajnos a műszaki fejlődés számos esete bizonyítja az állításunkat. Ezért a gyakorlatban a szabványok két családja létezik: a **de-jure szabványok**, amelyeket 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.

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

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 funkciójáról.

Ezen szabványok közül a legfontosabbakat a következőkben inkább illusztrációként felsoroljuk. 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 arra utal hogy a szabványt úgy fejlesztették tovább, hogy régebbi berendezések használata is lehetővé váljék.

E sorozat: Telefon szabványok
F sorozat: Távíró szabványok
G sorozat: Digitális hálózatok (PCM)
I sorozat: ISDN
Q sorozat: Digitális elérési jelzési rendszer
S sorozat: Távíró terminálok
T sorozat: Teletex, fax
U sorozat: Távíró kapcsolás-telex
V sorozat: Telefonvonalakon történő adat kommunikáció
X sorozat: Adatátvitel

V sorozat szabványai:

Alapok, általános előírások
V.1 A moduláció állapotspecifikációja
V.2 A teljesítményszintek előírásai
V.3 A jelstruktúrák előírásai
V.5, V.6 Szinkron vonalakon történő adatátviteli sebességek előírásai

Kétségtelen egy szabvány számos előnyt (köztük természetesen anyagi előnyt) biztosít annak a cégnek, aki kifejlesztette.

A didaktikus elvektől (nevezetesen, hogy csak már definiált fogalmakra építve vezetjük be az újakat) a szabványok ismertetésénél eltértünk, ezért ajánlott a könyv végigolvasása után ismételtten visszatérni ide.

V.7 A használt kifejezések definíciója

Interfészek, modemek

V.10 Aszimmetrikus áramhurkos vonalak elektromos viszonyai

V.11 Szimmetrikus áramhurkos vonalak elektromos viszonyai

V.15 Akusztikus adatátviteli csatolók

V.19 Párhuzamos átvitelű modemek

V.21 300 bps-os modemek

V.22 1200 bps-os duplex modemek

V.23 600/1200 bps-os modemek

V.24 Adatátvitelnél használt interfészvonalak definíciója

V.25 Automatikus hívó és válaszoló berendezések

V.25bis V.24 kiegészítése V.25-nek megfelelően

V.26 2400 bps-os félduplex modemek

Hibavédelem

V.40 Elektromechanikus berendezések által jelzett hibák definíciói

V.41 Hibavédelmi rendszer (átvitel független)

V.42 Hibavédelmi rendszer modemekre (MNP 1-4)

V.42bis Adattömörítési eljárás (MNP 5-7)

Az X sorozat szabványai:

Szolgáltatások és felhasználói kérdések

X.1 Felhasználói osztályok definíciói a köz- és ISDN alapú hálózatoknál

X.2 Átviteli szolgáltatások és kiegészítéseinek a leírása

X.3 Csomagok összerakása és szétbontása (X.25-höz)

X.4 Jelstruktúrák (V.4-hez hasonló)

X.10 Az átviteli szolgáltatások 14 elérési osztályának definíciói

Interfészek

X.20 DTE és DCE egységek közötti aszinkron folyamatok interfészének definiálása

X.21 DTE és DCE egységek közötti szinkron folyamatok interfészének definiálása (formátumok és eljárások)

X.24 V.24-el megegyezik

X.25 Felhasználó berendezések hálózati összekapcsolása dedikált vonalú csomagkapcsolt hálózaton keresztül

X.31 X.25 terminálok bekötése ISDN hálózatokba

Vezérlő jelek, kapcsolások

X.50 Nemzetközi szinkron adathálózatok multiplexelési módszerei

Hálózatok

X.121 Nemzetközi számozási előírás közhálózatokon

OSI modell (X.200-X.299)

Internetworking (Hálózatok közötti kommunikáció) (X.300-X.330)

Üzenetkezelő rendszerek (X.400-X.430)

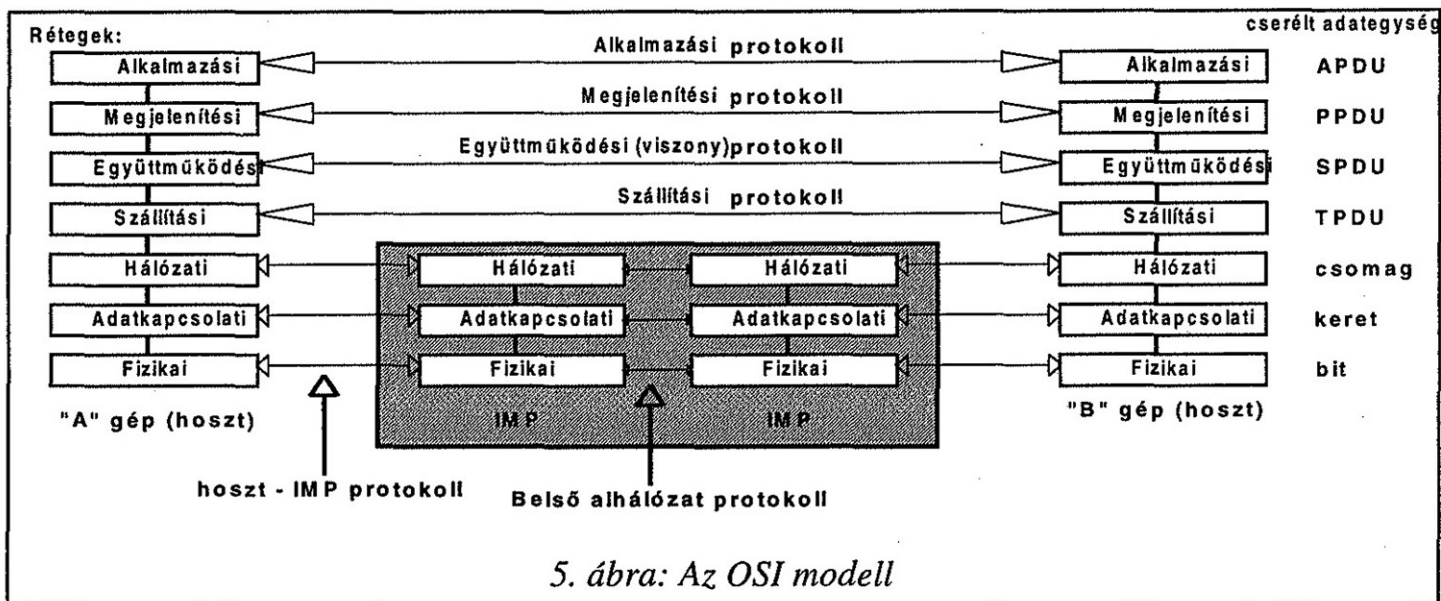
Katalógus szolgáltatások (X.500)

AZ OSI MODELL

Az előzőekben leírtak alapján már látható, hogy a számítógép-hálózatok rétegezett struktúrájú modell segítségével írhatók le. A Nemzetközi Szabványügyi Szervezet, az ISO (International Standard Organization) kidolgozott egy olyan modell-ajánlást (nem szabványt!!!) amelyet ma

Az OSI modell nem szabvány!

már minden hálózati, illetve kommunikációs rendszer tervezésekor követnek.



A modellt OSI-modell-nek hívják. Az OSI az Open System Interconnect — nyílt rendszerek összekapcsolása kifejezés angol eredetijéből alkotott betűszó.

Az OSI modell hét rétegből áll, és a kialakításuknál 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.

A következőkben röviden összefoglaljuk az egy-egy réteg által ellátott feladatokat, a legalsó szinttől felfelé haladva. A modell alsó három rétege a hálózattól függ, míg a felső négy réteg mindig alkalmazásfüggő, és mindig az alkalmazást futtató hosztokban történik a megvalósításuk (implementálásuk).

A kommunikációban résztvevő felek közötti üzenetváltást hálózaton átviendő kisebb egységek segítségével valósítják meg.

Fizikai réteg (physical layer): Valójában ezen a rétegen zajlik a tényleges fizikai kommunikáció. Biteket juttat a kommunikációs csatornára, olyan módon, hogy az adó oldali bitet a vevő is helyesen értelmezze (a 0-át 0-nak, az 1-et, 1-nek). A fizikai közeg, és az információ tényleges megjelenési formája igen változó lehet: pl. elektromos vezeték esetén, a rajta lévő feszültség értéke, vagy a



Nyílt rendszereknek olyan rendszereket hívunk amelyek nyitottak a más rendszerekkel való kommunikációra.

Az hogy a modell éppen hét rétegből áll (mint a gonoszok...) a bizottságban kialakított kompromisszum eredménye.

Pl. Az Internet TCP/IP protokollja csak négy rétegű, mégis jól működik.

Miért darabolják fel az üzeneteket kisebb egységekre? Például azért, mert az átvitel során bekövetkező hibák ilyenkor csak az üzenet egy részét érintik, és az a rész egyszerűen megismételhető.

feszültség változásának iránya. Információhordozó és közeg más és más lehet még: fénykábel, rádióhullám, stb.

Itt kell azt is meghatározni, hogy mennyi legyen egy bit átvitelének időtartama, 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). Ez általában úgy történik, hogy az átviendő adatokat (amelyek általában bitsorozatba kódolt formában — pl. bájtokban jelennek meg) adatkeretké (data frame) alakítja (tördeli), ellátja kiegészítő cím, egyéb és ellenőrző információval, ezeket sorrendhelyesen továbbítja, majd a vevő által visszaküldött az átvitt igazoló nyugtakereteket (acknowledgement frame) véve ezeket feldolgozza.

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

Hálózati réteg (network layer): lényegében a kommunikációs alhálózatok működését vezérli. Nagyobb hálózatok esetén a széttördelt üzenetrészeknek (ezeket szokták csomagoknak nevezni) a vevőtől a célba juttatása elvileg több útvonalon is lehetséges, feladat a bizonyos szempontból optimális útvonal kiválasztása. Ez a tevékenység az útvonalválasztás (routing), és több megoldása lehetséges:

- 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 való tartózkodása okozta torlódást, valamint különböző (heterogén) hálózatok összekapcsolását.

Szállítási réteg (transport layer): Feladata a hosztok közötti átvitel megvalósítása. Adatokat szükség esetén kisebb darabokra vágja, átadja a hálózati rétegnek. Fontos része a címzések kezelése. Egy viszonyréteg által igényelt összeköttetési kérés általában egy hálózati összeköttetést hoz létre, ha azonban nagyobb hálózati sebesség szükséges akkor több hálózati kapcsolatot is igénybe vehet. Fordítva, ha kisebb átviteli sebesség is elegendő, akkor egy hálózati

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

A hálózaton átküldött információ útvonalának meghatározása kiterjedt hálózatok esetén alapvető jelentőségű.

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.

összeköttetést lehet felhasználni több viszonyréteg kapcsolat lebonyolítására. Ezt a szállítási rétegnek a felsőbb rétegek felé nem érzékelhető módon kell megvalósítania. További feladatai: Több üzenetfolyam egyetlen csatornára nyálábolása, illetve forrás-cél összeköttetések létrehozása a névadási mechanizmus felhasználásával.

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

Megjelenítési réteg (presentation layer): a feladata az adatok egységes kezelése. A legtöbb alkalmazói program nem egy csupán egy bitfolyamot, hanem neveket, dátumokat, szövegeket küld. Ezeket általában adatstruktúrákban ábrázolják. A kódolás sem minden esetben egységes, pl. a karakterek kódolására az ASCII mellett az EBCDIC kód is használt. 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.

- adatábrázolás
- számábrázolás
- kódolás
- adattömörítés
- titkosítás

Alkalmazási réteg (application layer): Mivel ez kapcsolódik legszorosabban a felhasználóhoz, itt kell a hálózati felhasználói kapcsolatok megoldásait megvalósítani. Mivel számos termináltípust használnak a hálózati kapcsolatokban, amelyek természetesen kisebb-nagyobb mértékben egymástól eltérnek, ezért egy **hálózati virtuális terminált** definiálnak, és a programokat úgy írják meg, hogy ezt tudja kezelni. 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.

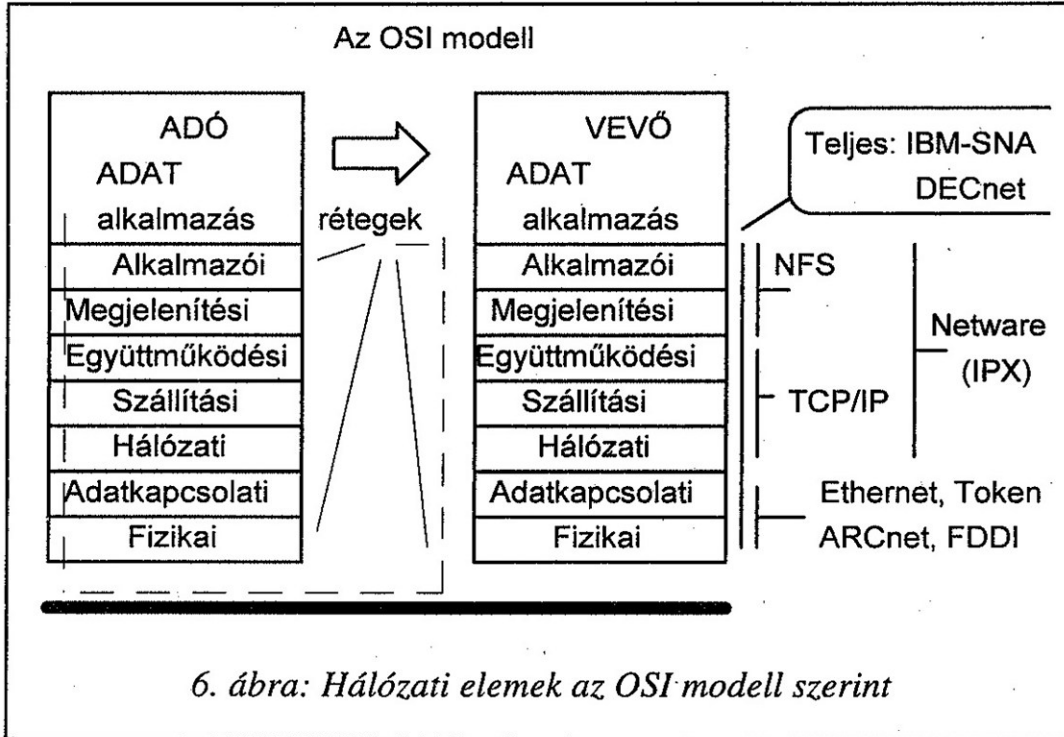
Mivel az OSI modell szemlélete számos hálózat kialakítása után, a belőlük leszűrt tapasztalatok alapján történt, ezért a **6. ábrán** összefoglaltuk néhány ismert hálózat és az OSI modell kapcsolatát.

SZOLGÁLATOK A RÉTEGEK KÖZÖTT

Minden rétegben vannak aktív, működő elemek ún. funkcionális elemek (más, elterjedt néven: entitás-ok), amelyek a rétegtől várt funkciókat megvalósítják. A kommunikációban résztvevő



Az entitás az adott réteg funkcióját valósítja meg.



berendezésben több, egymástól jól elkülöníthető egységet feltételez, amelyek mindegyike különböző feladatot valósít meg. Ez lehet egy program, vagy egy hardver elem (pl. egy be-kimeneti áramkör).

A rétegek közötti kommunikáció ún. szolgálatok segítségével valósul meg.

A szolgálatok a rétegek ki/bemeneti pontján ún. SAP-ján (Service Access Point) keresztül érhetők el. Ezek mindig két szomszédos réteg között található. Lényegében a két réteg közötti kommunikáció ténylegesen ezeken a pontokon keresztül valósul meg. 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. (7. ábra)

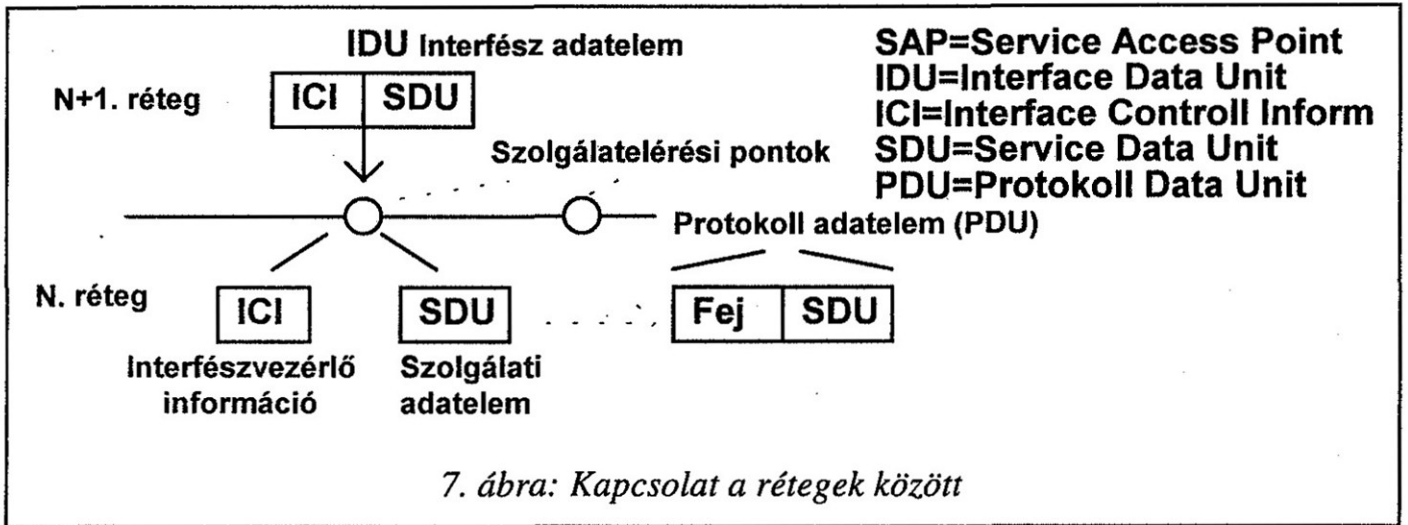
Az IDU két részből, a vezérlőinformációból (ICI) és az adatelemből (SDU) áll. Az ICI csak az interfész megfelelő működéséhez szükséges, a tényleges információt az SDU hordozza. Elképzelhető, hogy az

A szolgálatok a rétegek közötti kommunikációt megvalósító tevékenységek, a rétegek adott helyén (SAP) keresztül érhetők el.

A SAP például egy programban szereplő eljárás meghívása is lehet.

Egy adatcsomag esetén. (amely címet és adatot hordoz) ICI a csomag címe és SDU maga az adat.

adatelemet az N.-edik rétegbeli entitás még szétdarabolja és független protokoll-adatelemként küldi tovább.



A szállítási, viszony és alkalmazási protokoll adategységekre (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 szolgálat esetén az kapcsolat a közegen átviendő adatsomagok 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.

A folyamat részei: a kapcsolat felépítése, használata, majd bontása.

Az információ átvitel sorrendjét szigorúan az adó határozza meg.

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 hasonló 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.

Nagyon kellemes, mert az esetleg egymástól távol lévő kommunikációs felek pont-pont kapcsolatba kerülnek.

Melyik a jobb megoldás? — kérdezhetnénk. Mindkét megoldást annak megbízhatóságával minősíthetünk, ami azt jelenti, hogy az átvitel során nem veszünk adatot. A megbízhatóság megvalósításának az a módja, hogy a vevő az információvétel tényét visszajelzi a küldőnek, azaz nyugtázza (nyugtát küld).

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

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

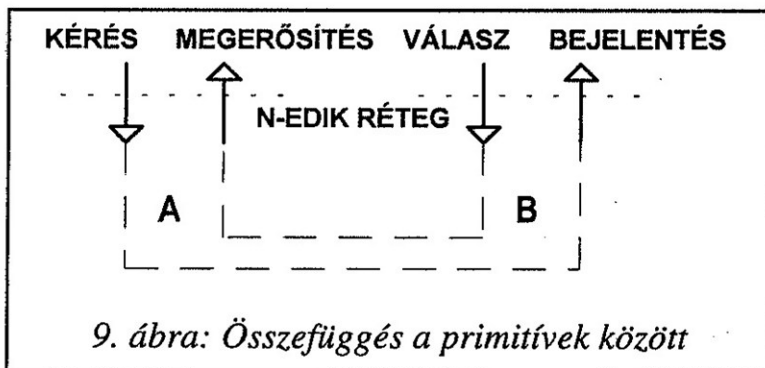
Ez természetesen nem minden esetben engedhető meg (például digitális hang- vagy képátvitel esetén), hiszen a nyugtázási megoldás késleltetést és külön adminisztrációt igényel.

Az előbbieket illusztrálására a **8. ábrán** összefoglaltuk a rétegszolgálatokat bemutató összefüggéseket.

Egy szolgálatot bizonyos alaplévelek (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 négy osztálya lehetséges:

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

A köztük lévő kapcsolatokat a 9. ábra mutatja.



Ha a kapcsolat létrehozását a **CONNECT**, az adatátvitelt a **DATA** és a lebontást a **DISCONNECT** szavakkal jelöljük, akkor egy

összeköttetés-alapú szolgálat nyolc szolgálat-primitívből áll [1]:

- | | |
|-----------------------|---|
| CONNECT.kérés | - Hívó összeköttetés létesítését kéri |
| CONNECT.bejelentés | - Hívó jelez a hívott félnek |
| CONNECT.válasz | - A hívott fél válasza a hívásra (elfogadja-elutasítja) |
| CONNECT.megerősítés | - Közli a hívóval, hogy a kérését elfogadta-e |
| DATA.kérés | - Hívott az adat küldését kéri |
| DATA.bejelentés | - Hívott az adat érkezését jelzi a hívónak |
| DISCONNECT.kérés | - Hívó összeköttetés bontását kéri |
| DISCONNECT.bejelentés | - Hívott jelez a hívónak hogy elfogadta |

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

ELLENŐRZŐ KÉRDÉSEK

1. Sorolja fel miért előnyös a számítógépeket hálózatba kapcsolni!
2. Melyek a számítógépes hálózatok legfontosabb jellemzői?
3. Mi az a hoszt?
4. Magyarázza meg, hogy mit takar az IMP fogalma!
5. Az összeköttetés kialakítása alapján hogyan csoportosíthatjuk az alhálózatokat?
6. Rajzolja fel a pont-pont kialakítás megoldási lehetőségeit!

1. A HÁLÓZATOK CÉLJA, ALKALMAZÁSA, ALAPFOGALMAK

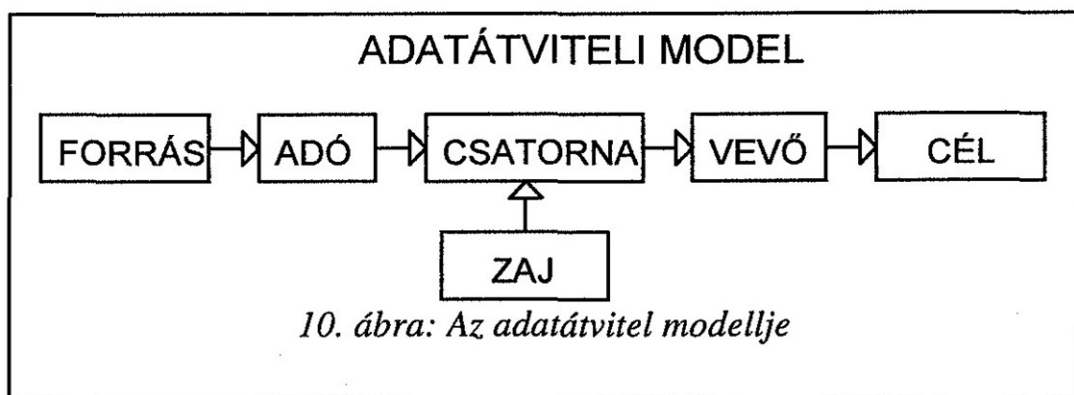
6. Rajzolja fel a pont-pont kialakítás megoldási lehetőségeit!
7. Mi az a csoportcímezés?
8. Rajzolja fel az üzenetszórásos kialakítás megoldási lehetőségeit!
9. Mi a topográfia és a topológia közötti különbség?
10. Határozza meg a protokoll fogalmát!
11. Mik azok a hálózati rétegek?
12. Mi a hálózati architektúra?
13. Adja meg a szimplex, félduplex és duplex átvitel meghatározását!
14. Miért fontos a hálózatok szabványosítása? Mik azok a de-jure és a de-facto szabványok?
15. Fogalmazza meg az OSI-modell filozófiáját!
16. Milyen rétegekből épül fel az OSI modell?
17. Mi a fizikai réteg feladata?
18. Mi az adatkapcsolati réteg szerepe?
19. Mit biztosít a hálózati réteg?
20. Milyen célokat valósít meg a szállítási réteg?
21. Mire használják az együttműködési rétegeket?
22. Milyen feladatokat lát el a megjelenítési réteg?
23. Ismertesse az alkalmazási réteg szerepét az OSI modellben!
24. Mi az entitás?
25. Határozza meg a rétegszolgálat fogalmát!
26. Mi az a SAP? Hol helyezkedik el?
27. Milyen részekből áll egy kapcsolati adatelem?
28. Ismertesse az összeköttetés alapú szolgálat lényegét! Milyen részekből áll a folyamat? Mikor célszerű alkalmazni?
29. Ismertesse az összeköttetés-mentes szolgálat lényegét! Milyen részekből áll a folyamat? 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

ÁLTALÁNOS ELMÉLETI ALAPOK

A csatornán történő információátvitel során az adó megváltoztatja a csatorna fizikai közegének valamilyen tulajdonságát, ami a közegen továbbterjed, és a vevő ezt a fizikai közegváltozást érzékeli. Például vezetékek esetén az átfolyó 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 a rabok is kommunikálnak egymással, itt a kommunikációs közeg a vízcső-vezeték hálózat és ez vezeti a kopogtatást, azaz a hangot, vagy gondoljunk az indiánok által használt füstjelzésre.



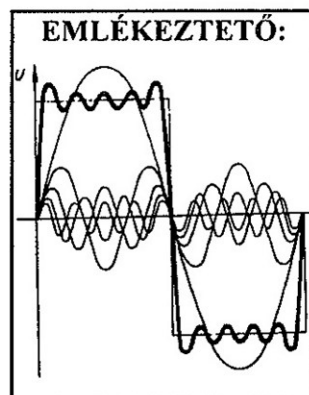
az átvitel során fellépő tényleges fizikai korlátok: A közeg fizikai jellemzői változásának mértéke, a változtatás lehetséges sebessége, a tovaterjedés során fellépő jelgyengülés.

CSILLAPÍTÁS, SÁVKORLÁTOZÁS ÉS A ZAJ HATÁSA A JELEK ÁTVITELÉRE.

A témához kapcsolódó részletes ismereteket a [16] irodalomból sajtátíthatjuk 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 **csillapítással**, **sávkorlátozással**, és a **zajokkal**.

A jelek átviteléhez — amely lényegében az átviteli közeg valamilyen fizikai jellemzőjének megváltoztatása (modulálása) — mindig energia kell. Ennek nagysága a jel összetevőitől függ. Ha jelet 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 komponensek közül a jelről a legtöbb információt az első ún. alapharmonikus hordozza, ezért, ahogy növeljük az adatátviteli sebességet, annál nagyobb lesz az alapharmonikus frekvenciája.

Ha ez megközelíti a vágási frekvenciát, ennek csillapodása a jel teljes eltorzulását okozza. Összefoglalva: **a közeg sávszélessége korlátozza az adatátviteli sebességet.**

Másrészről a csatornát a jel (S=Signal) és a zaj (N=Noise) teljesítményének arányával vagyis a **jel-zaj viszony**-nyal (**signal to noise ratio**) : S/N jellemezzük.

Általában ehelyett a $10 \cdot \lg(S/N)$ 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:

$$\text{Maximális adatátviteli sebesség} = H \cdot \log(1 + S/N)$$

(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 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 300 Hz, a felső frekvenciája 3300 Hz, így a sávszélessége: 3300-300=3 kHz

Digitális hálózatokat az adatátviteli sebességükkel: az időegység alatt átvitt bitek számával jellemezhetjük. Ezt célszerű bit/sec-ben mérni.

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

$$1 \text{ baud} = \log_2 P \text{ [bit/s]} \quad \text{ahol } P \text{ a kódolásban használt jelszintek száma.}$$

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



Nyugodtan kijelenthetjük: a hálózatok fejlesztése lényegében küzdelem a nagyobb sávszélességért.

Sokszor digitális hálózati terminológiaként használják a sávszélességet: azon maximális információátviteli sebességet értik alatta, amely egy adott kommunikációs csatornára jellemző.

Például olyan átvitelnél ahol ezt kétállapotú jelekkel valósítjuk meg, ott a baud és a bit/s azonos számértéket adnak, de ha a jelet négy szint felhasználásával vesszük át, ott a baud számértéke már csak fele a bit/sec-ban megadott valószínű adatátviteli sebességnek.

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

VONALAK MEGOSZTÁSA

A következők megértéséhez meg kell különböztetnünk a **csatornákat**, amelyeken az információcsere történik, és a felhasznált, tényleges, fizikailag létező összeköttetéseket biztosító **vonalak**at.

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. Mivel az ADÓ és VEVŐ oldal számára csak a végeredmény, az információ a fontos, ezért több csatorna is kialakítható egy vonalon, amelynek megvalósítására több lehetőség van.

Az egyik megoldás az, mikor a fizikai közeget osztjuk meg több csatorna között. Ezt az **adott vonal felosztását csatornákra több adó, illetve vevő között multiplexelés-nek nevezzük**. A multiplexelés olyan eljárás, amelynek során egy adatvonalat előre meghatározott, rögzített módszer szerint osztunk fel elemi adatcsatornákra. Minden bemenő elemi csatornához egy kimenő csatorna is tartozik, ezért a multiplexelés nem okoz csatorna-foglaltságot. Ezek a **frekvenciaosztásos** és az **időosztásos multiplexelési módszerek**, illetve ezek kombinációja.

A másik lehetőség a vonalak maximális kihasználására, az átvendő 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 adatvezetékeket 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.

Jelenleg az analóg átviteli vonalakat felváltották illetve folyamatosan felváltják a digitális átviteli utak, amely azt jelenti, hogy nem kellene a közbenső analóg-digitál és digitál-analóg átalakítók. Ez azt is jelenti, hogy a frekvencia osztásos multiplexelést az időosztásos multiplexelés váltja fel. A következőkben a vonalmegosztási módszereket fogjuk bemutatni.



Egy vonalon több információs csatorna is kialakítható.

Fontos a csatornák és a vonalak megkülönböztetése!

MEGOLDÁSOK:

- multiplexelés
- frekvenciaosztásos
- időosztásos
- üzenet és csomag kapcsolási módszerek
- vonalkapcsolás

A multiplexelést szokták magyarul nyalábolásnak is nevezni.

Egy kicsit később láthatjuk majd, hogy míg a beszéd analóg átviteléhez 300-3300 Hz=3kHz-es sávszélesség elegendő, ugyanazon beszéd digitális átvitele 64 kbit/s-os adatátviteli sebességet igényel.

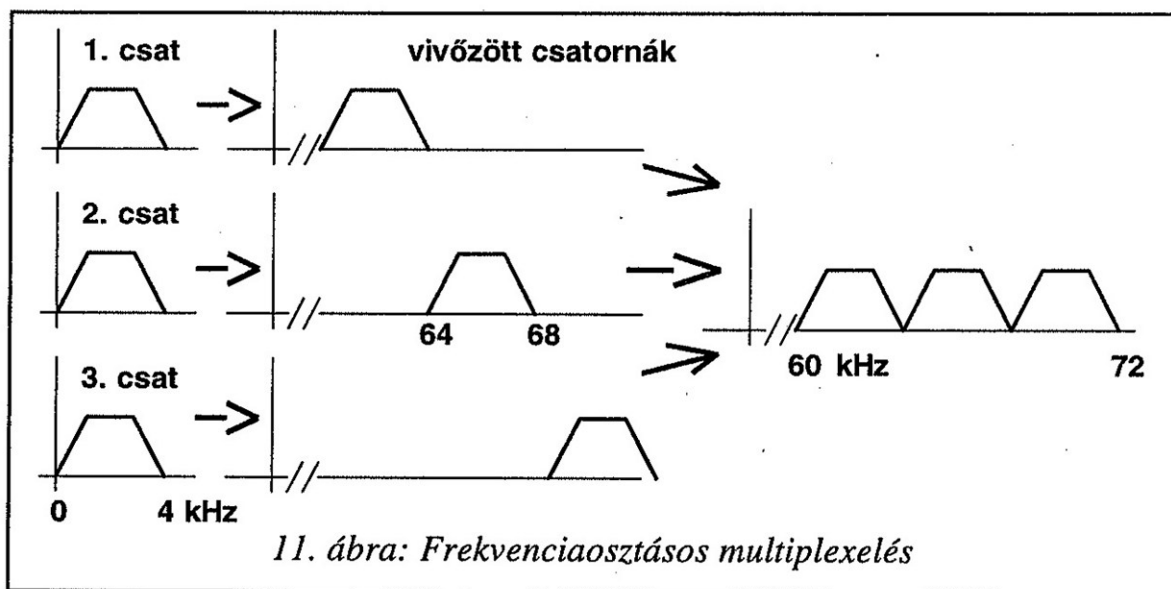
MULTIPLXELÉS FREKVENCIAOSZTÁSSAL

Frekvencia osztásos multiplexelés (FDM — Frequency-Division Multiplexing) üzemmódban elsősorban a távbeszélő-hálózatok vivőfrekvenciás rendszereinek szélessávú fővonalait használják. A széles frekvenciasávban időben is egyszerre haladnak a különböző vivőfrekvenciákra ültetett jelek. A módszer alapelve azon a tényen alakul, hogy szinuszos hullámok összegéből bármelyik összetevő egy megfelelő szűrővel leválasztható. Az adó oldalon a csatornák jeleit egy-egy vivőfrekvenciára ültetik (a vivőfrekvenciát a jelekkel modulálják), ezeket összegzik, az összegzett jelet átviszik a vevő oldalra, és ott ezeket szűrőkkel választják szét.

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 kihasználhatatlanok.

Például a szinuszos jel amplitudója hordozza az információt.

A jelösszegzés más néven a multiplexelés, a szűrőkkel történő szétválasztás a demultiplexelés.



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

Tizenkét beszédcsatornát távbeszélőcsoportba fognak össze, amely nemzetközileg a 60-108 kHz-es frekvenciatartományban fekszik. Öt távbeszélőcsoport egy távbeszélő főcsoportot alkot, ez 240 kHz sávszélességű. Tíz főcsoportból jön létre egy 2.4 MHz sávszélességű bázis-mestercsoport. Ezeket is 3-as vagy 6-os mestercsoportokba lehet foglalni.

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.

Az összefoglalóból az is nyilvánvaló, hogy ez a módszer nem igazán alkalmas számítógépek közötti információátvitelre, a csatornák emberi beszédre alapozott sávszélessége miatt.

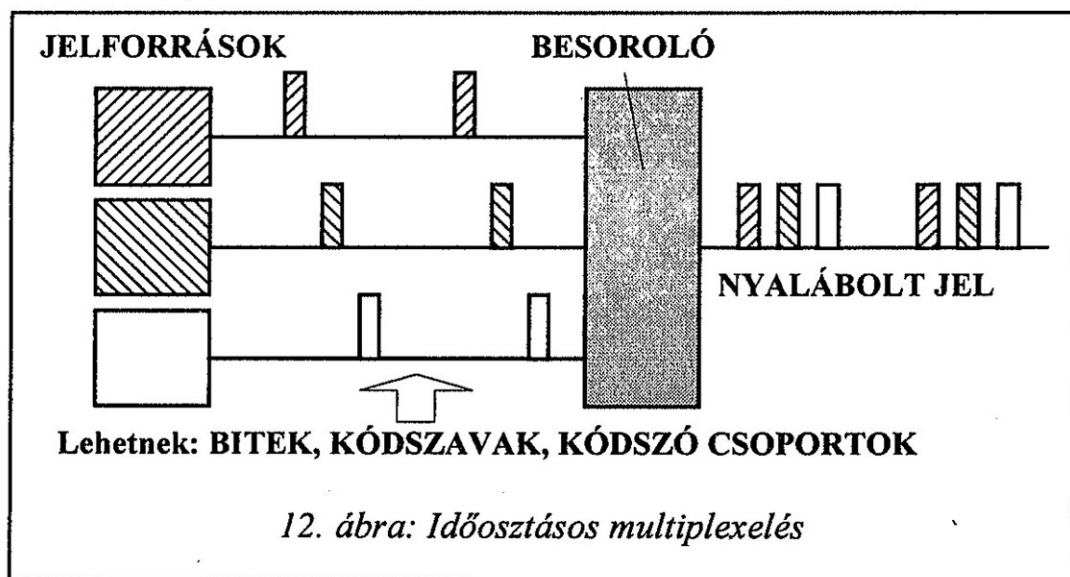
Természetesen a multiplexelt vonal minden egyes megcsapolásánál külön demultiplexer szükséges.

MULTIPLXELÉS SZINKRON IDŐOSZTÁSSAL

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

Lényegében csatornákat alakítunk ki a fizikai vonalon.

Minden elemi, digitális jeleket szállító adatcsatorna egy-egy időszelvet 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 vonalakat.



Néha a nagy sebességű fővonal közvetlenül a feldolgozó számítógéphez (illetve annak adatátviteli vezérlő egységéhez) csatlakozik és a demultiplexelés feladatát a számítógép látja el.

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 kihasználtságát. 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.)

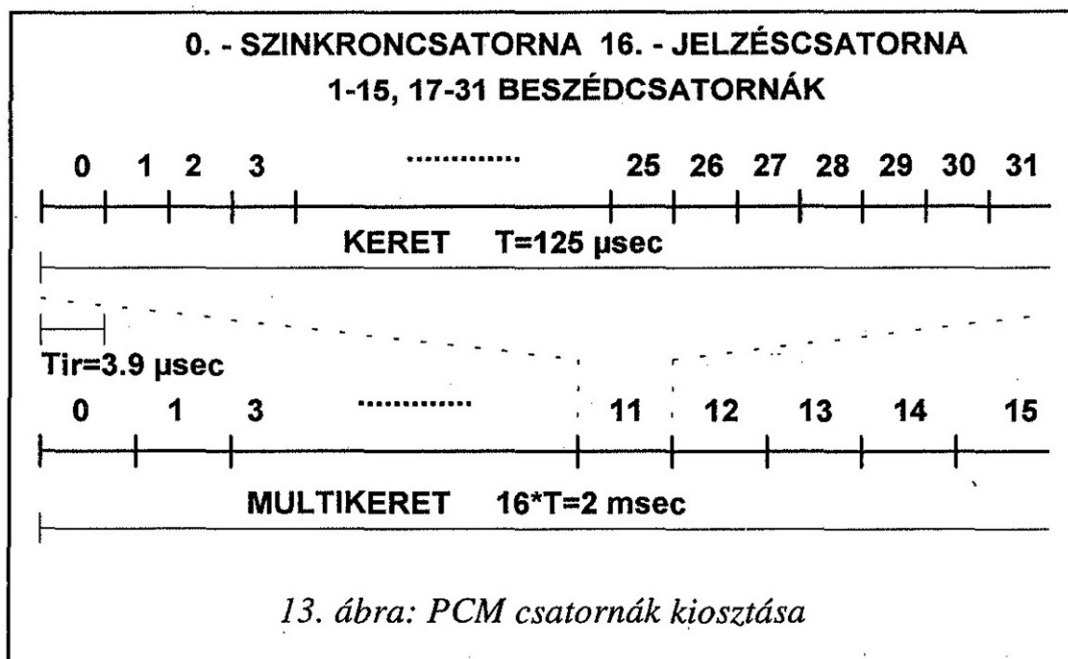
Például a telefontechnikában használt PCM (Pulse Code Modulation) impulzus kód moduláció esetén az eljárás a következő:

A mintavételezés (mivel a telefon sávszélessége 300...3400 Hz, és a Nyquist elv alapján, a maximális frekvencia legalább kétszeresével kell mintavételezni) szokásos értéke $f_v=8000\text{Hz}$, illetve a periódusidő

$T = 125 \mu\text{sec}$. 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 PCM átvitelben mivel minden impulzushoz $n = 8$ bit tartozik, az átviteli sebesség $8 \cdot 8000 = 64 \text{ kbit/s}$. Multiplexelés esetén a CCITT szerint az ún. primer csoport $N = 32$ csatornával. Az átviteli sebesség: $N \cdot n \cdot f_{\text{minta}} = 32 \cdot 8 \cdot 8000 = 2048 \text{ kbit/s}$.

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



13. ábra: PCM csatornák kiosztása

Természetesen ez a rendszer már más, nem beszéd jellegű digitális információ átvitelére is alkalmas.

A digitális multiplexelés ezen szintjét Európában E1-el jelölik és $32 \cdot 64 \text{ kbit/s}$ -os csatornából áll, 2048 kbit/s adatátviteli sebességű. Az USA-ban ennek megfelel a T1-el jelölt kialakítás, amely $24 \cdot 64 \text{ kbit/s}$ -os csatornából áll, és ez 1544 kbit/s adatátviteli sebességű.

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.

8000 darab 8 bites mintát viszünk át másodpercenként.

Egy másik példa: audio CD-knél a HIFI sztereo hangminőség 22 kHz-es sávzélességet igényel, ezért a mintavétel 44 kHz-el történik sztereo csatornánként, 16 bites felbontással. Ezért az átviteli sebesség:

$$2 \cdot 44\,000 \text{ minta/sec} \cdot 16 \text{ bit/minta} = 140 \text{ kbit/sec}$$

Ezt szokták egyszeres CD sebességnek hívni,

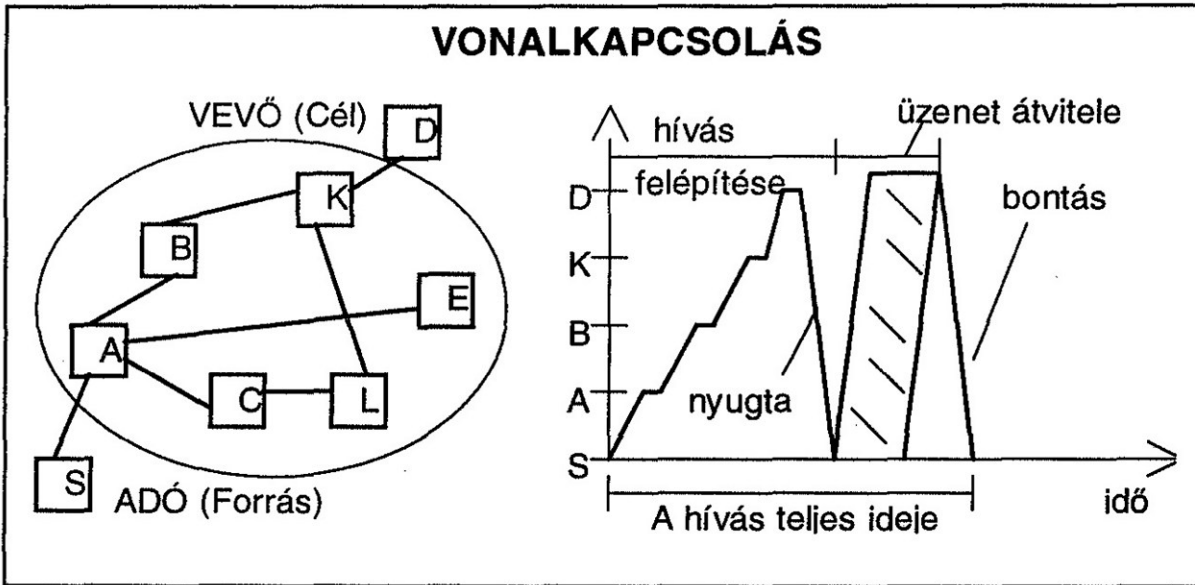
Vegyük észre, hogy az adatátviteli sebességnek nincs felső korlátja.

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.

A vonalkapcsolás esetén a rendelkezésre álló sávzélességet csak akkor tudjuk kihasználni, ha az információ áramlás folytonos.

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

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



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 mégis kisajátítja a vonalat. Ha a csatorna nem teljes kapacitással üzemel (telefonnál: hosszú csend), akkor ez, a vonal kihasználtságát rontja.

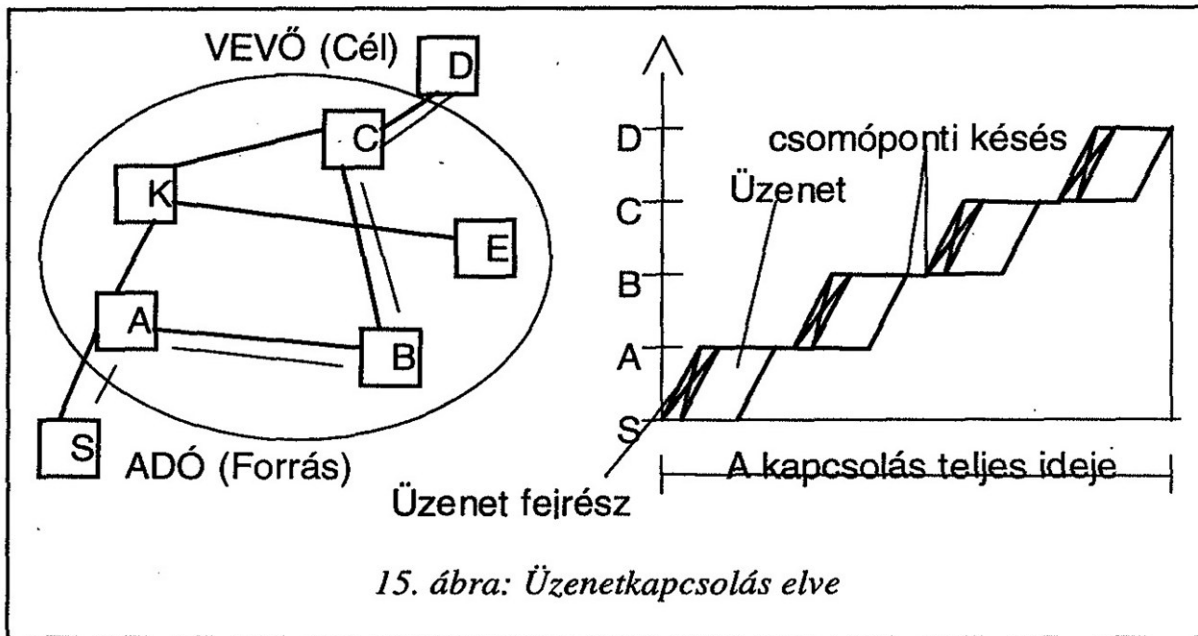
ÜZENET ÉS CSOMAGKAPCSOLÁS

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-eket igényel a teljes átviteli szakasz csomópontjain.

Másik hátránya az, hogy egy nagy üzenet akár percekre lefoglalhatja a közreműködő IMP-eket és a közöttük lévő átviteli csatornát. Ezért gyakrabban használatos (számítógépes hálózatoknál csaknem kizárólagosan használt) az a módszer, mikor az **átviendő adatblokk méretét korlátozzuk, és csomagokká bontjuk**.



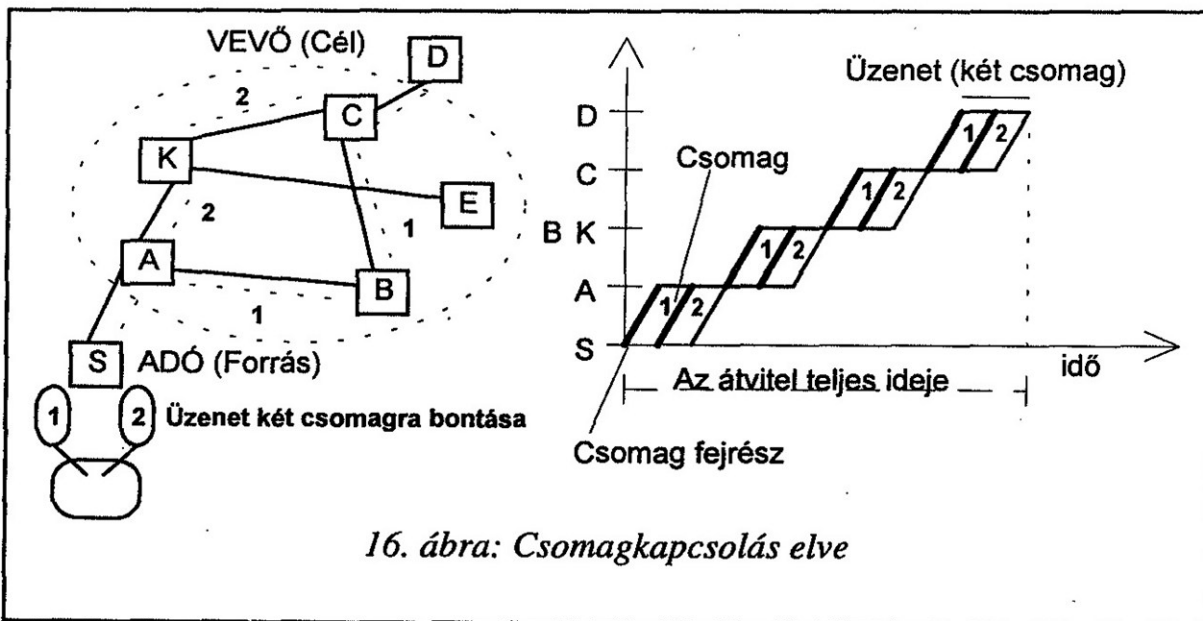
Figyeljük meg, hogy mielőtt az adatátvitel megkezdődne, a hívójelnek egészen a hívott pontig kell jutnia, majd onnan egy nyugtának kell visszaérkeznie



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. Másrésztől fennáll annak a veszélye, hogy a bemenő adatforgalom csomagjai úgy elárasztanak egy IMP-t, hogy korlátozott tárolókapacitása miatt

A vonalkapcsolás és a csomagkapcsolás sok tekintetben különbözik egymástól. A döntő különbség az, hogy a vonalkapcsolás a szükséges sáv szélességet statikusan előre lefoglalja, míg a csomagkapcsolás az igényekhez mérten hol lefoglalja, hol felszabadítja azt.



csomagokat veszít. Míg vonalkapcsolás esetén az üzenet lényegében egyben kerül átvitelre, csomagkapcsoláskor a csomagok sorrendje megváltozhat, és a sorrendhelyes összerakásukról is gondoskodni kell.

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:

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 forgalmas főút két oldalának összekötése — ha nincsenek kábelalagutak — kábelekkel szinte lehetetlen.

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.

A vezetékes rendszerek lehallgatás ellen védettebbek, kisebb távolságokon olcsóbbak lehetnek a telepítési költségei, de a kapcsolódó eszközök sokkal nehezebben helyezhetők át.

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)

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.

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 érpárt kívülről egy árnyékoló fémszövet burokkal is körbe vesszük, akkor árnyékolt sodrott érpárról (Shielded Twisted Pair = STP) beszélünk. A csavarás a két ér egymásra hatását küszöböli ki, jelkiszugárzás nem lép fel.

Általában több csavart érpárt fognak össze közös védőburkolatban.

Kétségtelen, hogy a már kialakított telefonvonalak megléte miatt a legszélesebb körben a rézvezetékeket használó huzalpárok vannak elterjedve.

Azt is tényként kell leszögezni, hogy a meglévő távbeszélő rendszerek nagy része csaknem kizárólag vezetékes kialakítású, és ezek felhasználása adja az összeköttetés mikéntjét.



Pontosan a sodrás biztosítja azt, hogy a szomszédos vezetékpárok jelei ne hassanak egymásra (ne legyen interferencia).

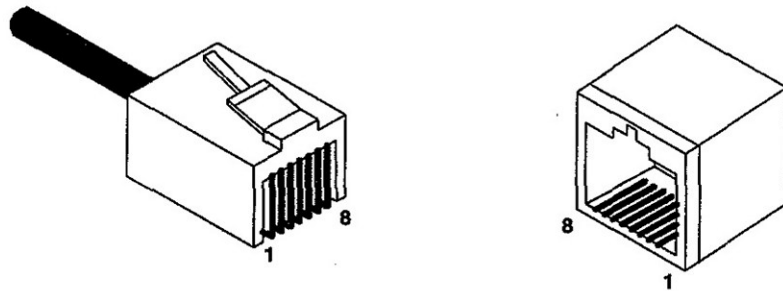
Az épületekben lévő telefon-hálózatoknál is csavart érpárokat használnak. A felhasználásuk számítógép-hálózatoknál is ebből a tényből indult ki: ezek a vezetékek már rendelkezésre állnak, nem kell új vezetékeket kihúzni a munkahelyekhez.

Ma már akár 100 Mbit/s adatátviteli sebességet is lehet ilyen típusú vezetékkel biztosítani. (Terjed az 1Gbit/sec sebességet biztosító Ethernet technológia is!)

Alkalmasak mind analóg mind digitális jelátvitelre is, áruk viszonylag alacsony. A zavarokkal szemben való érzékenységük tovább növelhető, ha árnyékolást alkalmazunk a csavart érpár körül. 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:

Típus	Használati hely
1. kategória	hangminőség (telefon vonalak)
2. kategória	4 Mbit/s -os adatvonalak (Local Talk)
3. kategória	10 Mbit/s -os adatvonalak (Ethernet)
4. kategória	20 Mbit/s -os adatvonalak (16 Mbit/s Token Ring)
5. kategória	100 Mbit/s -os adatvonalak (Fast Ethernet)

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 telefoncsatlakozót használják a csatlakoztatásra.



Az adó és vevő adatjelek a sodrott érpárú szegmensen polarizáltak, az egyik érpár vezetője a pozitív (+) a másik a negatív (-) jelölésű.

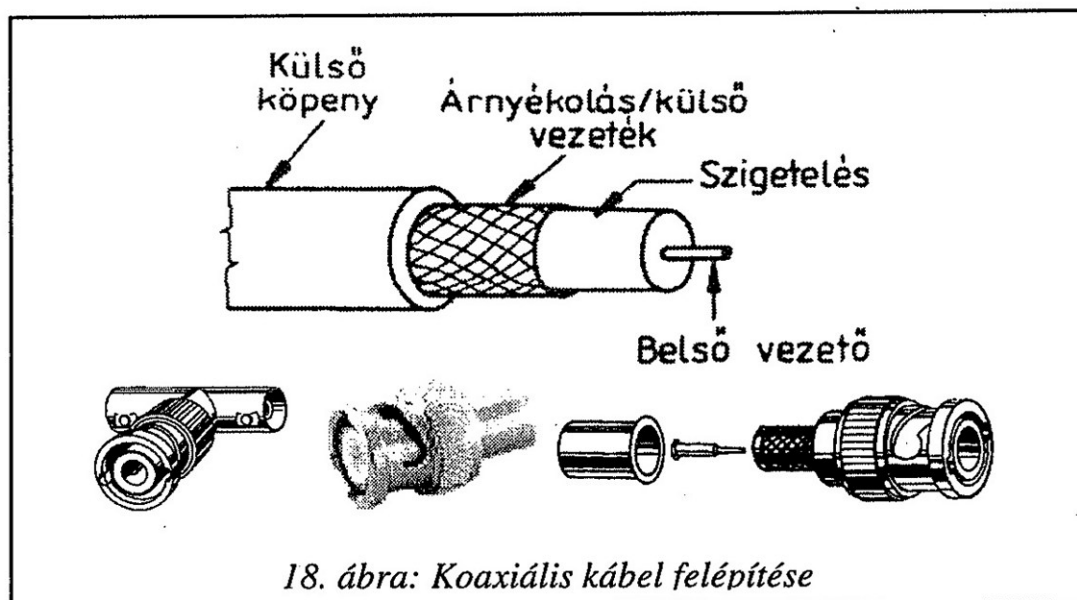
Csatlakozási Pont	Hozzárendelés
1	Kimenő adat +
2	Kimenő adat -
3	Bejövő adat +
6	Bejövő adat -
4,5,7,8	Más célokra fenntartva

17. ábra: Csavart érpár bekötése RJ-45-ös csatlakozóban

Ethernet hálózatokban 3.-5. kategóriájú kábeleket 10BaseT néven specifikálták. A 10Base-T jelölésben a T a sodrott(twisted) szót jelöli.

érpár adásra, a másik vételre szolgál. 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.

KOAXIÁLIS KÁBELEK



18. ábra: Koaxiális kábel felépítése

A másik vezeték kialakítási megoldás a koaxiális kábelek használata. Felépítése a 18. ábrán látható. Széles körben két fajtáját alkalmazzák: Az egyik az alapsávú koaxiális kábel, amelyet digitális jelátvitelre alkalmaznak, a másik az ún. szélessávú koaxiális kábel amelyet pedig analóg átvitelre használnak. Csatlakozásra BNC, (Bayone-Neil-Councilman) dugókat (RG58), aljzatokat és lezárókat használnak.

A koaxiális kábelek 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 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 ezt még az anyag szerkezetének lyukacsossá tételével tovább csökkenthető.

A kábel okozta veszteség az ohmos komponensekből, a dielektrikumban keletkező és a sugárzás okozta veszteségekből

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

Az egyszeres árnyékoló harisnya nem fed tökéletesen, nem véd teljesen a környezet zavaraitól, ezért kettős árnyékoló harisnyát vagy egyszeres és kétszeres alumíniumfólia árnyékolást használnak olyan kábelekben, amelyeket zavarokkal erősen terhelt környezetben alkalmaznak.

tevődik össze. A frekvencia növekedésével a bőrhatás 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 számítógéphálózatok kialakítására alkalmazzák. Az alapsávú koaxiális kábelek jellemző maximális adatátviteli sebessége 100 Mbit/sec 1 Km-es szakaszon. Az átviteli sáv szélesség nagymértékben függ a távolságtól. Tehát kisebb távolságon nagyobb sebesség is elérhető.

A digitális átviteltechnikában **vékony koaxiális kábeleket** Arcnet és Ethernet helyi hálózatok kialakításánál használnak. 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: ún. **vámpírcsatlakozókat** 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.

SZÉLESSÁVÚ KOAXIÁLIS KÁBELEK

A másik fajta koaxiális kábelrendszer a kábeltelevíziós szabványos kábeleken keresztüli 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 sáv szélesség, ezért a szélessávú rendszereket általában több csatornára osztják.

Ethernet hálózatokban az alapsávú koaxiális kábelek két típusa ismert az ún. vékony (10Base2) és a vastag (10Base5). A típusjelzésben szereplő 2-es és 5-ös szám az Ethernet hálózatban kialakítható maximális szegmenshosszra utal:

- vékony kábelnél ez 200 méter,
- vastagnál 500 méter lehet.

SZÁMÍTÓGÉP - HÁLÓZATOK

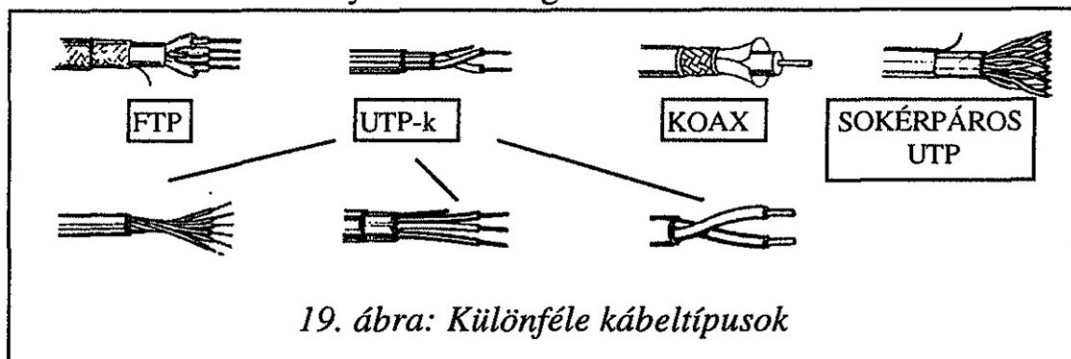
Az egyes csatornák egymástól függetlenül képesek pl. analóg televíziójel, csúcsminőségű hangátviteli jel, vagy digitális jelfolyam átvitelére is.

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.

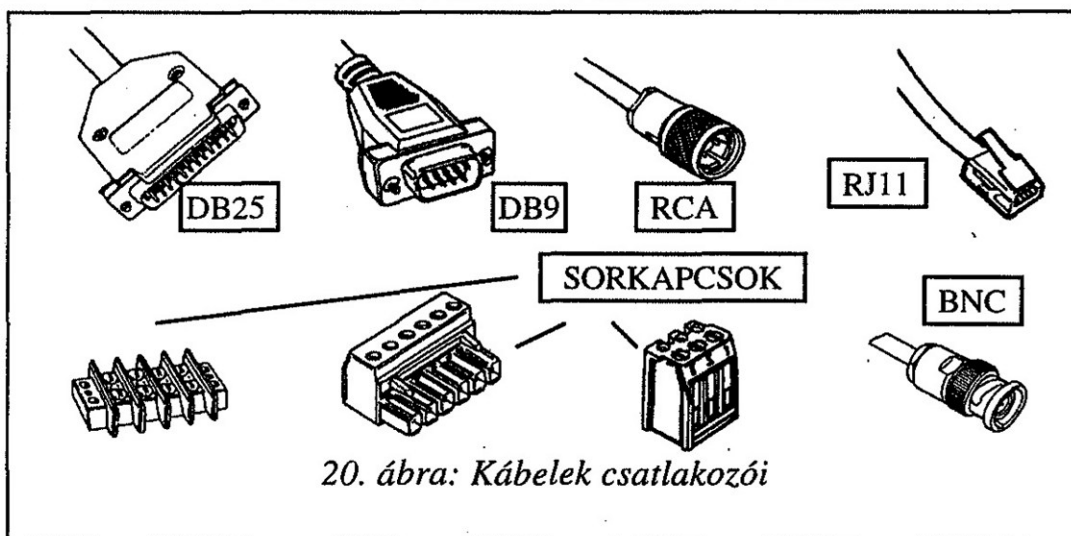
A kétkábeles rendszerben két azonos kábel fut egymás mellett. A két kábelben ellentétes irányú az adatforgalom.

Az alapsávú és a szélessávú technika közötti egyik legfontosabb különbség az, hogy a szélessávú rendszerekben analóg erősítőkre van szükség.

Egykábeles rendszerben egyetlen kábelben két különböző frekvenciatartomány van az adó (adó-sáv) és a vevő (vevő-sáv) részére.



19. ábra: Különböző kábeltípusok



20. ábra: Kábelek csatlakozói

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

ÜVEGSZÁLAS KÁBEL

A jelenlegi legkorszerűbb vezetékes adatátviteli módszer, az üvegszál technológia alkalmazása. Az információ fényimpulzusok formájában terjed egy fényvezető közegben, praktikusán egy üvegszálon. Az átvitel három elem segítségével valósul meg:



fényforrás — átviteli közeg — fényérzékelő.

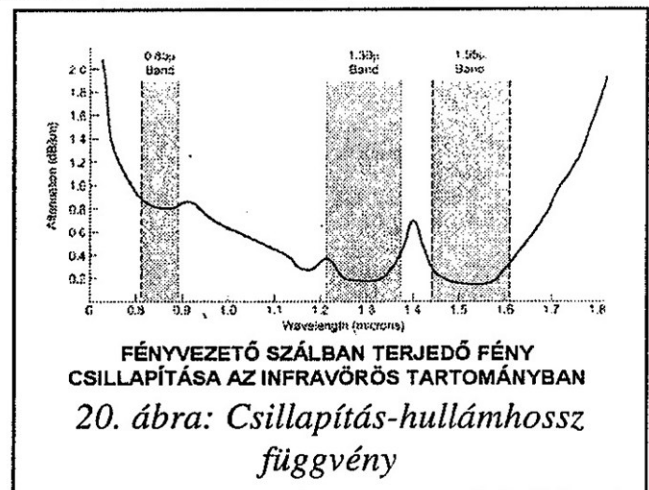
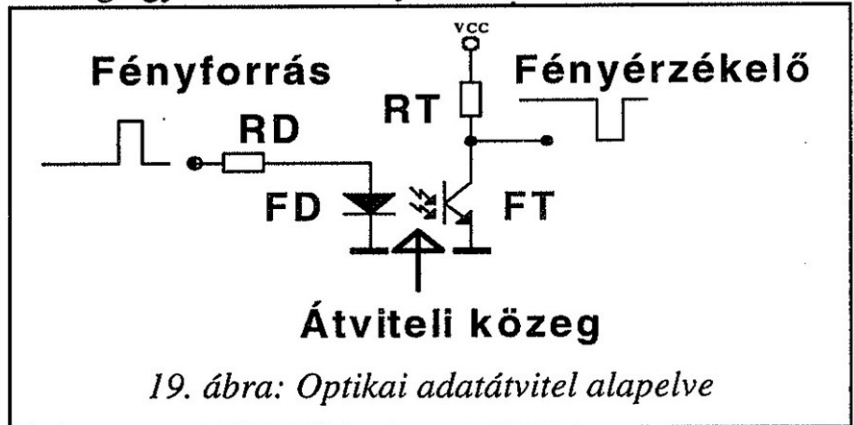
A fényforrás egy LED dióda, vagy lézerdióda. Ezek a fényjeleket a rájuk átfolyó áram hatására generálják. A fényérzékelő egy fototranzisztor vagy fotodióda, amelyek vezetési képessége a rájuk eső fény hatására megváltozik. Az átviteli közeg egyik oldalára fényforrást kapcsolva a közeg másik oldalán elhelyezett fényérzékelő a fényforrás jeleinek megfelelően változtatja az vezetőképességét.

Az elektronikában használt optikai kapu működése jól illusztrálja a működési elvet: A fotodiódára az RD ellenálláson keresztül kapcsolt pozitív feszültség a diódát nyitja, az átfolyó áram hatására fényt bocsájt 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 a fényvesztéseket minimálisra kell csökkenteni.

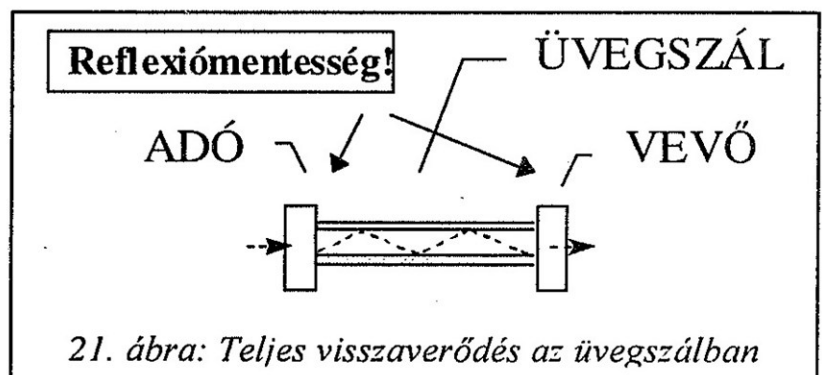
Fényvesztés három részből áll: a két közeg határán bekövetkező visszaverődés (reflexió), a közegen létrejövő csillapítás és a közegek határfelületén átlépő fénysugarak. Az első hatás a határfelületek gondos összeillesztésével minimálisra csökkenhető. Döntő jelentőségű az a tény, hogy a csillapítás nem az üveg alapvető tulajdonsága, hanem azt az üvegben lévő szennyeződések okozzák. A csillapítás megfelelő anyagválasztással minimalizálható. A csillapítás, vagyis a fényerősség csökkenése a használt fény hullámhosszától is függ.

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 21.

Az optocsatolókat az elektronikában széles körben használják áramkörök galvanikus leválasztására.



Az üvegszálas kábelek két alapvető típusa:
 többmódusú üvegszál (multimode fiber)
 egymódusú üvegszál (monomode fiber)

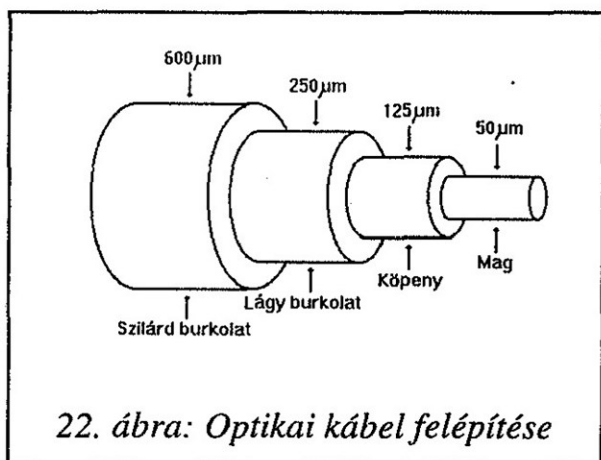


ábrán foglaltuk össze az elmondottakat. Az üvegszálaban az adóból kibocsátott számos fénysugár fog ide-oda verődni, az ilyen optikai szálakat többmódusú üvegszálnak (multimode fiber) nevezik.

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

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

Gondoskodni kell arról, hogy az optikai szál 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”, meg kis sugárban meghajlítani, mert az üvegszálak megszakadhatnak.)



A hagyományos, rézvezetékeket tartalmazó kábel és a fénykábel konstrukciós követelményei között az alapvető különbség az, hogy míg a rézvezetéknel 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.

A **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 kivenni illetve jeleket tudunk a kábelbe bejuttatni.

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 kábelben haladó fényjel villamos jelként is megjelenik, ezért ez közvetlenül elektromos jelillesztésre is felhasználható.

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

1. Csatlakozók használata (10-20%-os veszteség)
2. 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%).
3. 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 összeforrasztott szálakat.

Az ábra csak egy üvegszál felépítését mutatja, ilyen szálból egy közös védőburkolatban többet helyeznek el.

Az illesztés természetesen fényvesztéssel (és így csillapítással) jár, ezért adott távolságon csak adott számú illesztő használható.

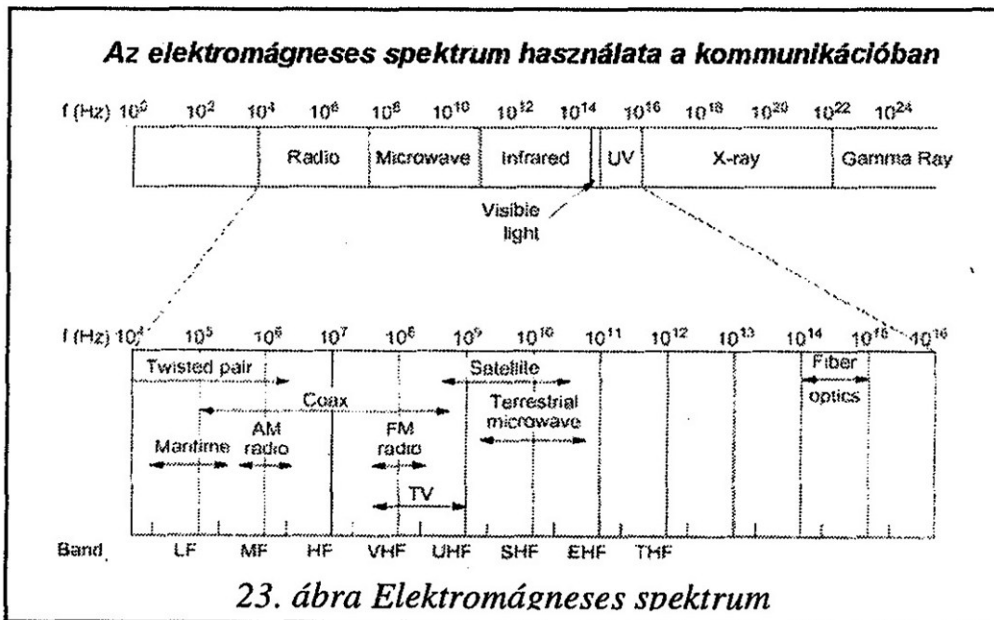
Ahogy az eddigiek szerint is nyilvánvaló, az üvegszálon adott hullámhosszú fényt használva csak egyirányú adatátvitel képzelhető el. Gyűrű kialakítású topológiánál az állomások illesztővel csatlakoznak a hálózatra, így egy vonalon is képesek venni (jel az illesztőbe bejön) és adni (illesztőn továbbadni).

Kétirányú pont-pont átvitel esetén már két üvegszálkapcsolat 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.

Ha az üvegszálon több eltérő hullámhosszú fényt viszünk át, akkor hullámhossz multiplexelést valósítunk meg, és több csatorna alakítható ki egy üvegszálon. Természetesen ilyenkor a fény be- és kicsatolása fényszűrőkön, prizmákon keresztül valósítható meg.

Ethernet hálózatokban az üvegszálkapcsoló kábelt 10BaseF néven definiálták.

VEZETÉK NÉLKÜLI ÁTVITELI KÖZEGEK



Az nyilvánvaló, hogy sok előnye van a vezeték nélküli kapcsolatnak, de vannak hátrányai is:

- lehallgatható
- a kialakítható csatornaszám erősen korlátozott,
- drága (még...).

Az ábrán foglaltuk össze, hogy a különféle kommunikációs módszerek az elektromágneses sugárzás melyik tartományát használják.

Hálózat kiépítéskor 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.

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 energiakoncentráció miatt rendkívül jól irányított, amely szinte teljesen védetté teszi az illetéktelen lehallgatás, illetve külső zavarás ellen.

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

Például kaphatók olyan nyomtatók, amelyeket már nem kell kábellel a számítógéphez kötni, az adatátvitel IrDA interfészen keresztül történik.

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 sugárnyalábokat küldenek és akár száz kilométert is átfoghatnak. A jelisméltést itt relézó állomásokkal oldják meg, azaz a vett jelet egy más frekvencián a következő, relézó állomásnak továbbítják.

Problémaként jelentkeznek a viharok, villámlás, egyéb légköri jelenségek. A frekvenciasávok kiosztása átgondolást igényel, és hatósági feladat.

Erősen korlátozott a kiosztható frekvenciasávok miatt

SZÓRT SPEKTRUMÚ SUGÁRZÁS

Kisebb távolságokra (kb. 1 km távolságig), lokális hálózatoknál használt megoldás, Széles frekvenciasávot használ, amit egy normális vevő fehér zajnak érzékel. (Azonos amplitúdó minden frekvencián.) A szórt spektrumú vevő felismeri és fogja az adást. Antennaként megfelel egy darab vezeték.

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, Az ún. geostacionárius pályára állított műholdak a Földről állónak látszanak. A mai technológia mellett 90 geostacionárius műhold helyezhető el ezen a pályán (~4 fokenként). A frekvenciatartományok a távközlési műholdaknál: 3,7...4,4 GHz a lefelé, 5,925...6,425 GHz a felfelé irányuló nyaláb számára.

A geostacionárius pályára állított műholdak a Földről állónak látszanak

A műholdak tipikus sáv szélessége 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.

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 úrszegmens és a földi szegmens.

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

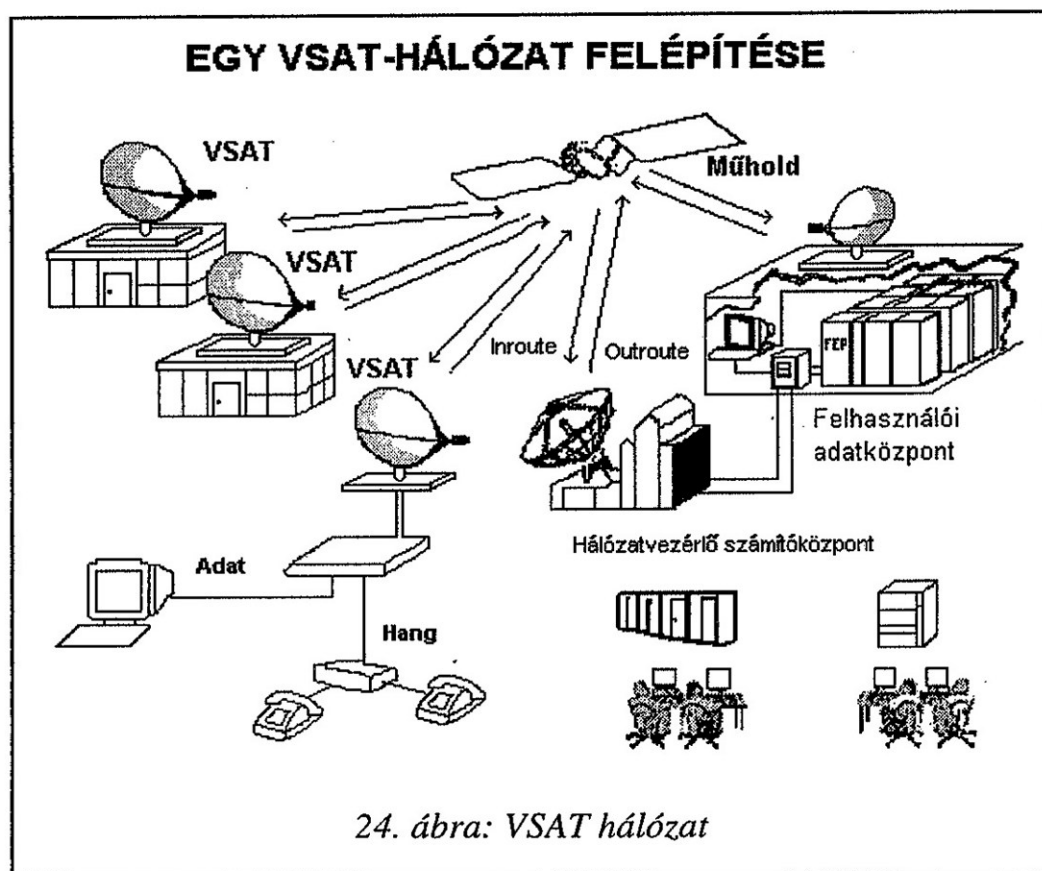
Magát az összeköttetést egy geostacionárius távközlési műhold biztosítja. A tipikus VSAT-hálózat lényegében csillaghálózat, amelynek a középpontjában a hub helyezkedik el.

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.

Ha a transzponderek az adást polarizálják, több transzponder is használhatja ugyanazt a frekvenciát.

Tudnunk kell, hogy a műholdas átvitel késleltetése a földi mikrohullámú illetve a vezetékes rendszerekhez képest jelentős a nagy távolság miatt: 250-300 msec.

Míg az első kisméretű terminálokat adatszórás célra használták, a 80-as évek elejétől a kétirányú VSAT-ok kezdtek elterjedni.



Az analóg átvitel a folyamatosan (nem diszkrét módon) változó jelek átvitelén alapszik. Ezeknél a jeleknél egy folytonosan változó fizikai jellemző hordozza az információt.

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

ANALÓG ÁTVITEL

A múltat teljes egészében az analóg átvitel jellemezte. Berendezések, az átviteli módszerek mindegyike analóg volt, gondoljunk a telefonra, a rádióra és a televízióra. A kialakított kommunikációs infrastruktúra is döntően analóg. Még évtizedek fognak eltelni míg a digitális átviteli rendszerek széles körben elterjednek, ezért az analóg rendszerek tanulmányozása fontos.

TELEFÓNIA

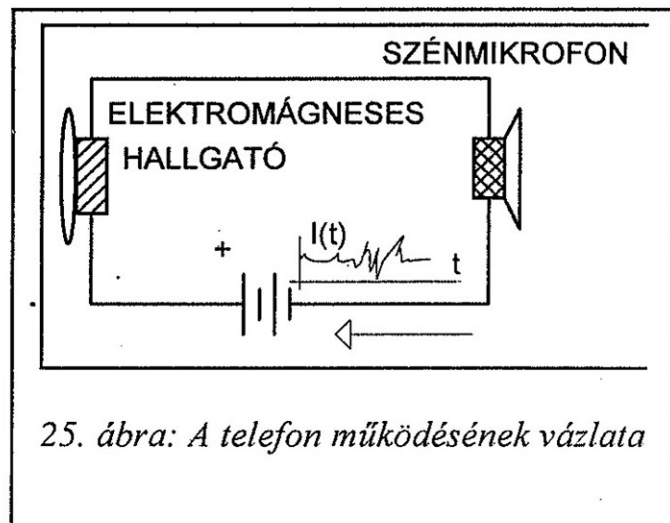
A nyilvános távbeszélő hálózatokat az emberi hang többé-kevésbé felismerhető módon való átvitelére tervezték. Számítógépek kommunikációjára való felhasználásuk csak igen nagy nehézségek árán lehetséges, de mégis szükséges, mert egyszerűen ez áll rendelkezésre. Számítógépeket összekötő adatátviteli kábeleken az adatátviteli sebesség minimum Mbit/s-os nagyságrendű, igen kis hibaarány mellett. Telefonvonalon keresztül ez mindössze 10 kbit/s nagyságrendű, jelentősen nagyobb hibaarány mellett, amit vonal és a kötések öregedése folyamatosan növel.

Kiemelten kell foglalkozni a távbeszélő rendszerekkel, mivel sok felhasználó hálózati elérése csak ezeken keresztül valósulhat meg.

Ezen hatalmas ellentmondás miatt rendkívül sok erőfeszítést fejtenek ki a távbeszélő vonalak felhasználási hatékonyságának a növelésére.

Hogyan is működik a telefon?

A szénmikrofon ellenállása (amely egy membránnal lezárt szénpor réteget tartalmaz) a rábeszél hang hatására változik. A hanghullámok a membránba ütközve mozgatják azt, és a szénzemcsék változó mértékben összepréselődnek. Ezért a körben folyó áram a hang 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.



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.

Minden előfizető két vezetékkel a hozzá közeli helyi központhoz 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 szerepek.

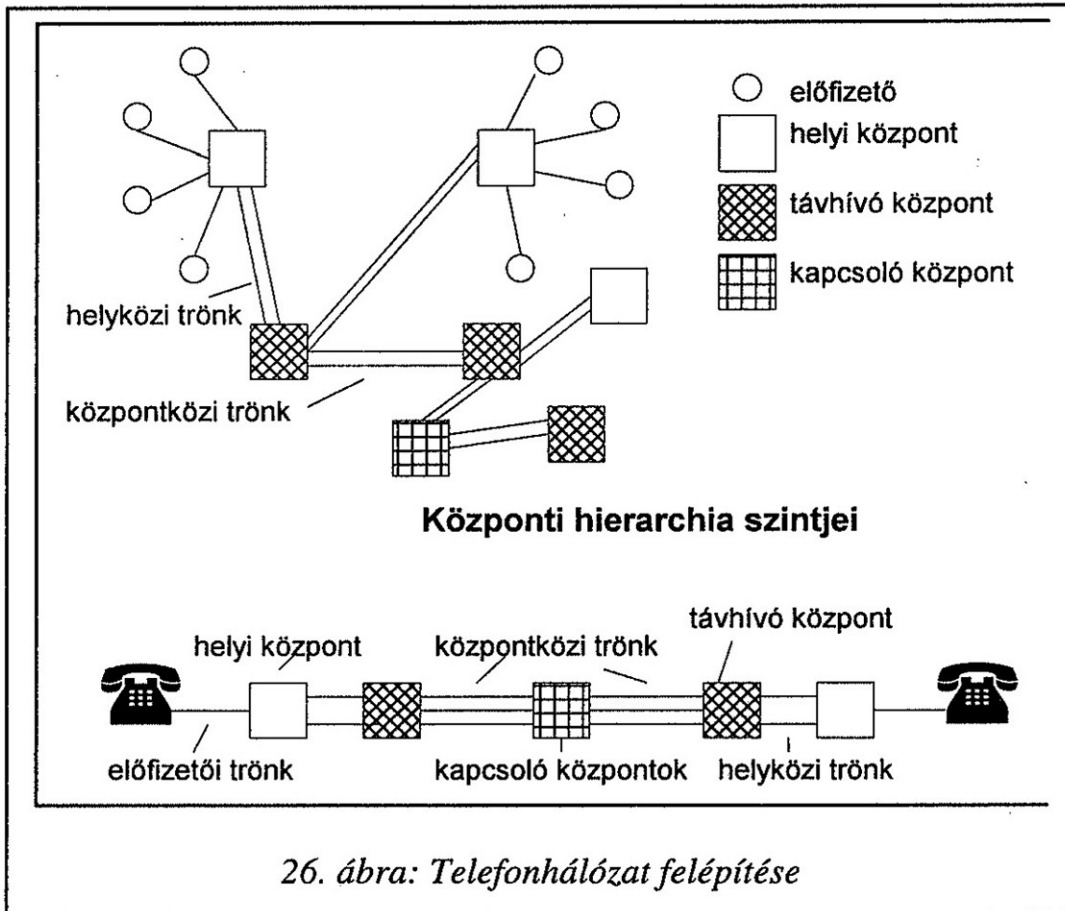
A helyi központok több vezeték-párral (nevük: helyközi trönk) kapcsolódnak a távhívó központhoz. Ezeken keresztül a helyi központok közötti információcsere valósul meg. Természetesen a két

Elnézést azoktól, akik tudják...

Bár elvileg a világ összes telefonja egy gigantikus központon keresztül összeköthető lenne, a központok mégis többszintű hierarchikus rendszerként épülnek fel.



előfizető távhívó központon keresztüli összekapcsolása csak akkor lehetséges, ha mindkét előfizető helyi központja ugyanazon távhívó központhoz kapcsolódik.



Ha a távhívó központ nem közös, akkor az összeköttetés kialakítása a kapcsolóközpont hierarchia következő szintjén történik. Ezek a magasabb szintű kapcsolóközpontok segítségével valósulnak meg.

Legáltalánosabban a lakásokban levő telefon-vonalak használatosak, ezek két vezetékes **kapcsolt vonalak (dial-up-line)**. Ez azt jelenti, hogy csupán az összeköttetés idejére kapcsolódik a telefonközponton keresztül a hálózathoz.

A **bérelt vonalak** dedikáltak, 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 hertz között van, a sávszélessége tehát 3100 hertz.

Lehetséges a négy vezetékes vonal is, azt azonban inkább a bérelt vonalakon használják. Ez utóbbiban az egyik érpáron folyik az adatküldés, a másikon a vétel.

A bérelt vonalat adatvonalnak is hívjuk, miután alkalmasabb az adatátvitelre.

CELLÁS MOBIL RÁDIÓTELEFONOK

GSM=Global System for Mobile Communication (mobil kommunikációs világszisztem) Vívőfrekvenciaként 450, 900 vagy 1800 MHz-et használ, idő-multiplexelést végző multiplex-csatornát használ (egyidejűleg többen is beszélhetnek ugyanazon a csatornán).

A cellás technika a cellaosztáson és a frekvenciák ismételt felhasználásán alapul.

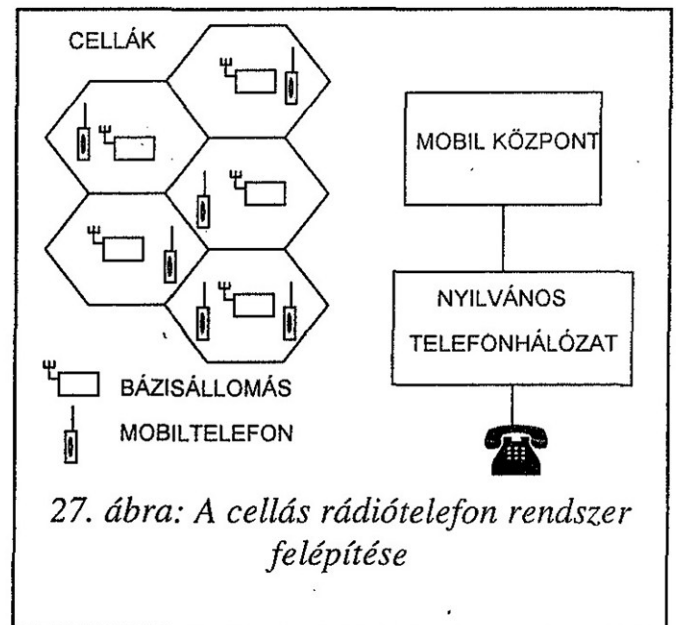
A cellás szerkezetű rádiótelefon rendszerek az igényeket a rendelkezésre álló frekvenciatartomány kihasználtságának növelésével elégítik ki. A cellás technika a cellaosztáson és a frekvenciák ismételt felhasználásán alapszik. A területet kisebb részekre osztják. A cellákon belül egy központi rádióállomás 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 kiosztható.

A gyakorlatban a cellák tényleges alakját az antenna típusa és a helyi körülmények befolyása határozza meg, de elméleti célokra általánosan elfogadott a szabályos hat-szöggel való közelítés. A celláknak azt a legkisebb csoportját, ahol a használható frekvenciákat tartalmazó csatornákészlet kiosztásra kerül cellacsoportnak (clusternek) nevezik.

Az azonos frekvenciákat használó cellák közötti távolságot úgy kell megválasztani, hogy az azonos csatornák kölcsönhatása (interferenciája) megfelelően kicsi legyen.

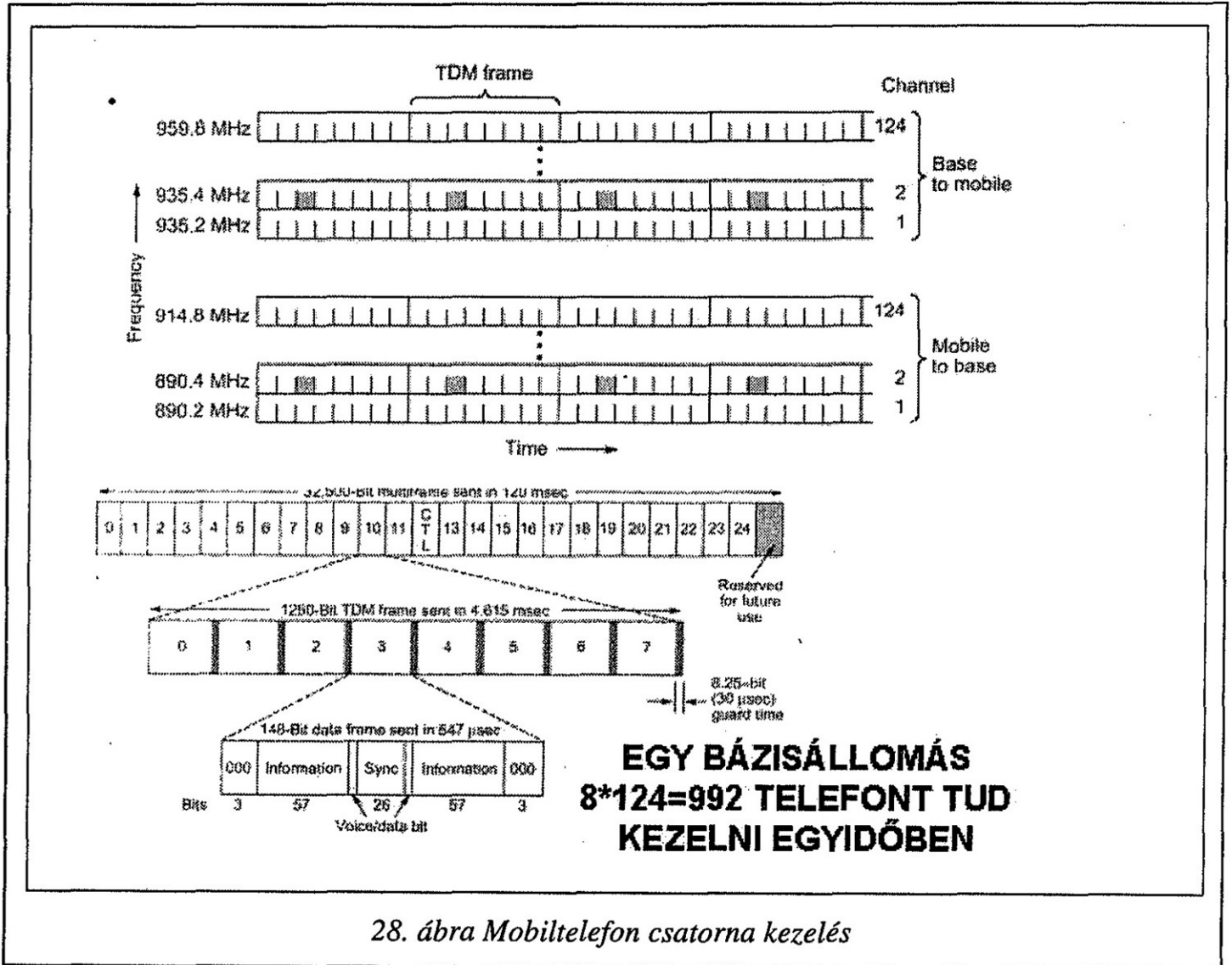
A felhasználók egy cellán belül a helyi bázisállomáson keresztül tartják a rádiós kapcsolatot. A bázisállomás hálózat a mobil központhoz csatlakozik rádiós vagy vezetékes összeköttetéssel. A mobil központ feladata a cellás rendszer működésének vezérlése, és a nyilvános postai távbeszélő hálózathoz való 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 kiosztható beszédcsatornával. Ezt a váltást handovernek vagy handoffnak nevezzük.



SZÁMÍTÓGÉP - HÁLÓZATOK

[1] A cellák 200 kHz széles frekvenciasávú duplex csatornával rendelkeznek, amelyek száma maximálisan 200 lehet. A **28. ábrán** egy ilyen, 124 csatornát tartalmazó bázisállomás frekvenciakiosztása látható, amely csatornák mindegyike időosztásos multiplexeléssel egyidejűleg 8 összeköttetést támogat. Ilyen módon a cellában összesen $124 \cdot 8 = 992$ mobil lehet aktív, de ezek közül számos a szomszéd cellák közelsége miatti zavarás miatt nem használható.



Például az ábrán a 935.4 MHz-es frekvenciasáv besötétített időszelleteit használja egy mobiltelefon a bázisállomástól történő vételre, és a 890.4 MHz-es frekvenciasávot adásra. Az időszeltesben lévő 148 bites információ tartalmazza az adat vagy hanginformációt. 8 időszeltes együtt egy TDM csoportot (keretet alkot), amelyben 1250 bit van.

Ezek még 120 msec-onként elküldendő 26 TDM csoportot alkotó ún. multikeretbe vannak szervezve, amelyben a 12.-ik csoport a vezérlésre, a 25.-ik csoport továbbfejlesztésre van fenntartva.

A csatornák kijelölésre négy módszer használatos:

Fix csatornakiosztás

A jelenleg működő rendszerek fix csatornakijelöléssel dolgoznak. A fix kijelölés a csatornákat úgy rendeli hozzá az egyes cellákhoz, hogy ezen a kiosztáson a későbbiek során már nem változtat. Előnye, hogy a kiosztást csak egyszer kell elvégezni, hátránya, hogy nem tud az egyes cellák forgalmi ingadozásaihoz alkalmazkodni.

Ha a fix kiosztású rendszerben egy cellában minden csatorna foglalt, a hívás letiltódik. Elkerülésére vezették be a csatornakölcsönzést. Ekkor a szomszédos cellák valamely szabad csatornája fogja kiszolgálni a hívást, ha a kölcsönzés nem zavarja a már folyó beszélgetéseket. Az eljárás hátránya, hogy nehéz forgalmi feltételek mellett a kölcsönzés további kölcsönzések sorozatához, végül a későbbi hívások letiltásához vezethet.

Dinamikus csatornakiosztás

Ez az eljárás az igényeknek megfelelően rendeli a csatornákat a cellákhoz az igény kiszolgálásának időtartamára. Bármelyik csatornát bármelyik cella megkaphatja, feltéve, hogy a csatorna újrafelhasználási távon belül lévő más cella az adott pillanatban nem használja a kiosztandó csatornát. Előnye, hogy rugalmasan alkalmazkodik a forgalom ingadozásaihoz. Hátránya, hogy nagy terhelések esetén nem lehet teljesíteni a sűrű csatornakiosztást. Ez újrendezéssel csökkenthető: az egymástól távolabbra kiosztott csatornákat amikor lehet, úgy rendezik át, hogy az azonos csatornákat használó cellák a megengedhető legkisebb távolságra legyenek egymástól.

Hibrid csatornakiosztás

A csatornákat két csoportba sorolják. Az egyik részt fix módon megkapják az egyes cellák, a másikat pedig dinamikus kiosztásra fenntartják.

Adaptív csatornakiosztás

Az adaptív eljárásnál a csatornakiosztás csak egy adott hosszúságú időintervallumban érvényes. Az időintervallumok elején a csatornák például fix módszerrel kerülnek kiosztásra a forgalmi igények pillanatnyi területi eloszlása alapján.

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 kód megadását, rendszertől függően, párszor meg lehet ismételni, majd az **Illetékességet Meghatározó Központ** letiltja a készüléket, amit csak szerviz tud újraéleszteni.

A csatornakijelölési módszereknek négy típusa van:

- fix
- dinamikus
- hibrid
- adaptív.

Egyszerű kölcsönzés esetén egy csatorna csak akkor adható kölcsön, ha egyidejűleg szabad mindhárom legközelebbi azonos csatornájú cellában.

A hibrid módszer átmenet a fix és a dinamikus csatornakiosztás között.

Ennek segítségével pl. fel lehet deríteni egy ellopott készülék helyét (vagy egy mobil telefontal felszerelt ellopott autó helyét).

A mobil készülék csak a SIM-kártya behelyezésével és a személyi azonosító kód begépelésével válik üzemképesé.

HÁLÓZAT ALRENDSZER

Ez az alrendszer végzi a kapcsolásokat és az adatbázis funkciókat. A tárolt adatbázisok:

Honos Helyzet-meghatározó 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 Helyzet-meghatározó 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. Ezen központok hívásfelépítéskor használják az előbb említett adatbázisokat.

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.

ÜZEMELTETÉST TÁMOGATÓ ALRENDSZER

Ez a hálózati software, 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.

A készülék és a SIM-kártya tulajdonlása ilyen módon elválasztható. Maga az előfizetés a kártyához kötött, az is vásárolhat SIM-kártyát, akinek nincs saját készüléke, és fordítva (bár ez utóbbinak nincs sok értelme ;-)).

(pl. ha egy Zalában nyilvántartott előfizető Békés megyében kezdeményez hívást, vagy beszélgetés közben tér át egy másik területre).

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).

Ezeket a méhsejteket cellának nevezzük.

- **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.
- **Biztonság kezelés:** 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 — HÍVÁSFELEPÍTÉS

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 Helyzet-meghatározó 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.

A beszélgetés útvonala:
hívó fél - a hívó területén levő Mozgó Kapcsoló Központ - a hívott felet tároló Látogató Kapcsoló központ - Bázis Állomás Vezérlő - Bázis Állomás (antenna, méhsejt) - hívott (mobil) fél.

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 illetékeség 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

A jelen kiépítettségben egy adott hálózaton belül (pl. adott országban) az előfizető szabadon mozoghat, de beszélgetés közben nem lehet hálózatot váltani (pl. más országba átmenni). Az egyes hálózatok (pl. országok) közötti összeköttetést a hagyományos telefonrendszer biztosítja.

Mobilmentan

A Westikett nyomán

- Ne hivalkodjon mobiltelefonjával, ne hordja a készüléket kihívóan!
- Legyen mindig elérhető – közvetlenül, vagy hang, illetve írott üzenettel, de örködjön mások nyugalma! Társaságban csak akkor kezdeményezzen és fogadjon hívást, ha megbizonyosodott róla, hogy ez másokat nem zavar!
- A hívó fél mutakozzon 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 vacsorapartneréitől 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.

Amikor bolyongás során a mobil állomástól érkező jelek alapján egy Mozgó Kapcsoló Központ érzékeli, hogy az állomás a területére jutott, elküldi az információt az illető állomás Honos Kapcsoló központjának, majd megkeresi a régi Látogató Központot, és onnan törli az állomás információit.

Bár ez nem tananyag... de nagyon fontos!

bolyongás műszaki feltétele, hogy a GSM-rendszer ki legyen építve. Kereskedelmi és jogi feltétele, hogy a honos és az 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.

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 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.

A könyv írásakor jelentette be a WESTEL a 777sms.com

Az SMS-el vezérlések kialakítása, rövid információk automatikus lekérdezése is megvalósítható, aminek igen nagy távlatai vannak.

weblapjának működését, amelyről bárki ingyen küldhessen SMS-t a magyar mobilhálózatokba.

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 lehet az interneten lévő adatokat továbbítani. Elsődlegesen grafikai megjelenítést nem, vagy igen kis mértékben igénylő alkalmazások használhatók, többszörös menürendszer felhasználásával.

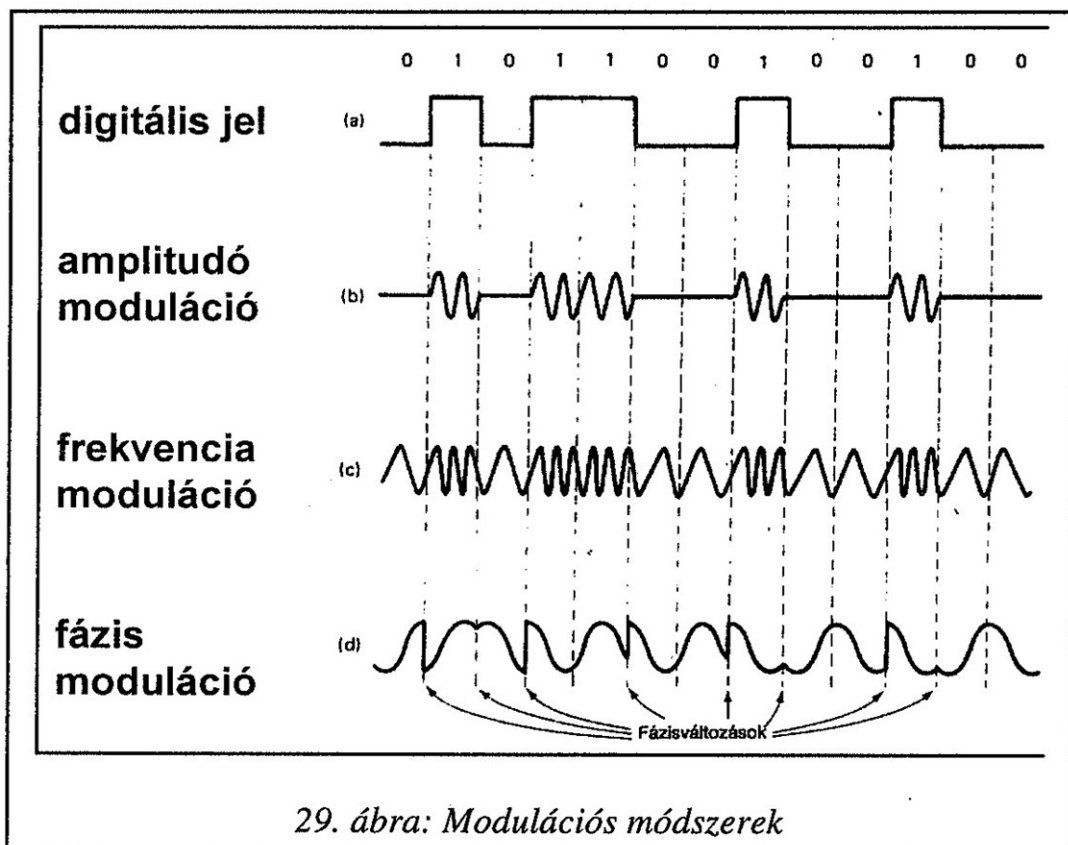
Lehetséges banki információk, szolgáltatások használata, adatlekérdezések, információs adatbázisok használata..

MODEMEK

Az „előfizetői hurok”, ami az előfizetőt a központtal összeköti: 300-3300Hz váltófeszültség átvitelére van korlátozva.

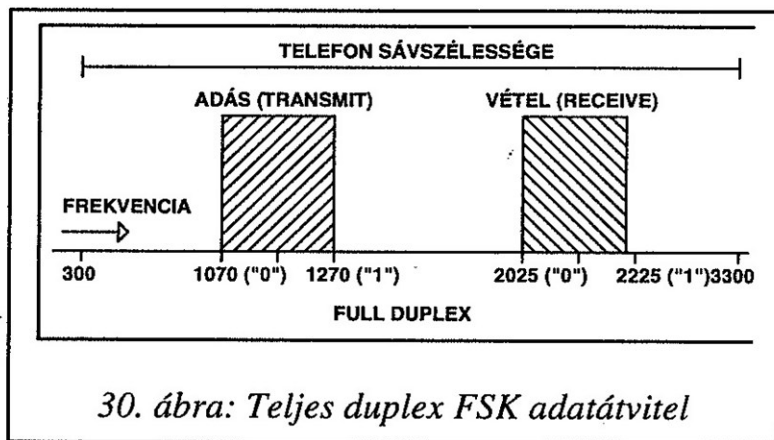
Bár az előfizetők a helyi központhoz egy fémes vezető párral csatlakoznak, és elvileg ezeken akár 1-2 Mbit/s-os adatátviteli sebesség is lehetséges, azonban a telefonbeszélgetések váltakozó áramú jeleit szűrők segítségével 300 Hz – 3.3 kHz között tartják. Ezért az információt ebbe a hangfrekvenciás tartományba eső szinuszos hullám valamelyik jellemzőjéhez (és annak diszkrét változásához) célszerű hozzárendelni. (29. ábra)

Legegyszerűbb eszköz számítógép-hálózat kiépítésére, a meglévő telefonhálózat felhasználásával.



A moduláció tetszőleges fizikai folyamat egy paraméterének megváltoztatása valamilyen elsődleges vezérlőjel segítségével. Szinuszos jel esetén annak amplitudóját, frekvenciáját, illetve fázisát lehet modulálni.

Ezt a hullám modulálásával érhetjük el. Szinuszos jel esetén annak amplitudóját, frekvenciáját, illetve fázisát lehet modulálni. 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.



A modemek működését funkcionális protokollok határozzák meg. Ezek:

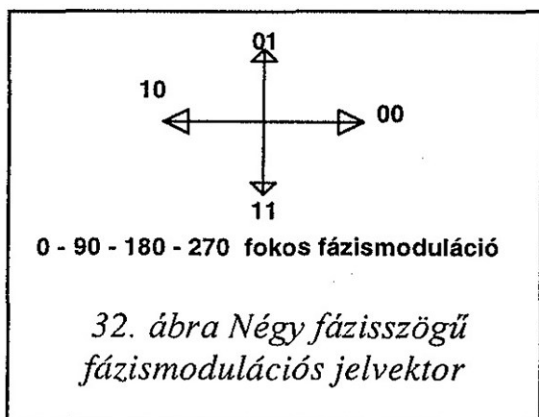
- modulációs protokoll (milyen modulációs módszert használ)
- hibajavító protokoll (error correcting)
- adattömörítő protokoll (compression)

MODULÁCIÓS PROTOKOLLOK

Az **amplitudó modulációra** jó példa a morze jelek használata. Egy adott amplitudóju jel (a hang), illetve annak hiánya (csend) hordozza az információt. Jól működő, de lassú átvitelt biztosító megoldás.

A frekvencia modulációt 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 32. ábrán látható.

A fázis-moduláció nyújt lehetőséget a telefonvonalon történő nagyobb adatátviteli sebesség elérésére.



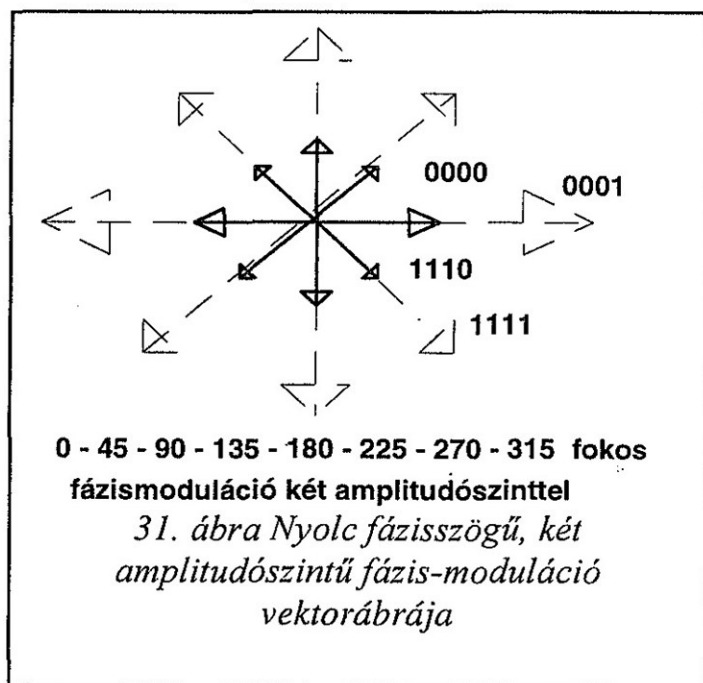
Az adónak 0 - 90 - 180 és 270 fokos fázisszög - kezdettel kell szinuszhullámot elküldenie, ami két bit kódolását, és így, egyszerre való elküldését biztosítja. Ha például 2400 Hz-es szinuszhullámot használunk, akkor másodpercenként

2400 darab szinuszhullámot küldünk át, amelynek négy különböző fázishelyzete lehetséges. Ezekhez két bites információt rendelve, az adatátviteli sebesség 4800 bit/s lesz! Az adatátviteli sebesség további növelése úgy lehetséges, ha növeljük a fázisszögek számát, illetve a hordozó szinuszhullámot különböző amplitudókkal küldjük. Például 8 fázisszög és 2 amplitudó esetén már egy jel 4 bitet képes kódolni !

Belátható, hogy a sikeres adatátvitel az amplitudók és a fázisszögek korrekt detektálását igénylik a vevő oldalán. Ezt a korszerű modemekben az analóg áramköri alkalmazásával lehet csak megoldani, és a

Az adatátviteli sebességet a használt alacsony frekvencia erősen korlátozza, mivel például a legkisebb, 1070 Hz-es frekvencián a minimális 1 teljes szinuszhullám átvitele ~1 msec, ami 1 kbit/s átviteli sebességet jelent.

Ilyenkor a modem folyamatosan küldi a szinuszhullámot. Ez viszi a fázis- és amplitudóinformációt. Ezért szokásos ezt a jelet vivőhullámnak (carrier) is hívni.



technikák kifinomult modemek minőségét

A modemek mint adatátviteli hálózati eszközök is tárgyalhatók az OSI modell alapján; a **33. ábra** ezt mutatja be.

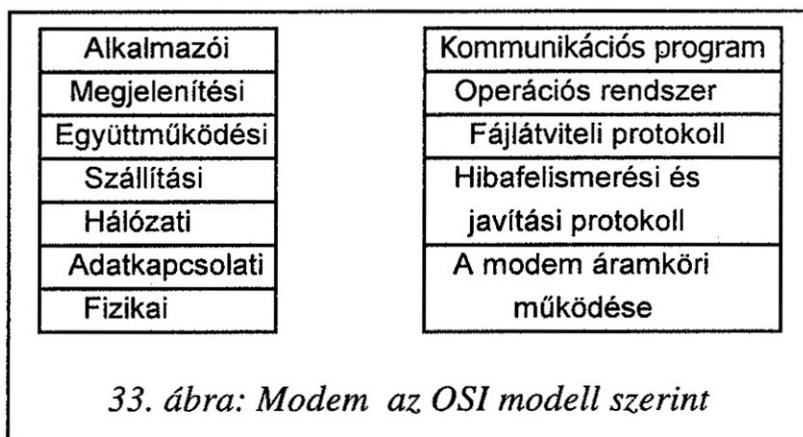
A modemek önállóan működő számítógépes perifériák, amelyeknek az adatátvitel megvalósításához a számítógépnek kell felprogramozni, parancsokkal vezérelni, és állapotát (státuszát) ellenőrizni.

Az összekapcsolás a később részletesen 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.

Az átviteli út 3 részből áll:

számítógép-modem
modem-modem
távoli modem-távoli számítógép

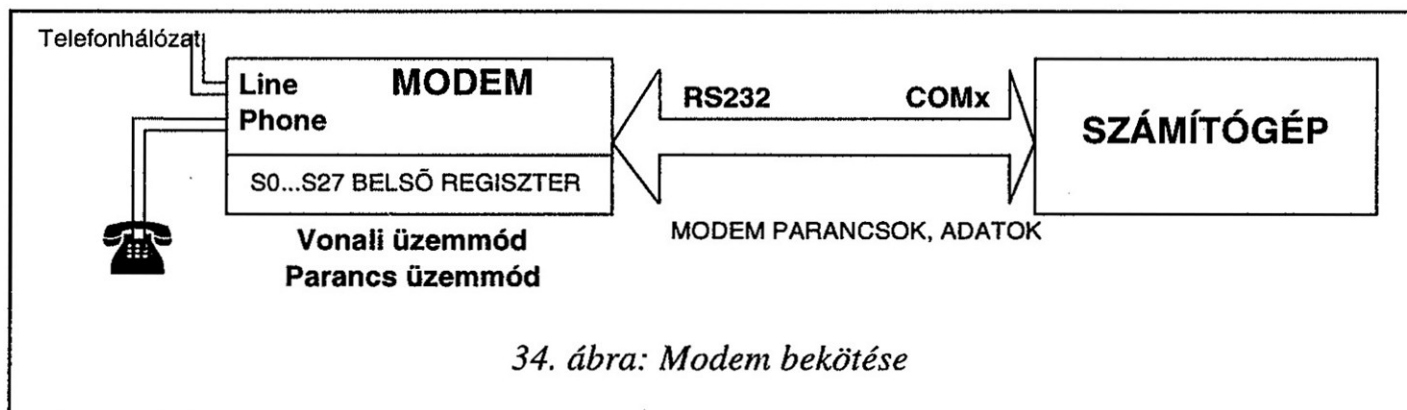
Két modem csak akkor tud kapcsolatba lépni, ha van közös modulációs protokolljuk. Ha több is van, akkor a leggyorsabb közt választják ki (kezdeti "összefütyülés" alatt, ill. később is).



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 (vagy XOFF) helyez az adatfolyamba az adatátvitel vezérlésére az adatvesztés megelőzése céljából.

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 vezetékkel igényel.

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 számítógép a szöveges formájú parancsokat soros vonalon keresztül adja ki a modemnek, a modem parancs üzemmódjában értelmezi azokat, és szintén szöveges, általában "OK" üzenettel válaszolva fogadja el, és esetleg egy eredménykódot is visszaküld.

2. FIZIKAI ÁTVITELI JELLEMZŐK ÉS MÓDSZEREK

Minden parancs az AT karaktersorzattal kezdődik, és ezt követi (betűköz nélkül!!!) a parancs további része. Csupán az AT utána Enter begépelésére a modem OK üzenettel jelzi a kapcsolat meglétét.

A modem és a számítógép egymással soros vonalon kommunikál szöveges parancsok segítségével.

A legfontosabb parancsok ismertetése egy sorban:

A parancsok csak azért szövegesek, hogy mi emberek könnyen tudjuk olvasni.

MODEM PARANCSONOK

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
;	Tárcsázási parancs végén a modemet parancs üzemmódban 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.

Az RS-232C szabványról később részletesen írunk. Valószínűleg az ott leírtak alapján lesz ez a rész teljesen érthető.

MODEM REGISZTEREK

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 törölhető memóriában (NVRAM) tárolhatók és a későbbiekben újra bekapcsoláskor ezek jelentik az alapbeállítást. A regisztertáblázat:

Regiszter	Érték	Gyári érték	Feladat
S0*	0-255	0	Csengetésszám, hányadik csengetés után válaszol automatikusan
S1	0-255	0	Csengetésszámláló, ha csengetés jön tartalma 1-el nő
S2	0-127	43	Kilépési karakter, utána adat üzemmódból helyi üzemmód
S3	0-127	13	Kocsi vissza karakter, ez van minden parancssor végén
S4	0-127	10	Soremelés karakt. ez van minden parancssor végén a kocsi vissza után
S5	0-32,127	8	Backspace karakter
S6	2-255	2 (sec)	Tárcsahang kivárási vonalra lépés után tárcsázás előtt. Mo.-n 30-50!
S7	1-255	30 (sec)	Vivőre várakozás, utána bont, NO CARRIER üzenet.
S8	0-255	2 (sec)	a ", " parancs szünetideje
S9	0-255	6 (0.1 sec)	CD válaszüzenet a vivőérzékelés válaszüzenete
S10	1-255	14 (0.1 sec)	Vivőhiány, Ha nincs vivő ennyi ideig, a modem bontja a vonalat.
S11	50-255	85	Hangtárcsa sebesség csak DTMF esetén
S12	20-255	50	Kilépési késleltetés
S13			Nem használt
S14*	Bitminta		Üzemmód regiszter. echo, tárcsázási mód, válasz v. kezdeményező mód
S15			Nem használt
S16	Bitminta		Modem teszt módok
S17			Nem használt
S18*	0-255		Tesztidőzítő, a diagnosztikai teszt hossza
S19			Nem használt
S20			Nem használt
S21*	Bitminta		Üzemmód regiszter.csatlakozó típus, DTR, DCD, DSR jelek hatása
S22*	Bitminta		Üzemmód regiszter. Hangszóró hangereje, vezérlése
S23*	Bitminta		Üzemmód regiszter. Sebesség, paritás
S24			Nem használt
S25*	0-255	5(0.01sec)	DTR késleltetés
S26*	0-255	1(0.01sec)	RTS-CTS késleltetés
S27*	Bitminta		Üzemmód regiszter. Üzemmód (szinkron, aszinkron), adatátviteli szabvány

A táblázatban *-al jelölt regiszterek tartalma az NVRAM-ban eltárolható. A regiszterek tartalmának módosítása és kiolvasása két modemvezérlő paranccsal lehetséges: (n = 0...27 és X = 0...255)

Módosítás: ATSn=X
Kiolvasás: ATSn? kírja az Sn regiszter értékét decimálisan

A modemek által használt vonalak 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 – ld. lent
CCITT V.42 (LAP-M, Link Access Procedure for Modems),

A hibajavítás lehetősége még rossz minőségű vonalak esetén is megteremti a megfelelő adatátviteli kapcsolat lehetőségét.

SZABVÁNYOS ADATTÖMÖRÍTŐ PROTOKOLLOK:

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

MNP (MICROCOM NETWORKING PROTOCOL)

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. Fokozatai:

MNP1 Aszinkron, bájt-orientált kapcsolatot valósít meg, fél duplex (half duplex) eljárással, ma már nem alkalmazzák. Egy 2400 bit/s sebességű modem ezzel az eljárással 1690 bit/s sebességet tud elérni.

MNP2 Aszinkron teljes (full) duplex átvitelt megvalósító eljárás. A Z80 és Intel 6800 típusú processzorokra dolgozták ki. Nem lassítja az átvitelt, zavart vonalakon az MNP2 egy 2400 bit/s-os modemen valóban eléri ezt a sebességet.

MNP3 Az MNP3 szinkron teljes duplex adatcserét valósít meg. 10 bites adatcsomagokat használ: 1 start-, 8 adat- 1 stopbit. Szinkron átvitelnél nincs start- és stopbit, ami gyorsítja az átvitelt. Az MNP3 már némi tömörítést is eredményez, tehát a modem fizikai

Hibajavító protokollal ellátott modemek hibamentes adatátvitel biztosítanak. A vonali hibák változatlanul jelen vannak, de a két modem - az adatok újraküldésével – kiszűri azokat.

Az adattömörítés előfeltétele, hogy működjön hibajavító protokoll. A modem az adatátvitel során folyamatosan végzi a tömörítést/kibontást.

Az amerikai hadsereg megbízására a kifejlesztője a MICROCOM cég.

A szinkron és aszinkron átvitelről a későbbiekben részletesen írunk.

sebességénél látszólag gyorsabb az adatátvitel: egy 2400 bit/s-os modem látszólagos sebessége 2600 bit/s lesz.

MNP4 Az MNP4-nél megjelent két új optimalizálási eljárás, amit Adaptive Packet Assembly(tm) és Data Phase Optimization(tm) neveken jegyeztek be. Ezek valamiféle csomag jelleget adtak az átvitelnek. Az egyes adatblokkok átvitele úgynevezett adatkeretekben, azaz csomagokban történik, és a keret tartalmazza a szükséges ellenőrző biteket. Szintén kerettel szinkronizálnak és nyugtáznak e rendszerben. Emellett bizonyos adattömörítés is végbemegy, így MNP4 alatt egy 2400 bit/s-os modem 2900 bit/s sebességet tud elérni, ami 20% nyereség.

MNP5 Az MNP5 tovább tökéletesítette az adattömörítést. A valós idejű tömörítés nagy hibája: nem ismeri fel azt, ha az alapinformáció eleve tömörített. Ilyenkor a különböző algoritmusokkal kísérletezve erősen lelassul.

MNP6 Sajnos nem kompatibilis számos MNP hibakorrekcióval dolgozó modemmel. Az MNP6 félduplex kommunikációt valósít meg, de teljes duplex szolgáltatásokat kapunk tőle. Ezt a Statistical Duplexing nevű eljárással érik el, amely az ellentétes irányú jelfolyamat az egyes keretek között, az adatáramlás szünetében továbbítja.

MNP7 Az MNP7 technológiánál az Enhanced Data Compression eljárást kombinálják az MNP4 szabványos kódolási eljárásával, aminek eredménye a szokásos fájlok továbbításának mintegy 300%-os felgyorsulása.

MNP8 Kimaradt a fejlesztésből.

MNP9 Az MNP9 esetében az Enhanced Data Compression eljárást kombinálták a V.32 szerinti kommunikációval, így egy ilyen modem 300%-kal gyorsabb, mint az eredeti CCITT V.32 szerinti modem.

MNP10 Használatának célja a korábbi eredmények felhasználásával a tömörítési eljárás intelligenssé tétele.

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 átvittel
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

A szokásos fájlok esetén egy MNP5-tel működő 2400 bit/s-os modem látszólag 4800 bit/s sebességgel kommunikál.

Ezek protokollok többet biztosítanak, mint a modem saját hibajavító protokollja, mert ez az átvitel út mindhárom részére (a számítógépes végpontokat is beleértve) kiterjedő védelmet ad, viszont csak a fájlok átvitele során.

MODEMEK FEJLŐDÉSE

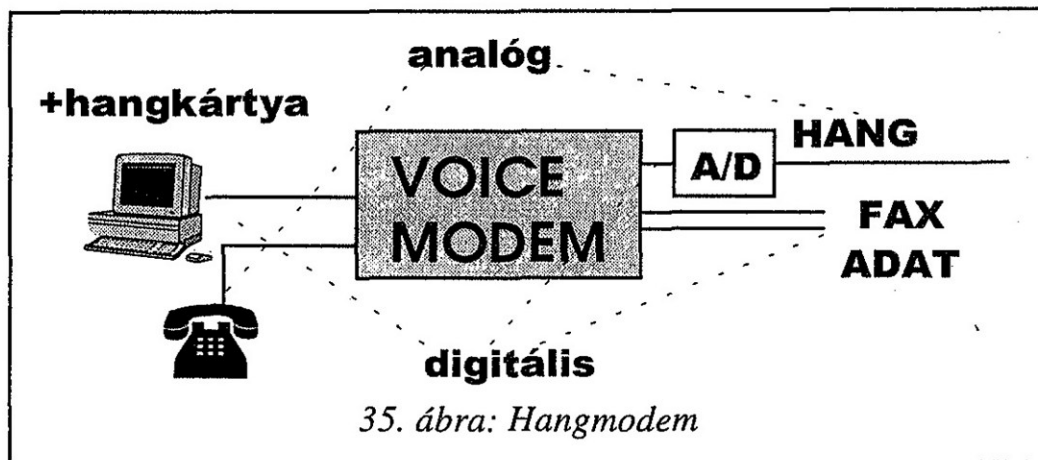
A modemek fejlődése az előbbieken kifejtett okok miatt töretlen. Folyamatosan jelennek meg az egyre nagyobb sebességű modemek (a könyv írása idején 56 kbit/s a sebesség), és a modemek segítségével újabb szolgáltatásokat is megvalósítanak.

Mit csinál a modem?

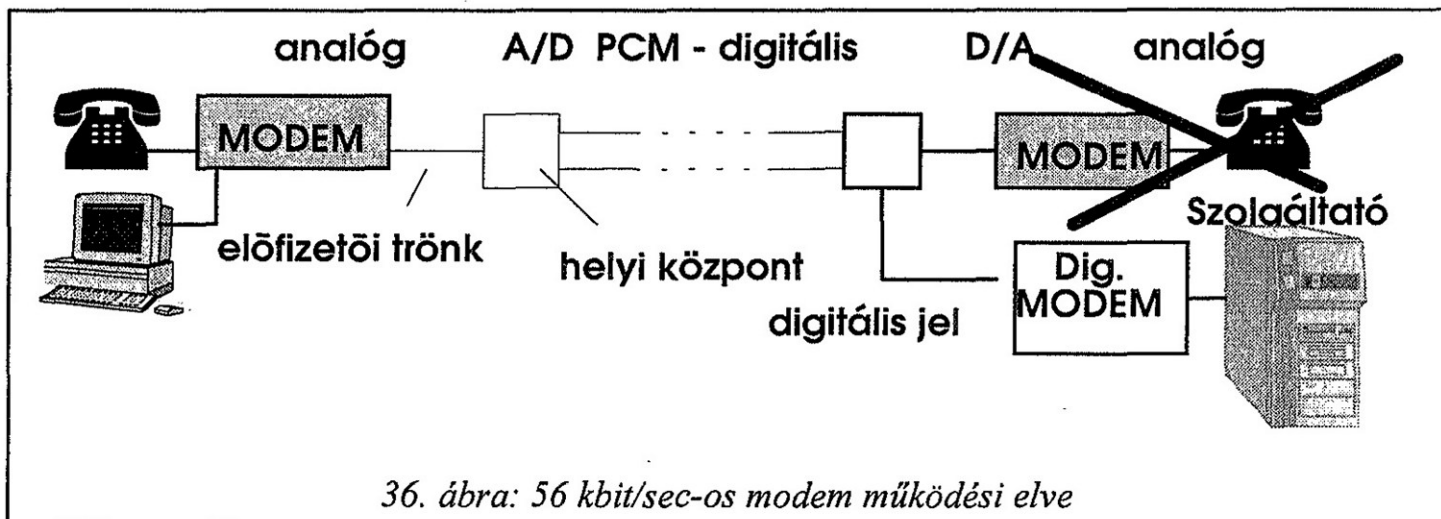
Digitális információt a telefonvonalon történő átvitel miatt analóggá alakít, majd a másik oldalon vissza.

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

Segítségével hangposta, üzenetrögzítő szolgáltatások valósíthatók meg.



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



SZÁMÍTÓGÉP - HÁLÓZATOK

A legkorszerűbb modemek már képesek 56 kbit/sec sebességű adatfolyam küldésére is. A sebesség növekedést a jel-zajt okozó analóg-digitál-analóg átalakítás egyikének az elhagyásával érték el.

A hagyományos modemeknél a számítógép digitális jelét a modem analóggá alakítva viszi át az előfizetői vonalon a központba. Mivel a legtöbb központ digitális, üvegszálalás átvitelt használ a központközi kapcsolataiban, ezért ezt az analóg fázis- és amplitúdó- 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 ún. digitális modemet alkalmazunk, amely képes a digitális PCM jelből előállítani az adatot.

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 amplitúdó-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	Adattömörítő szabvány, max. 4-szeres tömörítést lehetővé. Csak a V42 szabványú hibajavító protokollt alkalmazó MODEM-ekkel használható. Az előzetesen (pl. PKZIP-el) végrehajtott tömörítéstől függetlenül működik.
V90, X90	Ez jelöli az 56 kbit/sec-os adatátviteli sebességet biztosító szabványt

A 37. ábrán látható felsorolás tartalmazza mindazokat a jellemzőket, amelyek egy modem vizsgálatakor, vagy vásárlásakor figyelembe kell venni.

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 megjegyzés:

a leírtakból nyilvánvaló, hogy két 56 kbit/sec-os modem egymással nem tud ilyen sebességen kommunikálni,

egy ilyen modem sebessége csak akkor használható ki, ha a fogadó oldalon digitális modem van.



MODEMTULAJDONSÁGOK

- 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

37. ábra Modemtulajdonságok összefoglalása

DIGITÁLIS ÁTVITEL

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 trónkőkön folyamatosan a digitális átvitelre tértek át (azaz folyamatos jelek helyett 0-kból és 1-ekből álló sorozatok haladnak a vonalakon).

A digitális 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 digitális átvitel előnyei:
kicsi hibaarány: csak két állapotot kell megkülönböztetni.

Digitalizálva tetszőleges analóg jel (hang, kép, stb) átvihető.

Az átviteli sebességben még sok tartalék van.

A jel regenerálásához egy összehanlításra alapuló döntést kell hozni.

Ahogy a digitális számítógépek és integrált áramkörök ára tovább esik, úgy válik a digitális átvitel és a hozzá kapcsolódó kapcsolástechnika az analóg átvitelnél olcsóbbá.

A digitális átvitel egy másik előnye az, hogy egyetlen eszköz hatékonyabb kihasználásával, különböző típusú adatok (hang, zene, normál adat, kép pl. televíziós kép vagy videotelefon stb.) kevert átvitelét teszi lehetővé.

Ez természetesen a különféle típusú adatok bináris alakra kódolása segítségével valósul meg. További előny az, hogy a már meglevő vonalakon is nagyobb átviteli sebesség érhető el.

Az átvitel során mindig biteket viszünk át, de mivel eleinte szövegátvitelt valósítottak meg, ezért az átvitt információ egysége a bitsoport volt, amely a szöveg egy karakterét kódolta. Az ilyen, bitsoportokat átvivő módszert szokták **karakterorientált átviteli eljárás**-nak nevezni.

Az átvitt információ egysége a karakter, és speciális ún. vezérlő karakterek biztosítják az átvitel megfelelő megvalósítását. A hálózati szabványokban, leírásokban a bájtt kifejezés helyett az oktet (octet) fogalmát használják, ami egy 8 bites csoportot jelöl.

A hálózatok elterjedésével a szöveges jellegű információk mellett más jellegű információk átvitele is szükségessé vált, sokszor eltérő szóhosszúságú és adatábrázolású számítógépek között. Ezért a bitsoportos átvitel helyett a tetszőleges bitszámú üzenetátvitel került előtérbe, ezek a **bitorientált eljárás**-ok.

Az átvitel során, mivel az a legtöbbször sorban, bitenként történik, valahogy biztosítani kell az adó és a vevő szinkronizmusát, azaz azt, hogy pl. a ötödiknek elküldött bitet a vevő szintén az érkező ötödik bitnek érzékelje.

A **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 bitsoport (SYN vagy más terminológiában FLAG)) érzékelése biztosítja. A vevő ezt érzékelve, már helyesen tudja az ezt követő biteket vagy bitsoportokat (karaktereket) értelmezni.

Az átvitel során mindig biteket viszünk át!

Hogyan tudjuk biztosítani az adó és a vevőoldal szinkronizmusát?

(szinkronizmus: az adó 1-et küld, a vevő is annak érzékeli)

Karakterorientált átviteli eljárásnál az átvitt információ egysége a karakter



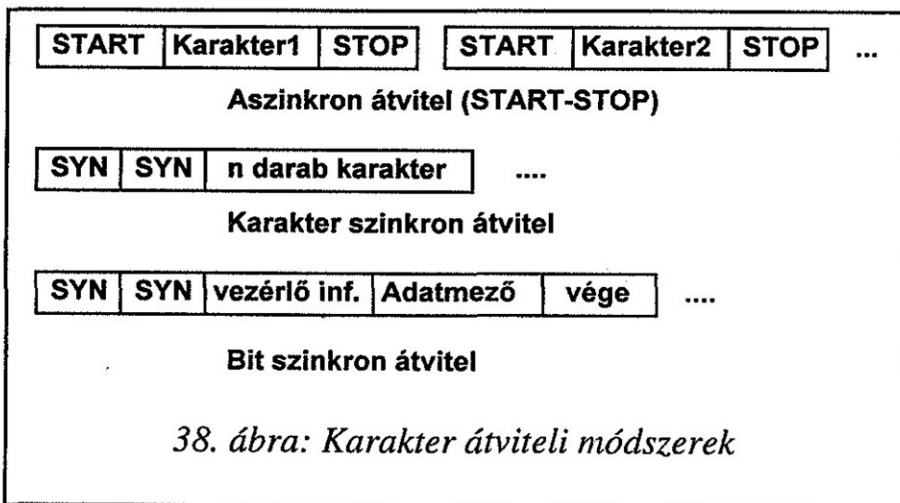
Bitorientált eljárásnál tetszőleges bitszámú az üzenetátvitel

Az átvendő bitfolyamba szinkronizációs pontokat kell beiktatni !

A SYN bitsoport (karakter):

01111110

hat darab 1-es van egymás után, a többinél nincs ennyi egymás után!



Az **aszinkron karakterorientált eljárások** legrégebbi módszere a START-STOP átvitel. Ennél a szinkronizmus az adó és a vevő között csak egy-egy karakter átvitelének idejére korlátozódik. A leírtakat a **38. ábrán** foglaltuk össze.

DIGITÁLIS JELEK KÓDOLÁSA

A fizikai vonalon való átvitelnél a bitek ábrázolására több lehetőség is van, amely közül a legegyszerűbb az, mikor minden bitet, értékétől függően két feszültség szinttel 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:

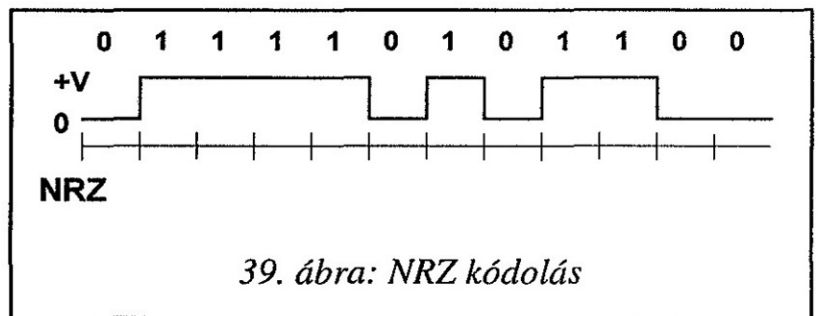
- Ha a használt kódolás kis sávszélességű (kevés váltást tartalmaz), akkor felhasználásával több információ is átvihető egy adott kommunikációs csatornán.
- Kicsi legyen a jelek egyenfeszültség összetevője, mivel a magas DC szintű jelek jobban gyengülnek, így az átviteli távolság csökken.
- Legyen elég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

Röviden tekintsük át a létező megoldásokat:

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.

Ha egy bit 1-es, akkor a feszültség teljes bit idő alatt H szintű, ha 0-ás, akkor 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 0Hz-től (ha csak csupa 1-est vagy csupa 0-át tartalmaz a sorozat) az adatátviteli sebesség feléig (ha sorozat: 10101010...). Polarizált jel.



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

Egy vonalon számos lehetőség van két megkülönböztethető állapot kialakítására!

Szokásos az „1” állapotot MARK-nak, a 0-át SPACE-nek is nevezni.

Ha a jel bitenkénti szinkronizáló információt hordoz, akkor az a sávszélességet növeli (bitenkénti állapotváltás!)



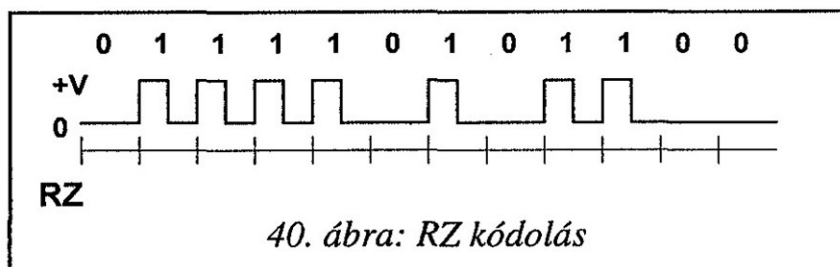
A legegyszerűbb megvalósítani ezt a jelet.

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).

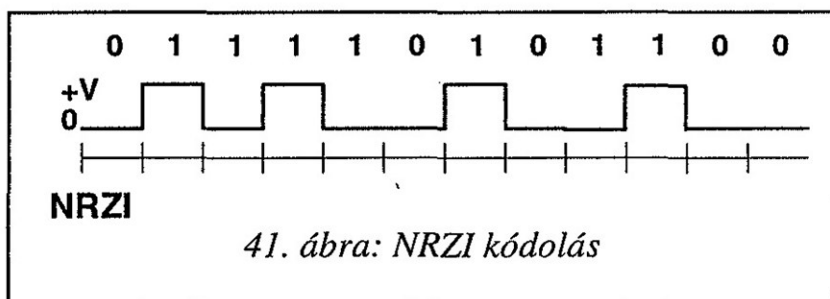
A legrosszabb a sávszélesség igénye: az maga az adatátviteli

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 azt a megoldást választják, hogy az adó pl. minden öt egymást követő nulla után egy 1 értékű bitet szúr be, amit a vevő automatikusan eltávolít a bitfolyamból.



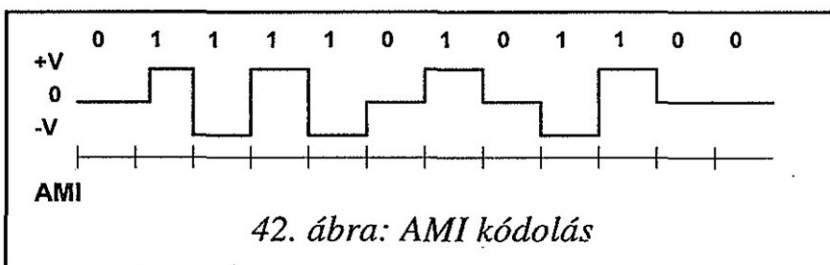
NRZI — Non Return to Zero Invertive: Nullára nem visszatérő, "megszakadásos".

A 0 bitnek nulla szint felel meg. Az 1 értékű bithez vagy nulla vagy +V szint tartozik a következő szabály szerint: ha az előző 1-eshez nulla szint tartozott, akkor +V lesz, ha az előző 1-eshez +V tartozott, akkor 0 szint lesz a bithez rendelt feszültség. 0 bitet követő 1 értékű bit mindig +V feszültségű. Ez a módszer az NRZ kisebb sávszélességét kombinálja a szinkronizálást biztosító kötelező jelváltásokkal. Sok nulla esetén itt is használható a bitbeszúrás.



AMI — Alternate Mark Inversion — váltakozó 1 invertálás

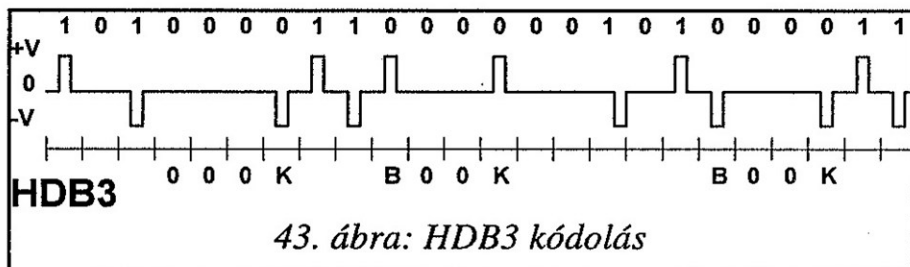
A módszer nagyon hasonló az RZ módszerhez, de nullára szimmetrikus tápfeszültséget használ, így az egyenfeszültségű összetevője nulla. Minden 1-es-hez rendelt polaritás az előző 1-eshez rendelt ellentettje, a nulla szint jelöli a 0-át. Természetesen hosszú 0-s sorozatok esetén a szinkronizáció itt is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



HDB3 — High Density Bipolar 3 — Nagy sűrűségű bipoláris 3

A módszer majdnem az AMI -val azonos, de a kódolásba beépítették a hosszú nulla sorozatok kezelését.

Mikor 4 egymás utáni „0” bit következnek, 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

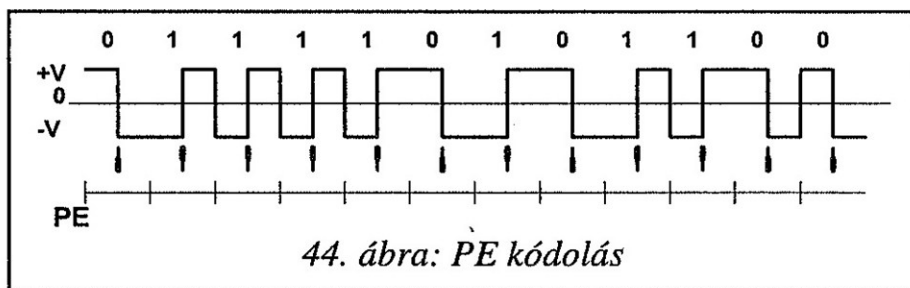


43. ábra: HDB3 kódolás

egyenfeszültségű összetevője keletkezne. 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.

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.



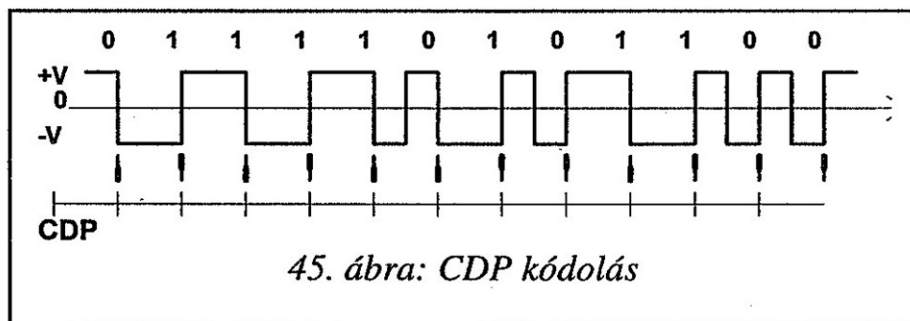
44. ábra: PE kódolás

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 következhesen. A jel detektálásakor, visszaállításakor, az alapfrekvenciás, bit értékeket hordozó átmeneteket el kell különíteni a kétszeres frekvenciájú "hamis" átmenetektől (a mai technikában ez nem okoz nehézséget).

A sok előnyös tulajdonsága mellett az egyetlen hátránya a gyakori jelváltások miatti nagy sávszélessége, mivel az információt ennél a formánál is a jelátmenetek hordozzák. Kiválóan alkalmas mágneses adatrögzítéshez is. Mivel minden bitnél van jelváltás, ezért a szinkronizálás nem okoz problémát.

CDP — Conditional Diphas — Feltételes kétfázisú jel.

A módszer az NRZI és a PE módszerek kombinációja: „0” bitet az előző bithez tartozó jelváltás azonos irányja, míg az „1” bitet az előző bithez tartozó jelváltás ellentétes irányja jelzi. Ez a módszer nem érzékeny a jel-polaritásra. A sávszélesség igény, a kábelezési rendszer, az adatátviteli sebesség mind meghatározza a kódolási rendszer kiválasztását.



45. ábra: CDP kódolás

Több módszert mutattunk be digitális jelek kódolására: a legegyszerűbb NRZ módszertől — amit az RS232 alapú protokolloknál is használnak — a PE módszeren keresztül — amit az Ethernet hálózati protokoll alkalmaz — eljutottunk a viszonylag komplikált HDB3 kódolásig, amit távbeszélő rendszerekben alkalmaznak.

KARAKTEREK ÁBRÁZOLÁSA — AZ ASCII KÓDRENDSZER

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 ezeket 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 egyeduralgoddá.

Az ASCII karakterkészlet 128 héttites, 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.

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 ezekbe a kategóriákba.



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

Az ASCII kódoknak két fő csoportja van:
grafikus karakterek
vezérlő karakterek, ezen belül:
információcsere vezérlők,
formátumot befolyásolók
információ elkülönítők.

Információcsere vezérlő karakterre példa a 04H kódú EOT karakter, amit annak a jelzésére használnak, hogy a karakterek átvitele befejeződött és ez a kód jelöli, 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 nyomtatónál a kocszi-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.

Ahogy ezt már az előbb is említettük, az ASCII vezérlőkódokat információátvitelnél használják. Az adatátvitel során az ADÓ valamilyen összeköttetésben van a VEVŐ-vel, és a kódolt információt átadja. Ez az információátvitel általában ASCII kódú karakterek segítségével történik és a vezérlőkarakterek szolgálnak az adatátvitel tényleges végrehajtásának a megvalósítására.

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) hajtodik 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 A szokásos. használata a DC1 és DC3 karaktereknek a különböző átviteli sebességű adók és vevők közötti adatátvitel vezérlése az ún. XON/XOFF protokoll szerint.

SYN (Synchronous Idle): átvitelvezérlő karaktert a soros szinkron adatátviteli rendszerekben használják.

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.

Az ENQ, ACK, és NAK általában a tényleges átviteli protokoll kialakítására szolgál.

Megállapodás esetén itt is lehetséges az első karakterpozícióra lépés (a gyakorlatban egyszerűen a következő lap elejére lép).

Szokásos megoldás nyomtatóknál, a nyomtató paramétereinek beállításához, felhasználó által definiált karakternek a nyomtató elektronikába való betöltéséhez, az ún. "escape szekvencia" használata.

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 konkrétan melyik vezérlőkaraktereket rendelték hozzá az adott funkcióhoz.

A PC-k megjelenésekor az IBM által hozzáadott 1 bites kiterjesztéssel újabb 128 karakter használatát szabványosította, amely kódrendszer Latin1 néven ismert.

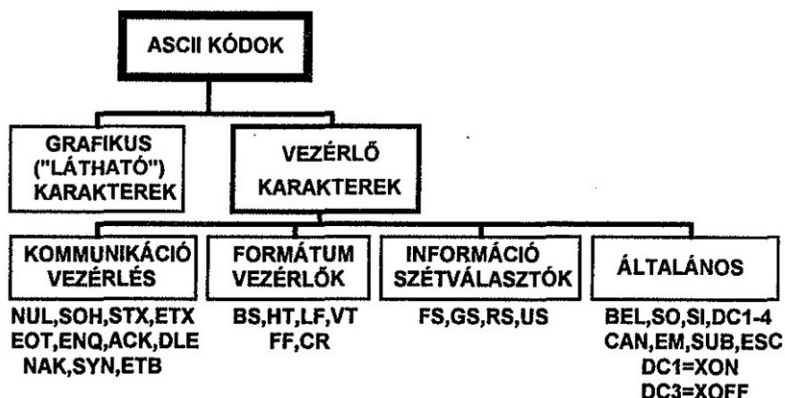
Ez tartalmazza számos európai nyelv — pl. francia, német spanyol, stb. — speciális nemzeti karaktereit, valamint a görög ABC betűit, táblázat-rajzoló karaktereket is.

SZÁMÍTÓGÉP - HÁLÓZATOK

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	á
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	Š	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

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 Acknowledge	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

ASCII vezérlőkarakterek



46. ábra: ASCII kódok osztályozása

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.

A UNICODE az MS OFFICE97-ben már megjelent !

Az ASCII kódrendszert 1977-ben az Amerikai Szabványügyi Hivatal megerősítése, és jóváhagyása után a Nemzetközi Szabványügyi Hivatal (ISO) is átvette és ISO646 néven regisztrálta.

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 beletartoznak 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.

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.

Milyen hatással lesz ez a szabvány a számítástechnikára? Remélhetőleg ilyen szabványokkal könnyebben lesznek gépek, és programjaik eladhatók. Ha a szabványt elfogadják, csak egy Unicode karakterkészlettel ellátott verziót kell gyártani.

A rendszerszoftverek íróinak a kétbájtos kódkészlet miatt újra kell írniuk a programjaikat, de ez részben automatizálható, és nem érinti a teljes programot.

PÁRHUZAMOS ÉS SOROS ADATÁTVITEL

Az előbbieken leírtak jól illusztrálják, hogy az információt általában bitsoportos alakok hordozzák. Ha egy bitsoportot egyszerre tudunk átvinni, akkor az információ átviteli sebesség nagyobb lesz, mintha a biteket sorban, egymás után vinnénk át. Ehhez azonban annyi, biteket átvivő adatutatót kell az ADÓ és a VEVŐ között kialakítani, ahány bitből áll a bitsoport.

Természetesen külön vezeték(ek) szükségesek a ADÓ-VEVŐ szinkronizmus megvalósítására is. Ügyelni kell a vezetékek helyes sorrendjére is.

Mivel ez a kialakítás jelentősen növeli az összeköttetés költségét és csökkenti a megbízhatóságot, ezért általában ezt az ún. párhuzamos átvitelt, csak kis távolságokra, illetve készülékek belsejében elhelyezkedő részegységek összekapcsolására használják. Ilyen megoldással működnek a számítógépek adat-, vezérlő- és címbuszai, vagy perifériák esetén a nyomtató, szkennerek.

Soros átvitel esetén az információs biteket egyenként, sorban egymás után visszük át. Ezért egy kódolt bitsoport átviteli ideje a párhuzamos átvitelhez képest megnő, de számos előnyt rejt ez a kialakítás: szélső esetben elegendő egy vezetékpár az összeköttetés fizikai megvalósításához, ami jelentős költségcsökkentő tényező. Az információ átvitel sebessége lassabb, de ha növeljük az adatátvitel sebességét (napjainkban folyamatosan ez történik) akkor ez a lassúság nem igazán korlátozó tényező.

A hardver oldaláról nyilvánvaló, hogy a hosszabb kód nagyobb RAM és merevlemez-tároló területet igényel. Egyes becslések szerint egy futó program a RAM 10-20%-át használja fel karaktertárolásra.

Ez nem korlátozó tényező, ha figyelembe vesszük, hogy terjednek a 16 Mbájtos RAM-al és 1-2 Gbájtos merevlemez egységgel ellátott PC-k. Az igaz, hogy egy tiszta szövegfájl mérete megduplázódik, de például egy szöveget-grafikát tartalmazó Word fájlnál ez a növekedés csupán 20-25%.

A párhuzamos átvitel a gyorsabb, de több vezetékkel igényel.

ASZINKRON SOROS ADATÁTVITEL

A nagyfokú és széleskörű elterjedése miatt egy számítógép és egy modem, vagy terminál közötti illesztés fizikai rétegének megvalósítása nagyon fontos. Ez teljes duplex, pont-pont típusú összeköttetés kialakítását igényli. Részletesen meg kell határozni a mechanikai-, a villamos-, a funkcionális-, és eljárás interfészeket.



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.

A szabványleírásban a számítógép és a terminál hivatalos neve:

adatvég-berendezés — DTE (Data Terminal Equipment),

a kapcsolódó modemé pedig

adatáramköri-végberendezés—DCE (Data Circuit-Terminating Equipment),

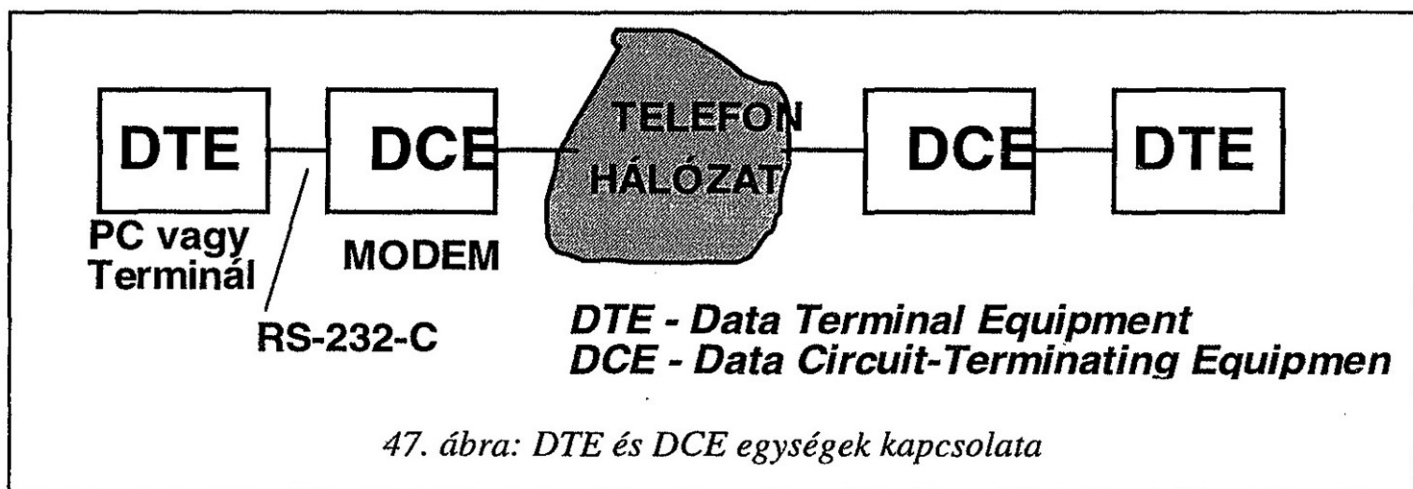
és a köztük zajló kommunikáció az RS-232 soros vonalon folyik.

A legtöbb gyakorlati esetben a DTE egy terminál, vagy egy számítógép, míg a DCE az analóg telefonhálózathoz kapcsolódó modem.

Általánosan fogalmazva egy DCE végzi a kommunikációs közeghez történő fizikai illesztést, azaz a kétállapotú bináris jeleket átalakítja a közegben átvihető fizikai jelekké.

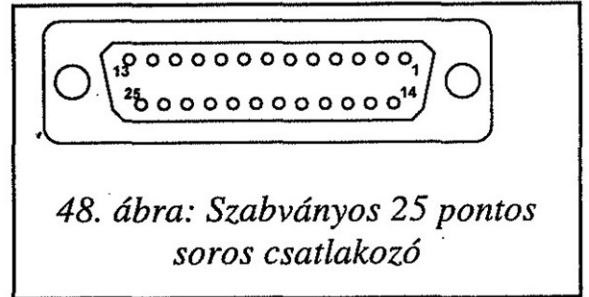
Ezért beszélünk:

- kábel modemről
- ISDN modelmről
- ASDL modemről



RS-232C SZABVÁNY

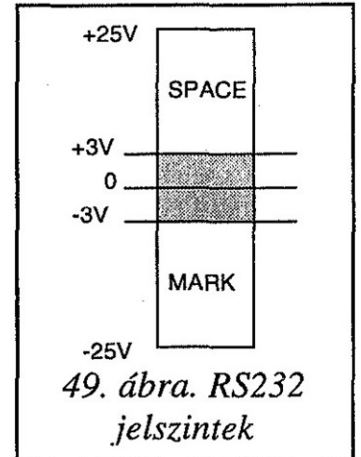
A DTE-DCE egységeket összekötő vezetékrendszer mechanikus csatlakozóját is definiálták: 25 pólusú csatlakozó (szokták DB-25-nek is nevezni). Két, egymásba dugható csatlakozó közül a dugós rész a DTE-n, a hüvelyes részt a DCE-n helyezkedik el.



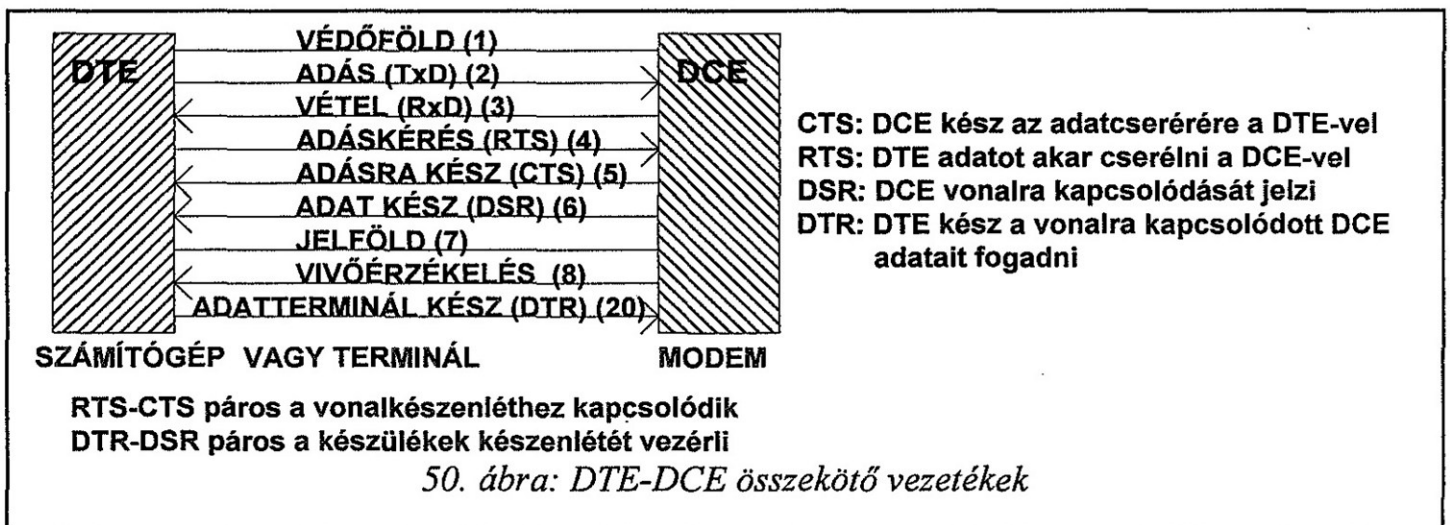
48. ábra: Szabványos 25 pontos soros csatlakozó

A villamos specifikáció szerint a -3V-nál kisebb feszültség a vonalon a bináris 1-et (MARK), míg a +3V-nál nagyobb feszültség bináris 0 -át (SPACE) jelent. A legfeljebb 15 méter hosszú kábeleken 20 kbit/s-os maximális adatátviteli sebesség a megengedett. A legtöbb gyakorlati esetben (pl. a számítógépek soros vonalánál) a feszültség $\pm 12V$.

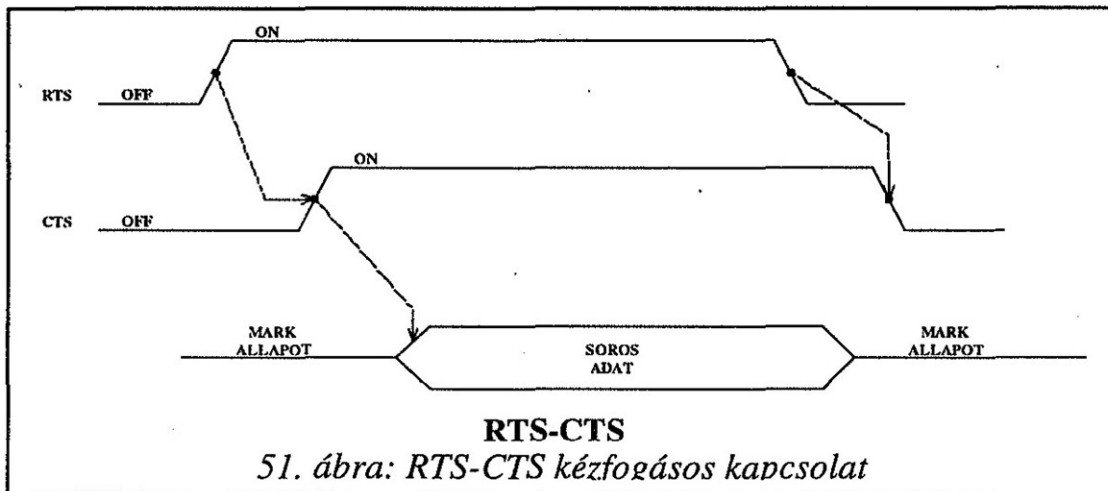
A funkcionális előírás a 25 ponthoz tartozó vonalakat megjelöli, és leírja azok jelentését. A 50. ábrán annak a 9 vonalnak a funkciója látható (a hozzá tartozó kivezetés számmal), amelyeket majdnem mindig megvalósítanak.



- Amikor a számítógépet vagy a terminált bekapcsolják, az aktiválja (MARK-ba állítja) az Adatterminál kész (Data Terminal Ready) jelet (20).
- Amikor a modemet kapcsolják be, akkor a modem az Adat kész jelet (Data Set Ready) (6) aktiválja.
- Ha a modem vivőjelet érzékel a telefonvonalon, akkor a Vivőérzékelés (Carrier Detect) jelet (8) aktiválja.
- Az Adáskérés (Request to Send) (4) jelzi, hogy a terminál adatot akar küldeni.
- Az Adásra kész (Clear to Send) (5) azt jelenti, hogy a modem felkészült az adatok fogadására.
- Az adatok adása az Adás (Transmit) vonalon (2), vétele a Vétel (Receive) vonalon (3) történik.

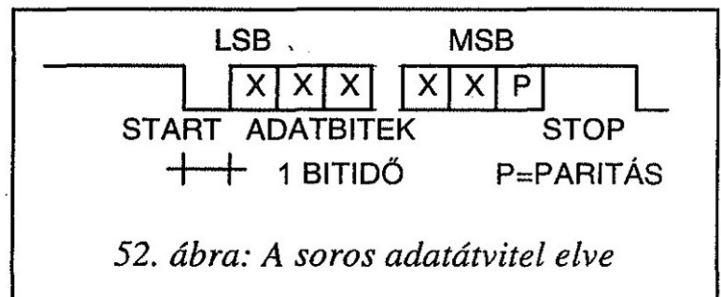


A többi, fel nem tüntetett áramkör a gyakorlatban alig használt funkciókkal rendelkezik: adatátviteli sebesség kiválasztása, modem tesztelése, adatok ütemezése, csengető jelek érzékelése, adatok másodlagos csatornán való fordított irányú küldése.



Az eljárásinterfész az a protokoll, amely az események érvényes sorrendjét határozza meg. A protokoll akció-reakció esemény-párokon alapszik. Amikor egy terminál kiadja 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.

Ahhoz hogy a soros adatátvitel során az ADÓ-ról érkező biteket a VEVŐ egyértelműen azonosítani tudja, szükséges, hogy azonosan értelmezzék a jeleket, azaz pl. egy bájt ötödik bitjét kiküldve, azt a VEVŐ is annak tekintse.



Az aszinkron soros átvitelnél a bitsoportos 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. Ezen biteket 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.

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 (Vigyázzunk! Ez a pozitív feszültség!) (START bit), majd utána történik meg az adatbitek átvitele. Az átvitt adatbitekből álló bitsoport végére az ADÓ STOP bit(ek)-ből álló aktív (MARK) szintű jelet helyez el. A VEVŐ az adás

A START és a STOP biteket szokták keretező biteknek is nevezni.

Itt a MARK negatív, a SPACE pozitív feszültséget jelent!

kezdetéről a vonal MARK-SPACE állapotváltozásából szerez tudomást.

Ezután mindig egy bit átvitelének idejéig várakozva, a VEVŐ az adatbitek veszi. A STOP bitek érkezése után már figyelheti a vonalon ismét megjelenő állapotváltozást, ami a következő bitsoport adásának kezdetét jelöli. Fontos kérdés a vonalon időegység alatt átvitt információ mennyisége, amit bit/s-ben mérünk. Tipikus, szabványosan használt értékeit a táblázat tartalmazza.

bit/s	Egy bit átvitelének ideje (msec)
150	6.6666
300	3.3333
600	1.6666
1200	0.8333
4800	0.2083
9600	0.1042
19200	0.0521
38400	0.0261

Az adatátvitel során az esetleges átviteli hibák felderítését megkísérelhetjük oly módon, hogy az átvendő 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.

Az előzőek alapján a soros adatátviteli protokoll konkrét kialakításánál a következőket 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 bitsoport 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 bitsoport 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.

Statisztikailag igazolható, hogyha kicsi a "hibacsomósodások" valószínűsége akkor a paritásellenőrzés az adatátvitelt sok esetben megfelelően megbízhatóvá teszi. Jó minőségű, zajmentes összeköttetésnél a paritásbitre pedig nincs is szükség.

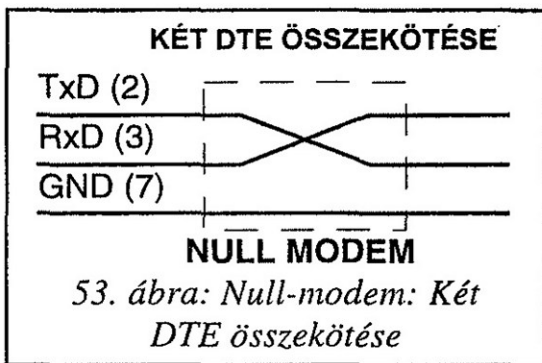


Mivel a soros adatátvitelt széles körben használják, ezért, megvalósítására céláramköröket fejlesztettek ki. Ezeknél az ADÓ oldalán csupán az adatbit-csoportot kell párhuzamosan a bemenetekre adni, az áramkör elvégzi a sorossá alakítást, a paritás, START és STOP bitekkel való kiegészítést, valamint az átvitelt. A vevőoldalon a vett soros adatokból vevőáramkör képezi a bitsoportot. 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 periféria

Magát a párhuzamos adatok sorossá alakítását, illetve a soros adatok visszaalakítását shiftregiszterek felhasználásával hardver úton lehet elvégezni.

áramköröket mindig ki kell egészíteni egy olyan szintátalakító áramkörrel, amely a TTL szintjeiket a szabványos RS232 jelszintekké oda- és visszaalakítja a 0V -> +12V; 5V -> -12V szabályok szerint.

Mivel majdnem minden számítógépnek van soros vonala, gyakran előfordul, hogy két számítógépet RS-232-C soros vonalon keresztül kötnék össze. Mivel nem DTE-DCE típusú az összeköttetés, ezért a megoldás egy null-modem-nek nevezett „eszköz” (hiszen csak egy keresztbe kötés), 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.

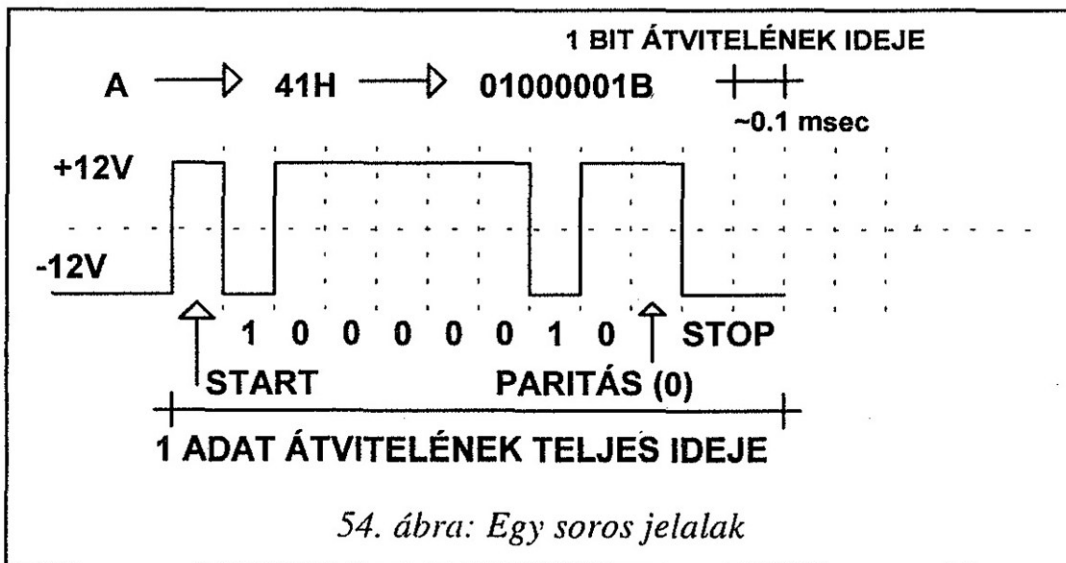
A tényleges adatátviteli sebesség számítására nézzünk egy konkrét példát: Ha például az adatátviteli

sebesség 9600 bit/s, és 8 bites adatokat (bájtokat) viszünk át páros paritásbittel kiegészítve, 2 STOP bittel a végén, akkor például másodpercenként:

$$9600 / (1 \text{ START} + 8 \text{ ADAT} + 1 \text{ PARITÁS} + 2 \text{ STOP bit}) = 9600 / 12 = 800 \text{ adat (bájt)}$$

kerül átvitelre. Ha például az „A” karaktert visszük át (ASCII kódja 41H), akkor a soros vonalon a következő jelformát láthatnánk:

A karakter bitjeit fordított sorrendben visszük át (LSB az első!), a páros paritásbit ennél a karakternél 0 mert eleve páros (kettő) 1 értékű bitet tartalmazott.



A null modemes megoldás lényegében két DTE típusú eszközt köt össze.



Még egy praktikus kérdés: Sokszor nehéz eldönteni, hogy a még össze nem kötött adatvezetékek közül melyik az adószál (TXD) és melyik a vevőszál (RXD). Ha nincs adás a vonalon, akkor az adószál nyugalomban (MARK állapotban, -12V-on) van, a vevőszálon pedig, mivel nincs bekötve nincs feszültség. Tehát ha egy voltmérővel a közös földvezetékhez képest mérünk, akkor az adószálon fogunk -12V-ot mérni.

ÁRAMINTERFÉSZ

Sokszor a kommunikációban részt vevő két oldal nem köthető össze galvanikusan. Az RS232 szabvány korlátjait nagyobb távolságú átvitel esetében a földvezetékeken átfolyó kiegyenlítő áramok okozta földhurkok is jelentik.

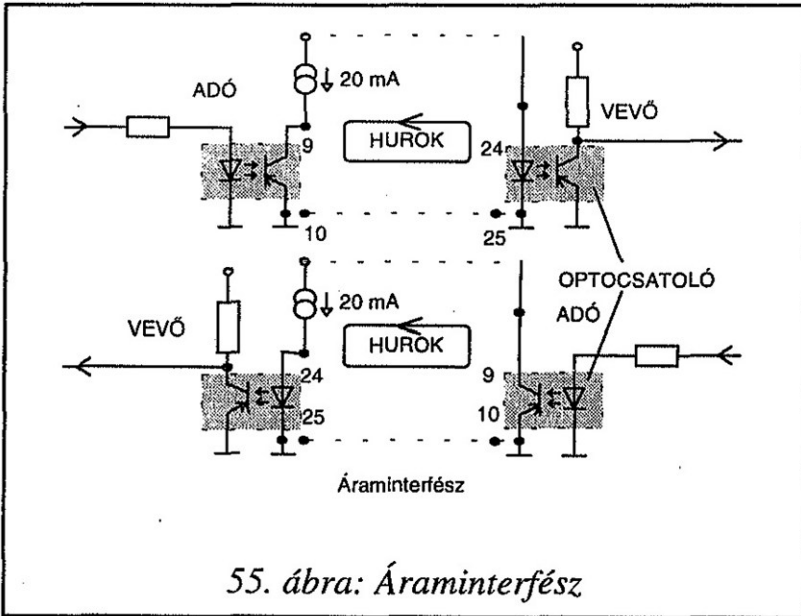
Ez kiküszöbölhető potenciál-leválasztással. Ennek megvalósítására kidolgozott megoldás TTY interfész, közismert nevén a "20mA-es áramhurkok". Lényege: az adó és a vevőoldal mindkét irányban egy hurkot alkotó vezeték-párral van összekötve. Az ADÓ 1 állapotként 20 mA-es áramot küld át hurkon amit a VEVŐ érzékel (áramgenerátorosan tápláljuk a hurkot.). Az információt az áram megléte, illetve hiánya hordozza. (1-van áram, 0-nincs áram).

A maximális sebesség kb. 9600 bit/s, a maximálisan áthidalható távolság 1000 m lehet. Ezen távolságon belüli egyszerű, gazdaságos megoldás, főleg pont-pont összeköttetés esetére.

Vegyük észre, hogy csak a jelet átvivő fizikai réteget cseréltük le!

A feszültségváltozások helyett áramváltozásokat (van –nincs) használunk a két állapot (0,1) kódolására. A logikai jelkezelés (START, ADATBITEK, STOP) változatlan!

Az elektronikában elterjedten használt optikai csatolók nagyon egyszerűvé teszik az alkalmazást, teljes duplex átvitelnél oldalanként két-két ilyen optikai csatoló szükséges.



55. ábra: Áraminterfész

Amint láttuk, az RS-232-C illesztés elsődlegesen számítógépmódem összeköttetésre tervezték, és mivel alkalmazhatósága miatt más területeken, 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 kompatibilis, vagy egy olyat, amely a régivel nem kompatibilis, de a korszerű követelményeket maradéktalanul kielégíti. A kompromisszumos megoldás mindkettőt tükrözi.

A következőkben azzal foglalkozunk, hogy a népszerű RS-232-C illesztést hogyan lehet továbbfejleszteni.

Itt is fontos megjegyezni, hogy csak a jelet átvivő fizikai réteget cseréljük le! A logikai jelkezelés változatlan.

AZ RS-449, -422, -423, ÉS AZ RS-485-ÖS SZABVÁNYOK

Az új, RS-449-nek nevezett szabvány valójában három szabvány egyé ötvözése. A mechanikai, a funkcionális és az eljárási interfész az RS-449 szabványban, míg a villamos interfész két további szabványban van megadva.

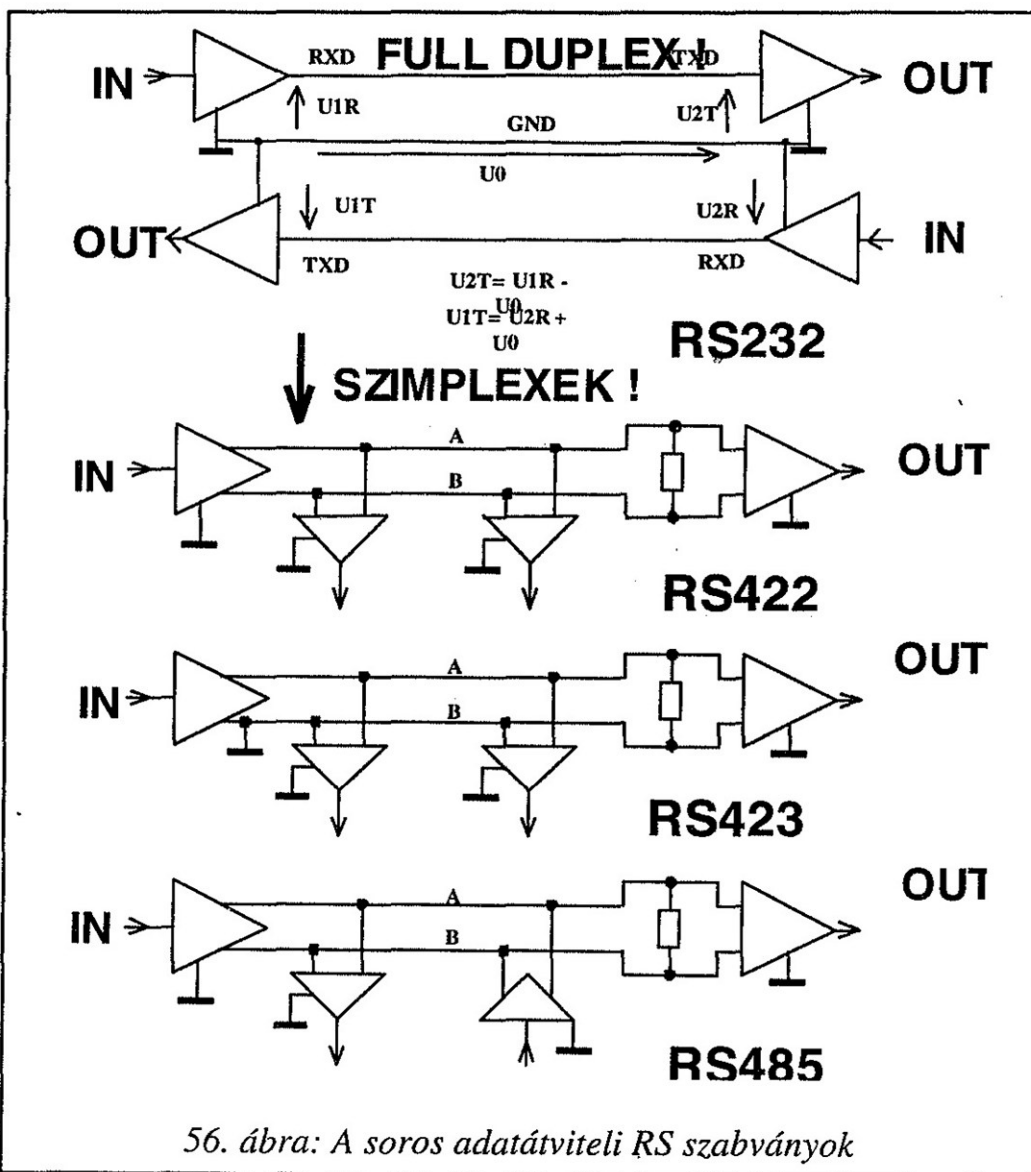
Mindkét villamos szabványnál a jeleket az összekötő vezeték-pár közötti feszültségkülönbség hordozza, és a vevők bemenetén lévő differenciálerősítő fogadja ezeket a jeleket. Mivel a zavart indukáló külső villamos zaj hatása mindkét vezetéken megjelenik, ezért a különbségképzésnél ezek hatása kölcsönösen kioltja egymást.

Vegyük észre, hogy ezen megoldásoknál is csak a jelet átvivő fizikai réteget cseréltük le!

A logikai jelkezelés (START, ADATBITEK, STOP) változatlan!

Miben adtak többet ezek a szabványok?

- nagyobb távolság hidalható át velük,
- az RS-232-C pont pont összeköttetése helyett, több készülék összeköttetése is lehetővé vált.



E kettő közül az egyik az RS-423-A, amely az RS-232-C szabványhoz hasonlít abban, hogy minden áramkörének közös földje van. Ezt a technikát **asszimmetrikus átvitelnek (unbalanced transmission)**

nevezik, mert a jeleket a közös földvezetékhez képest értelmezzük. A másik villamos interfész az RS-422 ellenben a **szimmetrikus átvitelt (balanced transmission)** használja, amelyben minden fő áramkör két, nem közös földű vezetékkel rendelkezik.

Ennek eredményeképpen az RS-422-A egy legfeljebb 60 méter hosszú kábelen 2 Mbit/s-os átviteli sebességet engedélyez, sőt rövidebb távolságokra még ennél nagyobbat is. 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.

Annak elkerülésére, hogy több ADÓ kezdjen a vonalon adni, az adási jogot az egyik kitüntetett eszköznek, az ún. MASTER-nek kell biztosítani. Ez az eszköz címzett parancsok segítségével szólítja meg a többi eszközt, a szolgákat (SLAVE), és szólítja fel őket esetleges adásra.

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 vonal-páron már több ADÓ és több VEVŐ is lehet és közöttük az egy vezeték-páron fél-duplex összeköttetést lehetséges. Természetesen a teljes duplex kommunikációhoz itt is a négyvezetékes kialakítás szükséges.

Összefoglalásul nézzük meg a fent részletezett szabványok leglényegesebb jellemzőit:

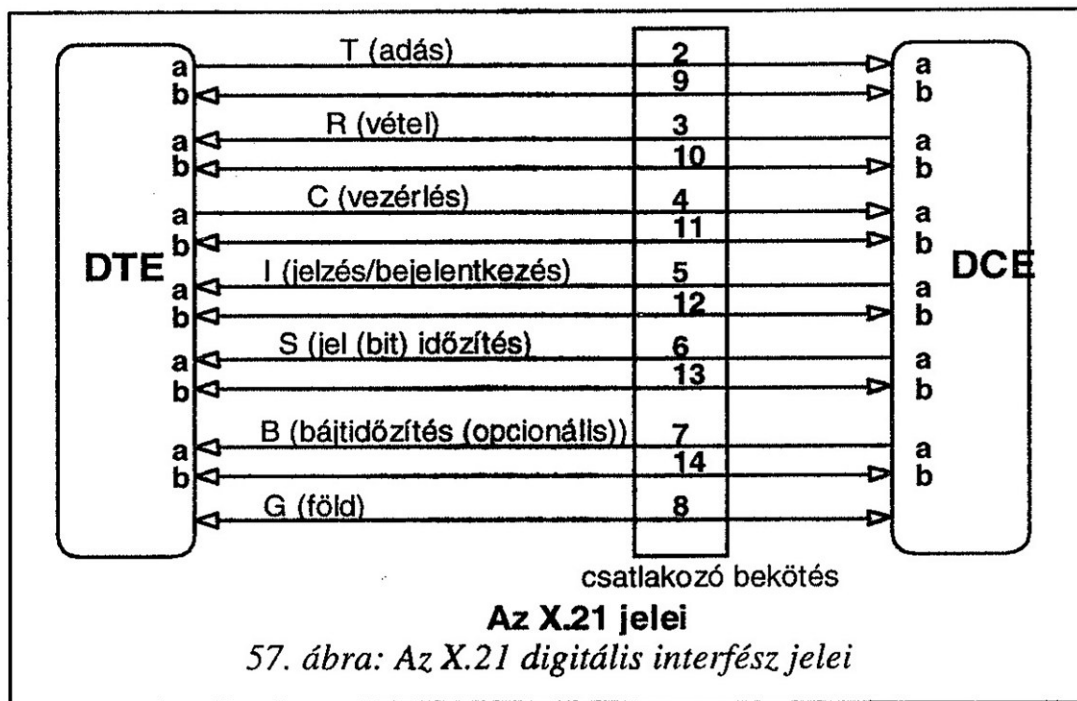
Jellemzők	RS 232C (V.24)	Áraminterfész	RS 423 (V.10)	RS 422 (V.11)
Átvitel	aszimmetrikus	szimmetrikus	aszimmetrikus	szimmetrikus
Kábel típus	sodrott érpár	sodrott érpár	koaxiális	sodrott érpár
Kábelhossz	15 m	300 m	600 m	1200 m
Adatsebesség (max)	20 kbit/s	10 kbit/s	300 kbit/s	2 Mbit/s
Meghajtó kimeneti szint (terheletlen)	+/- 25 V	20 mA	+/- 6V	+/- 6V (diff)
Meghajtó kimeneti szint (terhelt)	+/- 5...+/- 15V	20 mA	+/- 3,6 V	+/- 2V (diff)
Minimális vételi szint	+/- 3V	10 mA	+/- 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 az **57. ábra** tartalmazza.

Lényegében az X.21 az RS-232-C ipari, professzionális változata: adat-átvitel telefonvonalon.



Az alkalmazott csatlakozó 15 pontos, de nincs mindegyik kihasználva. A DTE a T és C vonalakat használja az adat- és vezérlőinformációk cseréjére. (A C vonal a telefonok on-hook/off hook (kagyló helyén/felvéve) jelével analóg.) A DCE az R és az I vonalakat használja az adatok, ill. a vezérlőadatok számára. Az S vonalon a DCE olyan időzítési információkat tartalmazó bitfolyamot küld a DCE-nek, amelyből az egyes bitintervallumok kezdete és vége megállapítható. A szolgáltató választásától függően létezhet egy B vonal is, amely bitek 8-bites keretökké való összefogására használható. Ha ez az opció él, akkor a DTE-nek és a DCE-nek minden karaktert egy kerethatáron kell kezdenie.

ISDN — INTEGRÁLT SZOLGÁLTATÁSÚ DIGITÁLIS HÁLÓZAT

[1] Már többször utaltunk rá, hogy a klasszikus távbeszélő rendszereket analóg hangátviteli célokra tervezték, és nem alkalmasak modern digitális távközlési igények kielégítésére (adat-, fax- vagy video-átvitelre). Az új digitális rendszerek elsődleges célja az, hogy integrálja a hang- és nem hang jelű átviteli szolgáltatásokat. Elnevezésük: ISDN (Integrated Services Digital Network — integrált szolgáltatású digitális hálózat).

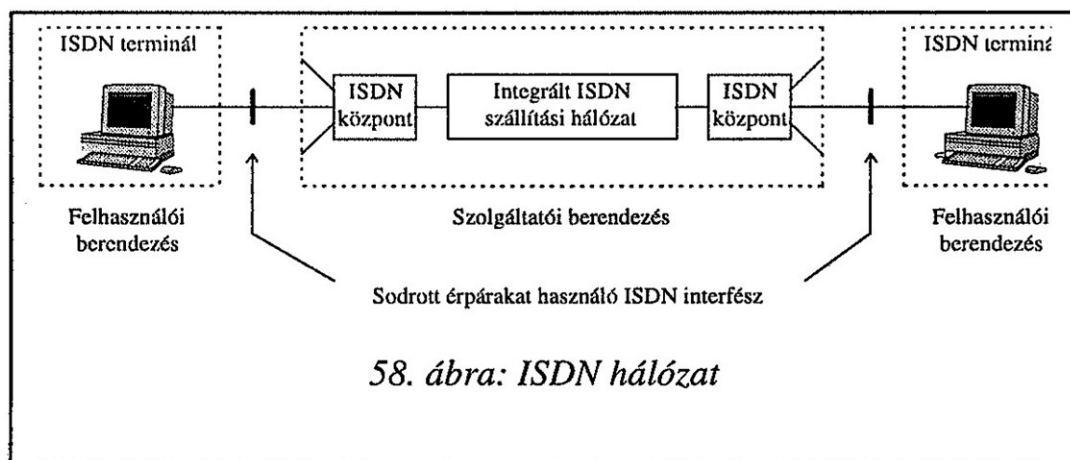
Elv: Ne az analóg működésű hálózaton vigyünk át digitális információt, hanem digitális hálózatokat alakítsunk ki, és itt a digitalizált hang csak az egyik átviendő komponens.

Az ISDN alap gondolata: olyan, digitális jeleket vivő hálózat kialakítása, amely a meglévő, analóg távbeszélő hálózatot hivatott felváltani. A meglévő telefonvonalak és új központok felhasználásával teljesen digitális kapcsolat kialakítása. Egyszerre képes hang, kép, adatátvitel lebonyolítására.

Kezdetben 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 **58. ábrán** bemutatott kétállapotú digitális jelátvitelű integrált hálózat váltotta fel.

Annak a megoldásnak: közös telefonvonalon vinni a hangot, a jelzéseket és modemek segítségével adatátviteli célokra is felhasználni, nyilvánvalóan sok hátránya van.



AZ ISDN SZOLGÁLTATÁSAI

A legalapvetőbb szolgáltatás továbbra is a hangtovábbítás, de számos új tulajdonsággal kiegészítve. Az ISDN telefonokon több azonnali hívásfelépítésre alkalmas gombokat helyezhetünk el, amelyekkel a világ bármelyik telefonját el lehet érni. A telefonok a kicsöngés ideje alatt a hívó telefonszámát, nevét és címét is kijelezhetik.

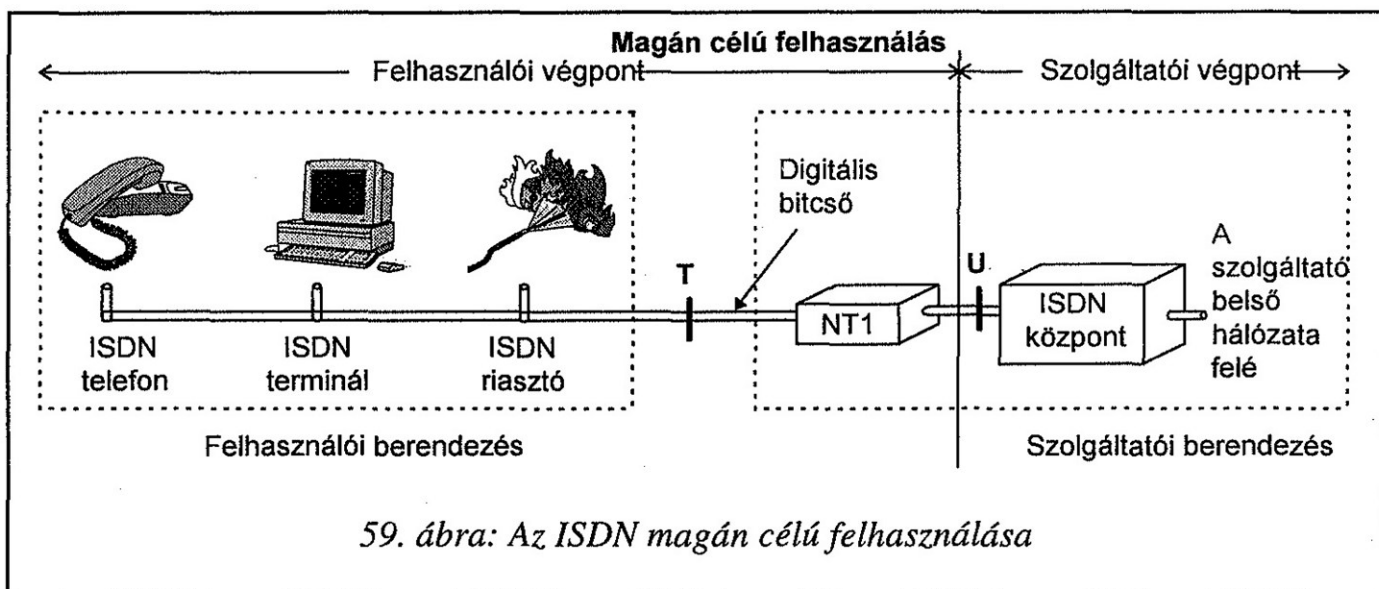
E sajátosság kifinomultabb változata szerint a telefonkészülék egy számítógéphez is hozzákapcsolódik azért, hogy egy bejövő híváskor a hívó adatrekordja képernyőn megjeleníthető legyen. Egy másik fontos adatátviteli sajátosság az, hogy zárt felhasználói csoportok alakíthatók ki, ami magánhálózatok létrehozását teszi lehetővé.

Egy csoport tagjai csak a csoport más tagjait hívhatják, és kívülről sem jöhet be semmiféle hívás (csak szigorúan ellenőrzött módon). Az ISDN várhatóan széles körben elterjedő új szolgálata a videotex, amely egy távoli adatbázis terminálon keresztüli interaktív elérését teszi lehetővé a felhasználó számára. (Áru kiválasztása és megvásárlása telefonon keresztül.)

Egy másik, várhatóan népszerűvé váló ISDN szolgáltatás a teletex, amely valójában házi és üzleti célokra átalakított elektronikus levelezési szolgálat. További korszerű hangátviteli szolgáltatások: világméretű konferencia-hívások lebonyolítása (kettőnél több partner között).

A beszéd-digitalizálási technikák lehetővé teszik a hívó számára azt is, hogy a foglalt jelzés vagy a hosszú idejű kicsöngés után üzenetet (hangposta) hagyjon.

Az ISDN adatátviteli szolgáltatásai által a felhasználók ISDN termináljaikkal, ill. számítógépeikkel a világ bármelyik másik ilyen gépéhez hozzákapcsolódhatnak.



Számos esetben kézzel aláírt szerződések, ábrák, grafikonok, fénymásolatok, illusztrációk és egyéb grafikus anyagok átvitele válhat szükségessé. Ehhez egy másik ISDN szolgálatot célszerű igénybe venni, a Csoport 4 módban működő színes vagy szürkerségi fokozatokat átvivő faxot. Kialakíthatók távmérési (telemetry) vagy riasztó (alarm) szolgáltatások is ISDN szolgálat segítségével.

AZ ISDN RENDSZERARCHITEKTÚRÁJA

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 támogatja 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 végközponthoz volt kötve, 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.

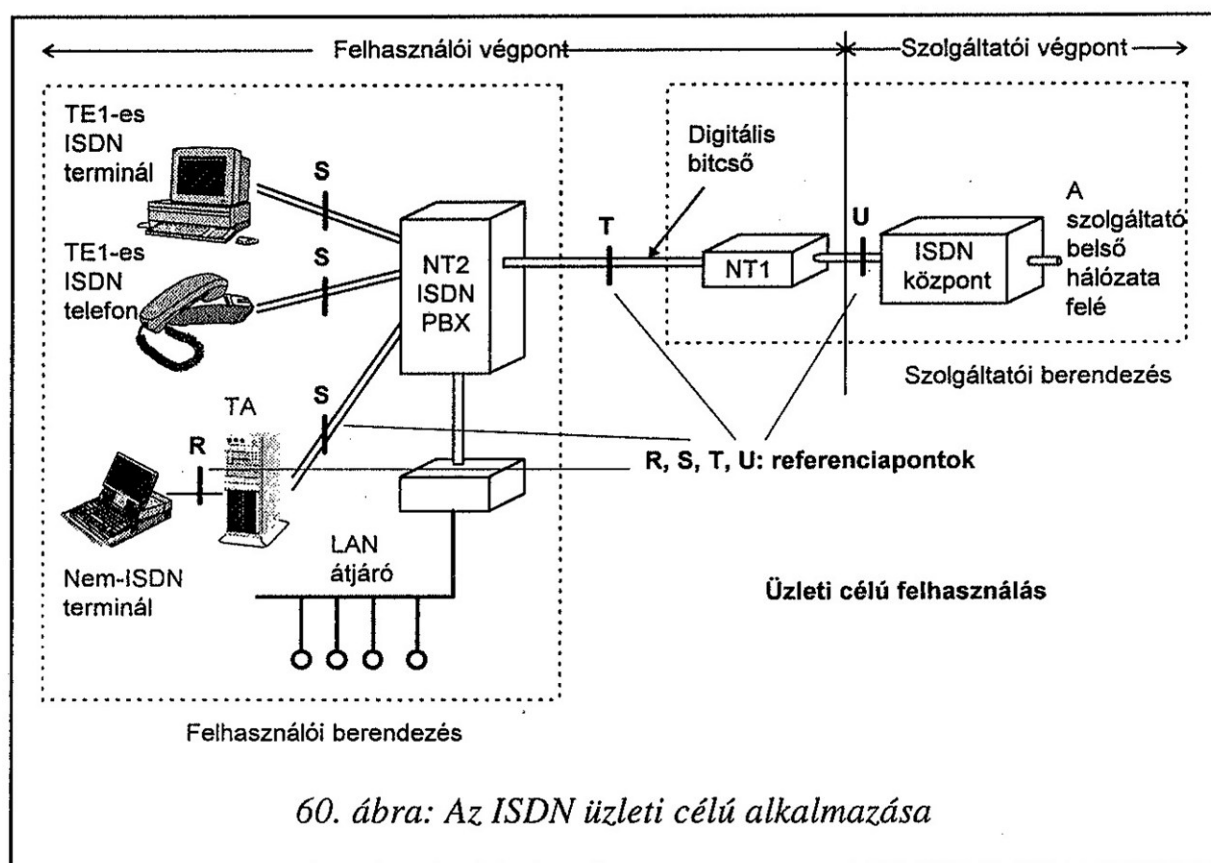
Nagyobb vállalatok számára a **60. ábrán** látható konfigurációt alkalmazzák. Ebben a modellben az NT1-el összekötve egy 2-es típusú hálózatzavghozást, egy NT2-t (Network Termination 2) is találunk. A modellben az NT2-t egy digitális telefon alközpont, a PBX (Private Branch eXchange) testesíti meg, ez nyújtja a telefonok, terminálok és egyéb berendezések számára a valódi interfészt.



Az ISDN összeköttetés megvalósítására a meglévő telefon vezetékeket használják fel, de természetesen az ISDN központok összeköttetése már üveggábelrel történik.

Ténylegesen azonban az NT1 doboz hálózati adminisztráció készítésére, helyi és távolsági hurok tesztelésére, hálózatfenn-tartásra és teljesítmény-figyelésre alkalmas elektronikát is tartalmaz. A passzív sínen lévő összes eszköznek címezhetőnek kell lennie, azaz egyedi címmel kell rendelkeznie. Az NT1-ben lévő sínhoz-záférés vezérlő, ha egyszer-re több eszköz is sínre akar kapcsolódni, akkor a versenyhelyzetet fel tudja oldani.

A CCITT négy referenciapontot határozott meg a különböző eszközök között: R, S, T, és U.



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 szátra 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.

Ezeket a módosított PBX-eket NT12 eszközöknek nevezik. Az NT1 előnye viszont az, hogy a felhasználó elől eltakarja az előfizetői hurkok technológiájában esetlegesen előforduló változásokat.

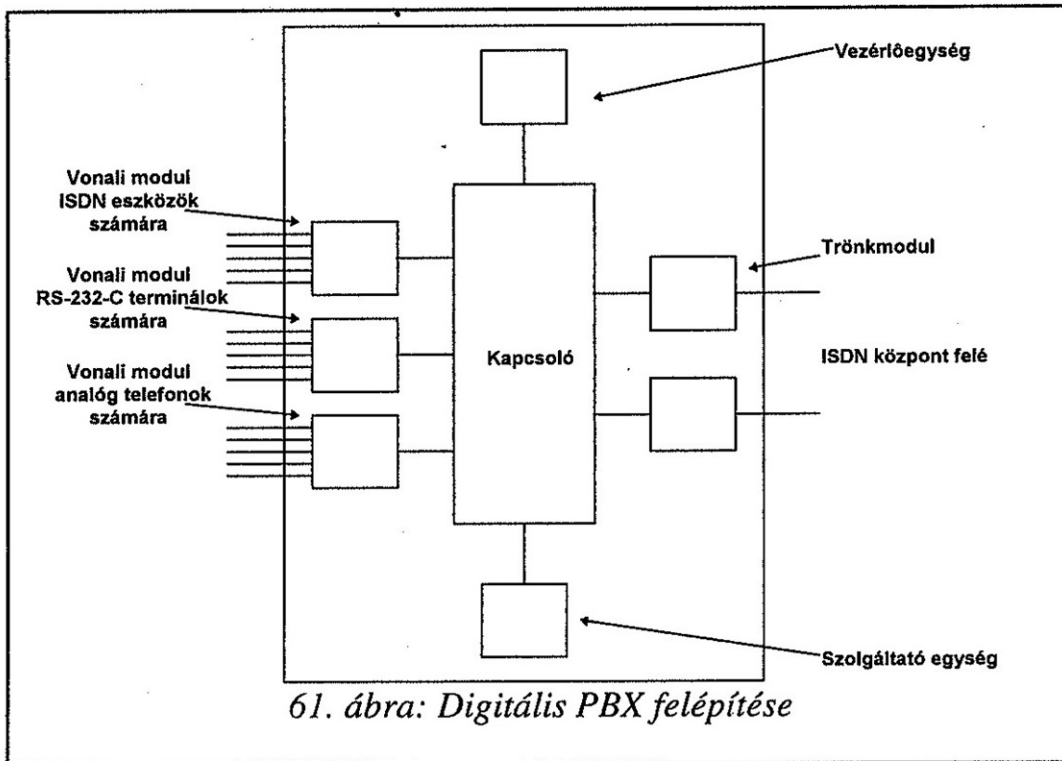
Ezek a referenciapontok lényegében szabványosan kialakított csatolófelületek. (Rétegszemplélet !)

Ha ma valaki magáncélú ISDN csatlakozást vásárol, akkor nála egyetlen dobozt szerelnek fel, amin öt csatlakozó pont van:

- DB9-es csatlakozó a PC soros vonalához (a dobozban benne van az ISDN modem)
- 2 db szokványos 4/6 pontos telefoncsatlakozó az analóg telefonhoz,
- 2 db RJ45-ös csatlakozó aljzat az ISDN készülékekhez, digitális csatlakozásra

Ha végül is megvalósul majd az áttérés az optikai szálakra, akkor az NT1-ek új környezetbe való illesztése sokkal egyszerűbb, mint a PBX-ek illesztése vagy esetleges kicserélése.

Egy másik kompatibilitási gond az S referenciapont körül alakulhat ki. Emiatt az egyes országokban különböző, egymással inkompatibilis ISDN rendszereket fedezhetünk fel.

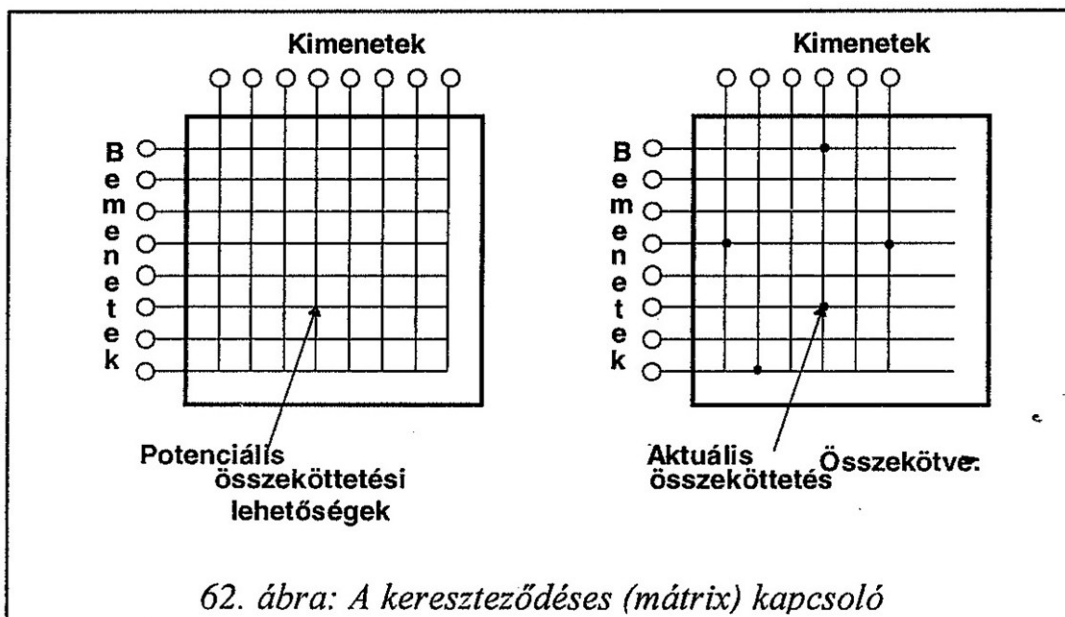


A DIGITÁLIS ALKÖZPONT (PBX)

A PBX-ek feladata az előfizetői hívások kapcsolása, amely kezdetben kézzel, a későbbiekben automatikusan, (rotary, crossbar kapcsolás) történt.

A modern PBX, amelyet PABX-nek (Private Automatic Branch eXchange — automatikus kapcsolású alközpont) vagy CBX2-nek (Computerized Branch eXchange — számítógép-vezérelt alközpont) is neveznek, már egy harmadik generációs rendszer. (**62. ábra**).

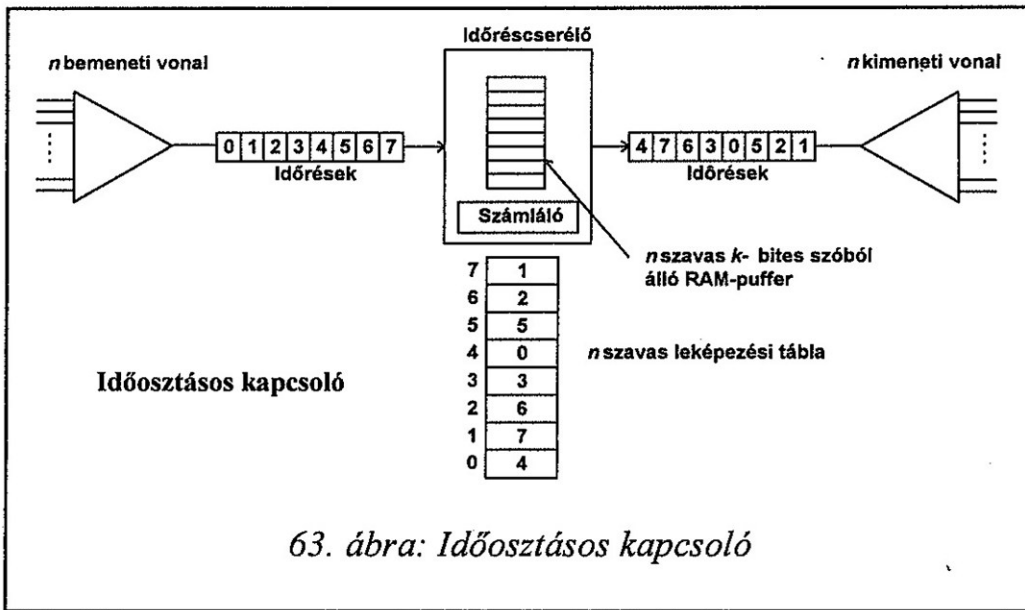
A telefonnal kapcsolatos részben már foglalkoztunk az alközpontokkal, amit most digitális adatátviteli vonatkozásai miatt ismét sorra kerül.



A PBX központi része egy áramkörkapcsoló, amelybe modulok illeszthetők. Minden modulkártya egy adott típusú eszköz számára biztosít kapcsolódási felületet, kimenetén mindig egy ISDN bitfolyamot előállítva. A vezérlőegység egy általános célú számítógép. A vezérlőegység összegyűjti a hívott szám számjegyeit, majd a kapcsolót a hívó és a hívott berendezések közötti kapcsolat létrehozására utasítja.

Kétféle típusú kapcsoló van.

Az egyik a mátrix alakban elrendezett kereszteződéses kapcsoló (crosspoint switch) ezt az **62. ábrán** láthatjuk.



Ennek mechanikus megfelelője a hagyományos telefonközpontokban használt crossbar kapcsoló.

A metszéspontokban a kimenő és bemenő vonalakat félvezetőkapcsolók köthetik össze.

A másik megvalósítási forma az **63. ábrán** látható időosztásos kapcsoló (time division switch). Az időosztásos kapcsoló az n darab bemeneti vonalat sorban egymás után letapogatja, és egy n (idő)résből álló bemeneti keretet állít össze.

Minden egyes rés k bitből áll. Az időosztásos kapcsoló legfontosabb eleme az időréscserélő (time slot interchanger), amely időréseket tartalmazó bemeneti kereteket fogad, és ugyancsak időréseket tartalmazó kimeneti kereteket állít elő.

AZ ISDN INTERFÉSZ

Az ISDN célja, hogy a felhasználó számára egy digitális bitsövet biztosítson, akár a T, akár az S referenciaponton keresztül. Új típusú csatlakozót definiál, aminek nyolc érintkezője (tüje) 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



2. FIZIKAI ÁTVITELI JELLEMZŐK ÉS MÓDSZEREK

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 bitsőben több csatorna is kialakítható, amelyek közül az eddig szabványosított csatornatípusok:

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

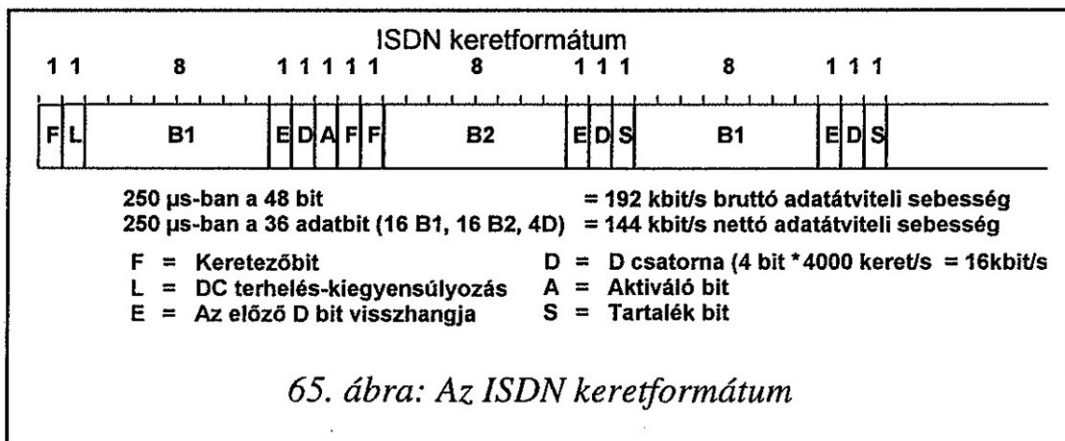
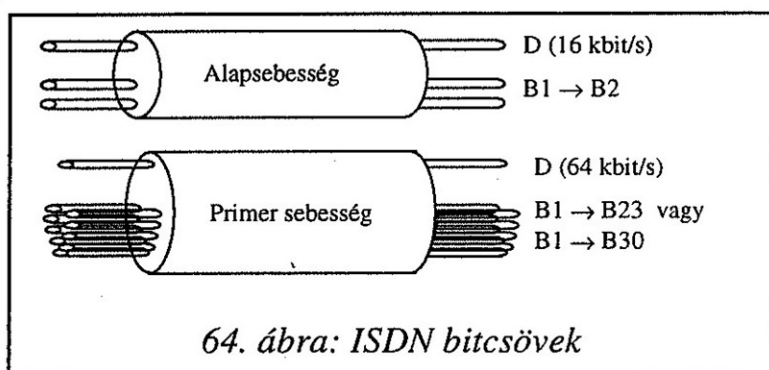
A szabványosításnak nem szándéka, hogy a digitális bitsövön a csatornák tetszőleges kombinációja létezzen. Eddig három kombinációt szabványosítottak (1994):

- 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 65. 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 μ sec 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 bitfolyam. 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.

AZ ISDN JELZÉSMÓD

Az ISDN a CCIS által meghonosított átvivő sávon kívüli jelzémódot használja. Minden jelzés (azaz vezérlőcsomag-küldés) 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.

A D-csatornán cserélt csomagok formátumát és tartalmát a CCITT SS #73 ajánlás definiálja.

ATM — ASYNCHRONOUS TRANSFER MODE

Az ISDN továbbfejlesztésében nagy szerepet játszanak az olyan átviteli módszerek alkalmazásai, 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 nem hatékony csatorna kihasználást okoz.

Az alkalmazásokat megvizsgálva, azok alapvetően kétféle digitális átvitelt igényelnek:

- **állandó bitsebességet biztosító (CBR — Constant Bit Rate)**, Ilyen például a digitális 64 kbit/s-os telefon, telefax, TV átvitel.
- **változó bitsebességet biztosító (VBR — Variable Bit Rate)**. változó sebességű adatátviteli sebességet igényel az interaktív szöveg és képátvitel. Ilyenkor sokszor lökészerűen egy állandó bitsebességű átvitel zajlik, amit hosszabb szünet

Egy tipikus híváskérési üzenet kijelöli a használandó B csatornát, és megadja a hívandó ISDN telefonszámot, és esetleg egyéb opciókat (pl. "R" hívás) is tartalmaz. A D-csatorna további három logikai alcsatornára van felosztva:

s alcsatornára, amely jelzési célokra szolgál (pl. hívás-felépítés),

t alcsatornára, amelyen telemetriás eszközöket (pl. füstérzékelők) használhatnak,

p alcsatornára, amely kis adatátviteli sebességen adatcsomagok átvitelére alkalmas.



Például a telefonvonalak tervezésénél csak az emberi beszéd 4 kHz-es sávzélességét vették alapul, nem gondoltak nagysebességű adatátvitel megvalósítására.

STM (Synchronous Transfer Mode) az ATM ellentéte, ott előre lefoglalt idő szeletekben történt az adatátvitel).

követ. Video átvitelnél is elegendő csak a kép teljes változásakor átvinni a képet, közben csak a változásokat.

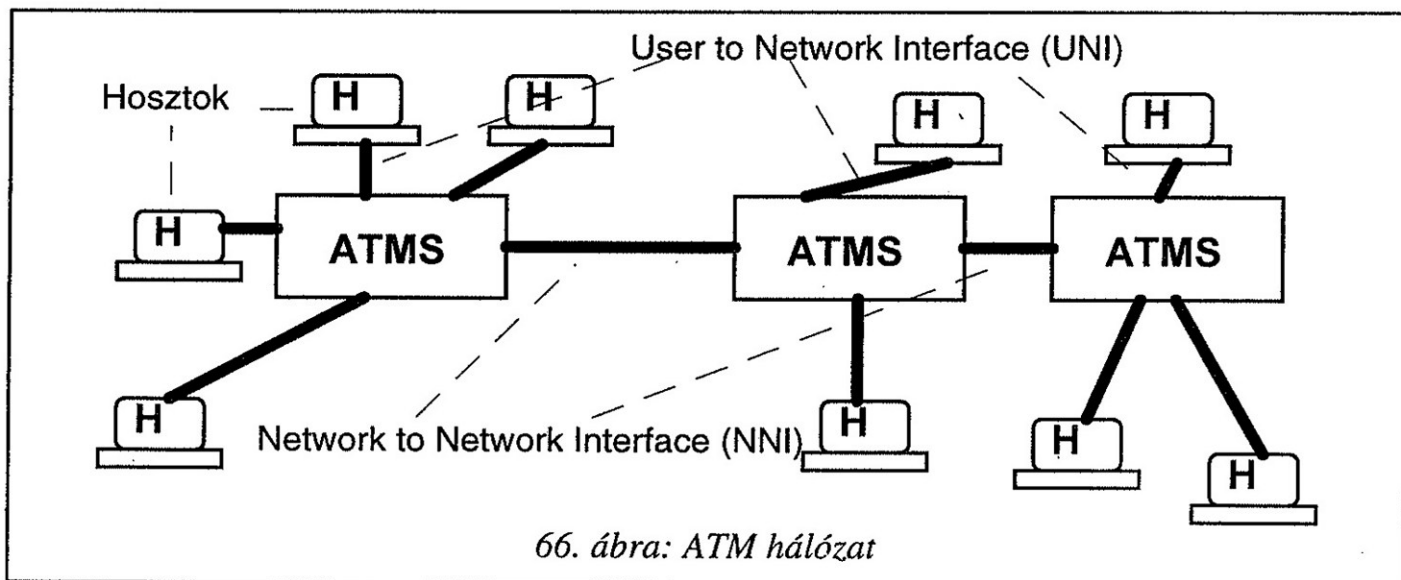
Az ATM-et, ez utóbbi VBR típusú adatátvitelre tervezték. Ezeket a követelményeket kielégítendő, az ATM aszinkron időosztásos multiplex adatátvitelt használ, viszonylag kis méretű cellákkal (csomagokkal.) A kis cellaméret előnyös, mert hiba esetén kevés információ veszik el, az átvitelt megvalósító kapcsolók egyszerűbbek és hardveresen megvalósítható.

A csomagok 53 oktet (oktet=8 bit) hosszúak, ebből mindössze 5 oktet a fejléc és 48 oktet az információ. A fejléc tartalmazza a csomagot vivő virtuális kapcsolat adatait. Az információ mező rövid, így a kezelő pufferek méretei kicsik lehetnek.

Az ATM architektúra ún. ATM kapcsolókon alapul (továbbiakban: ATMS). A kapcsolókat kétirányú üvegszálak kapcsolat köti egymással. A hosztok a kapcsolókhöz szintén üvegszálakon keresztül kötődnek. Az ATMS-ekből kialakított ATM hálózatot felhasználva bármely, a hálózathoz kapcsolódó hoszt tud a másik hoszttal kommunikálni.

Az STM időosztású multiplexelési sémát alkalmaz, ahol az egyes csatornák adatait az időben ismétlődő keretekben elfoglalt helyük azonosítja. Ez a séma azonban csak nehézkesen tudja kezelni a felhasználónál fellépő különböző sebességű csatornák iránti igényeket.

A kis csomagméret miatt a csomagok továbbítása a jóval nagyobb sebességet biztosító céláramkörökkel lehetséges.



Az ATM kapcsolat orientált összeköttetést használ a hosztok összekötésére. Ez azt jelenti, hogy először a két hoszt között egy virtuális áramköri kell kialakítani: ez lehet kapcsolt virtuális áramkör (Switched Virtual Circuits=SVC), vagy állandó virtuális áramkör (Permanent Virtual Circuits=PVC).

Virtuálisnak, vagyis látványosan azért nevezük az áramköröket, mert nem vezetékes összeköttetést, hanem pl. időosztásos csatornákat használ a pont-pont kapcsolat kialakítására.

Az SVC a szokásos telefonhíváshoz hasonlóan működik. A hoszt a hozzákapcsolt ATMS-el kommunikálva, SVC létrehozását kéri. A hoszt megadja a távoli hoszt címét, és a kért szolgáltatás minőségét, és várja, hogy az ATMS létrehozza a kapcsolatot. ez az ATM jelzésrendszer segítségével valósul meg. A kialakult útvonal esetleg több ATMS-en keresztül valósul meg.

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.

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:

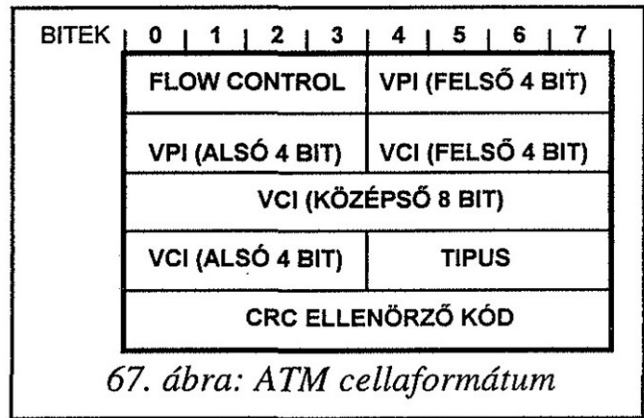
- 8 bites Virtual Path Identifier (VPI) és a
- 16 bites Virtual Circuit Identifier.

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 elvesztett cellák kezelése, hibakezelés és javítás. A **68. ábrán** látható a réteg elhelyezkedése.

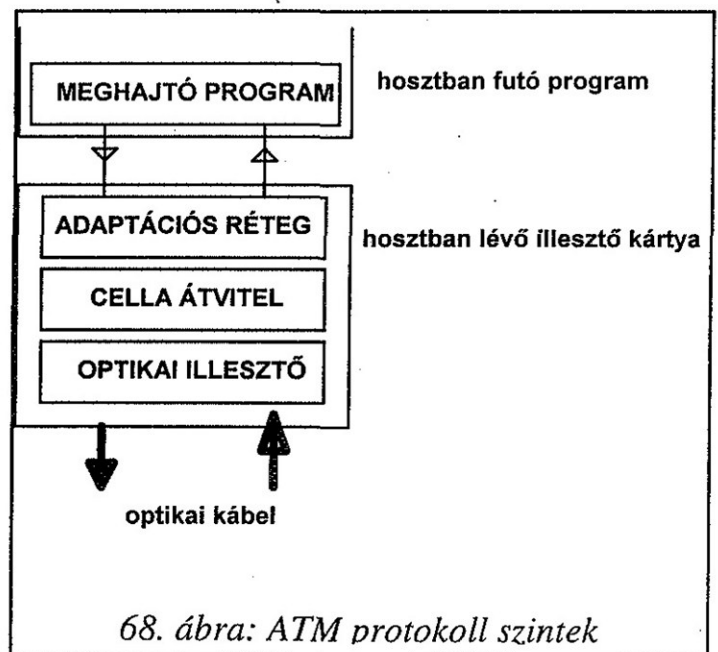
A kapcsolat kialakításánál a hosztnak azt is meg kell adnia, hogy milyen adaptációs rétegprotokollt használja. Csak két ilyen rétegprotokollt definiáltak eddig:

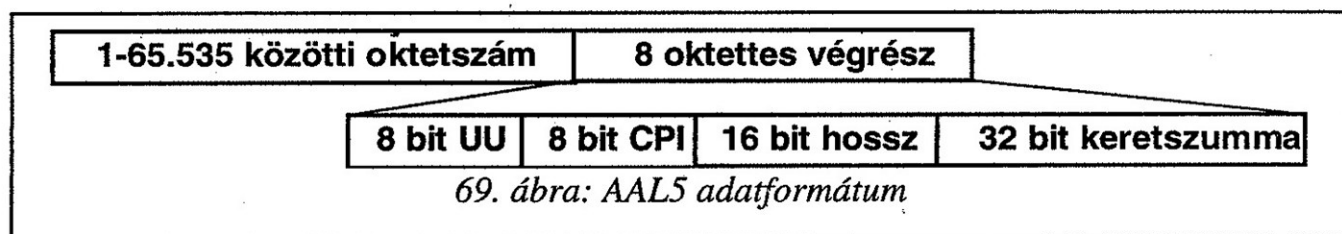
ATM Adaptation Layer 1 (AAL1): fix sebességgel visz át adatokat az ATM hálózaton keresztül (pl. audio és video átvitelek).



ATM protokoll szintek:

1. szint: fizikai szint
koax, üvegszál
2. szint: ATM datalink
53 bájtosos cellák,
CBR VBR
3. szint: ATM Adaptation
bizonyos felhasználások





ATM Adaptation Layer 5 (AAL5): adatcsomagok átvitele ATM hálózaton. Az

AAL5 a felső rétegek számára hosszú adatcsomagok átvitelét biztosítja. Érdekes, hogy míg legtöbb keretformátum fejléccet használ a keretek azonosítására, itt ez a végrészben (trailer-ben) történik. A végrész UU és CPI része jelenleg még nem használt.

Minden AAL5 keret az ATM hálózaton 48 oktetes cellákká alakítva megy át, ha az átvitt utolsó cellánál az adatrész 48 oktetenél kevesebbre adódik, akkor a maradék 0-biteket tartalmazó oktetekre kiegészítve kerül átvitelre (ún. zero padding).

Átvitelkor a hoszt átad egy adatcsomagot AAL5 rétegnek. A réteg egy végrészt generál, az adatcsomagot 48 oktetes részekre vágva a cellafejvel kiegészítve átviszi a hálózaton. A vételi oldalon a réteg a cellákból összeállítja az adatcsomagot, és ellenőrzés után átadja a fogadó hosztnak. Honnan tudja a vételi oldal, hogy mikor jön a végrész? A megoldás egyszerű: A cellafej TIPUS oktetjének legkisebb bitje jelzi. (keretvégljelző bit).

Az elérhető adatátviteli sebesség nagyobb 1 Gbit/s-nál, jellegénél fogva ISDN kommunikáció átvitelére alkalmas.

ADSL RENDSZER

Az ADSL (Aszimmetrikus Digitális Előfizetői Vonal) aszimmetrikus adatfolyamot továbbít (a két irány sávszélessége különbözik), és telefon szolgálatot biztosít egyidőben egyetlen előfizetői érpáron. Az eredeti alapsávi szolgálat biztosításához (telefon vagy 2B+1D ISDN) az ADSL rendszer mindkét végén jelen kell lenni leválasztó szűrőnek. Ez a speciális szűrő („splitter”) lehet az ADSL modem beépített része, de különálló egység is. A nagy sebesség miatt az előfizetői oldalon Ethernet és/vagy ATM illesztést használnak, illetve perspektívikusnak látszik a néhány éve megjelent, de mostanra beérett USB (Universal Serial Bus) technika, melynek nagy előnye, hogy a jelenleg kapható szinte valamennyi alaplapra rá van integrálva, így a PC használóknak semmilyen külön befektetést nem jelent az ilyen csatlakozású ADSL modem használata.

Aszimmetrikus adatfolyam :

a felhasználó felől a sávszélesség kisebb (100-800 kbit/sec), mint a felhasználó felé (5-20 Mbit/sec)

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.

Ennek megfelelően az előfizetői oldalon biztosított Ethernet interfészeletről érkező jeleket, az RFC 1483-ban leírt bridging technikát használva, ATM cellákba ágyazva viszik át.

BEÁGYAZOTT RENDSZEREK KOMMUNIKÁCIÓJA

A mikroprocesszor mint a számítógép központi egysége, a hozzákapcsolt memóriával és be- kimeneti egységekkel együtt használható mikroszámítógépként. Azokat az áramköröket, amelyek egy tokba integrálva tartalmazzák az előbb felsorolt elemeket mikrovezérlőknek vagy más néven mikrokontrollereknek nevezzük. Ezek egy nagyobb rendszer részeként abba beillesztve, „beágyazva” önállóan is képesek működni, az összehangolt rendszerműködés érdekében egymással és a környezetükkel is kommunikálniuk kell. Ezek végeredményben szintén számítógépes hálózatok, ezért célszerű áttekinteni ezt a területet. Használatukra számos példát lehet hozni: ilyen vezérlőt tartalmaz a PC billentyűzete, az egér, a hajlékony- és merevlemez meghajtó.

A következőkben három, ilyen rendszerben használt összeköttetés-módot mutatunk be, először egy párhuzamos, majd utána két soros megoldást.

BÁJT-SOROS BITPÁRHUZAMOS ADATÁTVITEL (GPIB)

Nagyon sok esetben kell mérőkészülékeket vezérelni, mérési eredményeiket feldolgozni, készülékek együttműködésének biztosításával mérőrendszereket kialakítani.

Mivel ilyen esetekben általában laboratóriumon belüli, nem nagy távolságokról van szó, ezért célszerű a párhuzamos adatátvitel előnyeit kihasználni.

Az IEC 625 szabvány ajánlásában egy általános célú csatlakozó rendszer leírását specifikálta, amely elsősorban automatikus mérőrendszerek készülékeinek összekapcsolásra szolgál, és biztosítja az összekapcsolt eszközök közötti kommunikációt.

Az IEC 625 ajánlás célkitűzései

- Gyártó független nyitott kommunikációs rendszer
- Eltérő képességű készülékek összekapcsolhatóságának megvalósítása
- Minél egyszerűbb, egységesebb kódrendszer kialakítása

A párhuzamos adatátvitel előnye a nagyobb adatátviteli teljesítmény, de több vezeték igényel.

A kommunikáció üzenetek formájában zajlik:

- Interfész üzenetek: a rendszer működését koordinálják
- Készülékfüggő üzenetek: az összekapcsolt berendezések üzemmódját befolyásolják.

Alapvető sajátosságok:

- Kieépíthetőség: maximum 15 készülék, melyek egymástól legfeljebb 20 m-re lehetnek
- Adatátvitel: maximum 1 Mbit/s, az adatkeretek rövidek (10-20 karakter)
- Mechanikai előírások: Speciális 25 pólusú csatlakozó
- Villamos előírások: Meghajtó és vevőáramköröknek a TTL specifikációt kell teljesíteni

Az összekapcsolt készülékek kommunikációs státusza háromféle lehet:

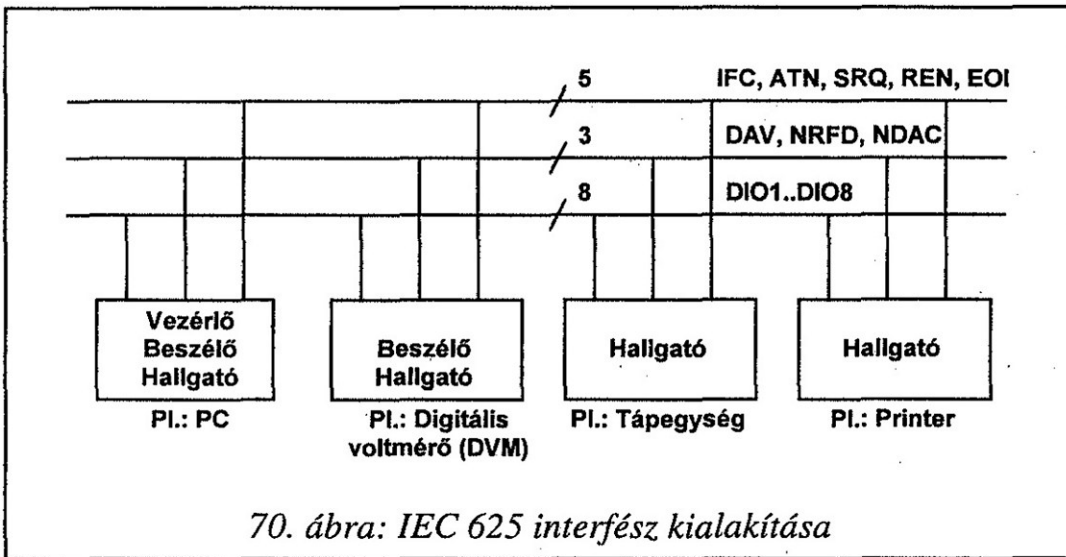
- **Beszélő (talker):** Egyszerre csak egy beszélő lehet aktív. Beszélő státuszban egy készülék készülékfüggő adatokat továbbíthat a buszon.
- **Hallgató (listener):** Címzett állapotában egy hallgató készülék készülékfüggő üzeneteket vehet.
- **Vezérlő:** Gondoskodik a beszélő és hallgató címek kiosztásáról.

Funkcionális előírások:

A buszra kapcsolt rendszer minden elemének funkciójától függően lehet beszélő, illetve hallgató címe mely őt adott minőségében egyértelműen azonosítja. Közös hallgató címe lehet az azonos információkat igénylő berendezéseknek, de beszélő cím csak egyedi lehet. Általában minden mérési összeállítás tartalmaz egy vezérlő készüléket amely koordinálja a kommunikációt.

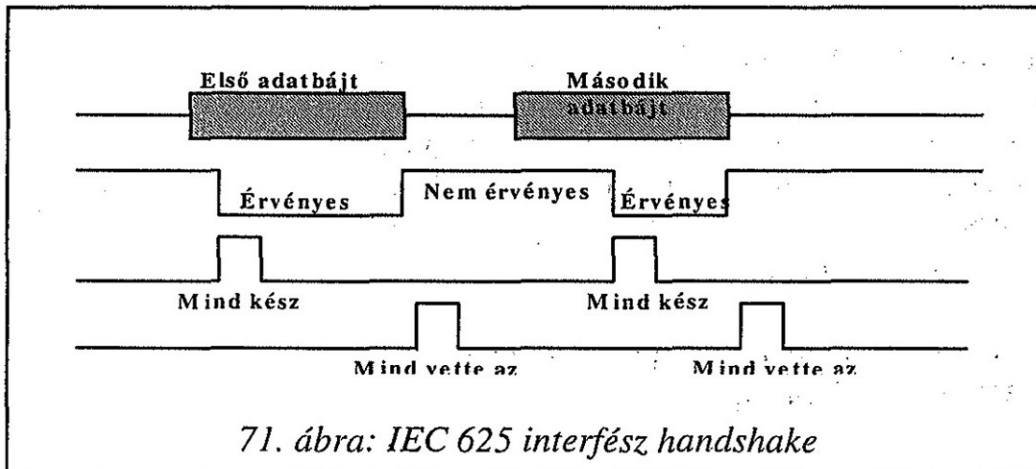
A rendszer felépítése: A rendszer minden készüléke rendelkezik buszillesztő egységgel a buszrendszer felépítése:

- 8 adatvezeték
- 3 vezérlő vezeték
- 5 kiegészítő vezérlő vezeték



Nézzük meg az egyes jelvezetékek szerepét:

- DIO1..DIO8 (Data I/O): üzenet továbbítása, (cím, program, eredmény, utasítás)
- DAV (Data Valid): a jelforrás jelzi, érvényes adat van a buszon
- NRFD (Not Ready for Data): hallgató jelzi vételkésztségét
- NDAC (No Data Accepted): hallgató készülék jelzi az elfogadást
- IFC (Interface Clear): valamennyi készülék alapállapotba állítása
- ATN (Attention): cím, üzenet, címzett parancs, elválasztása a készülékfüggő üzenettől



A 71. ábrán látható idődiagramon végigkövethető az adatátvitel handshake folyamata. Mivel a rendszer elemei eltérő sebességűek ezért a buszsebesség a vezérlőjelek segítségével mindig a leglassabb készülékhez igazodik.

- SRQ (Service Request): a vezérlőtől kérhető az éppen folyó művelet megszakítása
- REN (Remote Enable): a vezérlő jelzi, hogy üzenete távvezérlésre állít egy készüléket
- EOI (End or Identify): üzenet utolsó elemének jelzése, vagy vezérlő általi lekérdezés

AZ I²C BUSZ

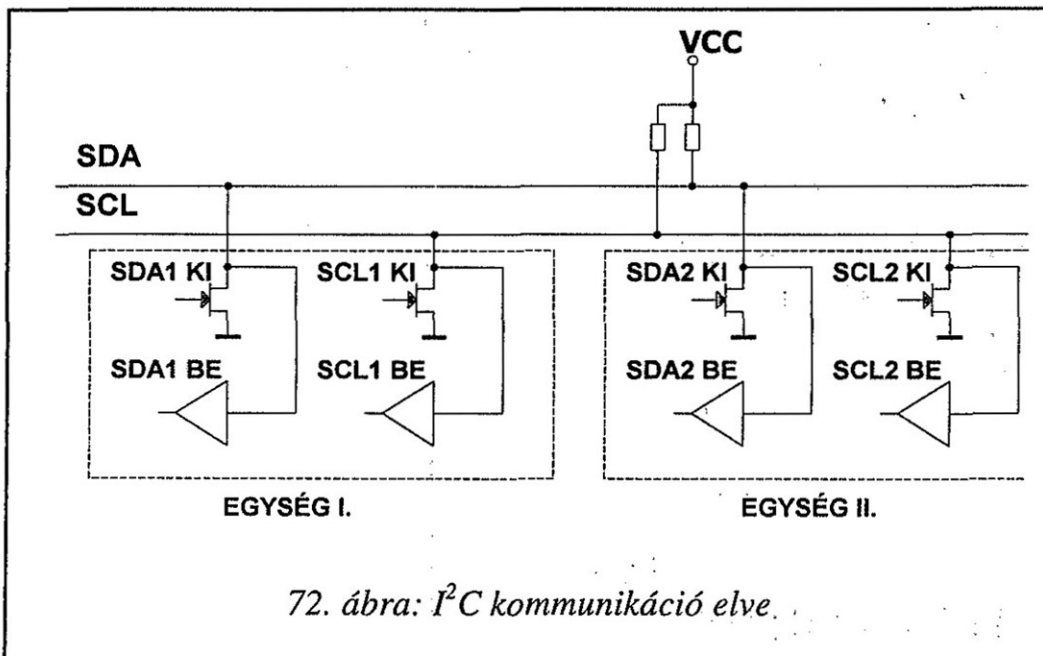
Az I²C, Inter IC azaz IC-k közötti busz. Az I²C busz nagybonyolultságú integrált áramkörök közötti soros információcserét biztosító, azt fizikailag három vezetékkel megvalósító sínrendszer. Az átviteli félduplex módon történik, sebessége kb. 100-400 kbit/s-ig növelhető. Az ilyen buszt tartalmazó nagybonyolultságú integrált áramkörök egymással könnyen, kevés vezetékkel tudnak sorosan kommunikálni.

A kommunikáció kétirányú adatvonalon (SDA=Serial Data) keresztül történik, és egy külön órajel (SCL=Serial Clock) szinkronizálja az adatvezetékén az adatokat.

A busz elvi felépítése a 72. ábrán látható.

A tranzisztorok kikapcsolt állapotában a felhúzó ellenállás miatt, a vonalak magas állapotban vannak. Ez az alaphelyzet. Ha bármelyik tranzisztort bekapcsoljuk, az a vezetékét a földre kapcsolja, így nulla állapotú. Ezt a megoldást az elektronikában huzalozott vagy kapcsolatnak hívják.

A vezérlési elvből következik, hogy mindig csak egyetlen egység



vezérelheti az adott vezetékét, a többi egység a tranzisztorát nem kapcsolhatja be. Az eddig tárgyalt adás mellett minden egység képes a vonalon lévő adatokat is venni egy erősítőn keresztül.

Az előzőekből következően minden egység lehet Adó ill. Vevő. Ezen felül megkülönböztetünk Master és Slave eszközöket. Így összesen két funkció és két szerep különböztethető meg:

A funkciók:

- **TRX = Transmitter (adó):** Az egység amelyik adatot küld a buszra.
- **RCV = Receiver (vevő):** Az egység amelyik adatot fogad a buszról.

A szerepek:

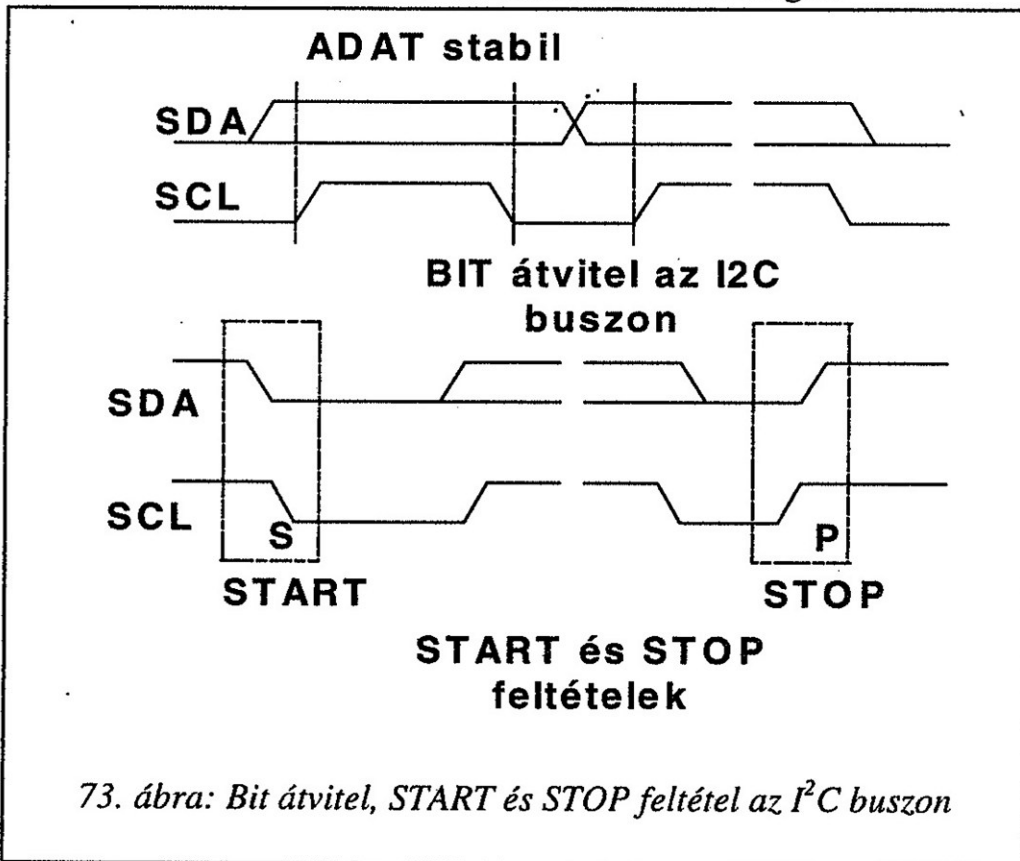
- **MST = Master (mester):** Az egység amelyik kezdeményezi az átvitelt, az átvitelhez az órajelet generálja, és be is fejezi az átvitelt.
- **SLV = Slave (szolga):** A mester által megcímezett egység.

Egy mikrokontroller I²C egysége mindegyik szerepre és funkcióra képes. Természetesen ha ez az egység vezérli a perifériákat (és a gyakorlatban ez a leggyakoribb eset), akkor szerepe: mester és a perifériák a szolgák.

A busz multi-master kialakítású. Ez azt jelenti, hogy buszra kapcsolódó eszközök közül nem csak egy, hanem több is átveheti az átvitel vezérlését, azaz az ütközések kezelése szükséges.

Még egy érdekes megállapítás: egy egység el tudja dönteni hogy a vezetékét más nem vezérli-e. Ha ugyanis az adatokat a vezetékre kapcsolja, a saját vevőerősítőjén ugyanazt az adatot kell vennie, mint amit kiküldött. Ha ez nem teljesül, valamelyik másik egység is „piszkálja” a vonalat, azaz a buszfoglaltság azonosítható.

Ezzel kapcsolatos fontos tulajdonság az arbitration, vagy döntés. Ez egy eljárás, ami biztosítja, ha egynél több mester akarja a buszt vezérelni, akkor ezt csak egyetlen egy tudja megtenni, így adatvesztés nem léphet fel.



73. ábra: Bit átvitel, START és STOP feltétel az I²C buszon

BIT ÁTVITEL

Az átvitel bit szinten a következő (73. ábra): az eredetileg magas szinten lévő SDA vonalra kerül a 0 vagy 1 értéknek megfelelő feszültségszint. Az SCL vonal magas szintje alatt érvényes az adat. Az adat csak az SCL vonal alacsony szintje alatt változhat.

A busz aktív és inaktív állapotát a START és STOP feltételekkel tudjuk definiálni.

START feltétel akkor lép fel és a busz aktív lesz amikor SCL magas állapotában az SDA vonalon egy H-L átmenet van.

STOP feltétel akkor lép fel, amikor SCL magas állapotában az SDA vonalon egy L-H átmenet van.

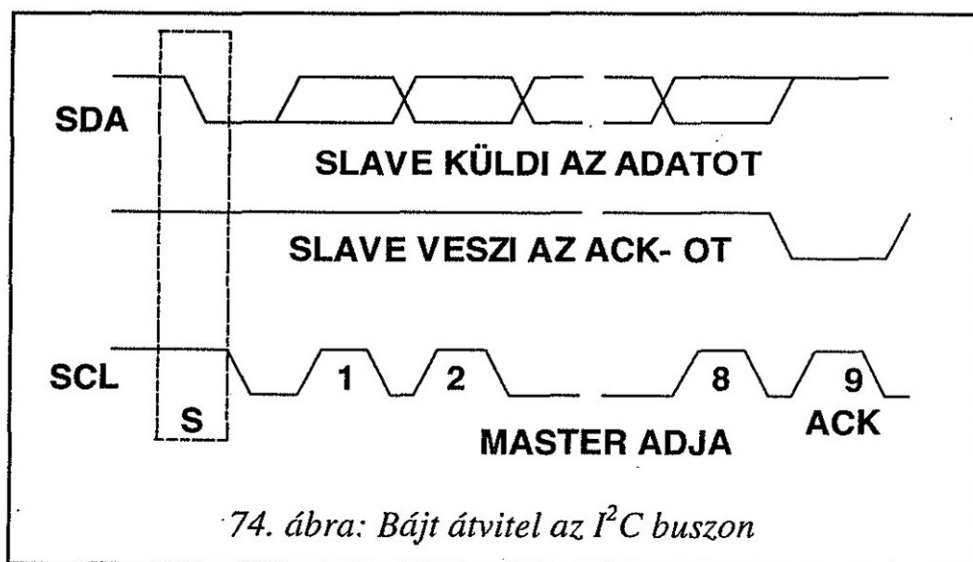
A START és STOP állapotokat csak a mester generálhatja. A busz aktív a START és STOP állapot között. Ezután válik a busz szabaddá.

BÁJT ÁTVITEL

Az SDA vonalon sorosan átvitt adat mindegyike 1 bájt = 8 bit hosszúságú. Az átvitt bájtok száma nincs korlátozva. Az adónak a vevő minden bájt vételét egy L szintű nyugtázó (ACK = acknowledge) bit küldésével igazolja. Az ehhez szükséges órajelet a mester generálja, az adó az SDA vonalat elengedi. A vevőnek ekkor az ACK generálásához le kell húznia az SDA vonalat. Az átvitel a legmagasabb helyiértékű (MSB) bittel kezdődik.

Az adatbitek az adó, az ACK bitet a vevő küldi. Ha egy vevő nem képes egy adatot venni, akkor az ACK bit küldése helyett az SCL vonalat 0 szinten tartja. Ez egy várakozó állapot.

Vegyük észre, hogy az adat- és az órajel vonalat az adó és a vevő felváltva használja. Ez megköveteli mind az adó mind a vevő számára a nagyon pontos kommunikációs feltételek betartását. A "minden bájt nyugtázása" szabály alól két kivétel van:



74. ábra: Bájt átvitel az I²C buszon

Az egyik akkor lép fel, ha a mester a vevő (MST/RCV). Ilyenkor jelezni kell az adatbájt sorozat végét, a küldőnek nem adva ACK-ot. Az ACK jelhez kapcsolódó órajelet a mester természetesen generálja, de az SDA vonalat nem húzza le L szintre. Ezt hívják negatív nyugtázásnak (NACK).

A másik kivétel: a szolga akkor küld NACK jelet, ha nem képes újabb adatbájtokat elfogadni. Ez akkor lép fel, ha olyan átvitelt kezdeményezünk, amit nem képes fogadni.

ADATFORGALOM A BUSZON

A buszon lévő minden eszköznek saját címe van. Mielőtt adatátvitel történne a buszon, a mester START állapotba hozza a buszt, majd kiadja a buszra a szolga címét, amelyikkel adatot akar cserélni. Az a szolga, amelyik felismeri a saját címét, ACK jelet küld vissza. A címzést a mester végzi közvetlenül START állapot után. Ez az első küldött bájt.

A cím hét bites. A nyolcadik bit dönti el a szolgálával történő adatcsere irányát. 0.bit jelöli az írást, ilyenkor a mester küld adatokat (W), 1 értékű bit pedig az olvasást (R).

A buszra kapcsolódó eszközök címei két kategóriába sorolhatók: Az egyik kategóriában a cím programozható, ezek általában a mikrokontrollerek. A másik kategóriát a különféle funkciókat megvalósító periféria áramkörök alkotják.

Ezeknél az eszközök címe két részből tevődik össze: egy típus címből (4 bit) és egy hardver címből (3 bit); az eszköz tokozásán a megfelelő lábak 0-ba ill. 1-be kötésével. A típus cím az azonos (típusú) tokoknál mindig megegyezik. Ezzel a címmel jelentkezik be a slave eszköz ill. ezzel a címmel szólítja meg a master eszköz a slave-et adatcsere előtt.

Az egy mester — több szolga struktúrájú buszon zajló adatátvitel a következő (M jelöli a mester által-, L a szolga által küldött adatbiteket):

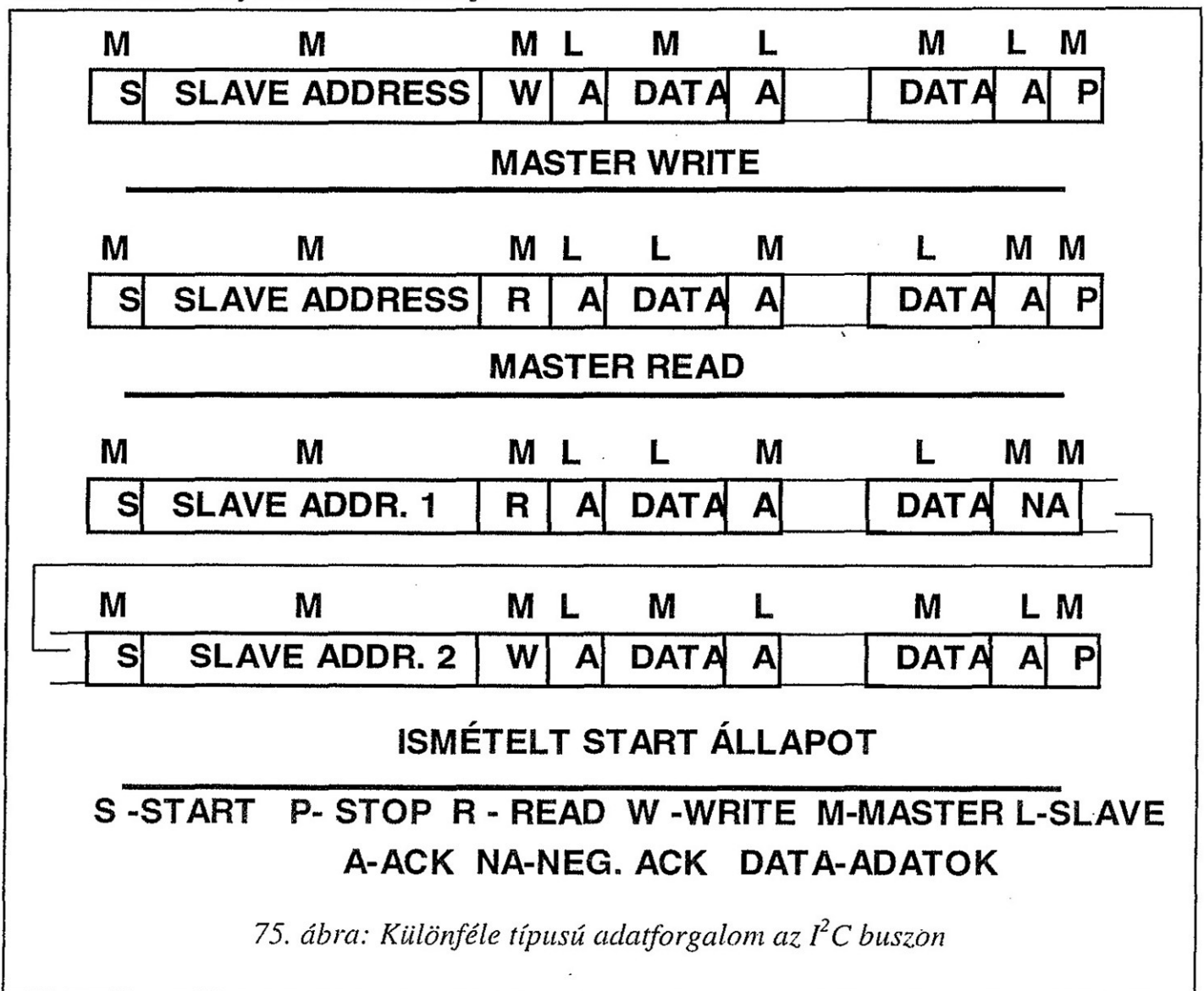
MASTER WRITE

A mester START állapotba hozza a buszt (S) és kiküldi a szolga címét. A cím legkisebb helyiértékű bitje $W=0$. Ezt a szolga az ACK jel visszaküldésével igazolja (A). Ezek után a mester küldi az adatokat a szolganak (DATA), és az minden bájt vételét (A) küldésével igazolja. Az utolsó adat küldése után a mester STOP állapotba hozza a buszt (P).

MASTER READ

A mester START állapotba hozza a buszt (S) és kiküldi a szolga címét. A cím legkisebb helyiértékű bitje R=1. Ezt a szolga az ACK jel visszaküldésével igazolja (A). Ezek után a mester fogadja az adatokat a szolgáltól (DATA), és minden bájt vételét (A) küldésével igazolja. Az utolsó adat küldését a mester negatív nyugtázással jelzi (NA) Ezek után a mester STOP állapotba hozza a buszt (P).

A fenti esetben a mester minden átvitelnél a buszt újból nyitja és zárja. Amennyiben a buszon több szolgálal akar a mester adatot cserélni, a minden átvitelt lezáró STOP, majd az indító újabb START állapot sokat lassít az átvitelen. Ilyenkor használható az ismételt START állapot generálása. Ez az jelenti, hogy az átviteleket nem STOP (P) hanem a következőt indító START (S) állapottal fejezzük be, azaz a mester a buszt folyamatosan használja.



Ismételt START állapot használata

Itt egy olyan esetet látható, amikor először a mester adatokat kapott egy szolgáltól, majd utána adatokat küld nem szükségképpen azonos című szolgának.

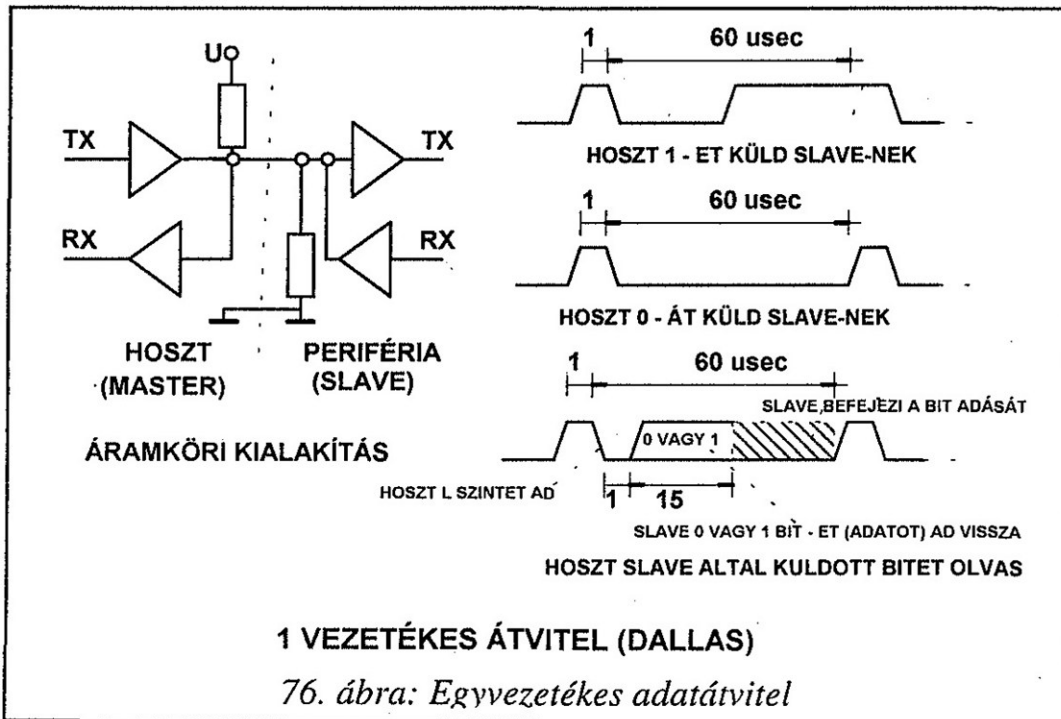
A fenti adatátviteli protokollokat két módon lehet megvalósítani: vagy beépített hardver segítségével (ilyeneket tartalmaznak az erre felkészített mikrokontrollerek és az I²C buszra kifejlesztett periféria áramkörök), vagy megvalósítására szoftver "bit-billegetéses" programot írhatunk. A fent ismertetett adatátviteli változatokat a **75. ábrán** foglaltuk össze. Az M illetve L betűk azt jelzik, hogy az adott bájtos a mester vagy a szolga küldi.

EGY VEZETÉKES ADATÁTVITEL

Az adatátvitel megvalósítható (a közös föld mellett) egy vezetékkel is. Ilyenkor a soros adat szinkronizmusát egy külön órajel nem tudja biztosítani, ezért a már az aszinkron soros átvitelnél ismertetett időzítési szinkronizálás jöhet számításba. Nézzük a **76. ábrát!**

A kétirányúságot az előzőekben már bemutatotthoz hasonló tranzisztor-erősítő kombináció biztosítja. Természetesen az átvitel itt is csak félduplex lehet.

Egy bit átvitelkor a mester az eredetileg alacsony szintű vonalat 1 μ s ideig elengedi, így az magas állapotba kerül. Utána, ha a mester egy 60 μ s-os tartományon belül végig nulla szintet kényszerít a vonalra, akkor ezt 0 értékű bitnek értelmezi a szolga.



Ha ezen belül (általában a felénél) 1-be váltja a vonalat, akkor 1 értékű bitet küldött. A szolga által küldött adást is a mester vezérli. Induláskor itt is a mester adja a vonalra az 1 μ s-os magas szintet.

Ezek után 1 μ s múlva figyeli a vonalat, ha a szolga legalább 15 μ s időre magasra állítja, akkor 1 értékű bitet küldött, ha nulla szinten tartja, akkor 0 értékű bitet küld vissza.

A most bemutatott két megoldás jól illusztrálja azokat a lehetőségeket, amelyeket egy controllerrel megvalósított kommunikáció biztosítani tud.

ELLENŐRZŐ KÉRDÉSEK



1. Mi az sávszélesség és az adatátviteli sebesség?
2. Magyarázza meg a baud és bit/s mennyiségek közötti különbséget!
3. Milyen vonalmegosztási módszereket ismer? Hogyan történhet több csatorna átvitele egy vonalon? Mi a multiplexelés?
4. Ismertesse a frekvenciaosztásos multiplexelés módszerét!
5. Ismertesse a szinkron időosztásos multiplexelés módszerét! Mi az a PCM?
6. Ismertesse a vonalkapcsolás elvét!
7. Mutassa be az üzenet és csomagkapcsolást! Mi köztük az alapvető különbség?
8. Melyek a fizikailag összekötött és össze nem kötött kapcsolatok jellemzői, előnyei, hátrányai?
9. Ismertesse a csavart érpáras összeköttetés jellemzőit! Milyen kategóriái vannak?
10. Ismertesse a koaxiális kábelt használó összeköttetés jellemzőit! Hogyan jellemezné az alapsávú és szélessávú kábeleket?
11. Ismertesse az üvegszál (optikai) kábelt használó összeköttetés jellemzőit! Mit jelentenek a egymódusú, (monomódusú) illetve többmódusú (multimódusú) fogalmak?
12. Milyen optikai kábel illesztő egységeket ismer? Jellemezze ezeket!
13. Mi az transzponder és mi a geostacionárius pálya? Milyenek a késleltetések egy műholdas rendszerben?
14. Hogyan működik a telefon? Hogyan épül fel egy hierarchikus telefonrendszer? Mi az a trónk?
15. Mi a cellás mobiltelefonok működési elve? Hogyan osztoznak a csatornákon?
16. Ismertesse a moduláció fogalmát! Szinuszos jel milyen jellemzőit lehet modulálni?
17. Mutassa be az FSK átviteli módszert!
18. Milyen részekből áll egy modem? Hogyan lehet a működésmódját beállítani?

19. Soroljon fel néhány jellemzőt, amit a modemnek adott parancsokkal beállíthatunk!
20. Mi az MNP eljárás lényege? Mit jelentenek az egyes fokozatai?
21. Mit takar a karakter- illetve bitorientált átviteli eljárás fogalmak? Mi a szinkron és aszinkron átvitel lényege?
22. Melyek a digitális jelek kódolásánál figyelembe veendő legfontosabb szempontok?
23. Mi az NRZ, illetve RZ digitális kódolási módszer lényege?
24. Mi az NRZI, illetve az AMI digitális kódolási módszer lényege?
25. Mi az HDB3, illetve a PE digitális kódolási módszer lényege?
26. Mutassa be az ASCII kódrendszert? Hogyan lehet a karaktereit csoportosítani?
27. Mutassa be az RS232C soros adatátvitelt! Mi a DCE és DTE?
28. Milyen modemvezérlő jelekkel kommunikál egymással egy modem és egy számítógép?
29. Mi az a null-modem?
30. Milyen adatokkal jellemezhető egy RS232C soros adatátvitel?
31. Hogyan működik az áramhurokot felhasználó soros adatátvitel?
32. Mutassa be, hasonlítsa össze a Az RS-449, -422, -423, és 485-ös szabványokat!
33. Mutassa be az X.21 interfészt!
34. Mi az ISDN? Milyen szolgáltatásai vannak?
35. Mi az a bitső, hálózati végződés? Milyen referenciapontok vannak a különféle eszközök között?
36. Mutassa be a digitális telefonközpont (PBX) működési elvét!
37. Milyen részekből áll egy ISDN interfész? Milyen keretformátumot használ?
38. Mi az ATM? Mi az alap gondolata?
39. Mi az ADSL?
40. Mi a párhuzamos adatátvitel legfontosabb előnye?
41. Mutassa be a GPIB (IEC625) párhuzamos adatátvitelt!
42. Mi a mester-szolga és az adó-vevő szerepek közötti különbség I²C busz esetén?
43. Rajzolja fel és magyarázza el az I²C busz működését!
44. Hogyan történik egy bit átvitele az I²C buszon?
45. Hogyan történik egy bájt átvitele az I²C buszon?
46. Mi jelzi egy átvitel kezdetét és végét az I²C buszon?
47. Magyarázza el az egyvezetékes busz működését!
48. Mekkora lesz az adatátviteli sebesség bit/s-ban kifejezve egyvezetékes adatátvitel esetén?

3. ADATKAPCSOLATI PROTOKOLLOK

Az adatkapcsolati protokollok feladata egy összeállított keret átvitele két csomópont között. Elsőként foglalkozunk az üzenetszórásos adatátviteli közegek hozzáférési módszereivel.

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 közegeken.

KÖZEG-HOZZÁFÉRÉSI 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 keret 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 keret tartalmaz, vagy több keret tartalmaz (de ez már ütközés!)
- az állomások vagy képesek (carrier sense = vivőjel érzékelés) vagy nem képesek a csatorna foglaltságát figyelni.

Közeg-hozzáférési kérdések csak akkor merülnek fel, ha csupán egy csatornán osztoznak az üzenetváltók.



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

SZÁMÍTÓGÉP - HÁLÓZATOK

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: ilyen esetben egy időpontban mindig csak egy állomásnak van joga adatátvitelre, és ez a jog halad állomásról-állomásra. Ez a módszer az elvében biztosítja az ütközések elkerülését.

Központosított vezérlés: ilyenkor 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.

Az átviteli közeg hozzáférésének három fő módszere:

- véletlen vezérlés
- osztott vezérlés
- központosított vezérlés



Ezen belül több megoldás lehetséges, a legfontosabbakat a következő felosztásban foglaltuk össze:

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

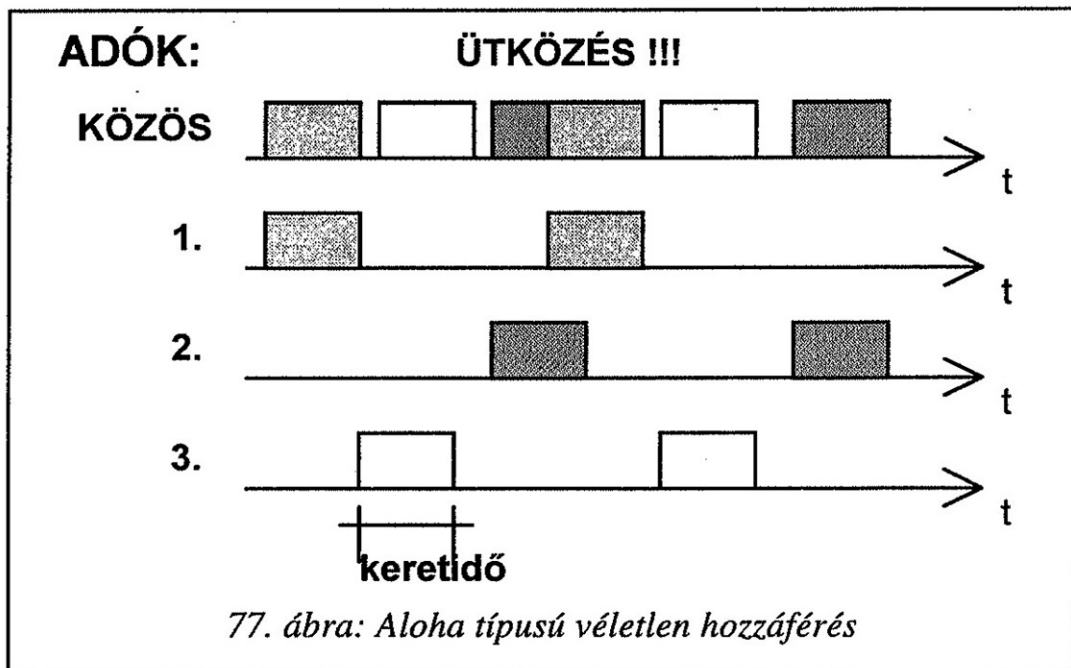
VÉLETLEN ÁTVITEL-VEZÉRLÉS

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 Hawaii 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 létrejövő átlagos N keretszám-nak 1-nél kisebbnek kell lennie: $0 < N < 1$. Kis terhelés esetén $N \sim 0$ és kevés ütközés történik, míg nagy terhelés $N \sim 1$ esetén sok ütközés.



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?

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).

ÜTKÖZÉST JELZŐ VIVŐÉRZÉKELÉSES TÖBBSZÖRÖS HOZZÁFÉRÉS (CSMA/CD)

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 behallgat a csatornába. Az állomás által küldött üzenet a 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. Az adás közben — mivel közben a csatornán lévő üzenetet veszi — el tudja dönteni, hogy az adott és a vett üzenetfolyam egyforma-e. Ha ezek különböznek, akkor azt jelenti, hogy valaki más is „beszél”, azaz a



Az előző Aloha módszerhez képest a változás az, hogy az állomás a küldés előtt először behallgat a közegebe.

küldött üzenet hibás, sérült. Ezt, ahogy már leírtuk, ütközésnek hívják, és ilyenkor az állomás megszakítja az üzenetküldést.

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 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 LAN-okkal foglalkozó fejezetben — mint az IEEE 802.3 szabvány — írunk róla.

Még egy módon lehet befolyásolni az ütközéseket. A fenti esetben amikor a csatorna szabad lett, azonnal megkísérelte 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, és 1-p valószínűséggel megvárja a következő időrés kezdetét. Ez a p-perzisztens CSMA protokoll.

RÉSELT GYŰRŰ (SLOTTED RING)

A gyűrűn felfűzött állomások rés-eknek 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.

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

Mivel önmagában a közeg nem biztosítja a résnek megfelelő lépkedéséhez szükséges késleltetést, ezért az állomásokon (és így a gyűrűn) a bitek átvitele léptetőregiszterek segítségével van lassítva.

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 figyel, hogy van-e olyan rés, amely a gyűrűben nem jut alaphelyzetbe, és ha ilyen van, egy idő múlva eltávolítja a gyűrűből.

REGISZTER BESZÚRÁSOS GYŰRŰ (REGISTER INSERTION RING)

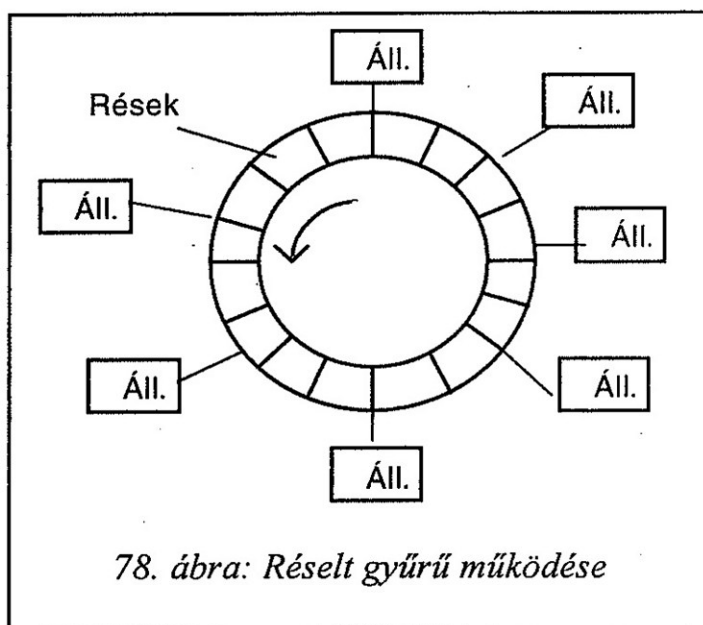
A gyűrű topológiájú hálózatoknál a másik alkalmazott eljárás a léptetőregiszter késleltető funkcióján túl, annak tárolási képességét is kihasználja. A hálózati illesztőben két regiszter: egy léptető- (shift-) és egy tároló- regiszter található.

A gyűrű indulásakor a mutató a léptető regiszter kezdő pozíciójára mutat. Ahogy jönnek a bitek a hálózatról, a pointert mindig bitenként balra lépteti, azaz a gyűrűben lévő biteket tárolja. Közben a keretben lévő címet a beérkezett bitekből megállapítja.

Ha nem az állomásnak szól, akkor a kapcsolón keresztül kezdi kiléptetni a biteket, miközben az újabb érkező bitek a mutató által jelölt helyre íródnak, amely a léptetés miatt mindig felszabadul. Ha a keret utolsó bitjei is beérkezett, akkor a maradékot még kilépteti és mutató ismét a kezdő pozícióba kerül.

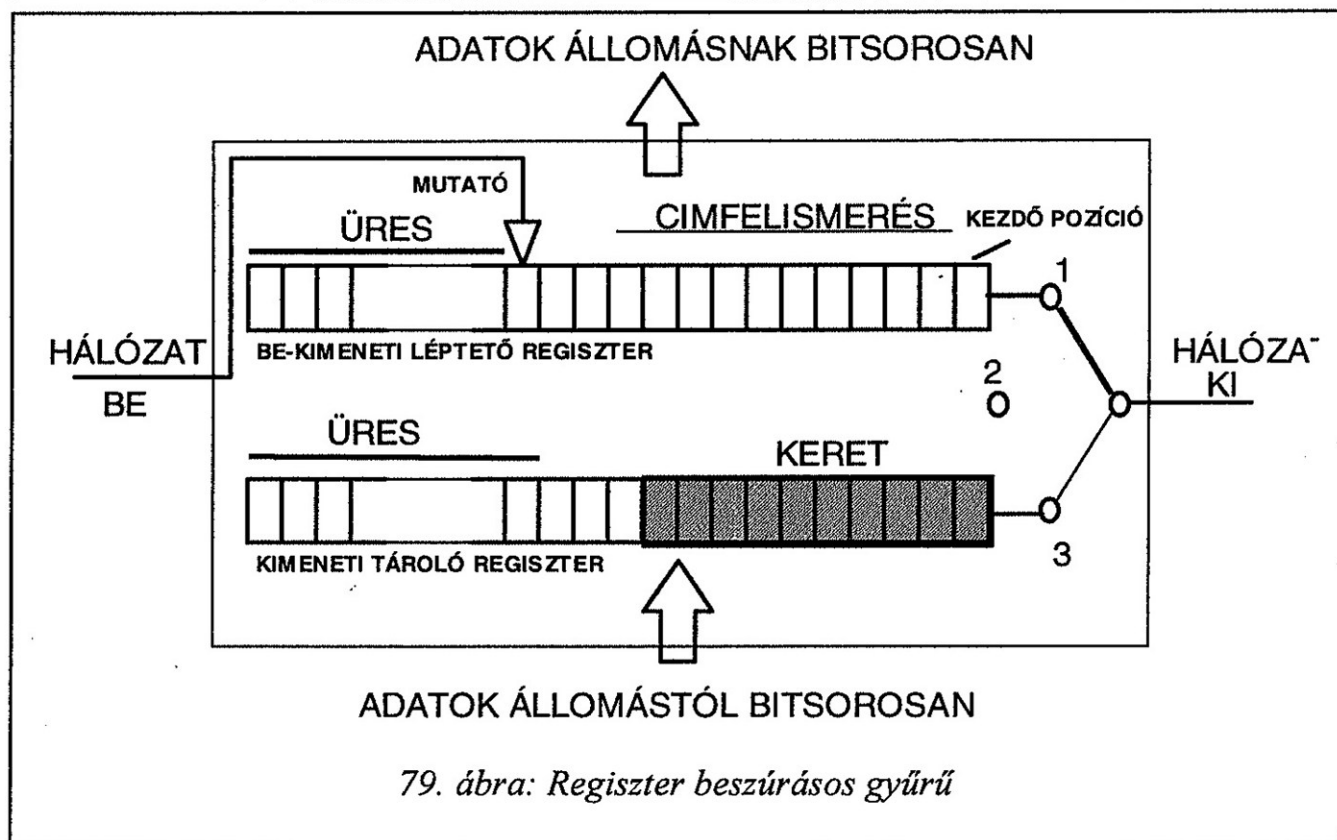
Ha a keret az állomásnak szól, akkor a kapcsoló 2-es pozícióba kerülve nem engedi a keret kijutását, azaz kivonja a keretet a gyűrűből.

Kivétel esetén az állomás által összeállított keret a KIMENETI TÁROLÓ REGISZTER-ben van. Kivétel csak akkor lehetséges, ha a az előzőleg vett, és továbbadandó keret utolsó bitjét is már kitolta a BE-KIMENETI LÉPTETŐ REGISZTER-ből a gyűrűre, és a regiszterben elegendő hely van a kimeneti keret fogadására. Csak ekkor kerül a kimeneti kapcsoló a 3-as pozícióba, és kerül a regiszter tartalma bitenként a gyűrűre, a bemenettel szinkronban. Az új bemenet eközben gyűlik a felső regiszterben. Ha a kimeneti tároló regiszter kiürült, a kimeneti kapcsoló ismét az 1-es helyzetbe billen, folytatva a vett bitek küldését.



78. ábra: Réselt gyűrű működése

A módszer előnye, hogy a gyűrű kisajátítást megakadályozza. Ha csak egy állomás aktív, akkor azonnal szinte állandóan adhat, ahogy ismét feltöltötte a kimeneti regiszterét. Ha azonban más állomás is használja a gyűrűt, akkor a keretének elküldése után valószínűleg nem küldhet újabbat, mert a be-kimeneti regiszterében nem lesz elég hely.



OSZTOTT ÁTVITEL-VEZÉRLÉS

Lényegében minden állomás a közeghez való vezérlés funkcióját is betölti, és ez a szerep váltakozva továbbadódik.

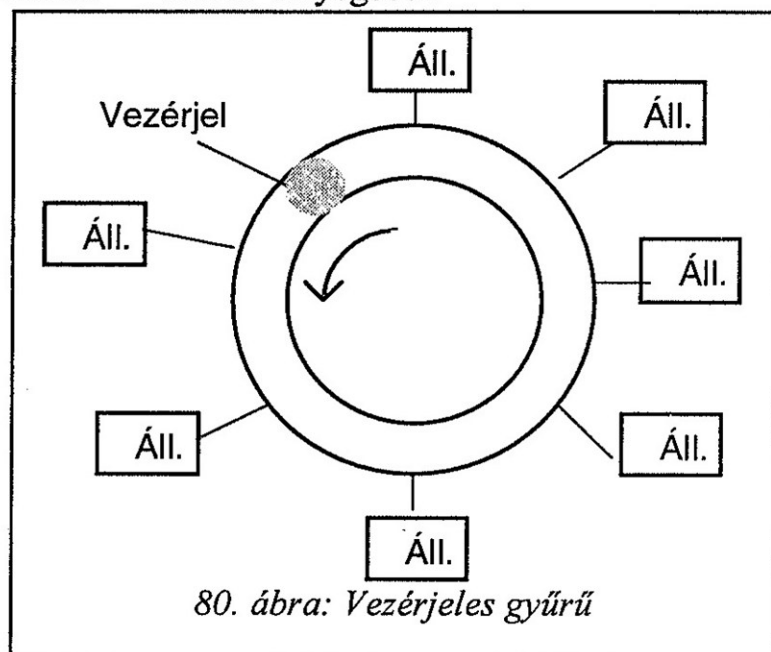
VEZÉRJELES GYŪRŰ (TOKEN RING)

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.

A vezérjel lényegében egy rövid üzenet, ami utal a gyűrű foglaltságára. Ha szabadot jelez, akkor a tokent vevő 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 vezérjel lényegében egy rövid üzenet, ami utal a gyűrű foglaltságára.



Az 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 szabadot jelző vezérjelet más állomás számára.



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.

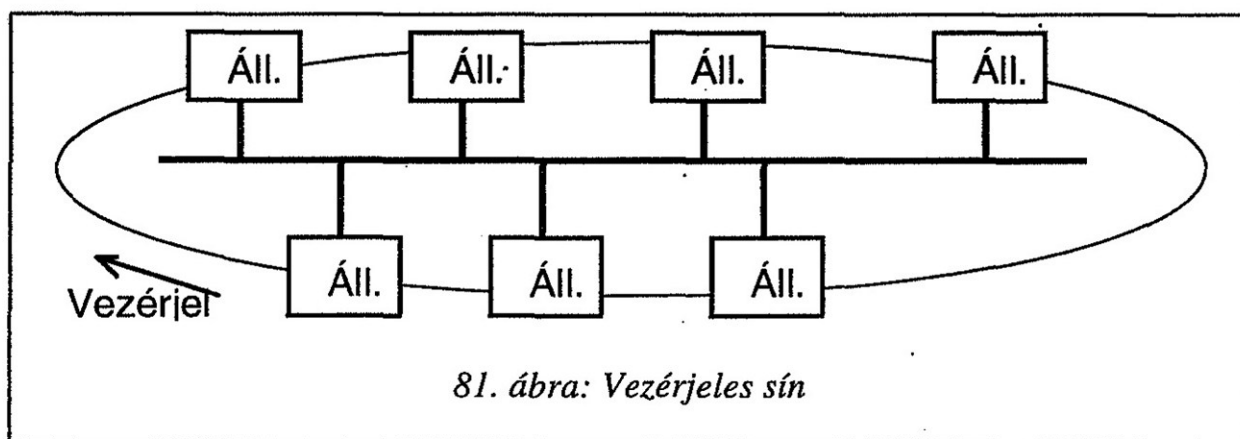
Az állomások között prioritás is kialakítható, azaz a nagyobb prioritású állomások az alacsonyabb szintű állomások előtt kaphatnak lehetőséget adataik továbbítására. Részletesebben a LAN-okkal foglalkozó fejezetben mint az IEEE 802.5 szabvány írunk róla.

VEZÉRJELES SÍN (TOKEN BUS — VEZÉRJEL BUSZ)

A vezérjel továbbítási eljárást két különféle topológiájú (busz illetve gyűrű) hálózati szabványban is használják. Busz topológiájú hálózat esetén vezérjel busz szabványról beszélünk.

Fontos megjegyezni, hogy bár az állomások logikai kapcsolata gyűrű, fizikailag mégis felfűzött busz topológiájú.

A vezérjel busz az átviteli közeget úgy vezérli, hogy az állomásról állomásra történő vezérjel (ún. token) továbbítása egy logikai gyűrűt képez.



Amikor egy állomás vette a vezérjelet, lehetőséget kap arra hogy adatblokkokat továbbítson a számára biztosított maximális időn belül.



Ha nincs adandó adatblokkja, akkor a token azonnal továbbadja. Részletesebben a LAN-okkal foglalkozó fejezetben mint az IEEE 802.4 szabvány írunk róla.

**ÜTKÖZÉST ELKERÜLŐ, VIVŐÉRZÉKELÉSES
TÖBBSZÖRÖS HOZZÁFÉRÉS (CSMA/CA)**

A módszer angol elnevezése: Carrier Sense Multiple Access with Collision Avoidance) =CSMA/CA. A véletlen közeghozzáférésekkel foglalkozó részben már a módszer alapgondolatát megismertük: Itt minden állomás adást figyelve „belehallgat” a csatornába. Az adás befejezése után minden állomás egy adott ideig vár, amit egy logikai listában elfoglalt helyük határoz meg. Ha ez alatt az idő alatt más állomás nem kezd adni, akkor elkezd az adást.

KÖZPONTOSÍTOTT ÁTVITELVEZÉRLÉS

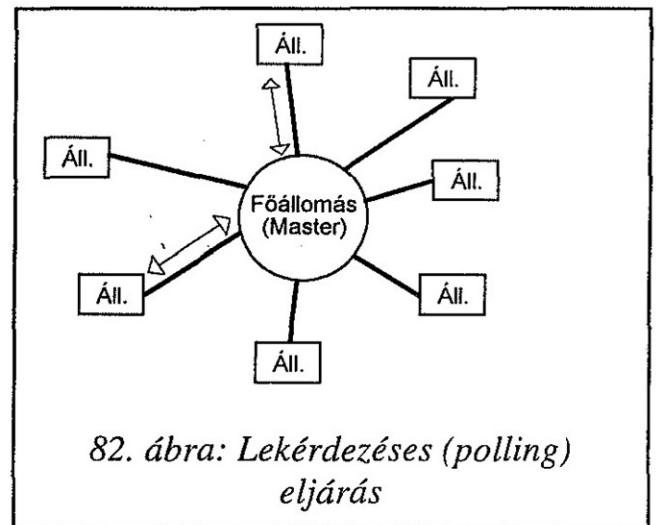
Ezeknél az eljárásoknál mindig van egy kitüntetett egység, amelynek feladata az egyes állomások hálózathoz való hozzáféréseinek a vezérlése.

Külön egység, a központ végzi a hozzáférésvezérlést.

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.

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



82. ábra: Lekérdezéses (polling) eljárás

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 mellékállomásokról 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.

A működési elv miatt elsősorban csillag kialakítású hálózatoknál használják.

Itt is igaz az, hogy az intelligens kapcsolóközpont meghibásodása az egész rendszer számára katasztrófát jelent.

VONALKAPCSOLÁSOS ELJÁRÁS

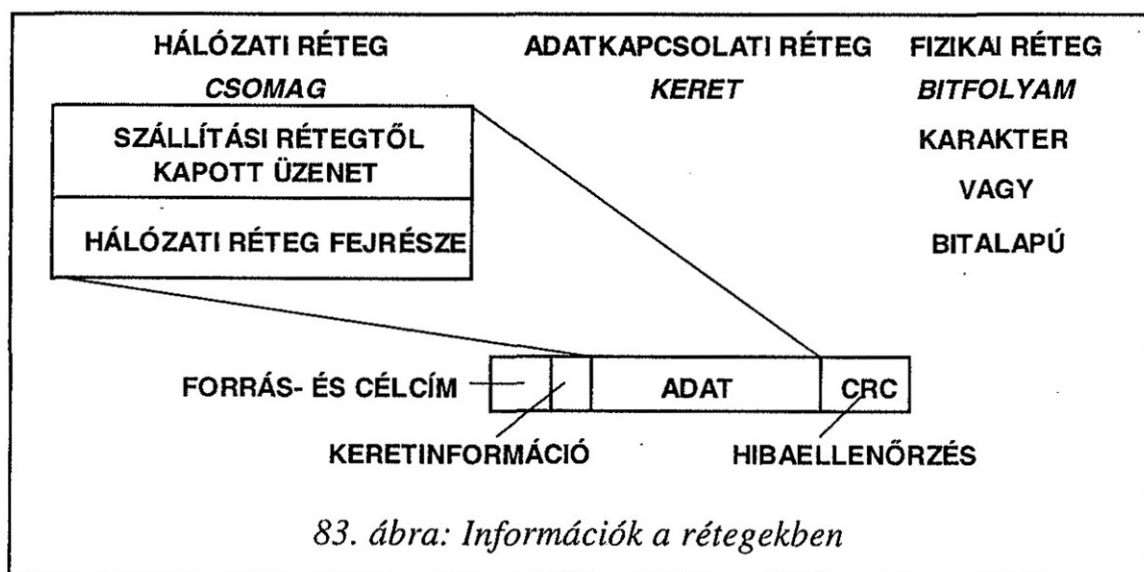
Az előbb ismertetett eljárásnál a főállomás fő funkciója a postás szerep volt. Mivel az elsődleges feladat a mellékállomások egymással való kommunikációja, ez más módon is megoldható. Ha lehetséges, akkor a két mellékállomást egy vonalon relék vagy elektronikus kapcsolók segítségével összekötjük, és 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.

IDŐSZTÁSOS 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 időszelvényben adhat. Ha nincs üzenete, akkor a szelet kihasználatlan marad.

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 szétválasztani (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.

A keretek átvitele két csomópont között első gondolatra egyszerűnek látszik, de az csak a látszat, hiszen az ADÓ és VEVŐ tulajdonságai, a keretek adatújtját 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 keretkezdet 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 bitsoportos átvitelt alkalmazni. A bitsoportok tetszőlegesen 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 biteit bitenként értelmezzük, bitorientált átvitelről beszélünk, ami általános információ átvitelekor rugalmasabb megoldás.

Nagyon sokszor viszünk át szöveges információt, ami indokolja a karakterorientált átvitel létét.

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

KERETEK KÉPZÉSE

A bitfolyam (illetve a ezt alkotó bitsoportok) keretké történésére több módszer használatos:

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.

Ezt a két módszert, a jellegükből adódóan csak karakterorientált átvitel esetén használhatjuk.

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 bitsoport 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:

ITTEZDLEVOLT

Az ADÓ adatkapcsolati réteg keretképzése és karakter beszúrása:

DLE STXITTEZDLE DLEVOLT DLE ETX

Az VEVŐ adatkapcsolati rétege leválasztja a kettőzött beszúrt karaktert:

DLE STXITTEZDLEVOLT DLE ETX

A VEVŐ hálózati rétegének átadott üzenet: **ITTEZDLEVOLT**

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 tartalmazó 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:

11111111110111

Az ADÓ adatkapcsolati réteg keretképzése és bitbeszúrása:

01111110 1111101111100111 01111110

Az VEVŐ adatkapcsolati rétege leválasztja a beszúrt biteket:

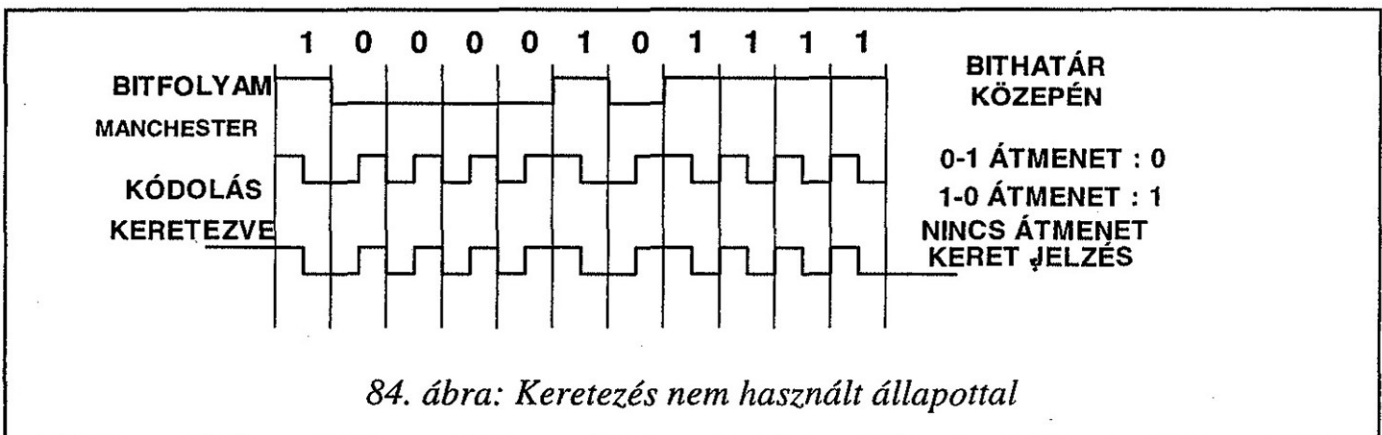
01111110 111111111110111 01111110

A VEVŐ hálózati rétegének átadott üzenet:

11111111110111

Ez a módszer már bitorientált átvitel esetén használható.

Speciális információ jelzésére használható még az RS232C soros szabvány brake jelzése is. Ez azt jelenti, hogy az eredetileg MARK állapotú vonalat nemcsak a START bit, és az azt követő bitsorozat átvitelének idejére tartjuk SPACE állapotban, hanem olyan hosszú ideig, hogy az egyértelműen külön állapotnak legyen tekinthető.



84. ábra: Keretezés nem használt állapottal

Ha a 0 és 1 bitek kódolásán kívül még létezik nem használt kód, ez a keretképzésre felhasználható. A **84. ábrán** az információt egy bit átvitelének közepén lévő átmenet (állapotváltás) hordozza. Mivel az

állapotváltás hiánya nem tartozik a kódoláshoz, ezért ez felhasználható a keret kezdetének és végének a jelzésére.

HIBAKEZELÉS

Az adatátvitel és a kommunikáció fontos kérdése az átvitel során fellépő hibák kezelése. 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 zajok időtartamából következően lehetnek egyedi és csoportos bithibák. A gyakoribb esetben a hibák fennállási ideje általában egy bit átviteli idejének a többszöröse, ezért ezek a hibák csoportosan, hibacsomók formájában jelentkeznek. Mivel az adatátvitel blokkos (keretes) formában történik, ezért az eredmény egy-egy blokk tönkremenése.

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 bits csoportot alkotó eredetileg m bites kódot r darab bittel egészítik ki, így a redundáns bits csoport (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.

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

Egy teljes kódszó rendszer Hamming távolságát úgy határozzuk meg, hogy képezzük minden lehetséges kódszó pár közötti Hamming távolságot, és ezek közül a legkisebb lesz az eredmény.

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 ellentétére változik, vagy egyszerre kettő azonos módon változik).



A vonalakon fellépő hibákat különböző fizikai jelenségek okozzák: termikus zaj, a vonalakat kapcsoló berendezések impulzus zaja, a légköri jelenségek (villámlás) okozta zajok.

A hamming távolság arról ad információt, hogy két kódszó mennyire különbözik egymástól, mekkora a különbségük.

(Hamming egy angol matematikus volt.)

ASCII kódrendszerben a Hamming távolság: 1

A paritásbittel történő kiegészítés során olyan kódszavakat generálunk, amelyek hossza eggyel nagyobb, mint az eredeti kódszó. Ezért a Hamming távolságuk 2, és egyszeres bithibák kimutatására alkalmasak.

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)

85. ábra Hibakezelés

A hibajelzés valószínűsége olyan módon növelhető, hogy a blokkot $n \cdot k$ elemű mátrixnak 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 inkább egy másik módszert használnak, Ez a hibavédelmi eljárás úgy működik, hogy egy keretnyi adatot egy előre meghatározott bitsorozattal „elosztunk”, és a „maradékot” a keret részeként továbbítjuk.

A vevő oldalon ugyanezt az osztást végezzük el, és ha ez a keret részeként átküldött maradékkal egyezik, akkor hibátlannak fogadjuk el a keretet. A módszer részletezésétől helyhiány miatt eltekintünk.

Az osztási elvből következik, hogy az osztandó bármelyik elemének a megváltoztatása az osztás eredményét befolyásolja.

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. $G(x) = x^4 + x + 1 \rightarrow 10011$

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. $T(x)/G(x) = [x^r M(x) + R(x)]/G(x) = Q(x) + R(x)/G(x) + R(x)/G(x)$

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

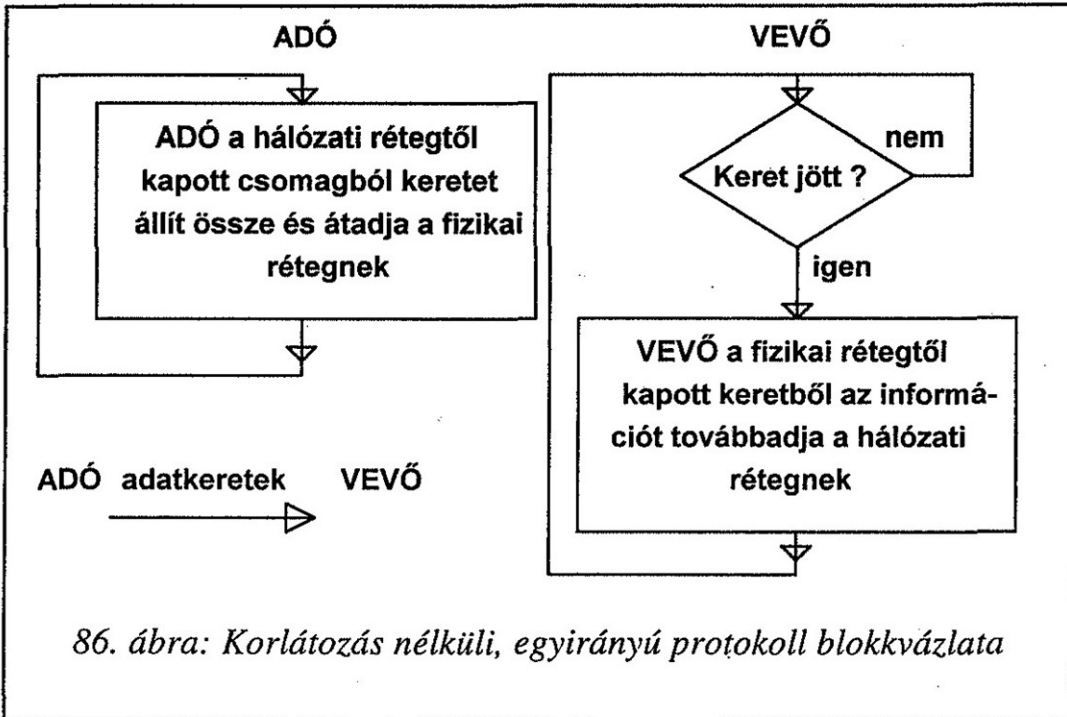


ADATKAPCSOLATI PROTOKOLLOK

A következőkben a röviden az ADÓ és VEVŐ közti keretek segítségével megvalósított adatkapcsolati protokollokat mutatjuk be. Az adatkapcsolati réteg tördeli keretökké a bitfolyamot, és látja el fejrészsel, amelyet a VEVŐ oldali adatkapcsolati réteg távolít el, és állítja vissza a bitfolyamot.

A hálózati és adatkapcsolati réteg teljes elkülönítése érdekében a hálózati rétegnek a keretekre tördelésről, annak fejléccel történő ellátásáról nincs információja.

KORLÁTOZÁS NÉLKÜLI, EGYIRÁNYÚ (SZIMPLEX) PROTOKOLL



A gyakorlatban ilyen esetnek tekinthető az a kapcsolat, amelyben egy gyors VEVŐ kapcsolódik egy lassú ADÓ-hoz, és nincs adatátviteli hiba (pl. azért, mert rövid a vezeték...)

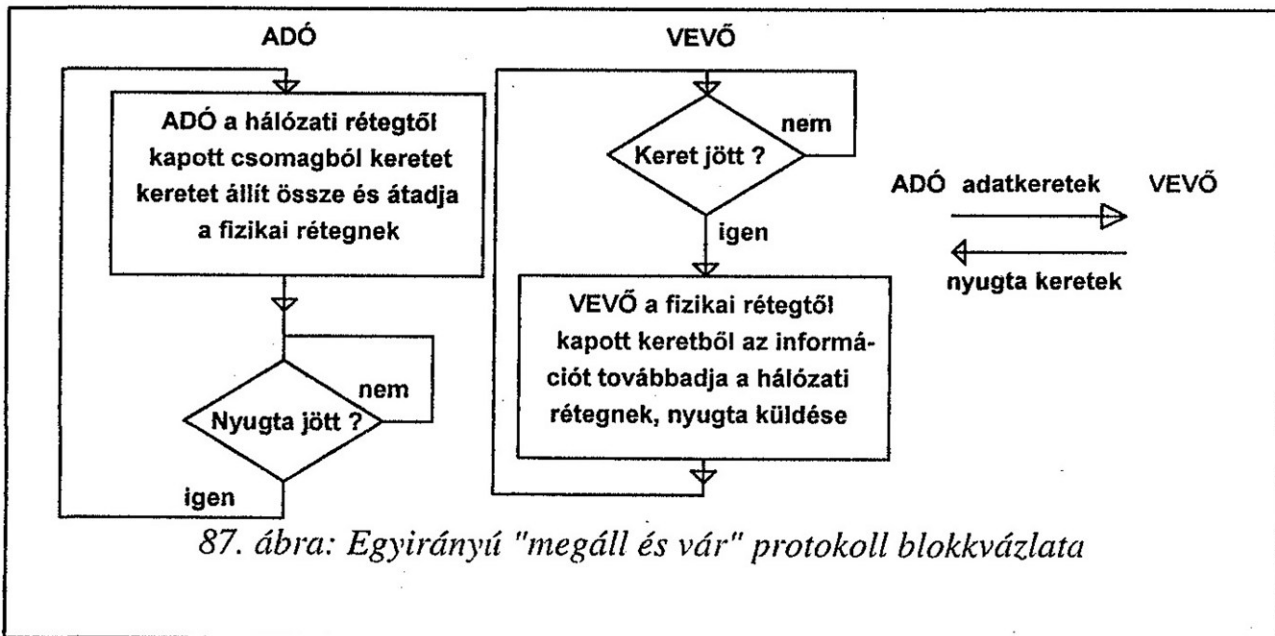
Pl. egér és a számítógép kapcsolata

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.



Látható, hogy bár az adatforgalom szimplex, azért a keretek már különböző időpontokban, két irányban áramlanak, ezért a fizikai réteg vonatkozásában fél-duplex csatorna kialakítást igényel.

EGYIRÁNYÚ „MEGÁLL ÉS VÁR” PROTOKOLL



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

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 „megállás 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 elküldeni a nem nyugtázott keretet, amit a VEVŐ ismételtten 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 újradaadásra kerültektől.

EGYIRÁNYÚ ÖSSZETETT PROTOKOLL

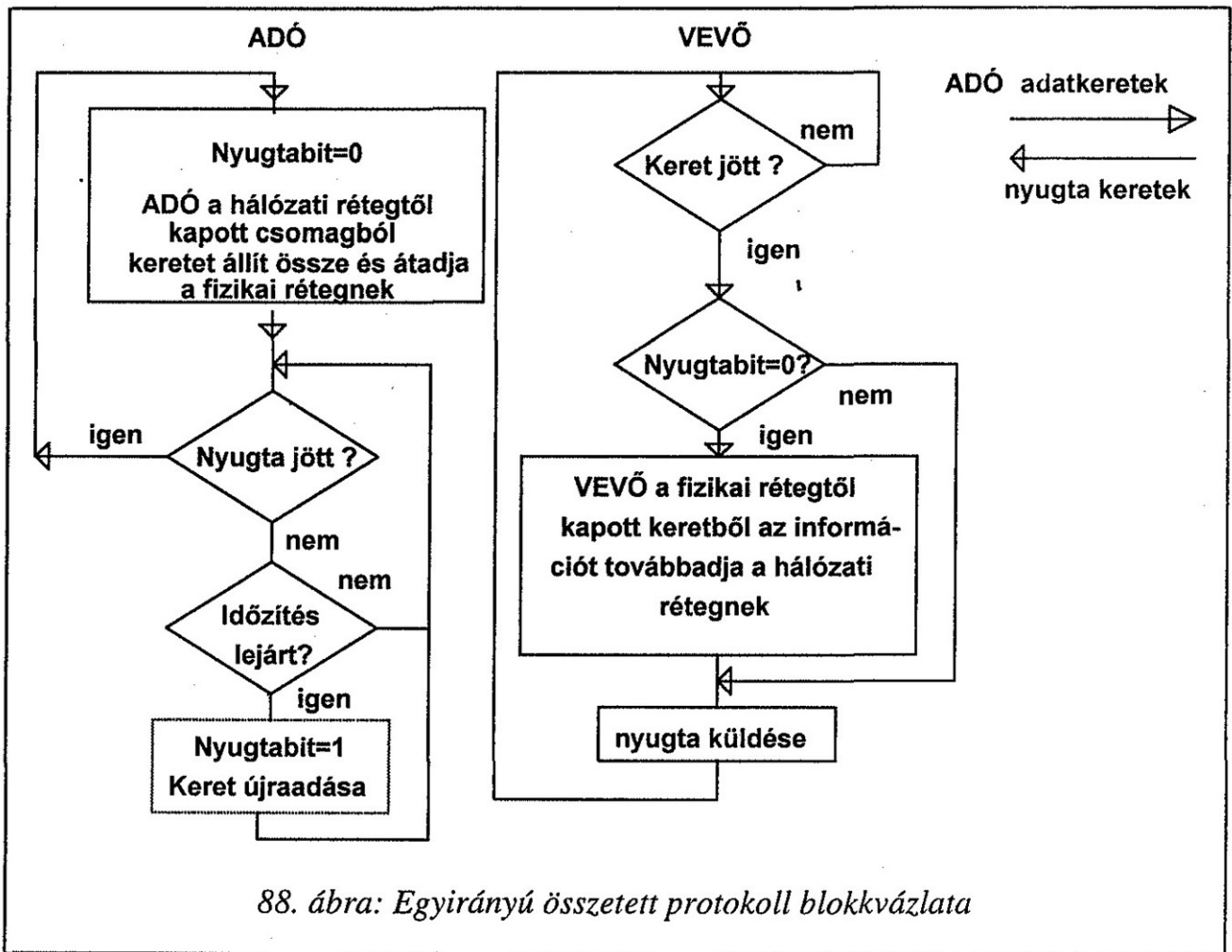
Ennek egyszerű megoldása az, hogy az ADÓ egy számot helyez el minden elküldendő keret fejrészébe, és ezáltal a VEVŐ eldöntheti, hogy először adott, vagy ismételt keretről van-e szó. Mivel a keretek és a nyugták egymás után vannak, ezért elegendő 1 bittel jelezni az újraküldés tényét.

Nézzük: a k-adik keretre (amelynek újraküldési bitje 0 volt, jelezve az első küldést) a VEVŐ nyugtát küld, de az elvesz. Az ADÓ mivel a k-adik keretet elküldte, de nem nyugtázták (legalábbis azt hiszi), egy adott időzítés lejáta után ismételtten 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 vissza az elveszett helyett, de a keretet eldobja.



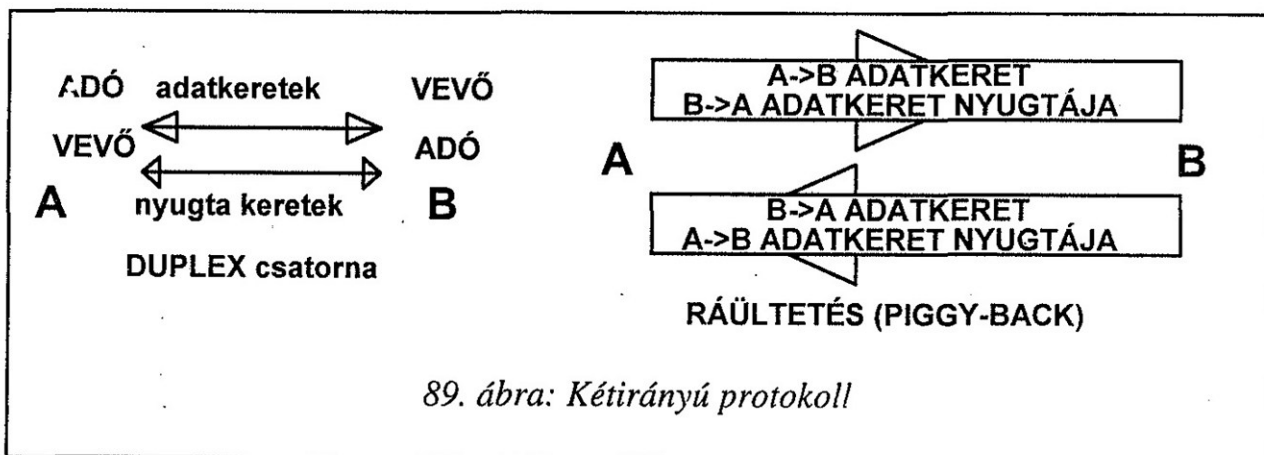
A megoldás lehetne két különálló, ellentétes irányú adatcsatorna használata, de az a nyugtázás miatt valójában négy információs utat jelentene, ahol a nyugtacsatornák kihasználása kicsi lenne.



KÉTIRÁNYÚ PROTOKOLLOK

Az előző esetekben az adatátvitel egyirányú volt, bár az utolsó két esetben 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.

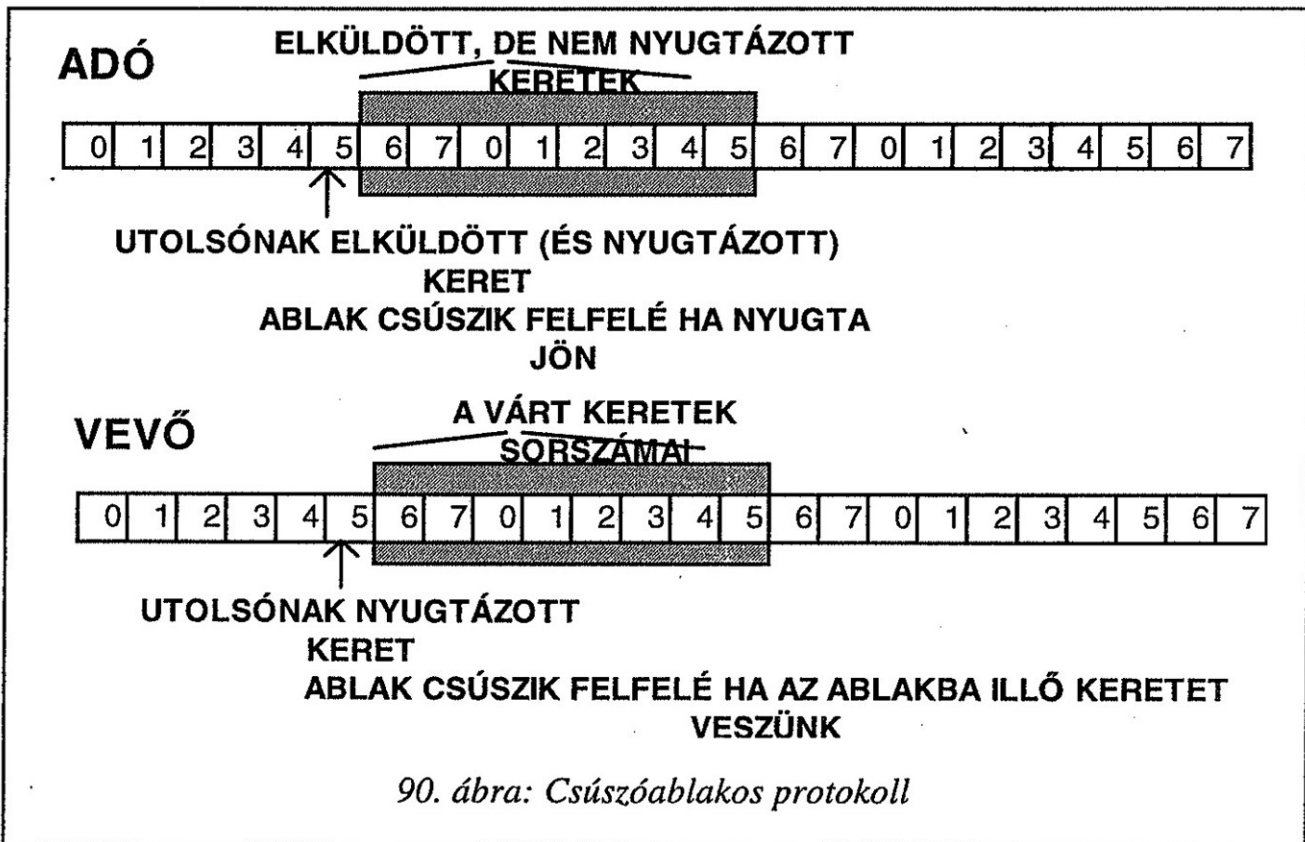
Jobb megoldás, ha mindkét irány számára ugyanazt a csatornát használjuk, hiszen az adatkereteket a nyugtakeretektől a keret fejrészében elhelyezett jelző meg tudja különböztetni, és ez a keret vételekor azonosítható.



SZÁMÍTÓGÉP - HÁLÓZATOK

Egy egyszerű megoldással az átvendő 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.

Az eddigiekben feltételeztük hogy a csatornán mindig egy adatkeret, majd rá válaszul egy nyugtakeret halad. A valóságban a csatorna jobb kihasználását teszi lehetővé, ha megengedjük, hogy a csatornán több keret is tartózkodhat. Az ezt lehetővé tévő eljárásokat **csúszóablakos (sliding window) vagy forgóablakos protokolloknak** nevezik. A könyvben az első megnevezést fogjuk használni.



A protokollban minden egyes kimenő keret egy 0-max (az n: 0-7) közötti sorszámot kap. A lényeg az, hogy a sorban elküldendő keretek sorszámaiból egy aktualizált listát tart fenn az ADÓ. A listában szereplő sorszámú keretek az adási ablakba (sending window) esnek. Az ADÓ adási ablakában az elküldött, de még nem nyugtázott keretek vannak. 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 az 1-es sorszámú keretre kap nyugtát, ez azt jelenti, hogy nyugtázott a 6,7,0,1 keret.

Ezenkívül az ADÓ az ablakban lévő minden keret elküldésétől eltelt időt nyilván tartja, és ha ez egy értéknél (timeout) nagyobb, akkor újra adja.

Mivel a kereteket esetleg újra kell adni, ezért az ablakban lévő kereteket ismételt adásra készen memória-pufferekben kell tartani.

A VEVŐ egy vételi ablakot (receiving 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.

Ha a keretek átviteli ideje hosszú, például műholdas átvitel esetén, akkor nem jó az a megoldás, hogy újabb keretet, csak az előző nyugtázása után indítunk. A megoldás az, hogy az ADÓ nem 1 hanem k darab keretet küld el nyugtázás nélkül. Az n . keret elküldése után kezdi várni a nyugtákat és folytatni az $k+1, \dots$ keretek küldését. Az ilyen esetben a csúszóablak mérete k kell hogy legyen. Ezt a megoldást csővonal-nak (**pipelining**) hívják, utalva arra a szemléletes képre, hogy a keretek egy csőbe haladnak, sorban egymás után.

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. Ez a stratégia 1 méretű vételi ablaknak felel meg. Zajos vonalak esetén ez a megoldás nagymértékben csökkenti az adatátviteli sebességet a sok újraküldés miatt.

SZELEKTÍV ISMÉTLŐ PROTOKOLL

A másik, csővonal esetén használható általános hibakezelési eljárást szelektív ismétlésnek (selective repeat) hívják, és működése már az előzőek és az elnevezése alapján már kitalálható: ennél a hibás keretet követő összes jó keret tárolásra kerül.

Ennél a protokollnál, mind az ADÓ mind a VEVŐ fenntart ablakot, a keretsorszámoknak. Az ADÓ ablaka 0-tól $sorszmax$ -ig növekszik. A VEVŐ ablaka rögzített méretű, a megfelelő működés érdekében 1-nél nagyobb.

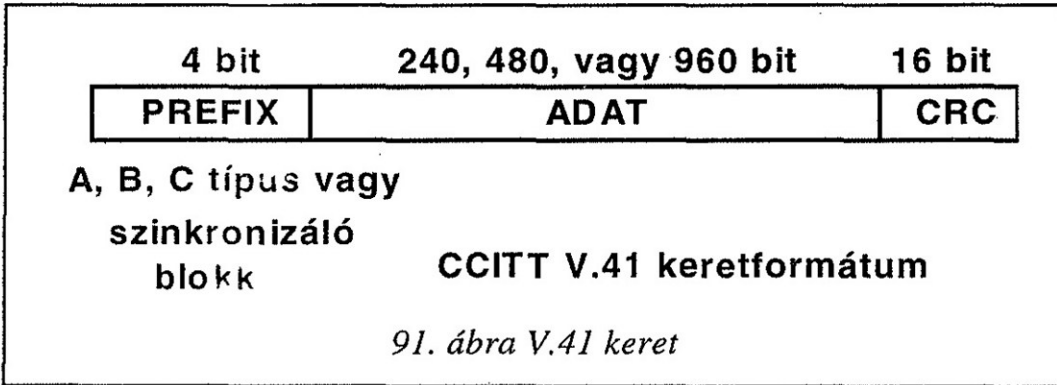
Amikor az ADÓ felfedezi, hogy volt hibás keret (nem kap nyugtát róla), akkor csak a hibást küldi újra.

A következőkben az előbbieket illusztrálására néhány konkrét gyakorlati megoldást mutatunk be.

CCITT V.41-ES AJÁNLÁSA

Az első nemzetközileg elfogadott ajánlás információcsere bitszinkron eljárásra. Az adó a vevőnek rögzített hosszúságú (260, 500, vagy 980 bites) blokkokat küld.

A 4 bites prefix határozza meg a blokk típusát, a fennmaradó 12 kombináció használata opcionális.



Az adó a szinkron blokk után sorban mindig A, B, és C típusú blokkot küld úgy, hogy az előző blokkot tárolja. Az elküldött blokk végén megvizsgálja a másik, nyugtacsatornán vevő által küldött választ.

A nyugtacsatorna csak egy bitet használ, lehet egy önálló vezetéken lévő feszültség szinttel reprezentálni.

Helyes vétel esetén ennek a szintje nulla (ezt állítja be a vevő). Ha a szint egy, akkor ismétlést kér a vevő. Az adó ekkor az éppen küldött blokkot elrontja, a CRC utolsó bitjének invertálásával, és újra küldi a nem nyugtázott blokkot.

A három blokk típus sorrendben való küldése biztosítja az adatvesztés elkerülését, és megakadályozza egy azonos blokk dupla vételét.

Előnyei:

- Szinkron átvitel alkalmazása.
- Ciklikus hibavédelmet használ.
- Számozott információs blokkok alkalmazása.
- Kód-független átvitel lehetősége.

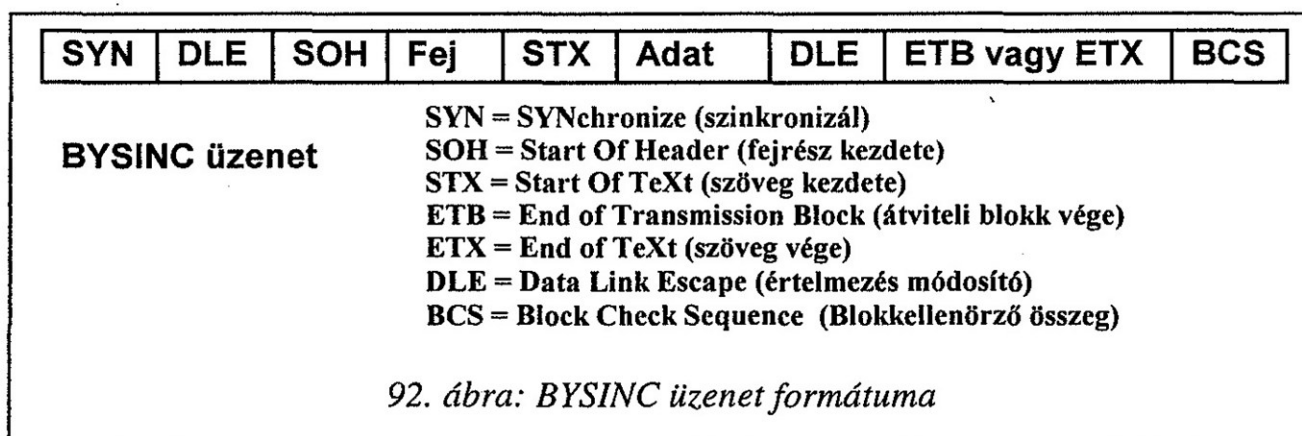
Hátrányai:

- Fix blokkhosszúság.
- A párbeszédes üzem bonyolult megvalósíthatósága.
- Többpontos üzem nem realizálható, a terminál címzés nem megoldott.
- Külön csatornán megvalósított egyidejű nyugtázás.

IBM BISYNC (BINARY SYNCHRONOUS COMMUNICATION)

[11] Karakterorientált szinkron eljárás, mely az ISO 1745 ajánlason alapszik. Ez is mint a többi ismert karakterorientált eljárás meglehetősen szabadsággal kezeli a vezérlő karaktereket, így ezek az eljárások általában nem teljes mértékben kompatibilisek egymással. Széles körben használják távoli terminálok lekérdezésére, valamint egyéb alkalmazásokra is. **Fél-duplex vonalához fejlesztették ki, és egyaránt működik többpontos és kétpontos típusú állomás kapcsolatok esetén is.** A BISYNC üzenetformátumát a **92. ábrán** láthatjuk.

A fejléc 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 megcímezését nem a fejlécben lévő cím, hanem egy külön vezérlőüzenet végzi. Minden blokk végén egy, vagy két karakternyi blokkellenőrző sorozat (Block Check Sequence = BCS) is átvitelre kerül.



A **93. ábrán** követhetjük végig egy üzenetváltás ütemét. 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 jelűt.

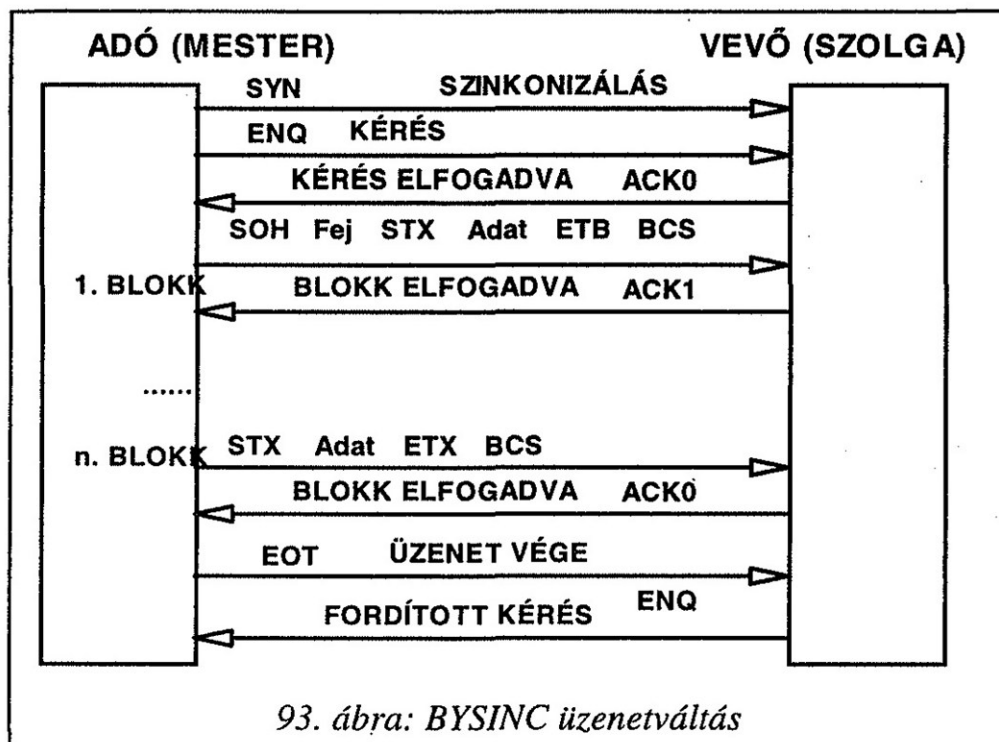
A kétféle nyugta biztosítja a keretvesztés elleni védelmet, mert felváltva kell jönniük.

Az IBM BYSINC terminológiában szokás még az adást kezdeményező és végrehajtó állomást mester-nek, míg a másik állomást, aki az üzeneteket veszi szolga-nak nevezni.

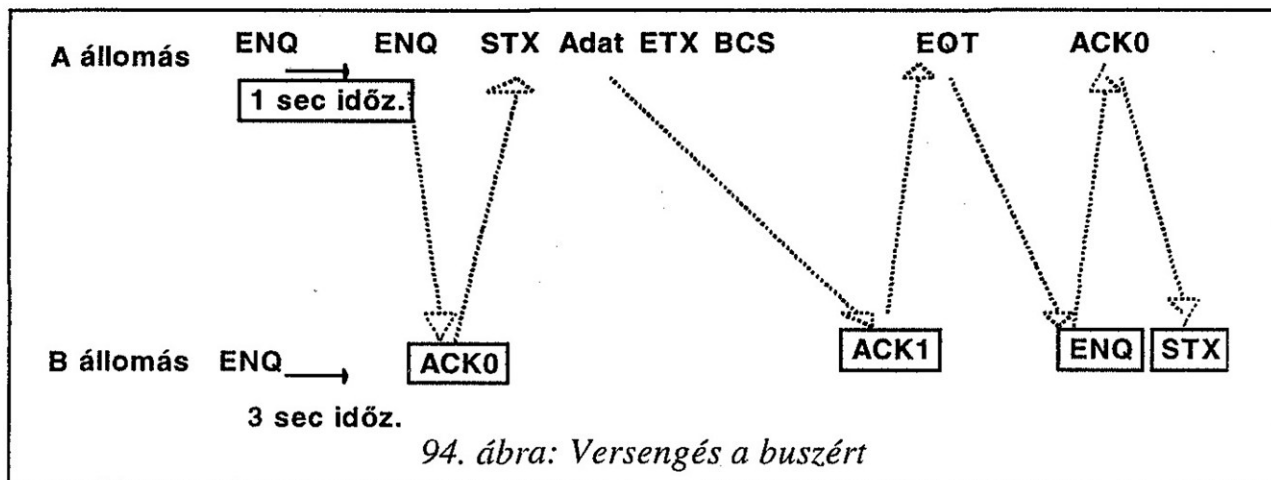
Ha egy vonalon két állomás található, és mindegyik adni szeretne a másiknak, mindkettő ENQ karaktert küldene el a kommunikációra való felszólításra és versenyhelyzet alakul ki. A közösen használt csatorna használatának megoldására, vagyis az arbitráció-ra a módszer

Ha az A és B jelű állomás egyszerre próbál adni, az ütközés után eltérő idő múlva próbálja meg ismét (természetesen számítógépterminál relációban az előbbinek kisebb a kivárása).

az, hogy az állomásokhoz különböző időzítéseket (time-out) rendelnek.

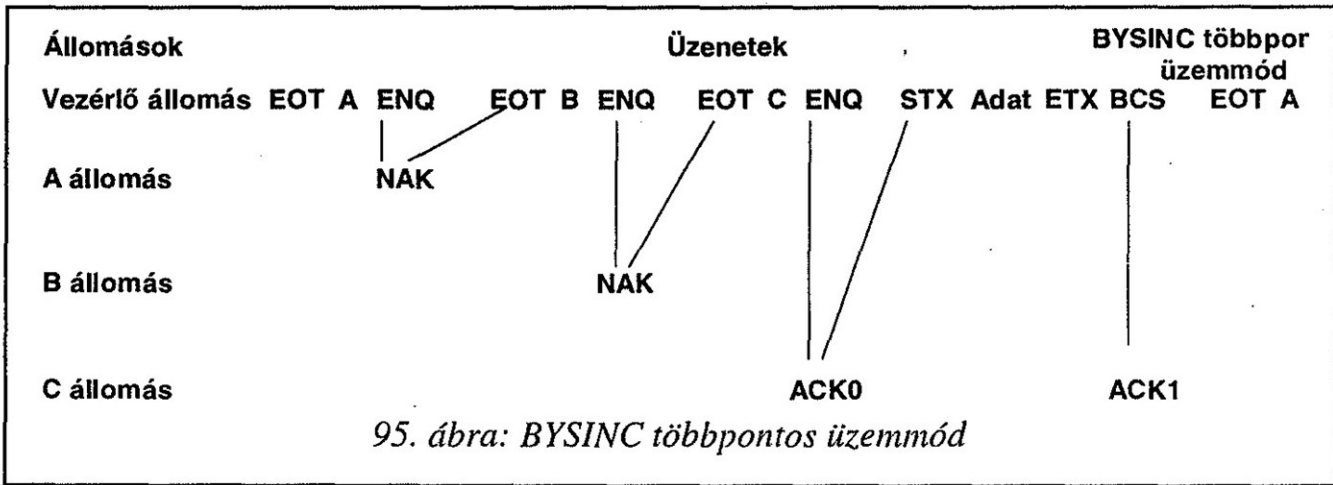


Mikor a „nyerő” állomás befejezi az üzenetét, akkor kísérheti meg a másik a kommunikációt.



Mivel a gyakorlatban gyakran használt az olyan elrendezés, hogy több állomást kötnék egy közös vonalra, vizsgáljuk meg a többpontos üzemmód kialakítását.

Ilyenkor a csatorna közös használatát felügyelni kell, és ezt a vezérlőállomás végzi, amely a legtöbbször egy számítógép. Ez lekérdezővel (polling) kérdezi a többi állomást, hogy van-e üzenetük. A lekérdező üzenet az ENQ karakterből és az állomás címéből áll. A lekérdezővel a kiválasztott állomás, adóvá válik, és elküldi üzenetét a vezérlőállomásnak, majd az adásának befejezését az EOT karakter küldésével tudatja. Ezt a vezérlőállomás véve újabb lekérdezőt indíthat el.



Az ISO 1745 előírás lehetővé teszi két alárendelt állomás egymás közötti kommunikációját is: a lekérdezővel mesterré vált állomás (adó) egy másik alárendelt állomást választ ki, amely ezzel a kiválasztással szolgálva (vevővé) válik. Ez az ún. multi-master üzemmód.

Két állomás közötti üzenetváltás eredetileg csak a vezérlőállomáson keresztül lehetséges, azaz a BYSINC ún. centralizált többpontos vagy más néven multidrop rendszer.

HDLC (HIGH LEVEL DATA LINK CONTROL)

Bitorientált eljárás. Hasonló elveken alapuló eljárás az **IBM SDLC (Synchronous Data Link Control)**. Az ISO kiegészítve **HDLC** néven fogadta el.

Eredetileg az IBM az SDLC-t szerette volna de-jure szabványként elfogadtatni.

Az adatkapcsolat szintű asszimetrikus működési módhoz fejlesztették ki, ahol egy mesterállomás (**főállomás (primary station)**) vezérli a szolgálállomásokat (**mellékállomás (secondary station)**).

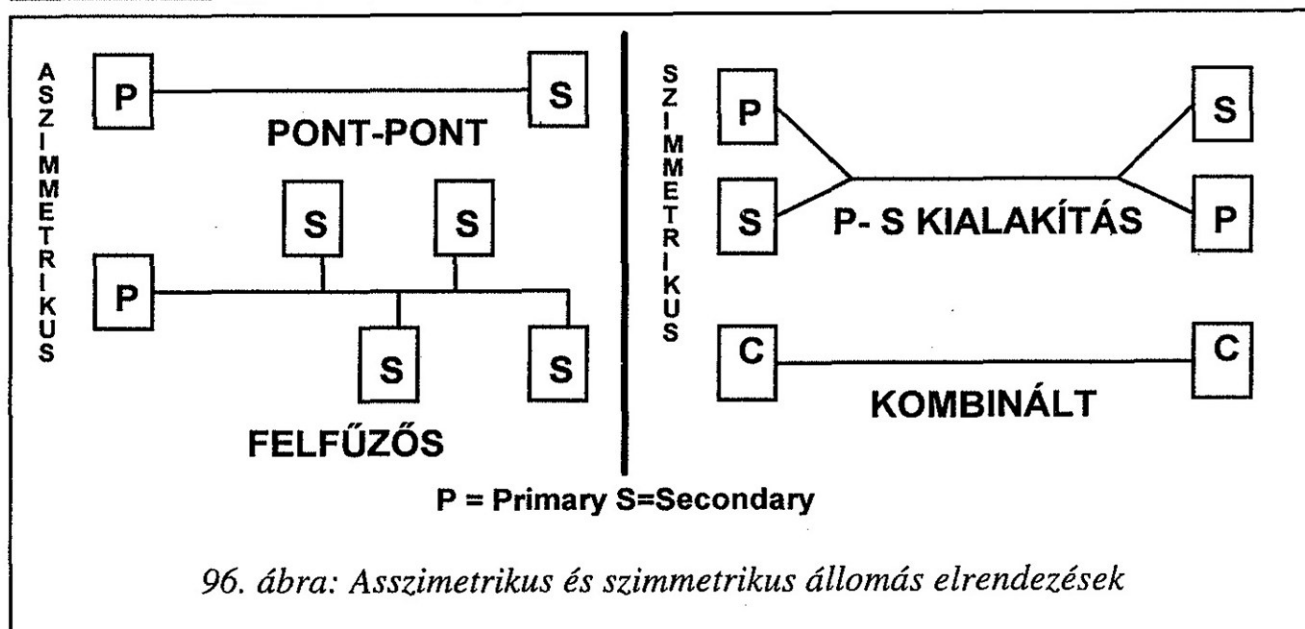
HDLC terminológia:
Főállomás-mellékállomás

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.

Ma a HDLC protokollt széles körben használják ipari rendszerekben is.

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 mellékállomást tartalmazó egységet **kombinált állomásnak** hívják.

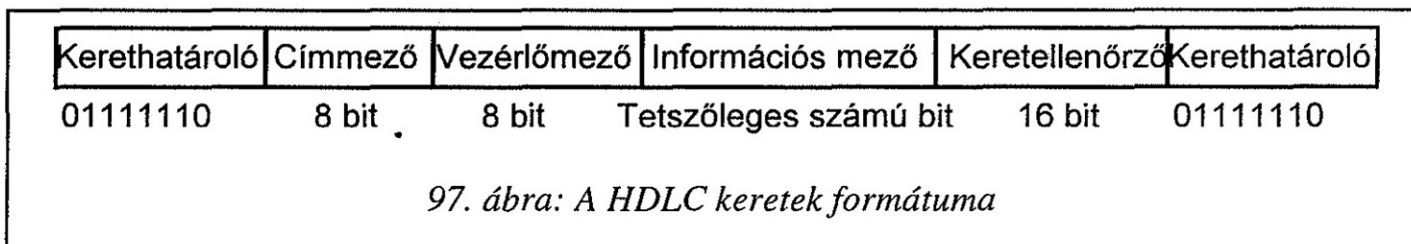
A CCITT később az X.25 szabvány részeként LAP (Link Access Procedure — kapcsolat elérési eljárás) néven majd később LAPB (LAP - Balanced) néven vette át. Ez utóbbi esetben már az állomások egyenrangúsága is lehetséges.



Legfontosabb előnyök, (összehasonlítva a karakteralapú eljárásokkal) 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 keretből 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.

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

3. ADATKAPCSOLATI PROTOKOLLOK

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 kerettípusokat a következő táblázatban foglaltuk össze:

Osztály	Megnevezés	Rövidítés	Funkció
C = parancs, R = válasz			
Információ	Információ	I	C/R
Felügyeleti			
	Vételkész	RR	C/R
	Nem vételkész	RNR	C/R
	Elutasítás	REJ	C/R
	Szelektív elutasítás	SREJ	C/R
Számozatlan			
	Normál válasz- üzemmód (kiterj.)	SNRM(E)	C
	Aszinkron válasz-üzemmód (kiterj.)	SARM(E)	C
	Aszinkron szimmetrikus üzemmód (kiterj.)	SABM(E)	C
	Szétkapcsolás	DISC	C
	Beindító üzemmód beállítás	SIM	C
	Beindító üzemmód kérés	RIM	R
	Sorszámozatlan lekérdezés	UP	C
	Reset	RSET	C
	Sorszámozatlan információ	UI	C/R
	Információcsere-azonosítás	XID	C/R
	Sorszámozatlan nyugtázás	UA	R
	Szétkapcsolt üzemmód	DM	R
	Szétkapcsolás kérés	RD	R
	Keretelutasítás	FRMR	C/R
	Parancselutasítás	CMDR	R

Három típusú keret 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.

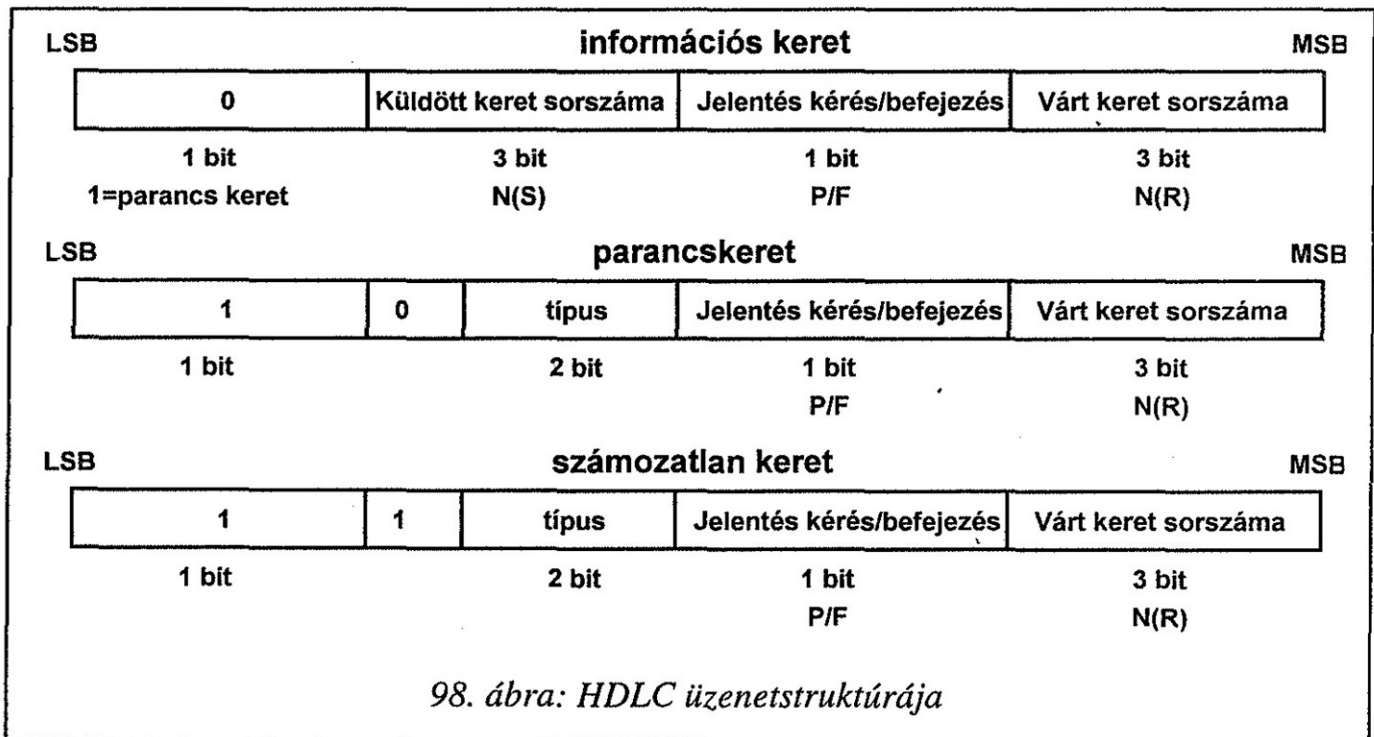
Ha az átviteli közeg jellemzői szükségessé teszik (pl. műholdas átvitel), lehetséges a keretszám hét bitre történő kiterjesztése is (128 hosszúságú csúszóablak).

SZÁMÍTÓGÉP - HÁLÓZATOK

A parancskeret típusát a 2 bites típusmező adja meg:

Típus	Megnevezés	Magyarázat
0	RECEIVE READY (nyugta)	akkor használják, ha nincs lehetőség ráültetett nyugtára
1	REJECT (negatív nyugta)	átviteli hiba jelzése, a várt keretsorszám a hibás (újraadandó) keretet jelzi
2	RECEIVE NOT READY	az eddigi kereteket nyugtázza, de a küldő nem küldhet újabb keretet
3	SELECTIVE REJECT	csak a kijelölt keretet kell újraküldeni

A kerettípusok felépítése:



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

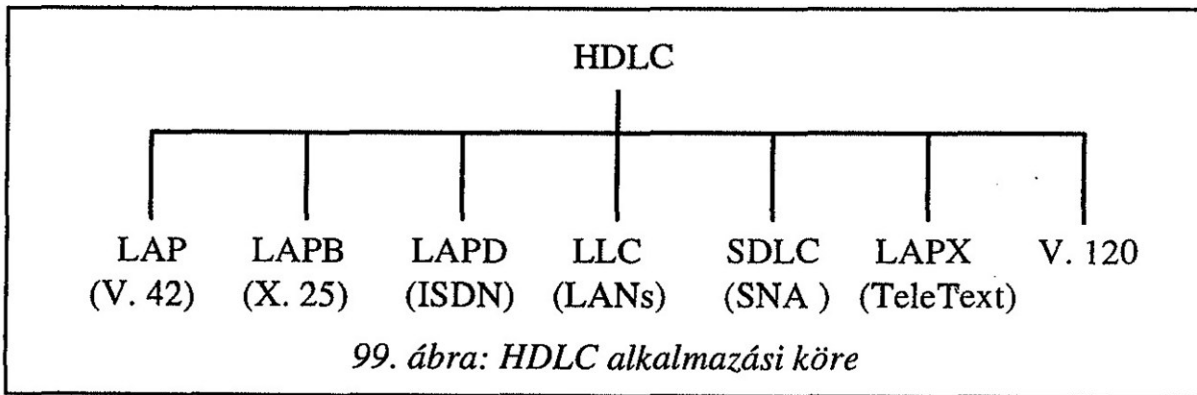


akkor küldhet, amikor akar, nem kell a főállomás felszólítására várakoznia. Jól használható pont-pont szimmetrikus, és üzenetszórásos (felfűzött) elrendezések esetén.

- **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), ilyenkor a vezérlő mező 16 bit hosszúságú, a megnövelt ablaksorszámok miatt. Számozatlan keretek esetén ilyenkor a második nyolc bit kihasználatlan.

A **99. ábrán** a HDLC jelentőségét jól igazolja számos adatátviteli szabványban való felhasználása.



ELLENŐRZŐ KÉRDÉSEK

1. Milyen feltételezésekkel tárgyalhatók a közeg hozzáférési módszerek?
2. Milyen hozzáférési módszerek lehetségesek a közeg elérési módja alapján?
3. Mi a véletlen, az osztott és a központosított átvitelvezérlés lényege?
4. Mi az az ütközés?
5. Hogyan működik az ALOHA protokoll?
6. Ismertesse a CSMA/CD módszert!
7. Hogyan működik a réselt gyűrű?
8. Hogyan működik a regiszter beszúrásos gyűrű?
9. Hogyan működik a vezérjeles gyűrű?
10. Hogyan működik a vezérjeles sín?
11. Ismertesse a CSMA/CA módszert!
12. Mikor és miért előnyös a lekérdezéses, (polling) eljárás?



13. Foglalja össze a vonalkapcsolásos és TDMA eljárás lényegét!
14. Mi a csomag és keret közötti különbség?
15. Milyen módszereket ismer egy bitfolyam keretké tördelésére?
16. Ismertesse a karakter-beszúrási módszer lényegét!
17. Ismertesse a bitbeszúrási módszer lényegét!
18. Mi az a Hamming távolság?
19. Mi az ECC és CRC?
20. Ismertesse az egyirányú "megáll és vár" protokollt!
21. Ismertesse az egyirányú összetett protokollt!
22. Miért előnyös a kétirányú protokollok használata? Mi az a piggy-back technika?
23. Mi a csúszóablakos protokoll lényege?
24. Ismertesse a visszalépés n-el technikájú protokollt!
25. Ismertesse a szelektív ismétlő protokollt!
26. Mutassa be a CCITT V.41 ajánlást!
27. Mutassa be az IBM BISYNC protokollt! Hogyan épül fel egy BYSINC üzenet?
28. Hogyan történik egy BYSINC üzenetváltás?
29. Mutassa be a HDLC protokollt! Milyen állomáselrendezések lehetségesek?
30. Hogyan épül fel és milyen részekből áll egy HDLC keret?
31. Milyen kerettípusokat használ a HDLC protokoll?
32. Milyen működési módjai vannak a HDLC protokollnak?

4. HÁLÓZATI RÉTEG

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 hálózatszerkezési 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. **Összeköttetés mentes hálózatokban az átvitel csomagok segítségével történik** (ezeket tárolják és továbbítják a hálózati csomópontok). Az áramló csomagokat datagramoknak nevezik.

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.

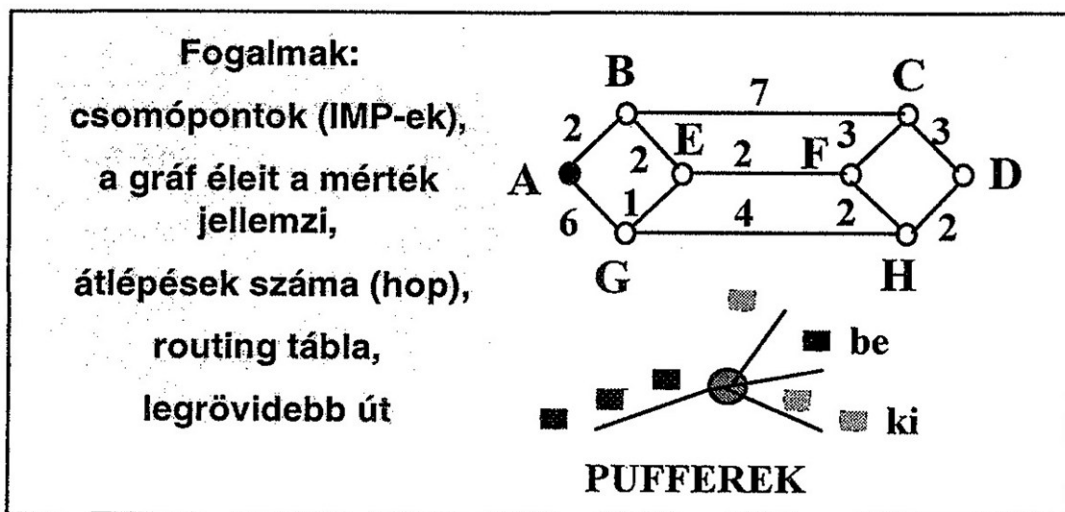
A virtuális áramkörök kialakításához minden csomópontnak fenn kell tartani egy olyan táblázatot, amely bejegyzései a rajta keresztül haladó éppen használt virtuális áramkörök jellemzőit (honnan jött—hova megy) tartalmazzák, és az azonosításukra egy sorszámot használnak. Minden hálózaton keresztülhaladó csomagnak tartalmaznia kell az

A hálózati réteg segítségével valósul meg az üzenetek eljuttatása több hálózaton keresztül.

Ha a forrás és a cél eltérő típusú hálózatokban vannak, a réteg feladata a hálózatok közti különbségből adódó problémák megoldása.



Fontos! A csatornákból felépített virtuális kapcsolat szintén csomagokat továbbít, de azok útvonalát a kapcsolat létesítésekor meghatározzuk.



Elméletileg a hálózatok a gráfelmélet segítségével tárgyalhatók, ahol a csomópontok az egyes IMP-k, és a csomópontokat összekötő élek az összekötő csatornák, amelyeket az átvitelre jellemző mértékekkel jellemezzük.

általára használt virtuális áramkör sorszámát.

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 hálózatban, 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 küldésért a küldő IMP-n futó program a felelős.

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 — 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 azokat a csomagokat kell újra adni, ami éppen továbbítás alatt volt. A következőkben, egy táblázatban [1] hasonlítjuk össze a két módszert:

Azonban olyan rendszerekben, amelyekben tranzakciókat dolgoznak fel pl. (hitelkártya kódellenőrzés), a kapcsolat felépítésének majd lebontásának időtartama olyan időtöbbletet jelent, amiért nem érdemes ezt az összeköttetési módot használni.

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 meghal
Torlódásvezérlés	Nehéz megoldani	Könnyű, ha elegendő puffer van
Összetettség	A szállítási rétegben	A hálózati rétegben
Alkalmas	Összeköttetés-alapú és összeköttetés mentes szolgálathoz is.	Összeköttetés-alapú szolgálathoz

FORGALOMIRÁNYÍTÁS

A forgalomirányítás (routing) feladata a csomagok hatékony (gyors) eljuttatása az egyik csomópontból a másikba, illetve a csomagok útjának a kijelölése a forrástól a célállomásig.

A hálózatot célszerű gráfként modellezni, ahol a csomópontok a csomagtovábbító IMP-k, és a csomópontokat összekötő élek az IMP-k 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 kialakít 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, a kerülő utak választékának kialakítása, az irányítási

A forgalomirányítási szemléletünket nagyon jól segíti az olyan analógia, ahol a hálózatot a közúti hálózat, míg a csomagokat az autók képviselik. A csomópontok pedig természetesen az útkereszteződések.

A 7. Fejezetben a TCP/IP protokollal kapcsolatban a forgalomirányításra még visszatérünk.



Egy másik ilyen szempontból egyszerű elrendezés a két irányú kommunikáció miatt duplán kialakított kettős gyűrű, hiszen csomópontból csak két irányba lehet elküldeni a csomagokat, bár a két lehetséges út közül az egyik általában rövidebb a másiknál.

Ezért vagy minden csomópont egy routing táblát tartalmaz, amiben az összes többire vonatkozó távolság be van jegyezve, vagy a csomópontok számozási rendszere olyan, hogy a címe alapján a távolság meghatározható.

Egy gyűrű esetén egyirányú pont-pont kapcsolat van, tehát a forgalomirányítás a másik pontba való küldésre egyszerűsödik.

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.



forgalomirányítás:

- adaptív (alkalmazkodó)
- determinisztikus (előre eldöntött)

Az említetteken kívül bevezethető még egy további forgalomirányítás-típus is, amelyet deltairányításnak neveznek. Ennél az eljárásnál a központi irányító egység munkáját a forgalomirányítási döntésekhez kizárólag abban az esetben használják fel, ha ezek a helyi információkra nem alapozhatók.

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, vagy a vonalhasználat költségeit.

Az objektív mérték megállapításához lehet olyan tesztek futtatni az adott szakaszokon, amely magadja az átlagos sorbaállási és átviteli késleltetési 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, ahol a csomópontok az egyes IMP-k, és a csomópontokat összekötő éleket jellemezzük az előbb említett mértékekkel. A feladat a gráf két csomópontja közötti olyan élekből álló útvonal meghatározása (shortest path), amelyre az érintett élek mértékeinek összege minimális. A megoldási módszer Dijkstra-tól (1959) származik [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.

A **véletlen forgalomirányító eljárás** alapján működő rendszerben a továbbítandó csomagot a csomópont egy véletlenszám 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.

Az **elárasztásos forgalomirányító eljárás** sem igényel semmi ismeretet a hálózatról. A csomópontok, mikor egy csomagot továbbítanak, a bejövő csomagot minden vonalra kiküldenek, kivéve ahonnan érkezett. 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.

ADAPTÍV ALGORITMUSOK

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 meglévő információjuk már időszerűtlen.



Érhető, hogy katonai alkalmazások esetén előtérbe kerülhet a módszer, mert erősen sérült hálózatban (sok csomópontot kilőnek) is nagy a valószínűsége egy üzenet célba jutásának. Nagy a megbízhatósága!

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 meg 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 érkezhett 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 az lehetett oka, 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 szimuláció 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 tartalékkapacitással 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.

A közúthálózatban az autóvezető a forgalmi információhoz rádióon jut hozzá; ideálisan szervezett úthálózatban, amelyben fejlett forgalmi információs rendszer működik, az autós ki tudja kerülni az akadályokat, mert kerülő utakat választhat.

Ez azért lehetséges, mert a forgalmi információkat külön rendszeren gyorsabban továbbítják, mint ahogyan maguk a járművek haladnak.

Az irányító központ veszi a rádióon érkező forgalmi jelentéseket, amelyeket a megfigyelőpontokról küldenek, és ezek alapján tanácsokat ad a közlekedők részére.

Még az is előfordulhat ennél a módszernél, hogy több csomópont egyszerre fedez fel egy gyengén terhelt hálózatrészt, és ezért valamennyi ide tereli a forgalmat, és abban erős torlódást okoz.

KÖZPONTI ADAPTÍV FORGALOMIRÁNYÍTÁS

Az eddig tárgyalt forgalomirányító algoritmusok vagy a helyileg rendelkezésre álló, vagy a szomszédos csomópontok között cserélt információt használják.

A **központosított adaptív forgalomirányításban** minden egyes csomópont helyzetjelentést állít össze, és abban a folyó sorhosszakot, a hálózat elemeinek meghibásodásait stb. elküldi a hálózat forgalomirányító központjába (RCC = Routing Control Center). A központ ezek alapján átfogó képet alakít ki a hálózatról, és valamennyi forgalmi áramlat részére meg tudja határozni a legkedvezőbb útvonalat. Ezeket a legjobb utakat a hálózat csomópontjai forgalomirányítási táblák formájában kapják meg.

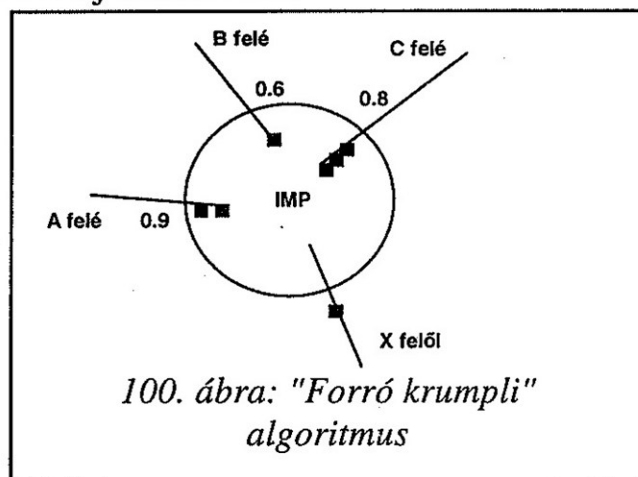
A központnak szóló helyzetjelentéseket és a csomópontoknak szóló új irányítási táblákat szabályos időközönként (szinkron üzemmódban) vagy csak jelentős változás hatására (aszinkron üzemmódban) küldik. Ha a szinkron üzemmódot választják, akkor az irányító algoritmus működtetése érdekében, a hálózatban áramoltatott vezérlő információ fantasztikus mennyiségűvé válhat. Különösen, ha a hálózat maga nagy, akkor a túlzott mértékű irányítási funkció jelentős többletterhelést okoz. Aszinkron üzemmódban 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 meg 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.

ELSZIGETELT 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 100. 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 a 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 felejtani”, 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.

Az algoritmus fő célkitűzése az adatforgalom részére a legkisebb késleltetéssel járó útvonalak keresése. E célból minden egyes csomópontban egy táblázatot hozunk létre, amely 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, mielőtt a csomagok célba értek, a becsült késleltetési időket felváltják a hálózatban ténylegesen mért továbbítási idők.

Az eredeti algoritmus szerint a késleltetési táblák adatait a szomszédos csomópontok rendszeresen megküldik egymásnak. Amikor a késleltetési táblákat megküldték, a csomópontok áttérnek a késéseket



újrászámító fázisba, amelyben a saját sorhosszaikat és a szomszédos csomópontok által küldött késleltetési értékeket figyelembe veszik.

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, és a lefoglalt kapacitás még a nagyobb sebességű vonalak esetén is észlelhető — bár kisebb — mértékű. A továbbított információról kimutatható, hogy az átvitt késleltetési táblák igen gyakran ugyanazt vagy majdnem ugyanazt az információt tartalmazzák, mint az őket megelőzők.

A táblák ilyen, szinkron 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. A késleltetési táblák újrászámítására csak akkor kerül sor, ha jelentősebb helyi változás történt, vagy ha módosított késleltetési tábla érkezik valamelyik szomszédos csomóponttól.

TORLÓDÁSVEZÉRLÉS



Azt hihetnénk, hogy ha a vonalak és csomópontok kapacitása elegendő az adatforgalom lebonyolításához, akkor a szabad információáramlás minden esetben garantálható. (A közúti forgalomban is így van?) A tényleges helyzet azonban más. Előfordul, hogy a rendeltetési helyen a csomagoknak a hálózattól 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 ezeket a

korlátozó intézkedéseket, ameddig a normális, korlátozás mentes üzem, ismét vissza nem állítható.

Ha egyes hálózatrészek túltelítődnek, akkor a csomagok mozgása lehetetlenné válhat. Azok a várakozási sorok, amelyeknek ezeket 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 egyszer s mindenkorra 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 teljesítménycsökkenés jelentkezhethet oly módon is, hogy a hálózat átbecsátóképessége (throughput) lecsökkent, anélkül, hogy a hálózat terhelését csökkentenénk, vagy pedig abban, hogy a hálózaton áthaladó csomagok késleltetése megnőtt. A teljesítménycsökkenés ezen jellegzetes tünetei többnyire együtt lépnek fel.

A torlódás lehet helyi jellegű, amikor a jelenség a hálózatnak csak bizonyos részét érinti, vagy súlyosabb, mikor az egész hálózatra kihat. A torlódás szélsőséges esetben olyan is lehet, hogy a forgalom egészen vagy csaknem egészen megbénul, amikor a hálózat egyáltalán nem vagy csak kevés adatot kézbesít a rendeltetésre és fogad el a forrástól. Nem lehet kérdéses, hogy ez olyan végzetes helyzet az adatátviteli hálózat számára, amelynek bekövetkezését bármi áron el kell kerülni.

A közúti forgalom viselkedése még közelebb áll a csomagkapcsolt hálózatok viselkedéséhez. A közutak hálózatot alkotnak, amelyben szállítási csatornák, utak, keresztezések stb. találhatóak. A forgalom az úthálózaton át a nagyszámú forrás-rendeltetés pár között járműfolyamok formájában áramlik. E folyamatok 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 jelenség jól illusztrálható a közúti körforgalomban lejátszódó hasonló események példájával.

Ha az elsőbbségi szabály a körforgalomba belépő forgalmat részesíti előnyben, akkor torlódás léphet fel. A forgalom csak akkor indulhat meg újra, ha a szabályokat megváltoztatjuk.

A csomagkapcsolt hálózatokban a helytelen pufferelosztás és a rossz prioritási szabályok hasonló befulladásokat okozhatnak.

A torlódás olyan állapot, amely a legtöbb szállítmányozó rendszerben előfordul.

Például a folyóban úszó farönkök akadálytalanul sodródhatnak mindaddig, amíg az egyes darabok mozgása nincs hatással a többiek előrehaladására.

Ha a rönkök számát eddig a szintig növeljük, akkor ezzel együtt az átbecsátott mennyiség is növekszik, de ezzel elérkeztünk egy olyan ponthoz, amelynél a rönkök már akadályozzák egymás mozgását, és a teljes átbecsátott 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-okban 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 puffer-foglalás. 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 általában az adatcsomagok általában ráültetett nyugtákat is tartalmaznak, ezért eldobásuk nem célszerű. Érdekes 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 árásszanak 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.

Stratégiák a torlódás elkerülésére:

- Pufferek foglalása
- Csomageldobás módszere
- Izometrikus torlódás-vezérlés
- Lefojtó csomagok használata.



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 puffert, de nem tudja mert a másik azt jelzi, hogy foglalt. Másik irányban is azonos a szituáció. Ezt az esetet **hívják közvetlen tárol és továbbít holtpontnak**.

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]

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.

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 (user) oldal		DCE(node)oldal
Packet Level	----- (virtuális kapcsolat)	Packet Level
Frame Level	----- (virtuális kapcsolat)	Frame Level
Fizikai Szint	----- tényleges kapcsolat	Fizikai Szint

A hálózatok, ahol alkalmazzák, csomagkapcsolású hálózatok. 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 bárhonnán fogadjon.

Az X.25 ajánlásban definiált hálózati szintek, és elnevezéseik a következők:

Physical Level (OSI 1. szint) X.21 vagy X.21bis (V.24) a fizikai interface, csatlakozók, feszültség-szintek, stb.

Link Level (OSI 2. szint), X.25 terminológia szerint "Frame Level" HDLC, LAPB a Frame szinten alkalmazott eljárások definíciója.

Network Level (OSI 3. szint), X.25 terminológia szerint "Packet Level" Virtuális hívás procedúra, adatátvitel módja, stb. a Packet szinten alkalmazott eljárások definíciója.

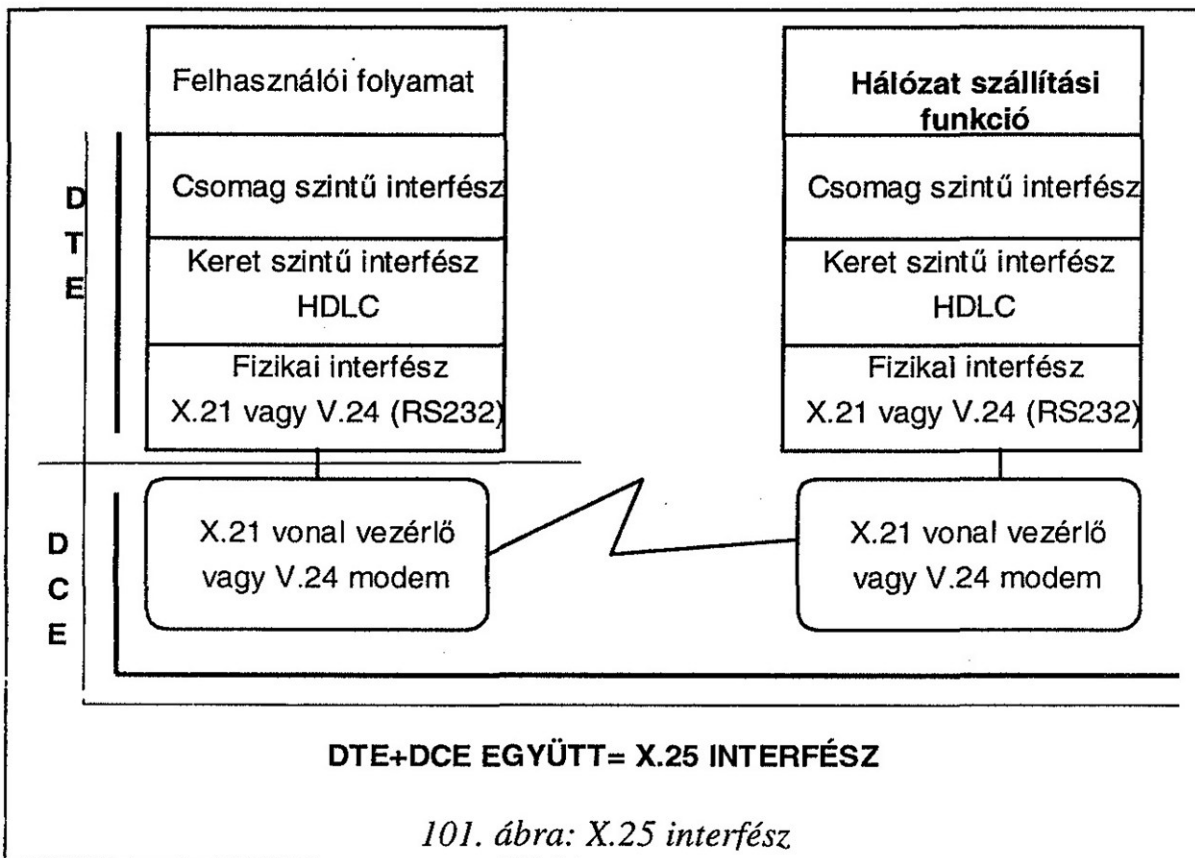
4. HÁLÓZATI RÉTEG

- 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.

A PAD valójában egy célszámítógép, amely egyszerű aszinkron soros vonalon keresztül képes egy terminállal kapcsolatba lépni

Pl. PC soros port - és az onnan érkező karakterekből X.25 ajánlásnak megfelelően adatsomagokat képezni és az X.25 hálózatba továbbítani, majd a visszaérkező adatokat karakterekre bontva a "buta" terminálhoz továbbítani.

Fontos szolgáltatás a nem csomagkapcsolt hálózatokkal való illesztést biztosító a csomagösszeállítás-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.



Az X.25 három protokollsintet 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

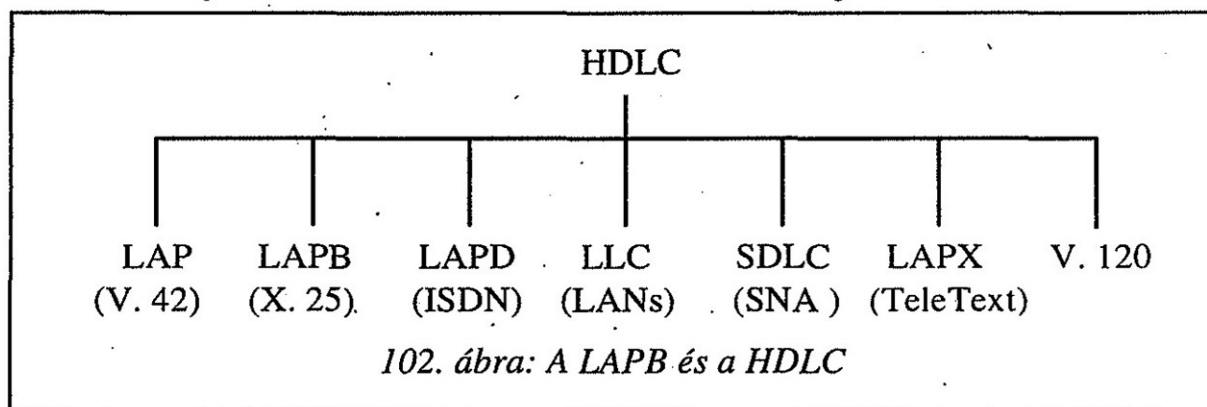
modemes összeköttetés V.24 ajánlása használható, amit X.21bis szabvány néven is emlegetnek.

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 Protocol-Balanced), ami egyenrangú állomásokat (kombinált állomás) definiál a két végponton. Sorrendtartó, hibamentes adatátvitelt biztosít. Az adatátvitel bájtokban csoportosított bitek segítségével zajlik, annak érdekében, hogy bármilyen bájt érték elő előfordulhasson fordulhasson az átvitt adatban, 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 átvendő adat, akkor a vonalon folyamatosan FLAG-ek mennek.

Ez egyrészt azt eredményezi, hogy a vevő beérkező FLAG-ek hiányában képes felismerni a vonalszakadást, másrészt a folyamatosan érkező FLAG-ek felhasználhatók az adó- és vevőoldali órajelgenerátorok szinkronizálására. Az órajelgenerátornak olyan stabilnak kell lennie, hogy a lehetséges leghosszabb frame átvitelének idejére biztosítsa a szinkron működést.

Ahogy ezt már leírtuk, a HDLC bitorientált 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-beszúrással ugyanis változik az átvendő adat, és a ténylegesen átvitt adat bitszáma.)

A LAPB helyét a HDLC családban a **102. ábrán** láthatjuk.



A **harmadik szint** a csomagszint, amely az előbbieken felsorolt (DG, PVC, VC) csomag típusú 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.

A **101. ábrán** egy 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. X.25 esetén a DCE a hálózat felsőbb rétegeibe is kiterjed.

4. HÁLÓZATI RÉTEG

VEZÉRLŐ MEZŐ											
Formátum	Parancsok	Válaszok	Kódolás								
			8	7	6	5	4	3	2	1	
Felügyelő	RR	RR	N(R)		P/F	0	0	0	1		
	RNR	RNR	N(R)		P/F	0	1	0	1		
	REJ	REJ	N(R)		P/F	1	0	0	1		
Sorszámozatlan	SAMB		0	0	1		P	1	1	1	1
	DISC		0	1	0		P	0	0	1	1
		DM	0	0	0		F	1	1	1	1
		UA	0	1	1		F	0	0	1	1
		FRMR	1	0	0		F	0	1	1	1
Információ transzfer	I		N(R)		P			N(S)	O		

INFORMÁCIÓ KERET

Flag 01111110	Cím mező	Vezérlő mező	Információ mező	Keret ellenőző sorszám	Flag 01111110
------------------	----------	-----------------	--------------------	---------------------------	------------------

ADATCSOMAG

Csomag fejrése	Felhasználó adat mező
-------------------	-----------------------

103. ábra: LAP-B jellemzők

A csomagszint egyedi azonosítókkal ellátott logikai csatornákból áll. A csatornák csoportokba vannak szervezve. 4 bit hordozza a csoportszámot és 8 bit a csatornaszámot, amely a csomagok fejréseben található meg. Az X.25 számos csomagformátumot definiál, amelyek lehetnek adatcsomagok, illetve az átvitelt vezérlő hívásfelépítő, bontó, forgalomvezérlő, megszakító, reset-et okozó, újraindító, diagnosztikai csomagok, illetve a datagram szolgálati jelzés-csomagok. Részletes leírásuk a [2] irodalomban megtalálható.

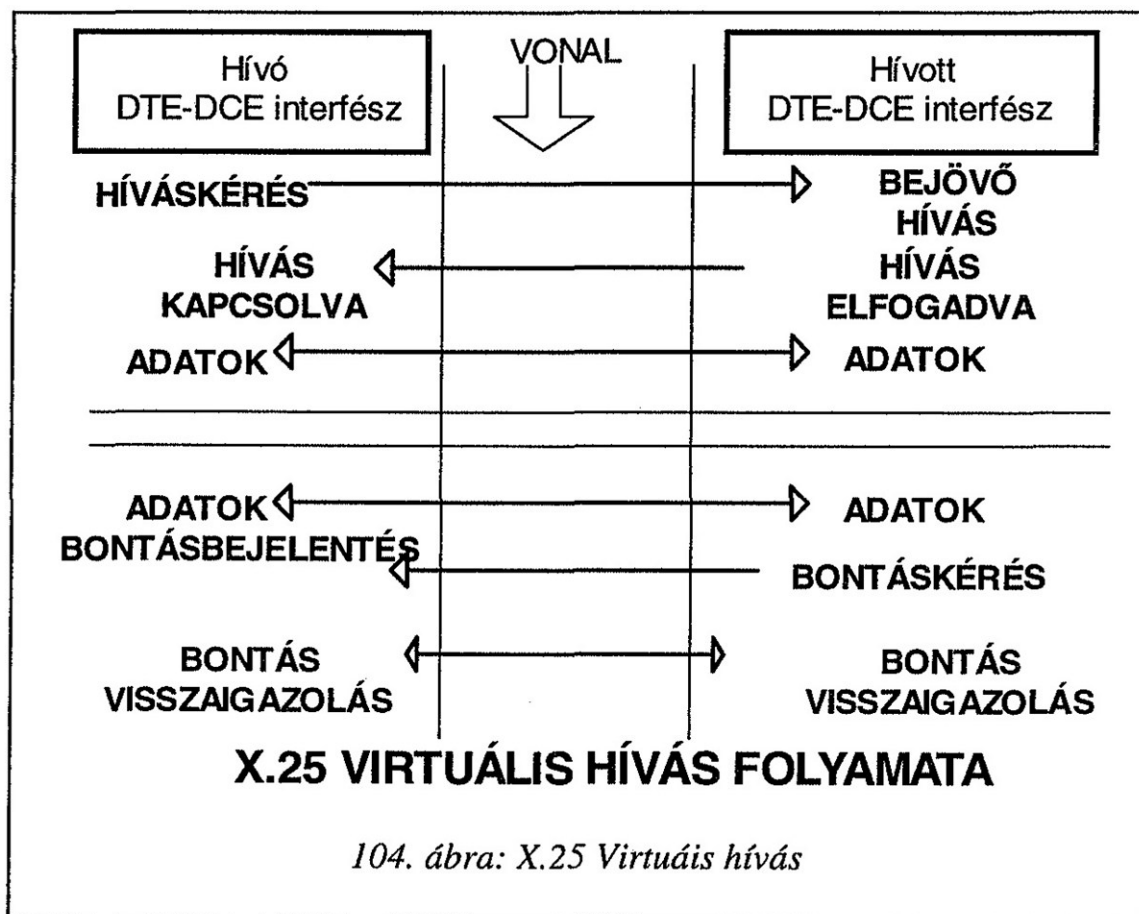
A DTE-DCE interfészek közötti átvitelt megvalósító és lebontó folyamat vázlatát az **104. ábrán** látható.

SZÁMÍTÓGÉP - HÁLÓZATOK

A három fázis: a hívás felépítése, adatátvitel és lebontás csomagok segítségével történik. A 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.

A Frame Relay az OSI struktúra második szintjén működő protokoll.

A jelenlegi protokoll szerkezeti irányzatnak megfelelően ez egyszerűbb és így megbízható átvitelt nyújtó közegen hatékonyabb mint a Frame Relay elődjének számító X.25.



KERET-RELÉZÉS (FRAME RELAY)

Mint a nevéből következik nem a csomagokat, hanem az adatkapcsolati szint kereteit viszik át a megfelelő minőségű hálózaton. A keret-relézés egy X.25-höz hasonló új 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 kb-ot 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 keret-relézést általában nem analóg (pl. telefon) vonalakon, hanem a kis hibaarányal működő digitális 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). Nagyobb sebességgel történő küldés esetén, az átvitel előtt néhány keret DE (Discard Eligibility) jelölést kap, és a sebesség túllépésekor ezeket fogja a protokoll először eldobni. Észak Amerikában az "európai" X.25 átvitel helyett használják.

ELLENŐRZŐ KÉRDÉSEK



1. 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?
4. Mi a routing tábla?
5. Mikor egyszerű a forgalomirányítás megvalósítása? Függ a topológiától?
6. Melyek a forgalomirányítás fő négy funkciója?
7. Milyen vezérlésmódokat különböztetünk meg?
8. Fogalmazza meg a legrövidebb út meghatározásának célját és módszerét!
9. Mi az a mérték, és mitől függ?
10. Ismertesse a véletlen forgalomirányítás módszerét!
11. Ismertesse az elárasztásos forgalomirányító eljárás módszerét!
12. Ismertesse a központi adaptív forgalomirányítás módszerét!
13. Ismertesse az elszigetelt forgalomirányítás módszerét! Mi az a „forró krumplic” algoritmus? Mi a fordított tanulás módszere?
14. Ismertesse az elosztott adaptív forgalomirányítás módszerét!
15. Mi a torlódás, és mi a torlódásvezérlés célja? Mi a befulladás?
16. Mutasson be néhány módszert a torlódás elkerülésére!
17. Mutassa be a lefojtó-csomagokat használó módszert!
18. Ismertesse a három alapvető csomag típusú szolgálatot!
19. Mi az a PAD?
20. Ismertesse az X.25 hálózat három (fizikai-, keret- és csomag szintű) protokoll szintjét!
21. Ismertesse az X.25 virtuális hívásának folyamatát!
22. Mi az a keretvezérlés?
23. Mi a flow-control (adatáramlás vezérlés)?

5. A FELSŐBB RÉTEGEK

A következőkben röviden összefoglaljuk az OSI modell felső négy rétegének feladatait. Ezek a rétegek már a hoszt-hoszt közötti kapcsolatok mikéntjét és milyenségét írják le.

Az OSI modellben a viszony-, a megjelenítési és az alkalmazási rétegek alkotják a felső rétegeket. Ellentétben az alsó négy réteggel, amelyek a megbízható két végpont közötti csupasz hibamentes csatornát valósítják meg, a felső rétegek további, az alkalmazások széles köréhez tartozó hasznos szolgálatokat biztosítanak.

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 nagyon fontos: megbízható adatszállítás biztosítása a forráshoszt és a célhoszt között, függetlenül az alatta lévő rétegek kialakításától.

A cél eléréséhez a hálózati réteg által nyújtott szolgálatokra támaszkodik. Itt már a feladat a tényleges hoszt-hoszt kapcsolat hibamentes megvalósítása.

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 címezés, 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.

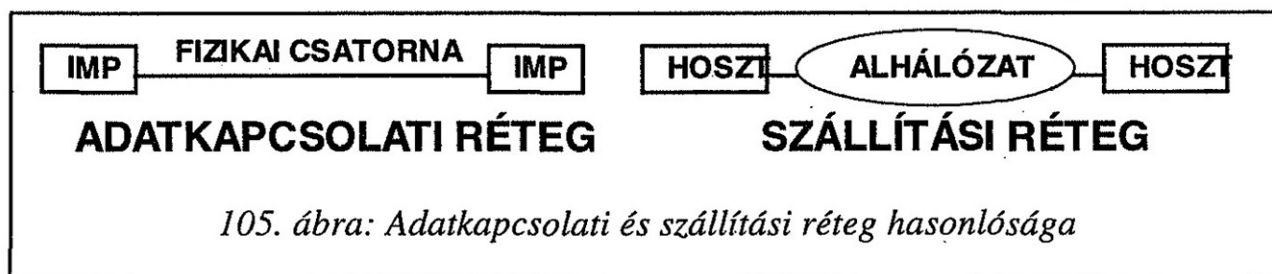
Az OSI modell tervezése előtt már sok hálózat működött, amelyek általában a hálózati réteggel jól átgondoltak voltak, és ezért jelentős mennyiségű működési tapasztalat és ismeret halmozódott fel. A felsőbb rétegek kialakításánál már nem lehetett az előző tapasztalatokra hivatkozni, tudták, hogy ezekre a rétegekre szükség van, de megvalósításukhoz csak kevés, és nem egy-egyedi reprezentáció szolgálhatott csupán alapul.



A szállítási réteg feladata nagyon fontos: megbízható adatszállítás biztosítása a forráshoszt és a célhoszt között, függetlenül az alatta lévő rétegek kialakításától.

Lényegében ez a réteg biztosítja, hogy az egymással kommunikáló két hoszt egymást úgy lássa, mintha pont-pont összeköttetés lenne közöttük.

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.

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. 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.

N-RESET: a hálózati réteg által kiadott, alaphelyzetbe állító, az összes függő csomagot törölő parancs (Net-Reset)

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.

A különböző szállítási protokollokat az előzőekben leírt hálózati szolgálatok minősége alapján öt osztályba sorolták [1].

Az üzenetek átviteléhez szükséges összeköttetés kezelése, a létesítés, és lebontás szállítási protokollok esetén 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 hoszt-hoszt kapcsolat többféle konkrét megvalósítását biztosítja: például fájlok átvitele a két hoszt között, vagy az egyik hoszt a célhoszt termináljaként kíván működni, stb.

Az ilyen 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 kétféle a módszert alkalmazzák:

- a küldő hoszt által igényelt szállítási alkalmazást mindig a célnál egy **alkalmazásslálgáltató (process server)** fogadja, és az üzenet tartalma alapján rendeli hozzájuk a TSAP címeket.
- vagy azt a módszert alkalmazzák, hogy 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.

Például egy banki átutalás kérését tartalmazó csomag, ha megkettőződik, és az egyik valahol rövid időre tárolódik, akkor először az első csomag hatására történhet meg egy átutalás, majd utána a másikra, hiszen ezeket lehet két független tranzakció kérésének tekinteni.

Az összeköttetés létesítése nem egyszerű C típusú hálózatszolgáltatások esetén, mert az elveszett, vagy kettőzött csomagok sok problémát okozhatnak. Ezekre, és hasonló problémákra két megoldás lehetséges:

- 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 ezt az időpont alapján a csomag korát meg tudják állapítani. Ha a csomag „túl öreg”, eldobják.
- **háromutas kézfogás** alkalmazása Ennek az a lényege, hogy az összeköttetés létesítése során a felek tetszőleges, véletlenszámként választott kezdősor számmal kezdik az adásukat, így kiszűrhető egy korábbi összeköttetésből származó késve érkezett csomag.

Az összeköttetés lebontásakor biztosítani kell az adatvesztés mentes lebontást. Nem lehet addig törölni az összeköttetést, amíg az összes elküldött adat meg nem érkezik. Ezt az előbb említett három-utas kézfogás alkalmazása biztosítja.

VISZONYRÉTEG

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.

Fontos feladata a viszonyrétegnek a párbeszédés 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 adat-vezérjel (token) bevezetése: mindig csak a vezérjelet birtokló küldhet adatot, addig a másik félnek hallgatni kell. Az adatküldés befejezésekor az adat-vezérjelet átadja a másik oldalnak, és így a helyzet megfordul.

MEGJELENÍTÉSI RÉTEG

A megjelenítési réteg felelős az információ megjelenítéséért és egységes értelmezéséért, a feladata a szállított **információ jelentéséhez** kapcsolódik:

- adatábrázoláshoz
- adattömörítéshez
- hálózati biztonsághoz
- védelemhez.

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.

ADATÁBRÁZOLÁS,
ADATTÖMÖRÍTÉS
HÁLÓZATI
BIZTONSÁG
VÉDELEM.

ADATÁBRÁZOLÁS

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 különbségek is.

A karakterábrázolással kapcsolatosan pozitív dolog, hogy szerencsére csaknem egységes az **ASCII kódrendszer** használata, éa ahogy ezt már az előzőekben leírtuk a nemzeti karakterek használatát a kódlapok, illetve a **UNICODE** használata teszi lehetővé.

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 formátumot használni, és erre átalakítva lehetne adatot a hálózaton átküldeni?

Ezeknek a kérdéseknek tárgyalása meghaladja a könyv korlátjait, többet az [1] irodalomban olvashatunk róluk.

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).

Didaktikai okok miatt (már előbb is felhasználtuk az ott szereplő ismereteket) ezek leírását a 3. fejezetben szerepeltettük.

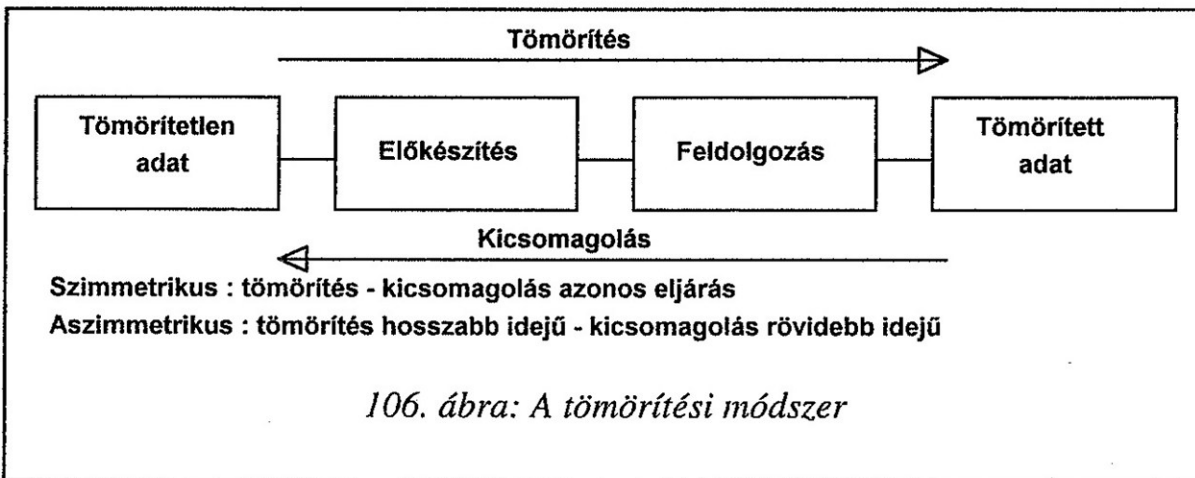


ADATTÖMÖRÍTÉS

Mivel a hálózatok használatáért általában fizetni kell, egyáltalán nem mindegy hogy időegység alatt mennyi információt viszünk át rajta. Az adatok ábrázolása általában redundáns. A csatornán elküldött információt szimbólumsorozatként is felfoghatjuk, amelyek egy adott szimbólumkészletből származnak, pl. decimális számjegyek készlete, karakterek készlete, stb. Az adattömörítés elvét a **106. ábra** mutatja.

Az adattömörítést az a tény teszi lehetővé, hogy az adatábrázolás és a kódolás általában redundáns.

A tömörítést gyakran becsmagolásnak, míg a visszaállítást kicsomagolásnak hívják.



A tömörítés során – ha jó tömörítést akarunk elérni – az információ egy, lényegien részét eldobhatjuk (pl. képtömörítésnél). Ezt

5. A FELSŐBB RÉTEGEK

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**.

A következőkben néhány tömörítési eljárást ismertetünk:

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

Minta helyettesítés: gyakori szimbólumsorozat helyettesítése speciális szimbólummal

Például a BASIC nyelvben szereplő definiált alapszavak (INPUT, FOR, stb.) a program futtatása előtt úgy bájtokban tömörítik, 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 Length Encoding = RLL): Sok nullát tartalmazó bináris sorozatokban a nullák számát bináris számként adjuk meg.

Például:

000100100000100001

18 bit

-> a nullák száma: 3 2 5 4

-> 3 bittel kódolva : 011 010 101 100

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. (Morse abc)

Huffmann kódolás: egyes jelek, vagy bájtsorozatok előfordulási gyakoriságát figyeli, és generált kód hossz ettől függ

Ilyen adathalmazt szolgáltat egy szkennert, vagy egy fax, hiszen a beolvasott képpontok jelentős része azonos (a papír fehér része).

Például ilyen a tabulátor (TAB) jel is! (8 betűközött ér)

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.

Aritmetikai kódolás: mint az előbbi, de a megelőző jeleket is figyeli

Transzformációs kódolás: Ilyen például a Fourier transzformáció: egy periodikus időfüggvényt adott amplitudójú és kezdeti fázisszögű szinusz-hullámok összegével írunk le.

Általánosabban fogalmazva nagyon sok folytonos összetett jelenség írható le vektorizációval: egy raszterpontokból álló kép leírható alapelakzatok (körök, vonalak, és koordinátáik) segítségével (vektorgrafika).

Subband kódolás: csak bizonyos frekvenciatartományba eső jeleket transzformáljuk (pl. telefon: 0-4 kHz)

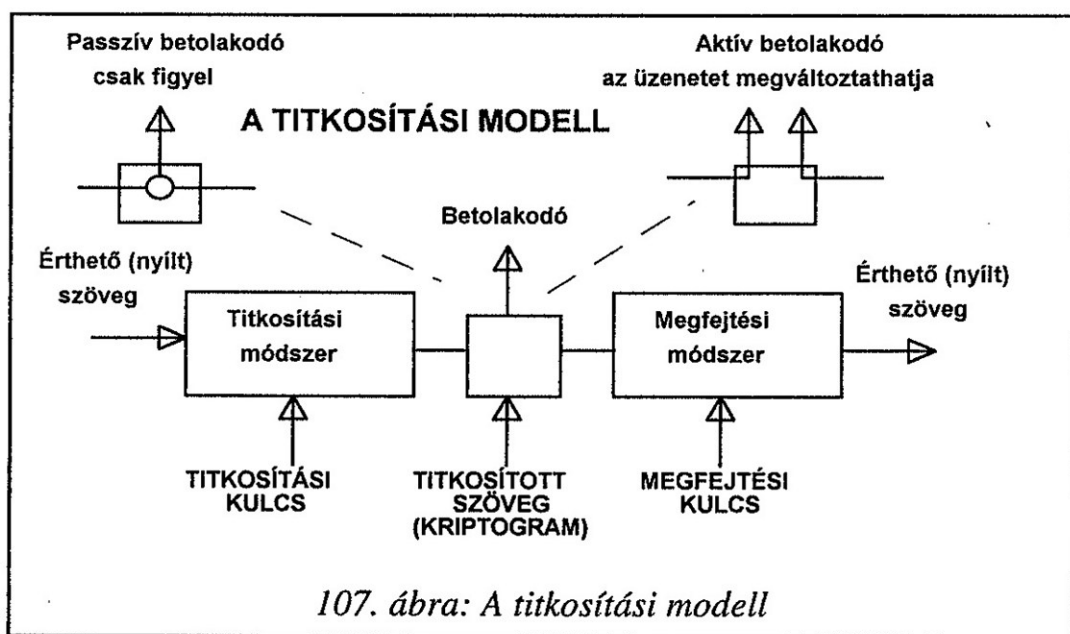
Predikció vagy relatív kódolás: ha az egymást követő jelek nem sokban térnek el egymástól, akkor elég a kis különbségeket kódolni.

Ezek az előbbieken felsorolt módszerek a gyakorlatban mind használhatók, a be- és kitömörítést programok, vagy jelenleg már egyre inkább hardver (egy chip) segítségével oldják meg.

Egy zeneszámot MIDI formátumban szintén alap hang és ritmusegységek írnak le, stb.

Például a PCM Pulzus Kód Modulációnál a beszédet 8 kHz-el mintavételezve 8 bites felbontással (kvantálással) -> 64 kbit/s adatfolyam keletkezik. Kisebb sebességű adatfolyam adódik, ha csak a mintavételezett jelek változását kódoljuk. Ez a Delta PCM.

HÁLÓZATI BIZTONSÁG ÉS VÉDELEM (TITKOSÍTÁS)



Gyakran előfordul, hogy bizalmas vagy titkos információt, banki átutalásokat kell továbbítani a hálózaton keresztül. Megoldandó, hogy az arra jogosulatlan személyek ne férhessenek hozzá a titkos adatokhoz. Megfelelő titkosítási algoritmus felhasználásával elérhető, hogy a titkosított adatok csak nem, vagy csak igen nehezen legyenek megfejthetők.

5. A FELSŐBB RÉTEGEK

A titkosítástan (kriptológia) alapvető szabálya az, hogy a titkosítás készítőjének feltételeznie kell, hogy a megfejtő ismeri a titkosítás általános módszerét (107. ábra). A módszernél a titkosítási kulcs határozza meg a konkrét esetben a titkosítást.

A titkosítási-megfejtési módszer régen nem lehetett bonyolult, mert embereknek kellett végigcsinálni. Két általános módszert használnak:

HELYETTESÍTÉSES REJTJELEZÉS:

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.

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 az 26-odik!) tagja T.*

A mátrix sorainak keverésével hatásosabb lesz a módszer, de ekkor a mátrix a kulcs részévé válik. Megfejtés alapja: a kulcs hosszának jó megsejtése.

Más: betűkódolás helyett két-három, vagy négy betű kódolása, vagy szavak használata. (kulcs egy könyv (pl. szótár), és elküldjük hogy adott oldal hányadik sorának hányadik szava. (számcsoportok).

FELCSERÉLÉSES REJTJELEZÉS:

A helyettesítéses rejtjelezések és kódolások a nyílt szöveg szimbólumainak sorrendjét változatlanul hagyják, csak álcázzák. A

A rejtjelezés két általános módszere:

helyettesítéses rejtjelezés
felcseréléses rejtjelezés

A szimbólumokat más szimbólumokkal helyettesítjük

Porta-féle rejtjelezésnél 26×26 -os mátrixot használunk, amelynek minden eleme betű pár. 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.

Egy adott könyvön alapuló rejtjelezés megfejthetetlen, hiszen nem tudjuk hogy a szavakat reprezentáló számcsoportok milyen szavakat jelentenek, és ugyanahhoz a szóhoz más és más számcsoport is tartozhat.

Betűk sorrendjének a megváltoztatása.

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 az oszlopok egymás után fűzésével kapjuk. Az oszlopok leírási sorrendjét a kulcs betűinek abc-beli sorrendje határozza meg.

Nyílt szöveg: **eztmostkodoljuk**
 Titkosítva: **odkmoutkj esoztl**

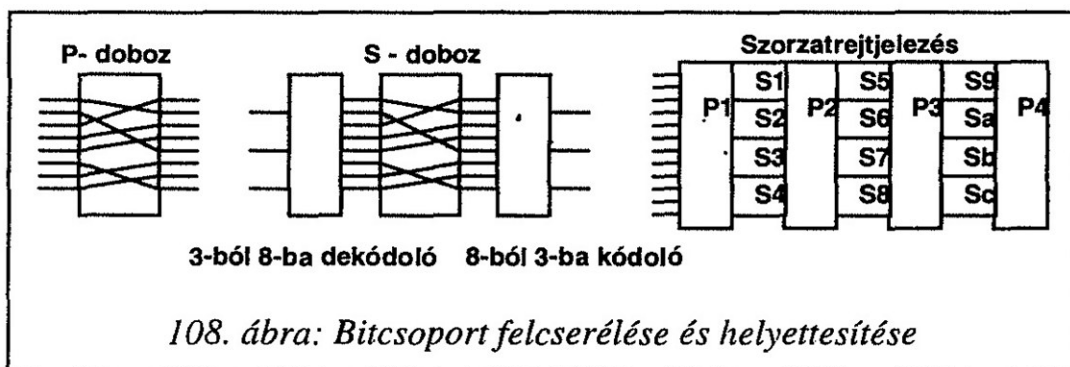
PROBA <- kulcsszó
 45321 <- kiolvasási sorrend
 eztmo
 stkod
 oljuk

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 ENCRYPTON STANDARD) — ADATTITKOSÍTÁSI SZABVÁNY

A számítógépek megjelenésével a hagyományos módszerek (helyettesítés és felcserélés) tovább élnek, de a hangsúly máshová került. Mivel régen emberek voltak a titkosítók, ezért a készítők egyszerű, emberek által jól megtanulható algoritmusokat és hosszú kulcsokat használtak.

A számítógépen csak két állapot van, ami a titkosítás szempontjából nem a legkedvezőbb.



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 sok titkosított szöveg birtokában sem tudja megfejteni.

A DES módszer lényegében egy 64 bites nyílt szöveget 64 bites titkosított szöveggé alakít, egy 56 bites titkosítási kulcs segítségével. Bináris elemek esetén a felcserélések és helyettesítések egyszerű áramkörök segítségével valósítható meg. (108. 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 össze-vissza kötött ki- és bemenetekkel, azaz a bemeneti 8 bit felcserélésével 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.

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.

NYILVÁNOS KULCSÚ TITKOSÍTÁS

A kulcsok védelme a betolakodóktól egy újabb problémát vet fel. A legbonyolultabb algoritmusnak semmi haszna, ha az általa használt kulcsot megszerzik a betolakodók.

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.

Ezekre az algoritmusokra az alábbi követelményeket kell figyelembe venni:

- $D(E(P))=P$ (Itt P a nyílt szöveget jelenti.)
- Rendkívül nehéz D-t E-ből leszámaztatni
- E-t nem lehet választott nyílt szöveggel feltörni

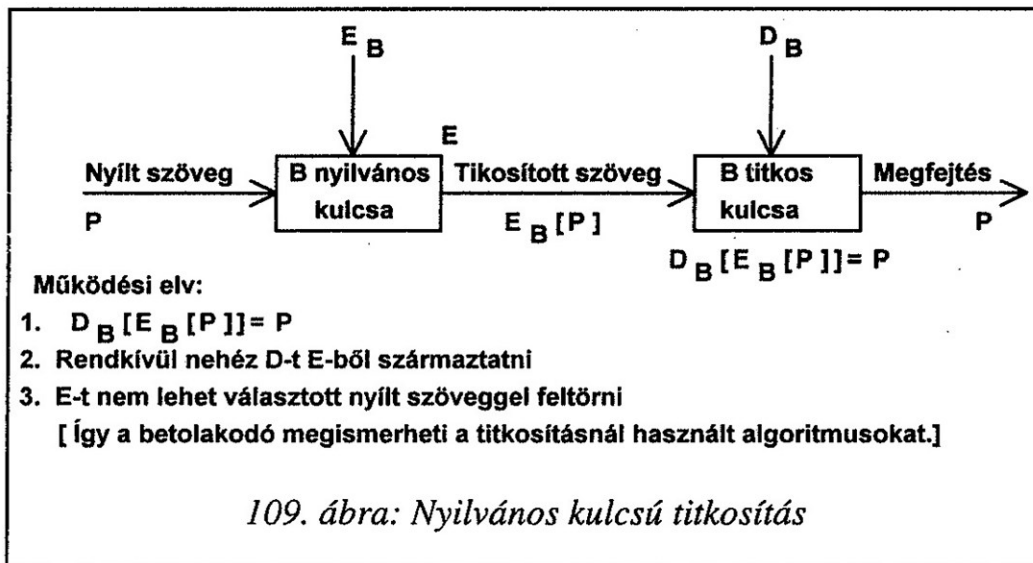
DES megfejtése. – mivel az algoritmus ismert — a összes kulcskombináció kipróbálásával oldható meg. Természetesen ehhez információval kell rendelkez-nünk a titkosított szövegről (pl. hogy ASCII karakterekből áll).

A feltörési folyamat automatizálható, és egyszerre sok számítógépen, kiosztott kulcscsoportok végigpróbálásával lehetséges.



A másik probléma, hogy két megelőzően nem érintkező fél, hogyan létesítsen egymással titkos kommunikációt.

A kulcsot vagy kulcsokat nem mindig adhatják át egymásnak személyesen.



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.

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 és kulcsokat ezután nyilvánossá teszik.

Innen származik a nyilvános kulcsú titkosítás elnevezés.

Egyetlen dolog maradt már hátra, hogy megtaláljuk a keresett algoritmusokat, amelyek az előző három követelményt teljesítik. Az algoritmusokat megtalálták, és az ezen alapuló számítógépes módszer a Pretty Good Privacy (Kellemes biztonság), röviden: PGP.

HOGYAN MŰKÖDIK PGP?

A hagyományos titkosítási módszereknél, (mint pl. az előbbieken bemutatott DES) ugyanazt a kulcsot kell használni a kódoláshoz és a dekódoláshoz. Ami viszont azzal jár, hogy a kulcsot először egy megbízható csatornán el kell juttatni a fogadó félhez, csak utána lehet nekikezdeni a titkosított üzenetek küldésének.

Azonban ha a kulcsküldéshez van biztonságos csatorna, tulajdonképpen miért is van olyan nagy szükség a titkosításra?

Ha hagyományos rejtjelzéssel küldeni akarunk valakinek titkosított üzenetet, a kódoláshoz egy kulcsot használunk, amit a címzettnek is használnia kell az üzenetet visszafejtéséhez

A nyilvános kulcsos rejtjelzésnél mindenki két, egymással szorosan összefüggő kulccsal rendelkezik. Az egyik kulcs nyilvános, azaz mindenki által hozzáférhető, míg a másik a titkos.

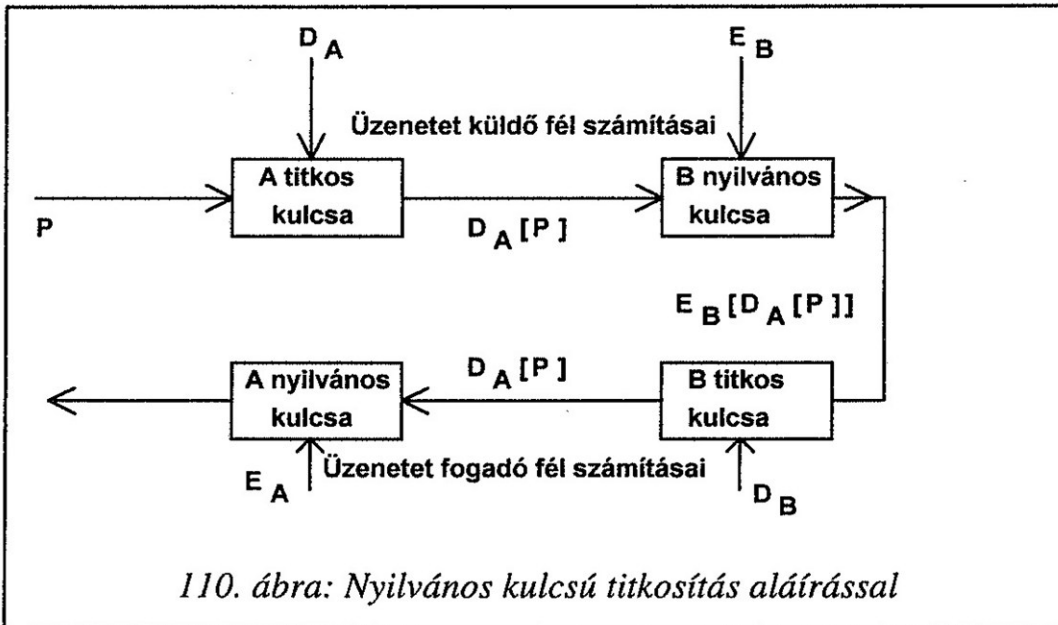
Lényeges, hogy a nyilvános kulcs nem nyújt lehetőséget a titkos kulcs kitalálásához.

A nyilvános kulcsot széles körben el lehet terjeszteni a kommunikációs hálózatokon. Egy nyilvános kulccsal bárki kódolhat üzeneteket.

5. A FELSŐBB RÉTEGEK

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.



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.

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. (110. ábra)

Számítógépes környezetet figyelembevéve, a kulcsok speciális struktúrában vannak tárolva, mely tartalmaz egy azonosítót (userId -- a személy neve), a kulcspár generálásának dátumát, és magát a kulcskódot.

A nyilvánoskulcs-struktúra tartalmazza a nyilvános kulcsot, míg a titkoskulcs-struktúra a titkos kulcsot. Ezeket a kulcsfájlokat ("key ring") szemléletesen kulcskarikának hívják, és egy vagy több kulcsstruktúrát tartalmaz. Az előbbieknél megfelelően vannak titkos és nyilvános kulcsfájlok (karikák).

Nyilvános kulcsokat tartalmazó kulcskarika
Titkos kulcsokat tartalmazó kulcskarika

A PGP kivonatokat (message digest) használ az aláíráshoz. A kivonat egy 128 bites szám, és a máshol használatos kontroll-összeggel vagy a CRC értékkel analóg abban a tekintetben, hogy jelzi az üzenet megváltozását. Ezt a kivonatot kell a küldő titkos kulcsával kódolni, ami ezzel az üzenet elektronikus aláírását adja ki. Az aláírás folyamán az üzenet szövege mellé kerül az aláírás struktúra.

Minden használónak két kulcsfájlja van: a nyilvános és a titkos. A nyilvános kulcsfájl esetén lehetőség van az egyedi kulcsok kinyerésére, exportjára, így azokat elküldhetjük ismerőseinknek, akik felvehetik az új kulcsokat saját nyilvános kulcsfájljukba.

Jelenleg ezek az eljárások már a korszerű levelező rendszerek részét képezik.

ALKALMAZÁSI RÉTEG

Az alkalmazási réteg feladata a felhasználó és a felhasználói programok számára a hálózati szolgáltatásokat biztosító illesztést 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.

Virtuális terminálokra azért van szükség, mert a valóságban számos inkompatibilis termináltípus létezik. Ezért egy olyan terminált definiáltak, amelyet a hálózaton futó megjelenítő programok használnak. Egy illesztőprogram végzi az adott termináltípus és az absztrakt terminál közötti megfeleltetést.

ELLENŐRZŐ KÉRDÉSEK

1. Mi a szállítási réteg feladata? Miért hasonlítjuk az adatkapcsolati réteghez?
2. Hogyan osztályozzuk a hálózati szolgálatokat minőségük alapján?
3. Milyen megoldások vannak a csomagtovábbítás hibáinak a kezelésére?
4. Mi a viszonyréteg feladata?
5. Mi a megjelenítési réteg feladata?
6. Milyen problémák léphetnek fel az adatábrázolással kapcsolatosan a hálózatokon?
7. Rajzolja fel az adattömörítés blokkvázlatát!
8. Ismertessen néhány tömörítési eljárást! Mi a darabszám kódolás? Mi a mintahelyettesítés? Mi a sorozathossz kódolás?
9. Mi a statisztikai kódolás? Mi a relatív kódolás?
10. Rajzolja fel a titkosítási modellt!
11. Ismertessen néhány helyettesítéses rejtjelezési megoldást!



12. Mi a felcserélési és helyettesítési rejtjelezés közti alapvető különbség?
13. Mi az a DES?
14. Milyen digitális áramköri elemekkel lehet egy bitcsoport felcserélését és helyettesítését megoldani?
15. Hogyan működik a nyilvános kulcsú titkosítás algoritmus?
16. Mi az a PGP?
17. Mi a nyilvános kulcsú aláírás módszere?
18. Mi az alkalmazási réteg feladata?

6. LOKÁLIS HÁLÓZATOK

A számítógép hálózatok egyik legdinamikusabban fejlődő területe a kisebb helyi hálózatok, azaz a lokális hálózatok. 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.

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 LAN-okon 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 sint 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 közeg-hozzáférési 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 legtöbb felhasználó a hálózatokkal ilyen formában találkozhat a csoportos munka, a kommunikáció, és az elosztott információkezelés hatékony eszközeként.



A lokális hálózati szabványok kidolgozása itt is egy „de facto” szabvánnyal, az Ethernet-tel indult, amelyet a XEROX cég fejlesztett ki, és nagyon gyorsan elterjedt.

6. LOKÁLIS HÁLÓZATOK

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

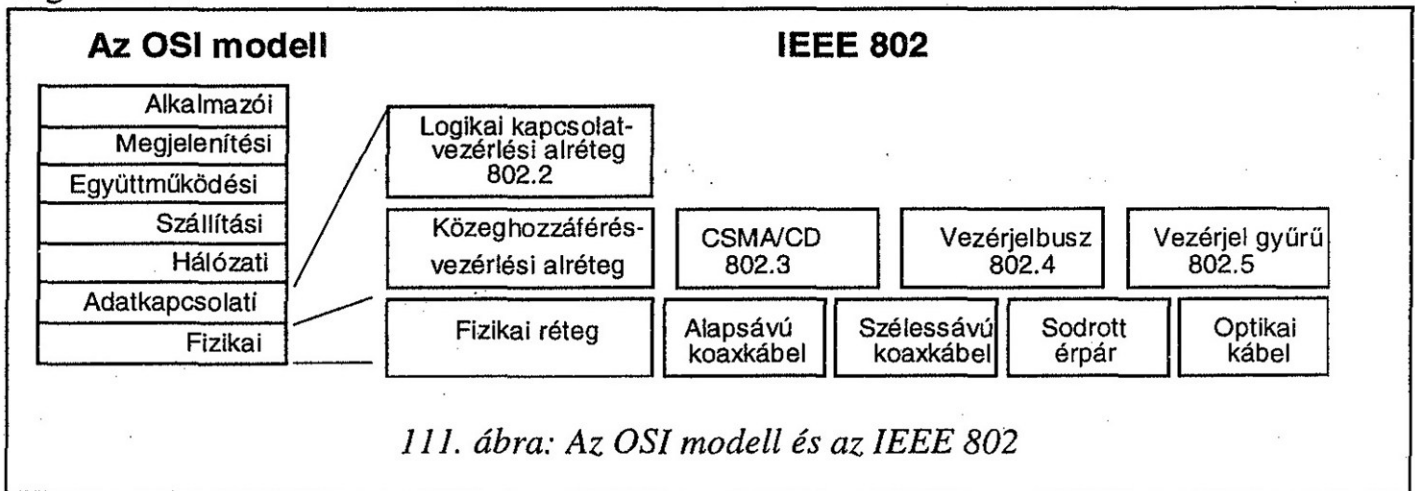
A 802.4-es szabvány a vezérjeles sín, és a

A 802.5-as szabvány a vezérjeles gyűrű leírása.

A 111. ábra jól mutatja az IEEE 802 szabvány és az OSI modell kapcsolatát, és ejtsünk most néhány szót a rétegekről.

Megjegyzés: 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.

Nagyon fontos itt megjegyezni, hogy a 802.3 szabvány és az Ethernet nem azonos fogalmak. Az Ethernet egy termék, azaz a 802.3-as szabvány megvalósítása.



A FIZIKAI RÉTEG

Az IEEE 802-es három olyan 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 bitek átviteléért. Meghatározza még átvitelre alkalmas formában az adatkódolást és dekódolást, vezérli az eszközök időzítését, hogy azokat az adott és vett jelek szinkronizálják.

Ez a réteg foglalkozik az átviteli közegen keresztül a jelek fizikai átvitelével. Definiálják a különböző típusú kábeleket, csatlakozókat.

KÖZEGHOZZÁFÉRÉS-VEZÉRLÉSI (MAC) ALRÉTEG

A lokális hálózatban lévő eszközök mindegyike a közös fizikai csatornán való hozzáférésért verseng. Mivel a LAN kialakításokban ezen a szinten számos hozzáférés-vezérlési módszert használnak ütközéssel és ütközés mentes egyaránt, a biztonság ezek közül — ahogy ezt már az előbbieken is leírtuk — a CSMA/CD, a vezérjel-

MAC=Media Access Control. Funkciói :

- Közeghozzáférés-vezérlés
- Keretezés
- Címzés
- Hibafelismerés

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

Az adatkapcsolati réteg logikai kapcsolatvezérlés szintjén az IEEE 802-es szabványt hozott létre: ezen a szinten minden IEEE 802-es szabvány közös, már amit a felsőbb rétegek felé mutat. Az alréteg szervezi az adatfolyamot, parancsokat értelmez, válaszokat generál, a hibákat ellenőrzi, és helyreállítási funkciókat hajt végre.

Ez az alréteg, a felette álló rétegeknek nyújt szolgáltatást ugyanolyan módon, ahogy azt a hagyományos adatkapcsolati protokoll nyújtja a távolsági hálózatban. Az OSI referenciamodellt követő LAN kialakításban, 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)

LLC= Logical Link Control
Funkciói:

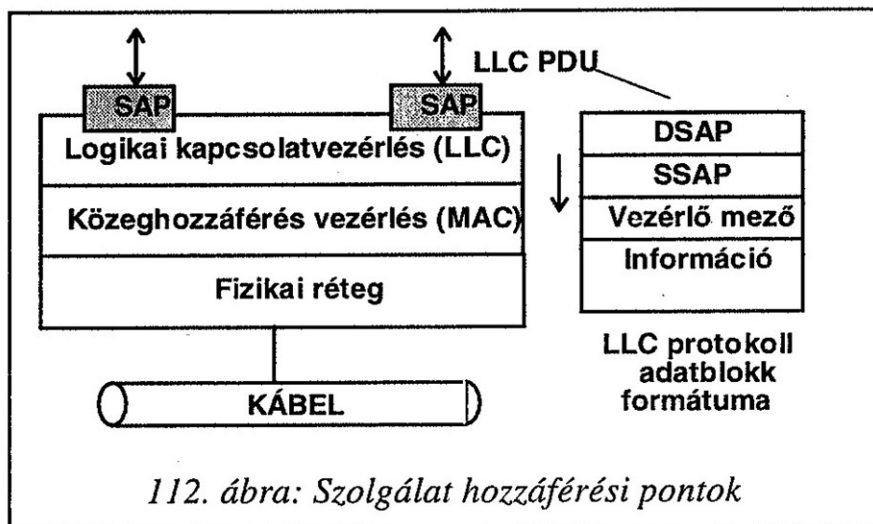
- a MAC rétegtől független egységes adatkapcsolati protokoll nyújtása a hálózati rétegnek,
- adatblokk csere,
- ehhez logikai kapcsolat létesítése

A szolgáltatás-hozzáférési pontot úgy is értelmezhetjük, mintha egy I/O port címe, vagy egy állomás magasabb rétegeihez való hozzáférési pont lenne. (Amin keresztül a rétegkapcsolat megvalósul)

A 111. 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 közeghozzáférés-vezérlő alrétegnek. Az átadott adatblokk felépítése is a 112. ábrán látható.

Ezek az adatblokkok (LLC PDU-k) parancsokat és nyugtákat is hordozhatnak, azaz az üzenet feldolgozásának a vezérlésére használják.



112. ábra: Szolgálat hozzáférési pontok

A forrás SAP cím mindig egyedi, ami egyetlen olyan SAP-ot azonosít, amely az eredeti adatblokkot küldte. A rendeltetési SAP cím vagy egyetlen SAP-ot azonosító egyedi cím, vagy csoportcím. A csoport SAP cím a rendeltetési SAP-ok olyan csoportját határozza meg, amelybe tartozó rendeltetési állomás mindegyike veszi az adatblokkot.

RÉTEGEK KÖZÖTTI SZOLGÁLATOK

A rétegek közötti szolgálatokról, az első fejezetben már írtunk, így itt csak összefoglaló jelleggel foglalkozunk velük.

Általában minden szolgáltatást megvalósító eljárásnak — primitívnek — a működéséhez paraméterekre is szükség van. Ezek összefoglalva: a célcím, a forráscím, az adatblokk vagy mutatója, az átvitel prioritási szintje, jelzőbitek az átvitel sikerességének jelzésére, stb.

HÁLÓZATI/LLC FELÜLETEK KÖZÖTTI SZOLGÁLTATÁSI ELŐÍRÁSOK

A felsőbb hálózati réteg és a logikai kapcsolatvezérlési alréteg felületek közötti kapcsolat számára két műveleti típust definiálnak: a kapcsolatmentes szolgáltatást, illetve a kapcsolatorientált szolgáltatást. Az első típus esetén nincs szükség logikai kapcsolat létrehozására az adó- és vevőállomás között, és minden elküldött adatblokkot

Lényegében ez a rész az 1. fejezetben leírtak jobb megvilágítására szolgál

SZÁMÍTÓGÉP - HÁLÓZATOK

közvetlenül dolgoznak fel. Nem végeznek sorszámellenőrzést (amivel biztosítanák, hogy az adatblokkot ugyanabban a szekvenciában vegyék, mint az a kiküldés, során volt), és a vevőállomás az adatblokk vétele után nem küld nyugtát. A kapcsolatmentes szolgáltatásoknál nincs folyamatvezérlés és hibajavítás. (azaz datagram szolgáltatás).

Itt csak két szolgálati primitív van:

L_DATA.REQUEST	A hálózati réteg az LLC-hez az adatblokkot továbbítja, és kéri elküldését.
L_DATA.INDICATION	Jelzi a hálózati rétegnek, hogy adatblokk érkezett, majd továbbítja is.

A második műveleti típus kapcsolatorientált szolgáltatás: Az adó- és vevőállomás között kapcsolatot kell létesíteni a műveletek megkezdése előtt, összeköttetést kell fenntartani majd az összeköttetést bontani.

Itt öt primitív osztályt definiáltak:

L_CONNECT	primitívek két SAP közötti logikai kapcsolat létesítésére szolgálnak, míg az
L_DISCONNECT	primitívek a kapcsolat bontására.
L_DATA_CONNECT	primitívek a felépített összeköttetésen keresztüli adatátvitelt vezérlik.
L_RESET	primitívekkel hozható alaphelyzetbe az összeköttetés.
L_CONNECTION_FLOWCONTROL	primitívek a két réteg közti adatcserét vezérlik, a sebességkülönbségeket egyenlítik ki.

LLC/MAC FELÜLETEK KÖZÖTTI SZOLGÁLTATÁSI ELŐÍRÁSOK

Az LLC/MAC interfész szolgáltatások lehetővé teszik, hogy az egyik állomás LLC alrétege adatot cseréljen a másik állomás ugyanezen alrétegeivel.

Ezt három szolgálati primitív segítségével valósítják meg:

MA_DATA.REQUEST	Adatblokk küldésének kérése,
MA_DATA.INDICATION	Adatblokk vétele és továbbítása megtörtént a MAC rétegből az LLC rétegbe,
MA_DATA.CONFIRM	LLC jelzi, hogy elfogadta.

A LOKÁLIS HÁLÓZATOK FIZIKAI EGYSÉGEI

A lokális hálózati kommunikáció feladatait az erre a célra tervezett hardver és firmware (ROM-ba égetett, a kártyán lévő program) látja el. Személyi számítógépekből álló hálózatokban használt fizikai összetevők 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 közeghozzáférést vezérlő funkciókat megvalósító hardvert és főmvert (firmware).

A hálózatok terjedésével az a tendencia, hogy a különálló adapter kártyákat a számítógép alaplapjára integrálják.

Kábelrendszer A kábelrendszer azt a kábelt, ill. vezetékét 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.

Koncentrátorok és erősítők Egyes lokális hálózati kialakítások koncentrátorokat, ill. hozzáférési egységeket használnak, hogy a hálózati jelek erősítése és elosztása megoldott legyen, illetve a hálózatban levő eszközök egy központi helyen kerüljenek összeköttetésbe egymással. Ezeket szokták HUB-oknak, jelisméltőknek is nevezni.

A lokális hálózatokban használt számítógépeket a hálózati funkciójukat tekintve, két csoportba sorolhatjuk:

- az információt felhasználó munkaállomások, ezeken dolgoznak a felhasználók, és az
- információt szolgáltató számítógépek az ún. szerverek.

Természetesen ez a két kategória — ahogy ezt a későbbiekben is látni fogjuk — fizikailag nem válik élesen ketté.

AZ IEEE 802.3 SZABVÁNY ÉS AZ ETHERNET

Az Ethernet közeghozzáféréseinek alap gondolatát már bemutattuk. Mielőtt egy állomás adni akar, behallgat a csatornába. Ha a kábel foglalt, akkor az állomás addig vár, amíg az üressé nem válik, máskülönben azonnal adni kezd.

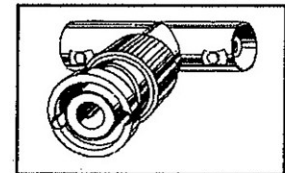
Ha egy üres kábelen két vagy több állomás egyszerre kezd el adni, ütközés következik be. Minden ütközést szenvedett keretű állomásnak be kell fejeznie adását, ezután véletlenszerű ideig várnia kell, majd az egész eljárást meg kell ismételnie.

Az Ethernet hálózatok átviteli sebessége a jeleleg még legtöbbet használt rendszerekben 10 Mbit/s. Ma már 100 Mbit/s sebességű Fast Ethernet vagy 100baseT hálózatok illetve 1 Gbit/s sebességű 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. Tehát az Ethernet optimális sebessége mintegy 4.5 Mbit/s. Ethernet hálózatokban többféle kábeltípus használható:

Elnevezés	A kábel fajtája	Jelregenerálás nélküli max. hossz (m)
10BaseT	Árnyékolatlan csavart érpár	100
10Base2	vékony koax kábel	185
10Base5	vastag koax kábel	500
100BaseTX	Fast Ethernet	100
10BaseF	üvegszál	>1000

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 illesztnek be, és ez csatlakozik a számítógép hálózati kártyájára.



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 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 (pl.: az ilyen kábelek csak meghatározott íveken hajlíthatók) csak olyan

A most következő részekben olyan jellegzetes lokális hálózatokat vizsgálunk meg, amelyek korábban tárgyalt és bemutatott különböző architektúrák és szabványok szerint épülnek fel.

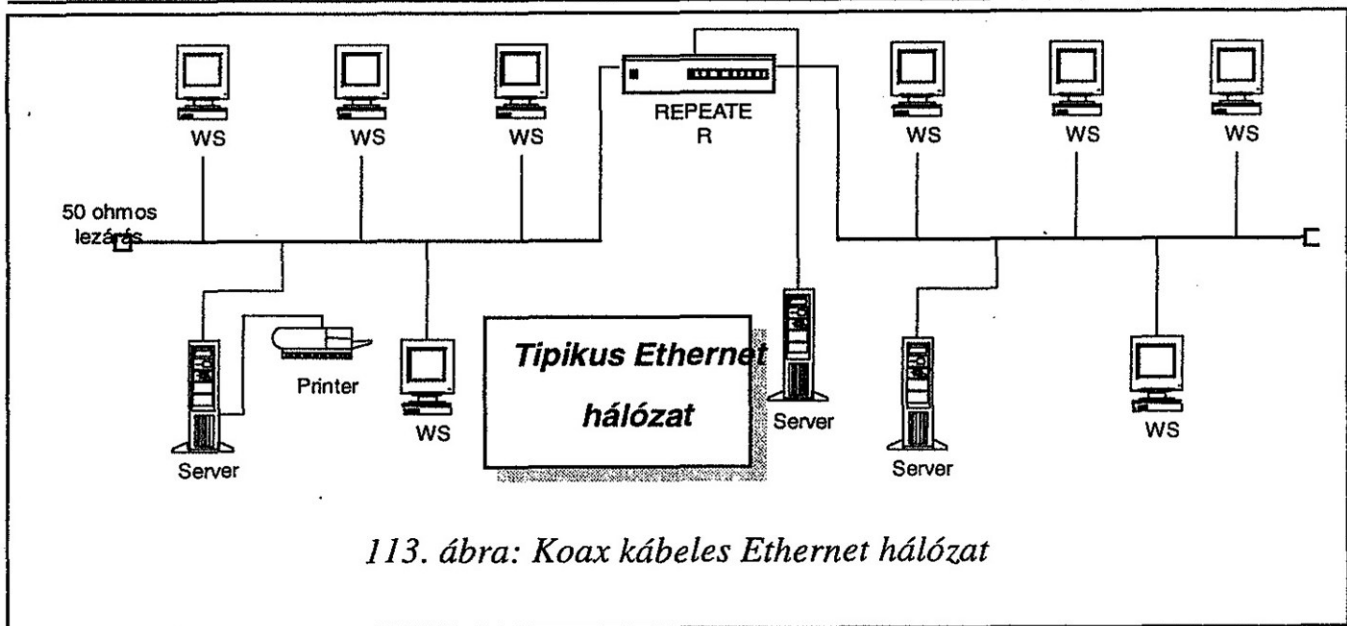
Az UTP és koax fizikai közegről már a 2. fejezetben is írtunk, de nem árt egy kis ismétlés...

Nagy hálózatok esetén, ahol gyakran kell új felhasználót a rendszerhez kapcsolni, vagy a rendszer leállítása nehezen oldható meg ez az eljárás erősen megkérdőjelezhető.

Továbbá, minél több ilyen csatlakozás van a hálózatban, annál valószínűbb, hogy valamelyiknél érintkezési hiba keletkezik.

Rendelkezése állnak ún. megszakítás nélküli csatlakozók is, ám ezek telepítése meglehetősen költséges.

6. LOKÁLIS HÁLÓZATOK



esetekben használják, ahol az erősebb külső zavarok miatt szükséges az erősebb árnyékolás (pl.: ipari felhasználás), illetve nagyobb az áthidalandó távolság.

A vastag koax kábeleknél a számítógép-csatlakoztatás módja az ún. **vámpír csatlakozó** használata. 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.

A vámpír csatlakozókat csak a kábel jelölt, meghatározott pontjain lehet elhelyezni.

AUI kábel bekötése

ÁRAMKÖR	NÉV	MAU-hoz	MAU-tól	BEKÖTÉS	MEGJEGYZÉS
DO	Data Out	x		DOA-3, DOB-10, DOS-11	Adat kimenet
DI	Data In		x	DIA-5, DIB-12, DIS-4	Adat bemenet
CO	Control Out	x		COA-7, COB-15, COS-8	Vezérlés kimenet
CI	Control In		x	CIA-2, CIB-9, CIS-1	Vezérlés bemenet
VP	Voltage Plus	x		13	12 Volt
VC	Voltage common	x		6	VP másik ága
PG	Protective Ground	x		14+fémház	Védőföld (árnyékolás)

A MAU csatlakozója (Canon DB-15) négy szimmetrikus jeláramkört, tápellátást és földelést szolgáló vezetéket tartalmaz. A jeláramkörök két jelvezetékéből (A és B) és az árnyékolásukból (S) állnak.

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 szegmensen 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 alkalmazásakor minden ilyen módon bekötött számítógép lényegében pont-pont kapcsolatot valósít meg az elosztó eszközzel. Ha a kialakítást megnézzük akkor a csillag formájú hálózat a küllős kerékhez hasonlít, ahol a küllők végén vannak az egyedi végpontok (munkaállomások, fájlserverek, stb.) Mindegyik munkaállomás a kábelhez egy hálózati adapteren keresztül kapcsolódik, és az összes kábel a közepen lévő dobozba — hasonlóan mint a küllők a kerékgyába — kapcsolódik. Ezt az egységet hívják **HUB**-nak. (Ejtsd: hab. Angolul a hub egyik jelentése: kerékagy, vagyis hasonlít egy küllős kerékhez.)

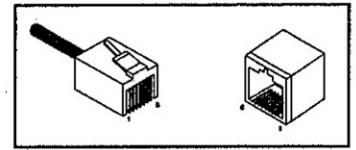
A hub egy doboz, rajta **port**-oknak nevezett, telefoncsatlakozókhöz nagyon hasonló 8 érintkezős (neve: RJ45) csatlakozó aljzatokkal. Minden port egy munkaállomástól, szervertől, vagy egyéb hálózati egységtől érkező kábelt fogad. A hub-ok számos formában és méretben kaphatók: 4 portostól akár 124 portosig.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

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

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 vételi, rendelkezésre álló sáv szélessége. 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

Ezek a megkötések a jelterjedési viszonyok kézbentartása és a megfelelő ütközésetektől való megvédés érdekében történtek.



Lényegében csillag topológiájú hálózat alakul ki.



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.
4. **SNMP- (távolról felügyelhető (menedzselhető)) hub-ok.** Ezek a hub-ok támogatják a hálózati menedzselhetőséget biztosító szabványos Simple Network Management Protocol (SNMP) protokollt (ld. 8. 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 előbbieken 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 kábelek csatlakoznak. Egyes megvalósításainál a függőleges gerincvezeték vastag, míg a vízszintesek vékony kábelek.

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 csatlakozóig) azonos érszámú kábel fut, csillag kialakítású hálózatot alkotva, amelynek középpontjában a rendező van.

A jelenlegi Ethernet kábelezési technika az üvegszálás (egyre ritkábban koax) gerincvezetékét részesíti előnyben, amelyhez jelismétlőkön (UTP multiport repeater = HUB) keresztül csavart érpárok csatlakoznak a számítógépek, csillag topológiát formálva. A különféle fizikai kábelezés megfelelő csatlakozás-párokat tartalmazó egységek használatát követeli meg. Ezek lehetnek az üvegszálnál alkalmazott csatlakozók (ST, FC/PC, SMA, SC, FDDI), vékony koax BNC csatlakozója, a 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 alkalmasak a telefon és a számítógép csatlakoztatására is. A fali csatlakozókat a berendezésekkel adapter kábelek kötik össze. A fali csatlakozókat sodrott érpáras kábelben 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 kábel** segítségével 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. Kinyomozásukra különböző technikákat fejlesztettek ki. Alapvetően egy ismert alakú jelet bocsátanak a kábelre. Ha a jel akadályba vagy a kábel végébe ütközik, akkor visszhang keletkezik, amely a jellel ellenkező irányba terjed. A jel kibocsátási és a visszhang visszaérkezési idejét precízen mérve a visszhang keletkezési helye

Bár a struktúrált hálózat kiépítése jóval drágább a hagyományos hálózathoz képest, ez a többlet évek során megtérül, hiszen bárhol történik változás, a módosítások csupán átkötésekkel megoldható.



meghatározható. Ezt a technikát időbeli reflektometriának (time domain reflectometry) nevezik.

Az összes 802.3 implementáció, beleértve az Ethernetet is, manchester kódolást használ, amelyet az 113. ábrán láthatunk.

A bitek közepén lévő jelváltás iránya jelenti a 0 vagy 1 információt, és ezen átmenet segítségével a küldő szinkronizálhatja a VEVŐ-t.

A számítógépben lévő interfészártya a csatlakozó kábeltípustól függő BNC, AUI, telefoncsatlakozó aljzattal van szerelve és olyan vezérlő integrált áramkört tartalmaz, amely kereteket vesz ill. kereteket küld a hálózatra. A vezérlő felelős a kimenő keretek adatokból való összeállításáért, a kimenő keretek ellenőrzőösszegének kiszámításáért és a bejövő keretek ellenőrző-összegének ellenőrzéséért.

HUB, BRIDGE (HÍD), SWITCH (KAPCSOLÓ)

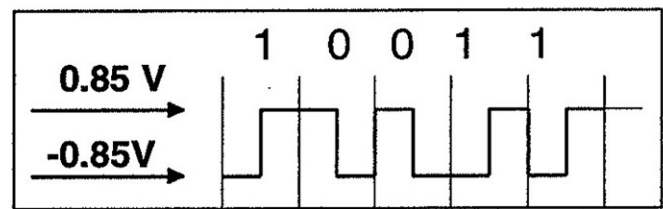
Az Ethernet rugalmasságát is növelik a hubok. Ezek az egyik portjukon vett keretet bit-ről bitre átmásolják a másik portjukra, mintegy meghosszabbítva ezzel az elektromos jellemzők miatt rövidre korlátozott szegmenst. Több porttal rendelkező (multiport) repeater-ek használata esetén minden portra átmásoljuk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

Az IEEE802.3 egyik fő tulajdonsága az osztott közeg, egyre inkább jelent hátrányt, mint előnyt. Az osztott közeg azt jelenti, hogy hálózat lelassul, és egy épületben mindenki munkáját akadályozza, ha valaki intenzíven másol vagy éppen letölt valamit a helyi kiszolgálóról.

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 határfoka. Ennek az irányába történik fejlődés, ha a **mikroszegmensekről** beszélünk. Ebben az esetben nagyon kevés munkaállomás csatlakozik a szegmensre. Ekkor az állomás akkor



114. ábra: Jelszintek az Ethernet hálózatban

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

Bármelyik időpontban a kábel a következő három állapot egyikében van:

- 0-ás bit átvitele (alacsonyból magasba való átmenet),
- 1-es bit átvitele (magasból alacsonyba való átmenet), vagy
- tétlen (0 V).

Ezen kívül, a kártya még kezeli a bejövő keretek számára fenntartott pufferterületet, a kimeneti puffer-sort, az átvitelt sok esetben DMA-val gyorsítva.

Ehhez természetesen tudni kell, hogy melyik állomás merre található, és el kell kerülni azt is, hogy a bridge-k körbekörbe adjanak egymásnak egy keretet.

Ráadásul minden ütközés csak tovább rontja a helyzetet, hisz az ütközés észleléséig eltelt időben nem történik hasznos kommunikáció.

adhat, amikor csak akar. Ezzel azonban a keretek ütközésének problémáját mindössze áttettük a kábeltől a hídra, 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

A switch-nek szinte mindig van egy vagy több nagysebességű portja is (FDDI, Fast Ethernet) melyen át egy nem Ethernet gerinchálózatra kapcsolható. Így 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 át 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 leadjuk
- **Cut through** állapotban a switch azonnal, a célállomás címének beérkezése után elkezd a keret továbbítását. Így csökkent a késleltetés, hiszen a címmező a keret elején található. Ha a kimeneti port foglalt, akkor természetesen a keret puffereljük és a port felszabadulása esetén adjuk le. Sajnos így nem képes ellenőrizni, 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 állomás 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 ezenkívül 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. Kísérletek 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

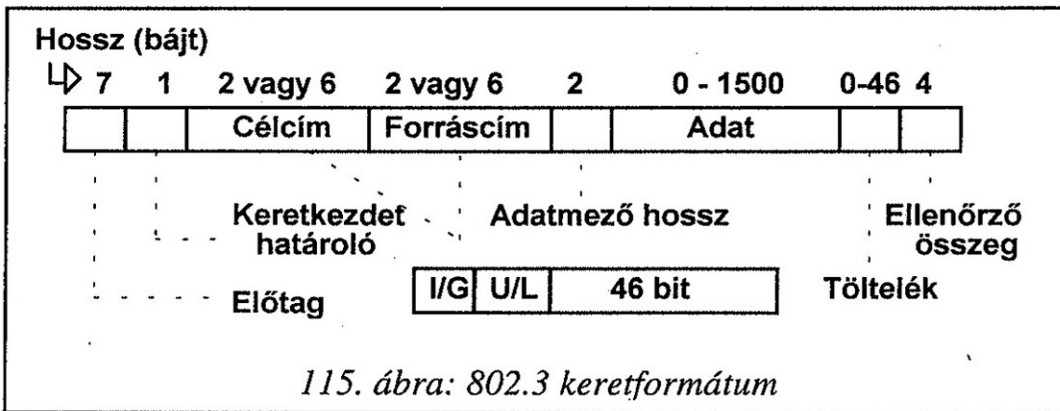
Azaz, ha egy switch-nek hét portja van, akkor egyidőben képes keretet továbbítani például az 5. portról a 6.-ra, és a 6.-ról a 7.-re. Csak akkor áll elő ütközés, ha egyszerre többen kívánnak ugyanarra a portra adni, ekkor a switch az egyik keretet pufferelemmel, és a másik után adja le a portra. Így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

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 bridge-k közbeiktatásával növelhető a méret. Ez azonban nem jelent komoly megszorítást, mert a HUB-ok egyre inkább kiszorulnak a bridge-k csökkenő ára és komoly előnyei miatt.

A 802.3 MAC-PROTOKOLLJA

A 802.3 keretszerkezetét az **115. ábrán** mutatjuk be.

Minden keret egy 7-bájtos **előtaggal** (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-bájtos 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 megcímezését. 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 üzenet-szórás (broadcast).

A címezésnél 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 $7 \cdot 10^{13}$ globális cím létezik.

A célok megtalálásának módja már a hálózati rétegre tartozik. Ezt a $6 \cdot 8$ bájtot megegyezés szerint hexadecimális alakban, bájtanként kettőspontokkal elválasztva adják meg, például:

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

A **adatmező hossz** (length field) az adatmezőben található adatbájtok számát adja meg. A minimum 0, a maximum 1500 bájt. 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 adata része 46 bájtnál rövidebb, akkor kitöltő 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 algoritmus 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, 1, 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.

Az alapgondolat az, hogy 46 bitet használva már a világ bármely két állomása megcímezheti egymást.

Bár egy 0 hosszúságú adatmező érvényes, de problémákat okozhat. Amikor egy adó-vevő ütközést érzékel, csonkolja az aktuális keretet, ami azt jelenti, hogy kóbor bitek, keretdarabkák mindig jelen lehetnek a kábelben.



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 választanak, akkor 100 egyszerre adni akaró állomás keretei addig ütköznenek, 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 az hibátlan, akkor erről a tényről egy nyugtakeret küldésével értesítenie kell a forrást.

Általában ez a nyugtázás egy másik keretet igényelne, amelynek elküldése érdekében, akárcsak egy adatkeret esetén, meg kell szereznie a csatorna-hozzáférési jogot. Ez megoldható úgy, hogy a sikeres adásokat követő közegért történő versengés során a célállomásnak prioritást biztosítunk.

Kétségtől jelenleg ma a legnépszerűbb hálózat az Ethernet, amelyet rugalmassága és egyszerűsége és olcsósága biztosít. A közeg-hozzáférési algoritmus is 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.

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 behatározni, mint a csillag kialakítású UTP-s rendszer esetén.

VEZÉRJELES SÍN (VEZÉRJEL-BUSZ)

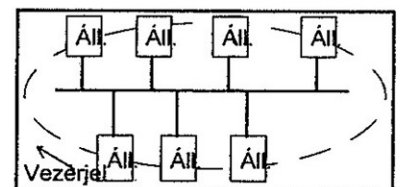
[1] Mint azt már az előbbieken leírtuk, a gyártásautomatizálás sokszor igényel valós idejű, vagy felülről korlátos válaszidejű

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

10 Mb/sec-os vezeték nélküli Ethernet kártya.

(Newsbytes) Az FCC (az Egyesült Államok telekommunikációt felügyelő hivatala) bemerte a Clarion 10 Mb/s átviteli sebességű vezeték nélküli Ethernet kártyáját és engedélyezte annak forgalomba kerülését.

Az eszköz kompatibilis bármilyen Ethernet-kártyával, és amellyel akár 100 méter távolság is áthidalható vezetékek nélkül.



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 időig tart egy keret átvitele, akkor bármelyik állomás $k \cdot T$ időn belül képes kommunikálni (felső korlát).

Sajnos a gyűrű, mint fizikai topológia kevésbé 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 gyűrű szervezés azt jelenti, hogy minden állomás ismeri a közvetlen bal és a 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. **Küldési joga csak a tokent birtokló állomásnak van, ezért ütközés nem jöhet létre.** A gyűrűhöz csatlakozó állomások minden üzenetet vesznek, de csak a neki szólót veszik figyelembe. A MAC protokoll nagyon bonyolult, 10 különböző időzítést, és számos belső állapotváltozót használ.

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

Három különböző modulációs módszer használható: fázis-folytonos ill. fázis-koherens frekvenciamoduláció, valamint amplitudó moduláció. A lehetséges sebességek: 1.5 ill. 10 Mbit/s. A modulációs technikák a kábel 0, 1 és tétlen állapota mellett még további három állapot fennállását is lehetővé teszik, amelyeket hálózatvezérlési célokra, jelzésátvitelre használnak.

A SÍN MAC PROTOKOLLJA

A gyűrű üzembe helyezésekor az állomások a gyűrűbe cím szerint csökkenő sorrendbe kerülhetnek be. A vezérjel küldés is mindig a



Fizikailag lineáris busz kialakítása miatt üzenetszórásos módot használ

Fontos megjegyezni, hogy a sínhez való fizikai csatlakozás nem jelent azonnal gyűrűhöz való csatlakozást is: az állomások gyűrűbe illesztése, illetve eltávolítása a vezérjel-busz MAC-protokolljának a hatáskörébe tartozik.

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 prioritás megvalósításának fontossága miatt négy, növekvő prioritási osztály van a forgalom számára: 0, 2, 4, 6. Ez utóbbi a legnagyobb prioritású.

Működési szempontból úgy is tekinthetjük, mintha minden állomás négy különféle prioritású alállomásból állna: az érkező kereteket a prioritásuk szerinti alállomás dolgozza fel.

Amikor a vezérjel megérkezik egy állomáshoz, annak 6-os alállomása aktivizálódik. Ha van kerete, azonnal kezdi küldésüket. Amikor végzett (vagy amikor az időzítése lejárt), a vezérjelet belül átadja a 4-es alállomásnak, amely szintén az időzítésének lejártáig küldhet kereteket. Ezután az is továbbadja a vezérjelet 2-es prioritású alállomásnak, majd az a 0-ás alállomásnak, ahol keretek elküldésre kerülnek. Ezek után a vezérjelet a következő állomásnak kell továbbküldeni.

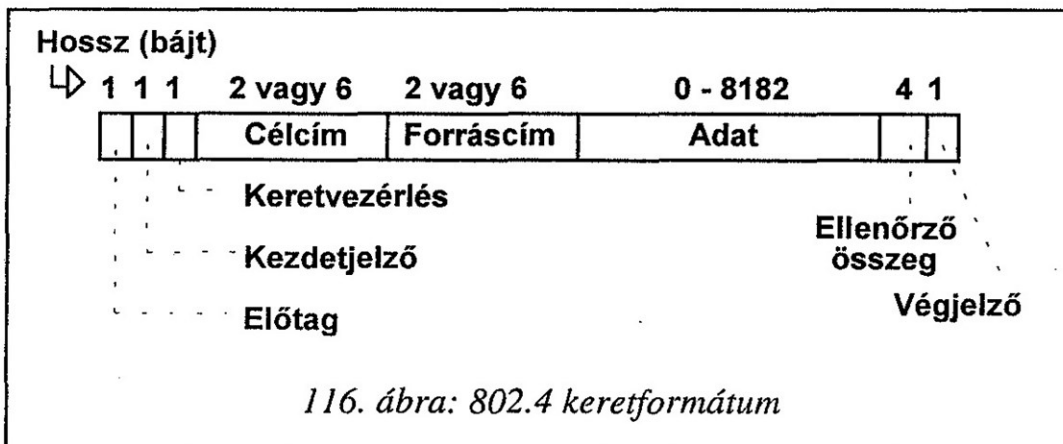
Az időzítések megfelelő beállításával elérhető például az, hogy a teljes vezérjel-birtoklási idő egy jól meghatározott része a 6-os prioritású forgalomé legyen.

Ez lehetővé teszi, hogy a hálózati adatátviteli kapacitás egy adott részét a 6-os prioritású forgalom számára tartsa fenn, és pl. hang vagy más valósídejű forgalom lebonyolítására használható. Az alsóbb prioritásoknak számára a maradék idő áll rendelkezésre. A vezérjeles sín keretformátuma a **116. 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

Ha például 50 állomásos, 10 Mbit/s-os sebességű hálózat paramétereit úgy állítjuk be, hogy a 6-os prioritású forgalom a teljes kapacitás 1/3-át foglalja le, akkor az alállomások $1/3 * 10.000/50 = 67$ kbit/s-os garantált sebességgel rendelkeznek a 6-os prioritású adatok átviteléhez.

Ez a sebesség állomásonként például egy kis, a vezérlőinformációk átvitelére alkalmas maradék kapacitással együtt, egy ISDN hangcsatorna megvalósításához elegendő.



kódolásától jelentősen különböznek. A speciális határoló jelek alkalmazása miatt nincs szükség adathossz mezőre.

A **keretvezérlés-mező** 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 gyűrű-karbantartási keretektől áll. Ez utóbbiak között vannak az állomásokat a gyűrűbe be- illetve kiléptető kerettípusok.

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.

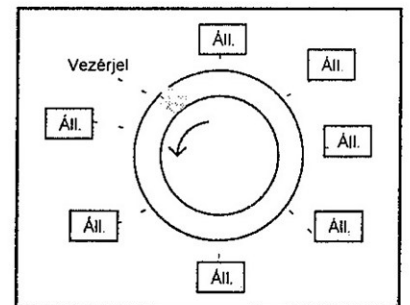
A vezérjeles sínen az időzítésekkel lehet korlátozni a hosszú keretek küldését, egyébként viszont nagyon kényelmes hosszú kereteket küldeni akkor, ha követelmény a valós időben történő feladatvégrehajtás. Az átviteli hibák kiszűrésére az **ellenőrzőösszeg-mező** szolgál. Ugyanazt az algoritmust használja, és ugyanúgy több tagú, mint a 802.3-é.

Megjegyezzük, hogy a 802.3 szabványban vezérlő keretek nincsenek.

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).

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 gyűrűtechnológia majdnem teljesen digitális, szemben pl. a 802.3-al, amely jelentős mennyiségű analóg elemet tartalmaz az ütközések érzékeléséhez. 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]



A gyűrűben zajló átvitel tervezésénél és elemzésénél alapvető kérdés egy bit "fizikai hossza". Ha egy gyűrű K Mbit/s-os adatátviteli sebességgel rendelkezik, akkor $1/K$ μ sec-onként kerül ki egy bit az átviteli közegre. A tipikus $200 \text{ m}/\mu\text{s}$ -os jelterjedési sebességgel számolva ez azt jelenti, hogy egy bit megközelítőleg $200/K$ métert foglal el a gyűrűn.

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.

Ha az állomások tétlenek, a vezérjeles gyűrűben, egy speciális bitminta, az ún. vezérjel (token) jár körbe. 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 egyes állomások 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. Az állomás megőrizheti — az eredeti bitekkel való összehasonlításhoz — vagy el is dobhatja azokat. Ez a gyűrűszerkezet nem korlátozza a keretek maximális méretét, hiszen az egész keret egyszerre úgy sem jelenik meg a gyűrűben.

Miután egy állomás az utolsó keretének utolsó bitjét is elküldte, a vezérjelet vissza kell helyeznie a gyűrűbe. Az utolsó bit visszaérkezése — és a gyűrűből való kivonása után az interfésznek azonnal vételi üzemmódba kell visszaállnia, nehogy ismét kivonja a vezérjelet, amely az utolsó bitet követően érkezik, hacsak addigra már egy másik állomás meg nem szerezte.

Emiatt például 1 Mbit/s-os gyűrű, amelynek kerülete 2000 m, csak 10 bitet tartalmazhat egyszerre.



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

Rövid gyűrű esetén ezért éjszakára mesterséges késleltetéseket illesztnek be, így teszik képessé a gyűrűt a vezérjel további fenntartására és keringtetésére.

Ha a keret üzenetszórásos típusú, az több állomásnak szól, akkor ennél sokkal bonyolultabb nyugtázási mechanizmust használnak

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.

Amikor a forgalom kicsi, akkor a vezérjel a működési idő legnagyobb részében a gyűrűben körbe-körbe fut. Alkalmasszerűen 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, ciklikus multiplexálás jelleggel, 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. Az IBM verziója, akárcsak a legtöbb vezérjeles gyűrű, 16 Mbit/s-os sebességen működik. A jeleket a különbségi Manchester-kódolással kódolják. A magas és alacsony logikai értékeket 3,0-4,5 V közötti pozitív, ill. negatív jelek képviselik.

Sajnos a gyűrűhálózatokban a kábel megszakadása esetén az egész gyűrű működése megszűnik.

A probléma megoldása: a huzalközpont (wire center), amely a **117. ábrán** látható. Minden állomás egy bejövő és egy elmenő vezetékkel kapcsolódik hozzá.

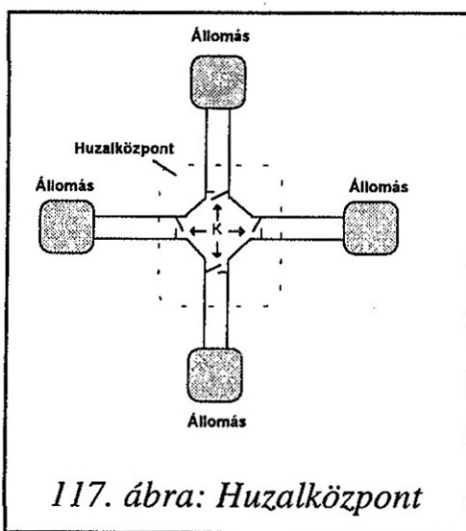
A huzalközponton belül egy állomás be és kimenő vezetékét rövidre záró ún. terelő relék (az ábrán K-val jelölve) vannak, amelyeket az állomások látnak el árammal. Ha a gyűrű megszakad, vagy egy állomás meghibásodik, akkor a tápáram hibája miatt a relé elenged, így az állomás kikerül a logikai gyűrűből.

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

Nagy terhelés esetén a hálózat hatékonysága a 100%-ot is elérheti.

Rendesen a különbségi Manchester-kódolás magas-alacsony és alacsony-magas váltásokat használ a bitek jelzésére, de a 802.5 bizonyos vezérlőbájtokban (pl. keretek elejének és végének jelzésére) alacsony-alacsony és magas-magas átmeneteket is használ. Ezek a nem adat jellegű jelek csak egymást követő párokban fordulnak elő azért, hogy ne idézzenek elő egyenfeszültségű komponenszt a gyűrűn.

Amikor a hálózat több egymástól messze fekvő állomáscsoportból áll, akkor logikusan bővítve több huzalközpontból álló topológia is létrehozható úgy, hogy az ábrán levő állomás kábelpárok egyikét egy távoli huzalközpontba tartó kábelpár váltja fel.)

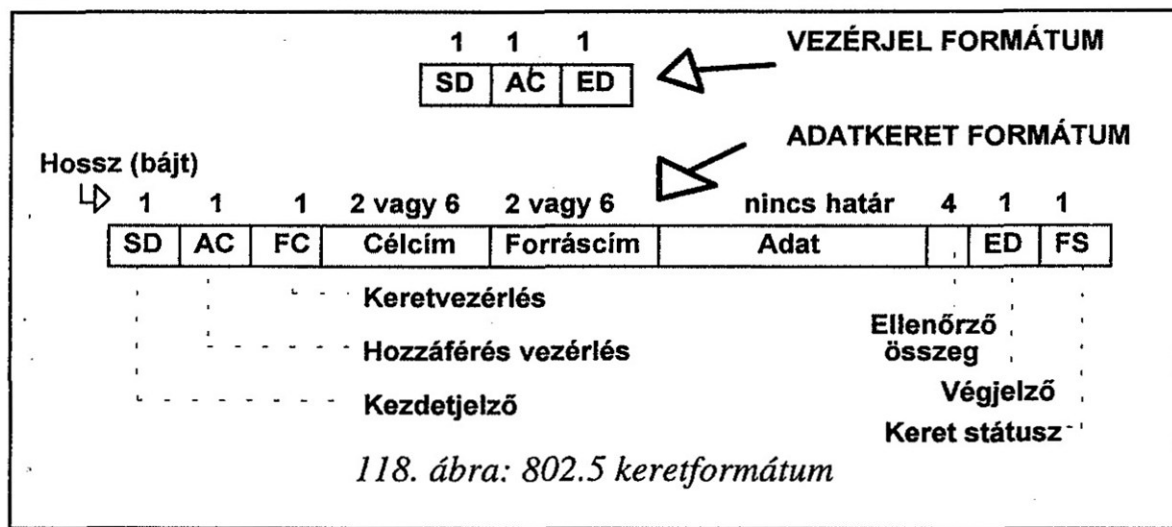


A VEZÉRJELES GYŰRŰ MAC PROTOKOLLJA

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

Ezáltal az első két bájttal keretkezdet szekvenciává alakul át. Az állomás ezután a 118. ábrán látható módon egy normál adatkeret további részeit kezdi el küldeni.

Bár a 802.5 szabvány formálisan nem követeli meg az ilyen huzalközpontot, de a gyakorlatban elvárják, hogy a 802.5 hálózatok a megbízhatóság és a karbantarthatóság növelése érdekében huzalközponttal rendelkezzenek. A kialakítására az IBM a strukturált kábelezést használja.



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 adó-állomásnak 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.

Az ábrán látható **Kezdetjelző** és **Végjelző** mezők a keretek elejét és végét jelzik. Az adatbájtoktól való megkülönböztetősége érdekében, érvénytelen különbségi Manchester mintákat (HH és LL) tartalmaznak. A **Hozzáférési vezérlés** mező tartalmazza a vezérjelet, valamint a **Figyelőbitet**, a **Prioritásbiteket** és a **Lefoglalásbiteket**.



Az adatkereteket a vezérlőkeretektől a **Keretvezérlés** bájttal különbözteti meg. Ezeket a **Célcím** és a **Forráscím** mezők követik, amelyek ugyanazok mint 802.3-ban és 802.4-ben. Ezután az adatmező következik, amely tetszőleges hosszúságú lehet (!), hosszát csak a vezérlő tartási idő korlátozza. Az **Ellenőrzőösszeg** mezője megegyezik a 802.3-aséval és 802.4-esével.

Egy érdekes, a másik két protokollban nem létező bajt a **Keretstátusz-bajt**. Ez tartalmazza az A és C biteket. Amikor egy keret megérkezik a célcímmel megegyező állomás interfészéhez, a keret elhaladása során az interfész bebillenti az A bitet.

Ha az interfész be is másolja a keretet az állomás memóriájába, akkor a C bitet is bebillenti. A keret bemásolása puffer-hiány vagy egyéb más okokból megghiúsulhat. Amikor egy állomás kivonja az általa elküldött keretet, megvizsgálja az A és C biteket. Három kombináció lehetséges:

1. $A = 0$ és $C = 0$ - a célállomás nem létezik, vagy nincs bekapcsolva.
2. $A = 1$ és $C = 0$ - a célállomás létezik, de nem fogadta a keretet.
3. $A = 1$ és $C = 1$ - a célállomás létezik és a keretet bemásolta.

Ez az elrendezés a keretek egyidejű nyugtázását is biztosítja. Ha egy keretet visszautasítanak, de a cél létezik, akkor a küldő opcionálisan egy kis idő múlva ismét próbálkozhat. Mivel a Keretstátusz bajt az ellenőrzőösszeg hatáskörén kívül van, ezért az A és C biteket a keret belsejében megismételték kompenzálva a megbízhatóság csökkenését.

A végjelző egy E bitet tartalmaz, amelyet akkor billent be egy interfész, ha hibát érzékel (pl. egy nem engedélyezett Manchester-mintát fedez fel). Tartalmaz még egy olyan bitet is, amelynek segítségével egy logikai sorozat utolsó keretét lehet megjelölni, azaz hasonló jellegű, mint egy állományvége (EOF=end-of file) jel.

A 802.5 többszintű prioritáskezelésre alkalmas. A 3 bájtos vezérlő középső bájtnak egyik mezője a vezérlő prioritását adja meg. Amikor egy állomás egy p prioritású keretet akar küldeni, akkor addig kell várnia, amíg egy olyan vezérlő el nem tud kapni, amelyiknek prioritása kisebb vagy egyenlő p -nél. Továbbá, egy állomás a következő vezérlő lefoglalását megkísérrelheti úgy is, hogy az éppen áthaladó keret lefoglalásbitjeit olyan prioritásúvá írja át, amilyen prioritással rendelkező keret el kíván küldeni.

Ha azonban ezekbe a bitekbe már nagyobb prioritást jegyeztek be, akkor az állomás lefoglalási kísérlete sikertelen lesz. Az aktuális keret elküldését követően a visszaállítandó vezérlő prioritásának meg kell egyeznie az eredeti lefoglalt vezérlő prioritásával.



Kis gondolkodással belátható, hogy ez a mechanizmus egyre följebb és följebb emeli a lefoglalási prioritást. A probléma megoldására a protokoll néhány összetettebb szabályt fogalmaz meg. A gondolat lényege, hogy egy prioritást emelő állomás az emelés végrehajtását követően, a prioritás csökkentés felelősségé válik.

A 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 nincs felügyelő állomás, egy Claim token vezérlőkeretet küldhet el. Ha ez a keret anélkül visszaér a küldőhöz, hogy valaki más ugyancsak Claim token keretet küldött volna, akkor maga a küldő válik felügyelővé (minden állomásba beépítik a felügyelővé válás képességét). A vezérjeles gyűrű vezérlő kereteit a következő táblázatban foglaltuk össze:

A vezérjeles gyűrű vezérlő keretei

Keretvezérlő mező	Név	Feladata
00000000	Duplicate address test	Ellenőrzi, hogy van-e két azonos című állomás
00000010	Beacon	A gyűrűszakadás lokalizálásához
00000011	Claim token	Próbálkozás felügyelővé válásra
00000100	Purge	A gyűrű újraindítása
00000101	Active monitor present	A felügyelő periodikusan bocsátja ki
00000110	Standby monitor present	Potenciális felügyelő jelenlétét hirdeti ki

A felügyelő felelős többek között a vezérjel-vesztés figyeléséért, a gyűrűszakadaskor 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.

A vezérjel-vesztést a felügyelő állomás egy, a lehetséges leghosszabb vezérjel nélküli intervallum értékére beállított időzítéssel ellenőrzi.

Ezt abból a feltételezésből kiindulva számítja ki, hogy minden állomás teljes vezérjel-tartási idejét kihasználva ad. Ha ez az időzítés lejár, akkor a felügyelő megtisztítja a gyűrűt, és egy új vezérjelet állít elő.

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

A felügyelő ekkor magán keresztül bocsátva felnyitja, majd megtisztítja a gyűrűt, és új vezérjelet bocsát ki. Az árva keretek kiszűrését úgy végzi el, hogy minden keresztülhaladó keret hozzáférési vezérlés mezőjében bebillenti a felügyelőbitet.

Ha egy bejövő keretben ez a bit már beállított, akkor ez arra hívja fel a figyelmet, hogy a keret eltávolításáért felelős állomás valószínűleg hibás, hiszen csak így fordulhat elő, hogy a keret már másodszor halad át a felügyelőn. A felügyelő állomás ekkor maga távolítja el ezt a keretet.

Az egyik monitorfunkció a gyűrű hosszával kapcsolatos. A vezérjel 24 bit hosszú, ami azt jelenti, hogy a gyűrűnek elég hosszúnak kell lennie ahhoz, hogy 24 bitet egyszerre tartalmazhasson.

A gyűrű szakadási helyének behatárolását a felügyelő állomás nem képes egyedül megoldani. Amikor egy állomás valamelyik szomszédját működésképtelennek érzékeli, akkor egy Beacon keretet bocsát ki, amelyben megadja a feltételezhetően hibás állomás címét. Ezeket az állomásokat a huzalközpontban levő terelő-relek segítségével emberi beavatkozás nélkül ki lehet iktatni a gyűrűből.

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 új állomásokat vehet fel 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 egy központi felügyelő állomással sokkal egyszerűbb. Továbbá egy normál rendszerben az állomások csak nagyon ritkán mennek tönkre, ezért az új felügyelő állomás alkalmoszerű versengéses megválasztása nem okoz nagy problémát. Ennek ára azonban az, hogy ha a felügyelő állomás egyszer valóban meghibásodik, de ugyanakkor Activ monitor present vezérjeleket küld periodikusan, akkor nincs olyan állomás, amely ezt megkérdőjelezhetné. A felügyelő állomást nem lehet felelősségre vonni.

Ha az állomások 1-bites késleltetése, plusz a kábel késleltetése kisebb mint 24 bit, akkor a felügyelő külön késleltetésekkel biztosítja a vezérjel keringtetetőségét.

FDDI

A ma már elavuló FDDI alap gondolata: üvegszál alapú lokális hálózat, ha drágán is, de kialakítható.

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 reléekkel van felszerelve, amelyek a gyűrűk összekapcsolására, és a meghibásodott állomások kiiktatására használhatók.

Az FDDI multi-módusú üvegszálakat használ olcsóbb volta és kisebb veszélyessége (nem lézertény, csak LED) miatt.

Az FDDI két állomástípust határoz meg:

az A osztályú állomást, amely mindkét gyűrűhöz kapcsolódik, és a B osztályút, amelyik csak az egyikhez.

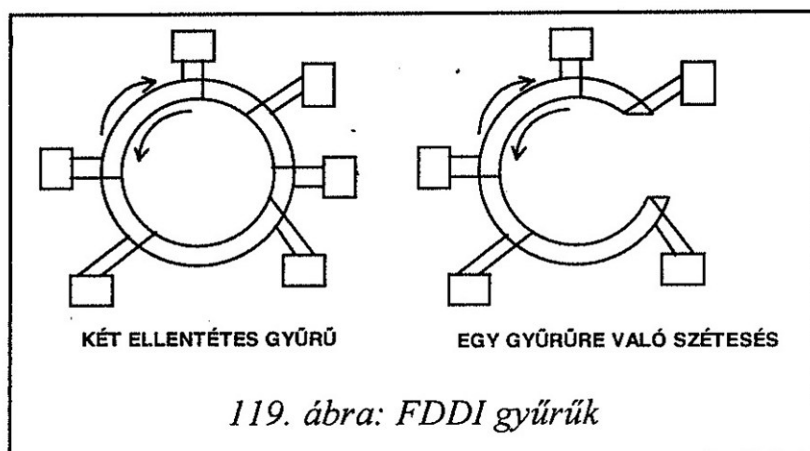
Az igényektől és a költségektől függően üzembe helyezéskor tiszta A, tiszta B, vagy kombinált típusú állomásokból építhetjük fel a hálózatot.

A fizikai réteg nem használ Manchester-kódolást, mert a 100 Mbit/s-os Manchester-kódolás 200 Mbit/s-ot követelne, és ez túl költséges lett volna. Ehelyett az ún. 4 az 5-ből (4 out of 5) kódolást használják. Minden 4 MAC szimbólumból (0-kból, 1-ekből, és bizonyos nem adat jellegű szimbólumból, pl. keretkezdetből) álló csoport 5-bites csoporttá kódolva jelenik meg a közegen. A lehetséges

32 kombinációból 16 az adatok, 3 a határolók, 2 a vezérlés és 3 a hardverjelzés számára van fenntartva, 8 egyelőre kihasználatlan.

E kódolásnak az az előnye, hogy kisebb sávszélességet igényel, hátránya viszont az, hogy elveszíti a Manchester-kódolás önszinkronizáló tulajdonságát, azaz a bitváltásokat nem lehet órajelként használni. Ennek kompenzálására a küldő a vevő órajelének szinkronizálására egy hosszú előtagot küld a keret elején. Emiatt az is követelmény, hogy az összes órajel frekvenciájának legkevesebb 0,005 %-on belüli pontosságúnak kell lennie.

Ilyen stabilitás mellett legfeljebb 4500 bájt hosszú keret küldhető el a szinkronizmusból való kiesés veszélye nélkül.

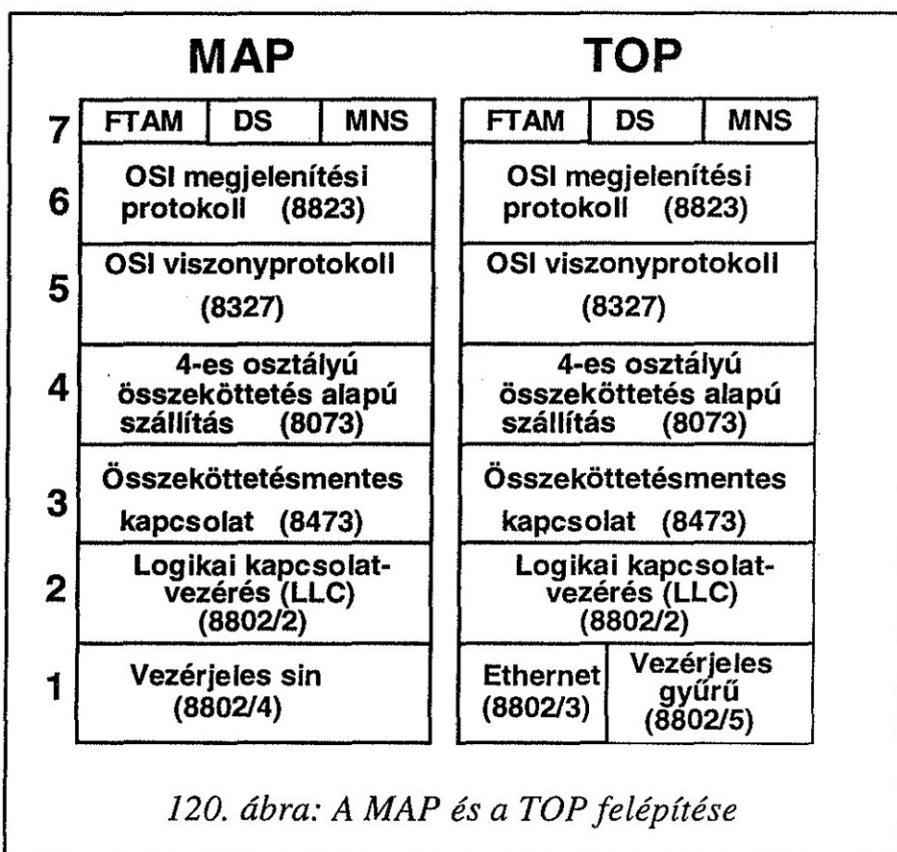


Az alap FDDI protokoll modellje a 802.5 protokollon alapszik.

A MAP ÉS TOP

Az IEEE 802 szabvány csak a hálózati réteget szabványosítja a LAN hálózatokhoz. Azért, mert a három szabványban a közeghozzáférési módszerek eltérnek, nem célszerű a teljes felépítést különállóan kezelni. Ez vezetett két, a szabványokon alakuló protokollhoz:

- a valós idejű működést követelő **MAP (Manufacturing Automation Protocol** — gyártásautomatizálási protokoll),
- és az ilyen igényt nem támaztó **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 kompatibilisek, azonos protokollokat használnak.

Az első két szintről már írtunk, nézzük a felsőbb rétegeket. Az összeköttetésmentes 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 forgalom szabályozást 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 fölülte alacsonyabb osztályba tartozó szállítási protokollt lehetne használni.)

A viszony- és a megjelenítési és alkalmazási rétegekben 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 protokoll készlete.

A TOP hálózatok ötféle fizikai eszközt használnak az összeköttetések megvalósítására: a hosztokat, jelisméltöket, hidakat, útválasztókat (routereket) és átjárókat (gateway).

A hosztok lényegében az információ forrásai és céljai.

Jelismétlőről (repeater), jelelosztóról (HUB) már az Ethernet hálózat kapcsán írtunk, feladatuk a jelregenerálás, bitek továbbítása az egyik hálózatból a másikba.

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. Lényegében egy kommunikációs számítógép.

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.

Összekötőréteg	Protokollok		
	Fizikai réteg	Adatkapcs. réteg	Hálózati réteg
Repeater	ugyanaz	ugyanaz	ugyanaz
Bridge	különböző	ugyanaz	ugyanaz
Router	különböző	különböző	ugyanaz
Gateway	különböző	különböző	különböző

LOKÁLIS HÁLÓZATI OPERÁCIÓS RENDSZEREK

Az OSI-modell két legalacsonyabb szintjének funkcióit már szabványosították, és a legtöbb LAN összhangban van a három IEEE-szabvány valamelyikével.

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.

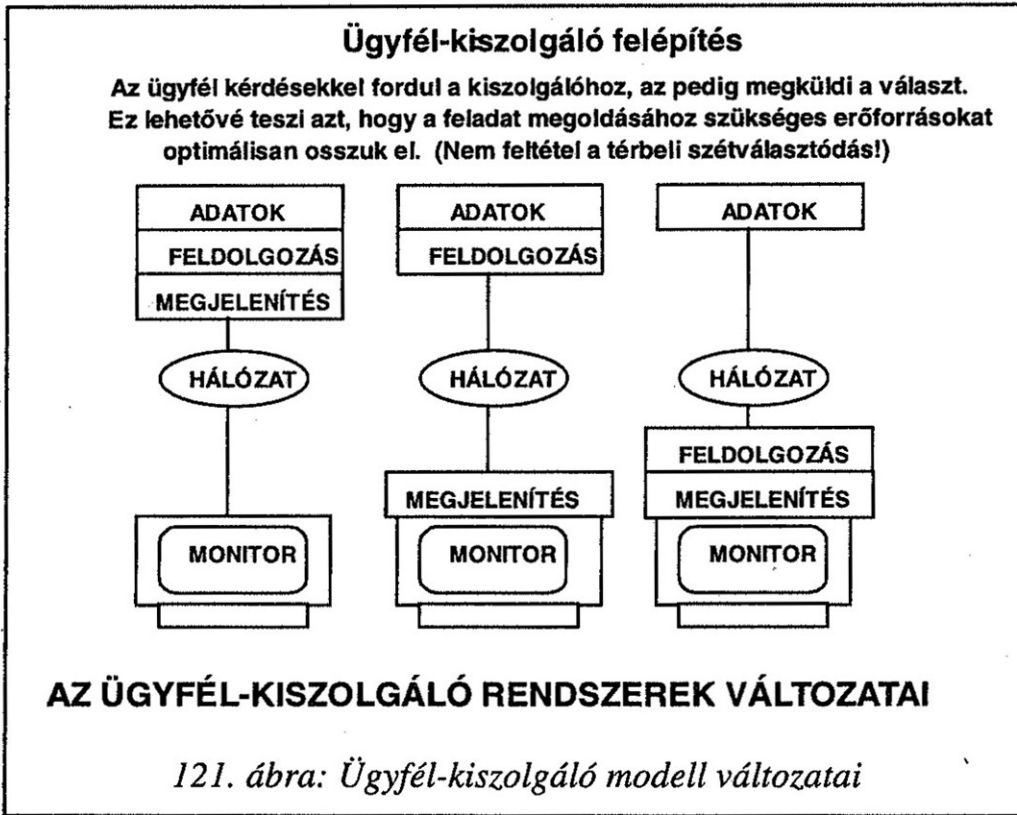
LAN-eszközökkel megvalósított fizikai hálózat az állomásai számára általános adatcserére alkalmas kommunikációs szolgáltatást nyújt. Ahhoz azonban, hogy megkapjuk a lokális hálózat valamennyi előnyét, az általános kommunikációs szolgáltatáson túl további funkciókat is meg kell valósítanunk. Ezek a funkciók a hálózat magasabb szintű rétegeihez tartoznak, és ezeket a hálózati operációs rendszerként ismert programrendszer szolgáltatja.

A 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.

A hálózati operációs rendszer egy szoftver, amely a hálózatba kapcsolt eszközökön fut, és feladata az eszközök közötti kommunikációs szolgáltatások biztosítása.

A hálózati eszközök kapcsolatában két megoldási mód lehetséges:

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 kliens 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.

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 begépelte adatokat átküldi pl. soros vonalon a szerver gépre, az ott futó program ez ily módon begépelte parancsok alapján a keresést végrehajtja, és a keresett rekordokat visszaküldve a soros vonalon a terminál azokat megjeleníti.



Egy lehetőség lehet az is, hogy a kikeresett adatokat a szerver csak „ömlesztve”, nyers formában küldi vissza a kliensnek, ahol a futó programmegfelelő formában megjeleníti. Egy másik esetben a kereső program a kliens gépen fut: a keresés végrehajtásához szükséges adatbázis rekordokat a vonalon a szerver elküldi a kliensnek, az leküldött részen végrehajtja a keresést, majd a továbbiakban leküldött részekkel folytatja.

Egy rendszerben természetesen egynél több szerver is elképzelhető.

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 ún. egyenrangú, vagy peer-to-peer hálózatok. Ilyen hálózatokban is elképzelhetők, hogy az egyik gép csak szerverként működik.

LOKÁLIS HÁLÓZATI OPERÁCIÓS RENDSZEREK FUNKCIÓI

Foglaljuk össze, hogy melyek a hálózati operációs rendszerek által kínált közismertebb funkciók:

Fájlszerver A nagy kapacitású lemez a legfontosabb erőforrás, amelyet a hálózat megoszthat. A fájl-szerver általában egy számítógép, amely a hálózati kapcsolatán túlmenően a nagy kapacitású merevlemez meghajtót kezeli. A fájl-szerver teszi lehetővé az állomások számára a tárolt fájlokhoz való hozzáférést. Ez a fájl-megosztás, 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. Egyes hálózati operációs rendszerek rendelkeznek rekordlezáró szolgáltatással, és így a program "le tud zárni" egy megadott információt, megakadályozva, hogy bármely más program ehhez hozzáférjen.

Nyomtatószerver 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 még akkor is a nyomtatóhoz rendelhető, amikor a

Ezt azért fontos megjegyezni, mert nem a gépek információ-szolgáltatásban nyújtott szerepe a döntő, hanem az, hogy az egyenrangúság értelmében bármelyik lehet ügyfél és szolgáltató.



A jogosultságoknak külön a fájlokhoz és a felhasználókhöz történő hozzáféréssel a hozzáféréseket finoman lehet szabályozni. Ilyen jogosultságok lehetnek a fájlok megnyitásának, módosíthatóságának, írhatóságának, létrehozásának, másolásának, törlésének engedélyezése.

nyomtató foglalt. Erre a megoldásra a nyomtató viszonylagos lassúsága miatt is szükség van.

Elektronikus levelezés Egyes hálózati operációs rendszerek lehetőséget nyújtanak elektronikus postai alkalmazásra: Ez a felhasználónak lehetővé teszi, hogy üzeneteket és dokumentumokat könnyen összeállítson, küldjön, fogadjon és tároljon. Ezen a módon a lokális hálózat átlátszó adatátviteli szolgáltatásával a felhasználók könnyen tudnak egymással kommunikálni. A hálózatba való belépéskor az ilyen rendszer a felhasználónak képes jelezni, hogy levele érkezett-e (ez a hálózat használata közben természetes).

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 megjelölésére 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 nevekkkel (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 számítógép számára lehetővé teszi, hogy nagy távolságú távközlési berendezésen, pl. telefonvonalon keresztül hozzáférjen a lokális hálózathoz. Ez az ún. távoli hozzáférés (remote access). A lokális hálózathoz tartozó állomás számára lehetőséget ad, hogy a hálózat egyik állomása által támogatott osztott kommunikációs szolgáltatás felhasználásával hozzáférjen a hálózathoz nem tartozó számítógéphez. Az osztott kommunikációs szolgáltatással rendelkező állomást néha kommunikációs szervernek nevezik. Az így elért számítógépet közvetlenül vagy távoli kommunikációs berendezésen keresztül kapcsolhatják a lokális hálózathoz.

Két vagy több lokális hálózat összekapcsolásakor a hálózatok lehetnek azonos vagy különböző típusúak. A hálózatok összekapcsolhatók közvetlenül vagy távolsági hálózaton keresztül.

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 elérésére, megbízhatóságára vonatkozó szervezési szolgáltatásokkal, de ezeknek a szolgáltatásoknak pontos természete hálózatról hálózatra változhat. Egyes esetekben a hálózatszervezés csak egyetlen lokális hálózattal foglalkozik. Máskor egy nagyobb szolgáltatás része lehet, amely egy csoport egymással összekötött hálózat és átviteli berendezés szervezésével foglalkozik.

NOVELL NETWARE FELHASZNÁLÓI ISMERETEK — ÖSSZEFOGLALÓ

A hálózati operációs rendszerek illusztrálására, egy rövid összefoglalót adunk az egyik legnépszerűbb ilyen rendszer a Novell cég PC-ken futó Netware 3.12-es operációs rendszeréről.

Bár jelenleg már az 5.0-ás verzió is kapható, de az itt bemutatott alapelvek változatlanok maradtak.

A PC-s hálózatokban mind a szerverek mind a munkaállomások (kliensek) eltérő teljesítményű IBM PC kompatibilis számítógépek. A munkaállomások általában önmagukban is használhatók, és rajtuk a DOS (illetve WINDOWS) operációs rendszer fut.

A szerver operációs rendszere a nagyobb teljesítményt nyújtó, többfeladatúságot biztosító **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 kb-ot foglalnak el a munkaállomás memóriájában.

Ezért ezen programok helyett ma már inkább a kártyafüggetlenséget biztosító ODI illesztést (IPX-et helyettesíti), illetve a NETX nagy rezidens helyfoglalása miatt a VLM (Virtually Loadable Modul) illesztést használják.

E programok lefuttatása után a szerverrel való kapcsolat már kiépült. Már létezik a fájlservert jelentő (általában F:) meghajtó, ahol a hálózatra 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 fájlserver és a rajta lévő kötetek nevével:

SZERVERNÉV/KÖTET: alkönyvtár\...\fájlnev.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

SZÁMÍTÓGÉP - HÁLÓZATOK

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 az ún. 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 alkönyvtáira 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] Kijelentkezteti a felhasználót a megadott szerverről.

Természetesen ez a rendszer nagyon rövid összefoglalása, bemutatása, részletes ismeretek a [6,7,8] irodalmakban található.

Az utána megjelent Netware 4.x verziója számos újdotságot tartalmazott:

- 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 az NDS (Netware Directory Service). Ez

Az új 5-ös verzióban a kommunikáció TCP/IP alapú, továbbfejlesztett NDS jellemzi.

lehetővé teszi a fájlrendszer és a hálózatot használó szervezet logikai felépítésének különválasztását.

- a nemzetközi alkalmazások miatt 16 nyelvre lehet installálni — sajnos a magyar még nincs közte,
- Méreténél fogva a mai követelményeknek megfelelően CD-ről installálható, a jogosultságot egy külön érvényesítő hajlékonylemez hordozza,
- adatok tárolása a merevlemezen a felhasználó által nem látható módon tömörítve történik,
- ritkán használt adatokat automatikusan átmenti egy lassú de nagy kapacitású háttértárra pl. mágnesszalagra, és igénylőskor tölti vissza a merevlemezre. Ez az adatvándorlás (migration),
- A fájlokat hordozó lemezblokkok mérete 4 és 8 kb-ot mellett 512 bájt lehet — (suballocation), amivel a fizikai blokkméretet az adatblokkok méretéhez lehet igazítani,
- 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.

ELLENŐRZŐ KÉRDÉSEK



1. Miért előnyös a számítógépek helyi hálózatba kapcsolása?
2. Ismertesse a 802.1-802.5 szabványokat!
3. Miért osztotta a szabvány az adatkapcsolati réteget két alrétegre. Mi ezek feladata?
4. Milyen típusú kábelek használható a fizikai rétegben?
5. Melyek a közeghozzáférés-vezérlési alréteg funkciói?
6. Mi a logikai kapcsolatvezérlési alréteg feladata?
7. Mi az a SAP? Milyen réteghatárom helyezkedik el?
8. Ismertesse a hálózati-LLC rétegek közötti szolgáltatási előírásokat!
9. Ismertesse a LLC-MAC rétegek közötti szolgáltatási előírásokat!
10. Milyen részekből állnak a lokális hálózatok fizikai egységei?
11. Ismertesse a 802.3 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Mi a kapcsolata az Ethernettel?
12. Mit takarnak a MAU és az AUI fogalmak?
13. Milyen kódolást használ a 802.3? Mi a lényege?
14. Ismertesse a 802.3 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?

15. Ismertesse a 802.4 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Milyen kábelezést használ?
16. Ismertesse a 802.4 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?
17. Ismertesse a 802.5 szabványt! Mi a közeghozzáférés módszere? Milyen a hálózat topológiája? Mi a gyűrű felépítés előnye? Milyen kábelezést használ?
18. Ismertesse a 802.5 MAC protokollját! Milyen keretformátumot használ? Milyen a címzés?
19. Hogyan történik a vezérjeles gyűrű karbantartása?
20. Ismertesse az FDDI felépítését. Miért használ két gyűrűt? Hogyan kapcsolódnak a gyűrűre az állomások? Milyen kódolási eljárást használ?
21. Mi az a MAP és TOP?
22. Milyen megoldásokat ismer a különböző hálózatok összeköttetésekének a megvalósítására? Hogyan illeszthetők ezek be a rétegmodellbe?
23. Mi a kliens-szerver felépítés lényege?
24. Mi az egyenrangú (peer to peer) hálózat?
25. Ismertesse a fájl szerver fogalmát és funkcióit!
26. Mit ért az összekapcsolhatóság fogalmán?
27. Röviden jellemezze a Novell Netware 3.12-es operációs rendszert!

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

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.

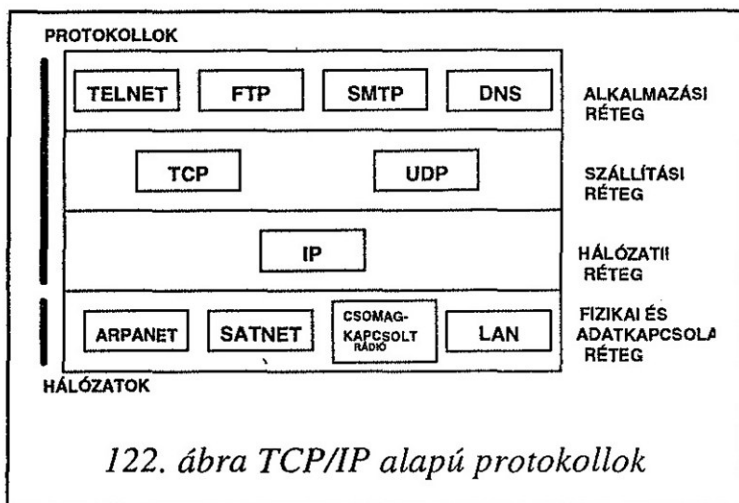
A későbbiekben már Internet-nek nevezett hálózat — bár rétegmodell szemlélettel tervezték az 1960-as években — nem követi az OSI hét rétegű felépítését. A tényleges hálózati kapcsolatot biztosító TCP/IP protokollt UNIX operációs rendszerhez fejlesztették ki, és alapvetően az OSI modell két rétegének a funkcióját valósítja meg: a hálózati és szállítási rétegét. A hálózati modell összesen négy rétegből áll:

Alkalmazási szint (Application) Itt vannak a felhasználói és a hálózati kapcsolatot biztosító programok.

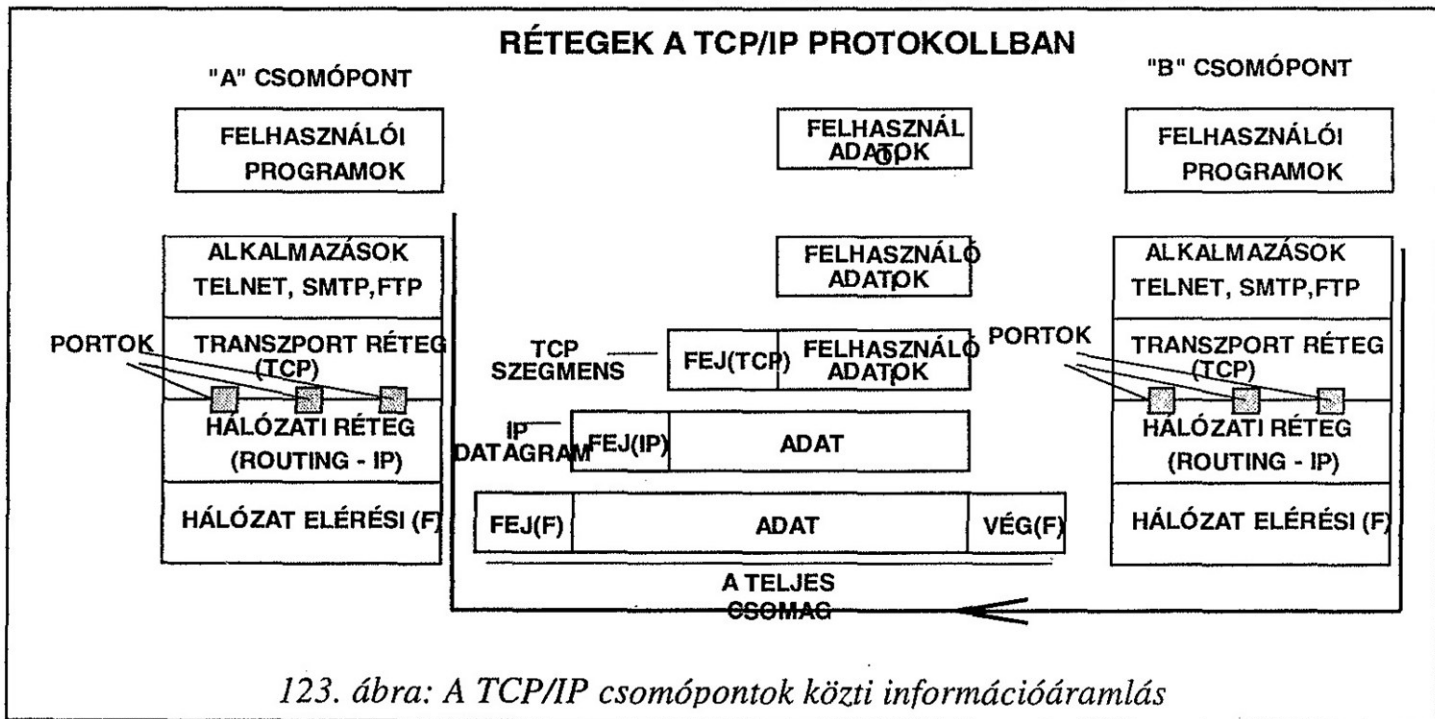
Mivel az Internet hatalmas léptekkel fejlődik, ezért egy külön fejezetet szenteltünk az Internet hálózati kapcsolatos legfontosabb ismeretek összefoglalására.

Itt természetesen már felhasználjuk az előbbieken szereplő ismereteket.

Tehát a TCP/IP protokoll a számítógép hálózat két tetszőleges csomópontja közötti adatátvitelre képes.



122. ábra TCP/IP alapú protokollok



123. ábra: A TCP/IP csomópontok közti információáramlás

Hoszt-hoszt réteg (Transport) Az OSI modell szállítási hálózati rétegének felel meg. A létesített és fennálló kapcsolat fenntartását biztosítja.

Két rétegprotokollból áll: az egyik a Transmission Control Protocol (TCP) azaz a továbbítást szabályozó eljárás, a másik az összekötésmentes szállítási protokoll User Datagram Protocol (UDP)

Hálózatok közötti (Internet) Az OSI modell hálózati rétegének felel meg, ez a réteg végzi az a csomagok útvonal kijelölését a hálózatok között. Ennek a rétegnek a protokollja az Internet Protocol (IP), az üzenetvezérlő protokoll cím meghatározó eljárása, a foglalt címet meghatározó eljárás. A rétegben előforduló események és hibák jelzésére szolgál az Internet Control Message Protocol (ICMP), az Internet Vezérlőüzenet Protokoll.

Hálózat elérési (Network Interface) Az OSI modell két alsó szintjének felel meg, és ez biztosítja a kapcsolatot a csomópontok között. (Pl.: Ethernet, Token-Ring, Token-Bus, PPP).

Az információ áramlása két (legtöbbször távoli) csomópont között az **123. ábrán** látható. Jól követhető a már leírt elv: az üzeneteket a rétegek kisebb részekre bontják, fejinformációkkal látják el, majd ezeket adják tovább.

Ezek közül az Ethetnet a legelterjedtebb

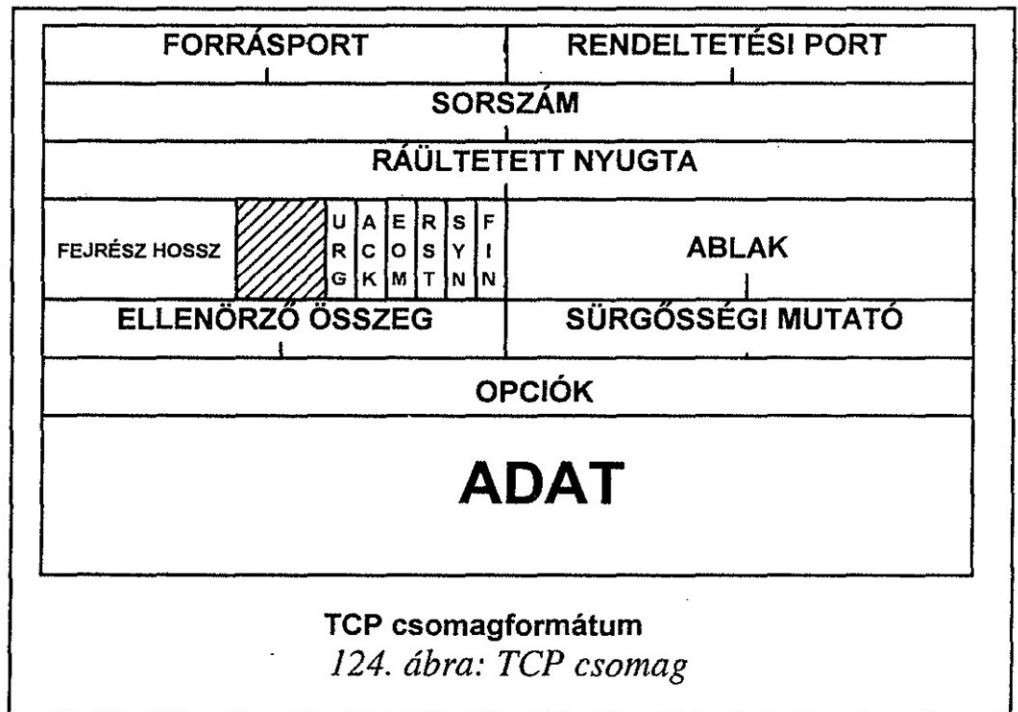
Az eredeti ARPANET-ben az alhálózattól virtuális áramkör szolgáltatást vártak el (azaz tökéletesen megbízhatót).

TCP =
Transmission Control
Protocol
Átvitel vezérlési protokoll

TCP/IP PROTOKOLL

AZ INTERNET SZÁLLÍTÁSI RÉTEGE: A TCP

Az első szállítási réteg protokoll az NCP (Network Control Protocol – hálózatvezérlési protokoll) elvileg egy tökéletes alhálózattal való együttműködésre épült. Egyszerűen átadta a TPDU-kat a hálózati rétegnek és feltételezte, hogy a megfelelő sorrendben kézbesítődnek a célnál. A tapasztalat azt mutatta, hogy az ARPANET-en belül ez a protokoll kielégítően működött.



7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

Ahogy azonban az ARPANET Internetté vált, amelyben már számos LAN, rádiós csomagszóró alhálózat; valamint több műholdas csatorna is működött, azaz a végpontok közötti átviteli megbízhatóság csökkent. Ezért egy új szállítási protokollt, a TCP-t vezettek be. 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.

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 fejléctet fűz, majd ezeket a darabokat egymástól független datagramokként küldi el. A hálózati réteg sem azt nem garantálja, hogy a datagramokat helyesen kézbesíti, sem a megérkezett datagramok helyes sorrendjét.

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 bájtnek saját sorszáma van. A sorszám tartomány 32 bit széles, vagyis elegendően nagy ahhoz, hogy egy adott bájt sorszáma egyedi legyen.

A TCP által használt fejlécszám a 123. ábrán látható. A minimális TCP fejlécszám 20 bájtos. A **FORRÁSPORT** és a **CÉLPÖRT** mezők az összeköttetések végpontjait (TSAP-címek az OSI terminológia szerint) azonosítják. Minden egyes hosznak magának kell eldöntenie, hogy miképpen allokálja (osztja ki) a portjait.

A **SORSZÁM** és a **RÁÜLTETETT NYUGTA** mezők a szokásos funkcióikat hajtják végre. A TCP minden bájtot megsorszámoz, ezért 32 bit hosszúak.

A **FEJLÉCSZÁM HOSSZ** kijelöli, hogy a TCP fejlécszám hány 32 bites szót tartalmaz. Erre az információra az **OPCIÓK** mező változó hossza miatt van szükség.

Ezután 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 sorszámától 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=1**. A **FIN** az összeköttetés lebontására használható, azt jelzi, hogy a küldőnek nincs több adata. A hirtelen hibák miatti nem jó állapotba került összeköttetéseit az **RST** bit használatával lehet



Figyeljük meg, a TCP fejlécszám a forrás és cél gép címét nem tartalmazza, a portok valójában a kommunikáló programok kapui.

A TCP protokoll segítségével a két kommunikáló folyamat „látja” egymást, a köztük lévő távolságtól függetlenül.

Azért, hogy egy állomás egy időben több TCP kapcsolattal rendelkezhesen, az ún. TCP portot is. Ez a 16 bites szám azonosítja a célállomáson belül megszólítandó kommunikációs partnert. A 0 és 1023 közötti portszámok foglaltak, ezeken találhatóak az ismert szolgáltatások (well-known-services), e fölött szabadon használhatóak a portszámok.

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 algoritmus egyszerű: 16-bites szavakként az adatokat összegzik, majd az összeg 1-es komplementjét veszik.

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 bitre ki kell egészíteni
- **Virtual Circuit Connection:** duplex átvitel előtt erőforrásokat kell allokálni mind a két oldalon, a kapcsolatot mindkét oldalon meg kell nyitni, adatátvitel során 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 adatátvitel 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 védelmi az újrásorszámozást alkalmazza, csúszóablakos technikával, vagyis egy időben 1-nél több nyugtázatlan csomag is úton lehet.

Vételkor a képződött összeghez az 1-es komplementst hozzáadva, hibátlan átvitel esetén nullát kapunk.

AZ INTERNET HÁLÓZATI RÉTEGE: AZ IP

A hálózati réteg IP protokollja a 80-as években jelent meg. A protokoll összeköttetés-mentes. A szállított csomagok a datagramok, amely a forrás hoszt-tól a cél hosztig kerülnek továbbításra, esetleg több hálózaton is keresztül.

A hálózati réteg megbízhatatlan összeköttetés mentes szolgáltatot biztosít, így az összes megbízhatósági mechanizmust a szállítási rétegben kell megvalósítani, ami biztosítja a két végállomás közötti megbízható összeköttetést.

Az IP — definiálja az adatátvitel legkisebb egységét, annak pontos formáját, az útválasztást (routingot), valamint néhány további olyan fontos szabályt, amelyek meghatározzák, hogy a hostok, IMP-k hogyan dolgozzák fel az IP csomagokat, mikor és hogy kell hibajelzéseket generálni, mikor kell csomagot eldobni.

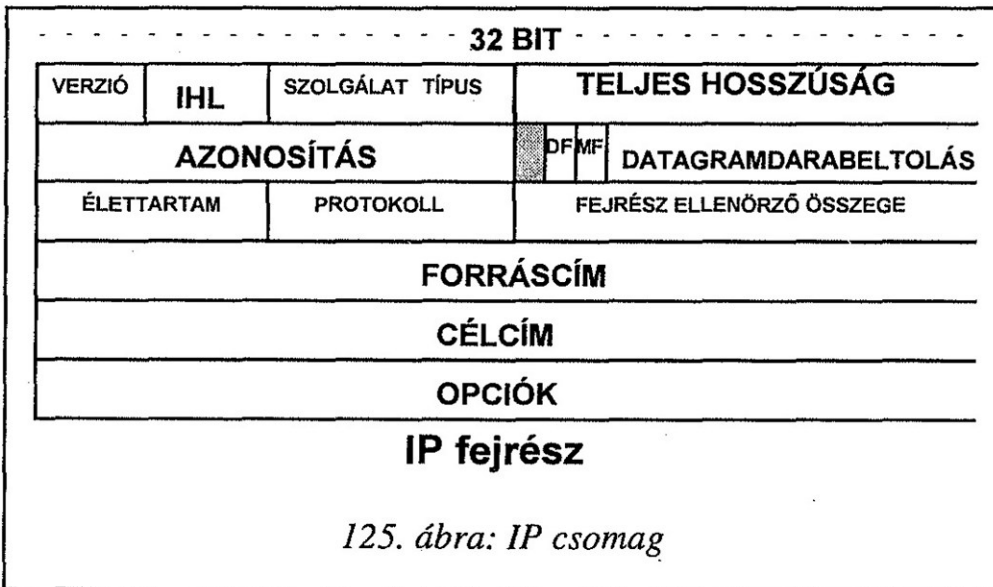
Az IP működése a következő: A szállítási réteg az alkalmazásoktól kapott üzeneteket maximum 64 kbájtos datagramokra tördeli, amelyek az útjuk során esetleg még kisebb darabokra lesznek felvágva. Amikor az összes datagram elérte a célgépet, ott a szállítási réteg ismét összerakja üzenetté. A datagram két részből áll: egy fejrészből és egy szövegrészből. A fejrészben 20 bájtt rögzített, és van egy változó hosszúságú opcionális rész is.

A **VERZIÓ** mező a protokoll verzióját azonosítja, így a protokoll módosítását is ezzel figyelembe lehet venni. Az **IHL** adja a fejrész teljes hosszát 32 bites egységekben (20bájtt+opció rész). Minimális értéke:5. (Nincs opció.)

Angolul Connectionless Packet Delivery Service: NEM megbízható, csupán a legtöbb erőfeszítést megtevő (best-effort).

Vagyis csomagok elveszhetnek, többszöröződhetnek, hibás sorrendben érkehetnek.

Ennek ellenére ez a szint ezeket az eseményeket nem javítja és nem is jelzi a magasabb szintek felé. Csomagvesztésre viszont csak ritkán és indokolt esetben kerül sor.



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.

IP CSOMAGOK TÖRDELÉSE (FRAGMENTÁCIÓJA)

A datagrammokat a különféle hálózati csomópontokon történő átvitel során esetleg kisebb darabokra kell szétbontani (tördelni, vagy más néven fegmentálni, azért, hogy az adatkapcsolati rétegben lévő keretekben elférjen. Minden hálózatra van egy jellemző maximális adatátviteli keretméret: network MTU (Maximal. Transfer Unit).

Például Ethernet MTU = 1500 oktetes (bájtos) IP csomag, X.25 esetén lehet hogy MTU csak 128 bájt. A széttördelt darab neve: Fragment és 8 bájttal oszthatónak kell lennie.

A **fragmentált IP csomagok egyesítése (reassembly)** — csak a végponton történik meg, időzítő felhasználásával, amelynek lejártakor a fragmenteket eldobja

Az **AZONOSÍTÁS** mező alapján állapítja meg a célhost, 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** mező 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** mező neve a More Fragments (több darab) rövidítése. A széttördelt datagramdarabokat jelzi, kivéve az utolsót. A Teljes hosszúság mező mintegy második ellenőrzésként használható, vajon nem hiányzik-e datagram darab, és hogy az egész datagram összeállt.

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.

Hostok és ICMP-ék esetén a minimális MTU 576 oktet

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

Ehhez a kapcsolódik a **DATAGRAMDARAB-ELTOLÁS** 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, amely az elemi datagramdarab hosszúsága.

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.

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.

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 hálózati cím – hálózaton belüli hoszt címet adják meg (ld. mindjárt!!!).

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.

CÍMZÉSI RENDSZER

Mivel az Internet lényegében hálózatok összekapcsolása, a címezési rendszer kialakításánál ezt a valóságos tényt vették figyelembe: *a címezés legyen hierarchikus; azaz vannak hálózatok, és ezen belül gépek (hosztok).*

Így célszerű a címet két részre bontani: egy **hálózatot azonosító**, és ezen belül egy, a **gépet azonosító** címre. A hálózatok közötti kapcsolatot az útválasztók (ruterek) biztosítják.

Mivel a cím hossza 32 bit ezért ezt kellett két részre bontani, olyan módon, hogy a nagy hálózatokban lévő sok gépet is meg lehessen címezni. Öt különböző formátum használható, ahogy ezt az 126. ábrán bemutatjuk.

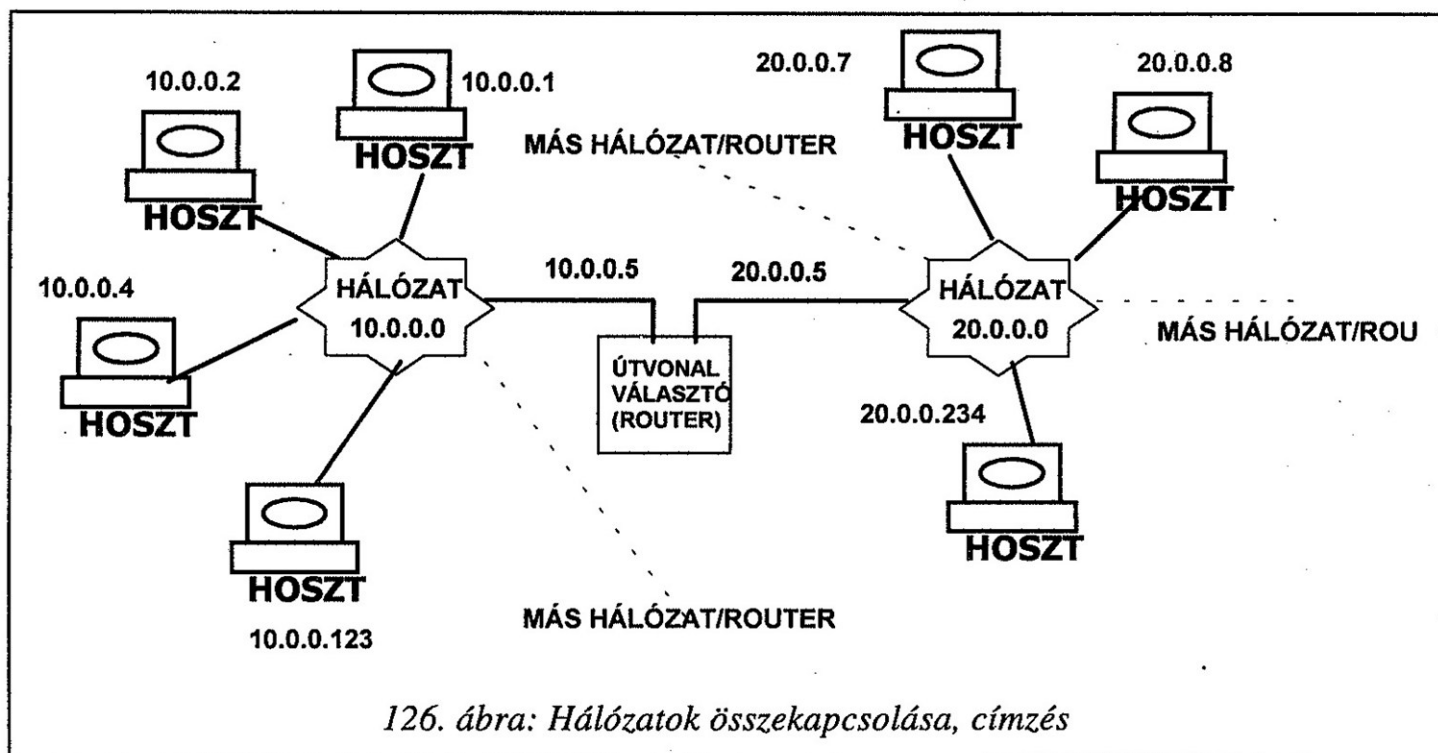
Mivel e mező 13 bit hosszú, ezért maximálisan 8192 darabból állhat egy datagram, amelyből a maximális datagramhossz $8 \cdot 8192 = 65\,536$ bájt.

Így a maximális élettartam 255 másodperc lehet..



Az Interneten egyetlen, mindenki által ismert tulajdonság létezik: az Internet Address

A ruterek a teljes cím hálózati címrésze alapján irányítanak.



A cím négy bájtot szokásos a közjük pontokat írva, a bájtok decimális megfelelőjével leírni. (dotted decimal notation)

Példa: legyen az IP cím:
 224.11.23.1,
 az alhálózati maszk:
 255.255.255.0.
 A hálózati cím leválasztásához a bitenkénti ÉS műveletet végezve a hálózat címe:
 224.11.23

Az első három címforma 128 hálózatot hálózatonként 16 millió hoszttal (A osztályú cím), 16 384 hálózatot 64 K-nyi hoszttal (B osztályú cím), illetve 2 millió hálózatot, (amelyek feltételezhetően LAN-ok), egyenként 254 hoszttal azonosít. Az utolsó előtti címforma (D osztályú cím) többszörös címek (multicast address) megadását engedélyezi, amellyel egy datagram egy hosztcsoporthoz irányítható.

IP CÍMFORMÁTUMOK

	8	8	8	8		
0	HÁLÓZAT (7)		HOSZT (24)		"A"	0 - 126 0.0.1 - 255.255.254
1 0	HÁLÓZAT (14)		HOSZT (16)		"B"	128.0 - 191.255 0.1 - 255.254
1 1 0	HÁLÓZAT (21)		HOSZT (8)		"C"	192.0.0 - 223.255.255 1 - 254
1 1 1 0	TÖBBSZÖRÖS CÍM (28)				"D"	
1 1 1 1	FENNTARTVA (RESERVED) (28)				"E"	

127. ábra: IP címek

Az utolsó címforma (E) fenntartott.

A címzéseknél a hálózat és hoszt címének szétválasztására cím-maszkokat (netmask) használnak. Alkalmazásakor bitenkénti ÉS műveletet végezve az IP cím és a cím-maszk között, a hálózati cím leválasztására. C osztályú címek esetén ezért a maszk: 255.255.255.0, míg B osztálynál: 255.255.0.0, stb.

Ahhoz hogy egy számítógépünk egy másik hálózaton lévő gépet el tudja érni, a helyi hálózatunkban kell lennie egy alapértelmezés szerinti átjárónak (default gateway) 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 használható.
- 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 193.13.2 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 jelöli. Például a saját gépről 0.0.0.0 címre küldött üzenet a saját gépre érkezik.

Sajnos a fejlemények azt mutatják, hogy ez a felosztás a tervezés gyenge pontja volt, és már ma is címhiánnyal küszködik az Internet. Természetesen ez nem azt jelenti, hogy már 2^{32} számítógép van a hálózatban, hanem a kialakított címrendszerben nagyon sok címet nem használnak fel. (Hol van a világon 128, egyenként 2^{24} gépet tartalmazó hálózat?) Ezért rövidesen új címzést fognak bevezetni (a tervezet neve: IPv6), ahol a címmező 128 bit hosszú lesz.

Az Internetben a rétegeknek megvan az egyedi azonosítója a címzéshez:

Réteg	Címzési módszer
Alkalmazási	Hoszt neve, portja
Internet	IP cím
Hálózat elérési	Fizikai cím

A hoszt neve, — amely valamilyen szimbolikus név — azonosítja a felhasználó számára a gépet — „így hívja”. Pl.: alpha1

Az Internet használata során két, egymástól akár sok ezer kilométerre lévő számítógép között alakul ki kapcsolat. Nyilvánvalóan ezért minden egyes gépet azonosíthatóvá, címezhetővé kell tenni.

Erre két, egymással egyenértékű módszer áll rendelkezésre. Az elsődleges módszer az amit IP címzésként már megismertünk, míg a másodlagos — a felhasználók által szinte kizárólagosan használt módszer az azonosító domén (domain) nevek rendszere. [13]

Ha egy alhálózat minden csomagjában lévő IP célcímre a fenti műveletet elvégezzük, eredményül vagy a saját hálózati címünket kapjuk, (és akkor a csomag a hálózaton belül marad), vagy idegen hálózati címet és akkor az útválasztónak kell elküldeni, amely majd továbbítja a többi hálózat felé.

Amikor a hálózathoz újabb gép csatlakozik, egy — az adott hálózatnak adott címtartományból — négy tagból (bájtból) álló azonosító számot, Internet címet (IP-address) kap. A címtartományok kiosztását az Internet központi adminisztrációja, az INTERNIC (Internet Network Information Center) végzi. A körzeti központok az adott gépet ezen a számon tartja nyilván. A tényleges címeket általában decimális alakban, pl. 193.224.41.1 használják.

A címbe szereplő egyes címrészeket ma már nem véletlenszerűen határozzák meg, hanem hierarchikusan felosztott földrajzi terület, domének alapján. Így a cím egyes oktetjei (8 bites csoportjai) a domént, az ezen belüli aldomént és hosztot, azaz a címzett számítógép helyét jelölik ki. A domén általában egy ország globális hálózati egysége vagy hálózati kategóriája, az aldomén ezen belül egy különálló hálózatrész, a hoszt pedig az adott hálózatrészen belüli felhasználókat kiszolgáló gép azonosító száma.

A felhasználó számára könnyebben használható a név alapján történő címzés, ahol a sok számjegyből álló IP cím helyett egy karakterlánc, az FQDN (Fully Qualified Domain Name) használható. Az FQDN, azaz a teljes domén-név, amelyet a DNS (Domain Name System), vagyis a domén-név rendszer szerint képeznek, ugyanúgy hierarchikus felépítésű, mint az IP cím, formailag pedig több, egymástól ponttal elválasztott tagból áll.

A hálózati altartomány, az aldomén több tagot is tartalmazhat, de akár hiányozhat is a cím domén-név részéből.

A domén-név egyes részeit néha eltérő kifejezéssel adják meg: a hálózati tartomány domén vagy network, az altartomány aldomén vagy subnet, a kiszolgáló gép a hoszt vagy hoszt-address.

A domén-nevek használata az Internet számára némi járulékos munkát ad, hiszen egy adatcsomag továbbítás előtt a hosztcímből meg kell határozni a vele egyenértékű IP címet, és a küldemény hosztcímét ezzel kell helyettesítenie.

Az összetartozó IP címeket és hosztcímeket a hosztgép először a helyi címtáblázatban (host table) keresi. Ha a keresés eredménytelen, a hosztgép az Internet valamelyik speciális szolgáltató-gépéhez, a névszolgáltatóhoz (Name Server-hez) fordul, amely az Internet gépeinek adatait tartalmazó, szabályos időközönként frissített sokszor hatalmas címtáblázatot kezel. A címtáblázatokban a host.aldomén.domén alakú hostcímhez a vele egyenértékű IP cím, esetleg hivatkozási (alias) alak is tartozhat [13].

Az IP cím kérésekor azt is közölni kell a névszolgáltatóval, hogy az mire kell. Ha levelezéshez kérjük, akkor a névszolgáltató a névhez

Például az **alpha1.obuda.kando.hu** címbe az egyes tagok sorrendben: a kiszolgáló gépet, a hosztgépet (egy DEC Alpha munkaállomást), az aldomént azaz hálózati altartományt (obuda.kando), végül pedig a domént, vagyis az adott ország globális hálózati tartományát (hu)

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

tartozó MX (Mail Exchange) adatrekordot adja vissza, különben a tényleges IP címet.

Az előbbi példa szerint az `alpha1.obuda.kando.hu` cím-meghatározása a következő: A gép Internet címének meghatározásához 4 potenciális kiszolgálót kellene megkérdezni. Először egy központi kiszolgálótól kellene megtudakolni, hogy hol található az `hu` kiszolgáló, amely nem más, mint a hálózatba kapcsolt magyar internet helyek nyilvántartása. A gyökéreként szereplő kiszolgáló több `hu` kiszolgáló nevét és Internet címét adná meg. (Minden szinten több ilyen névkiszolgáló van, hogy az esetleges meghibásodások ne okozzanak fennakadást.) A következő feladat lenne az `hu` kiszolgáló lekérdezése a `kando` névkiszolgálójáról. Itt is több kiszolgáló nevét és Internet címét kapnánk meg (az egyik közülük a SZTAKI). Ezek közül általában nem mindegyik található az intézmény területén (egy esetleges áramszünet fellepte miatt).

Ez után a `kando`-tól kérdeznénk le az óbudai telephely (`obuda`) névkiszolgálójának adatait, majd végül az ottani névkiszolgálók egyike adná meg az `alpha1` gép adatait. A végső eredmény a `alpha1.obuda.kando.hu` gép Internet címe lenne. A fenti szintek mindegyike egy tartományt (domain) jelöl. A teljes `alpha1.obuda.kando.hu` név egy tartománynév (domain name). (Ugyanígy a felsőbb tartományok nevei is tartománynevek: `obuda.kando.hu`, `kando.hu`, és `hu`)

Az esetek nagy többségében szerencsére nem kell a fenti lépések mindegyikét végrehajtani. A legfelső kiszolgáló (gyökér) ugyanis egyben a legfelső szinten lévő tartományok (pl. `hu`) névkiszolgálójaként is szerepel. Tehát a gyökér kiszolgáló felé irányuló egyetlen kérdéssel a HU névkiszolgálójához lehet eljutni. Az alkalmazott szoftverek pedig a már feltett kérdésekre kapott válaszokra emlékeznek, az így megkapott domén név és a hozzá tartozó IP cím eltárolódik.

Az IP cím — hosztcím átalakítást a TCP/IP automatikusan végzi, de a host operációs rendszer parancs kiadásával mi is lekérdezhajtuk egy ismert felhasználó számát.

Az előbbieket alapján már nyilvánvaló, hogy az egyes hosztgépekhez nemcsak IP cím vagy az azzal egyenértékű domén cím tartozik, hanem a hosztgépek a rajtuk futó alkalmazások eléréséhez tartozó portcímet (Application Selection Address) is használnak. Ezért a címeket ki kell egészíteni az alkalmazás elérésére szolgáló portcímmel is:

`hosztcím:portcím`

Persze minden ilyen információnak van egy megfelelő élettartama, ami tipikusan pár napnak felel meg. Az élettartam lejárta után az információkat fel kell frissíteni, amivel az esetleges változások is nyomon követhetők.

Mivel vannak a szolgáltatáshoz tartozó előre definiált ún. alapértelmezett portcímek, ezért ilyenkor a portcímek elhagyhatók.

Míg az egyes hosztokat a hosztcímük egyértelműen meghatározzák, addig a hosztokat több felhasználó használja, tehát a hozzájuk kapcsolódó felhasználókat is meg kell különböztetnünk egymástól. Erre azok felhasználói neve (login- vagy felhasználónév), vagyis az adott hoszton egyedi azonosító-név szolgál [13].

Egy személy Internet elérhető levelezési (E-mail) címe tehát két főrészből áll, és a következő alakú:

felhasználónév@host.aldomén.domén

DOMAIN NEVEK - ÖSSZEFOGLALÓ

- Gép azonosító: név (high level), cím (low level)
- Lineáris névrendszer (Flat namespace): központilag kellene kezelni, a világon nem lehet két azonos név, nem kezelhető
- Hierarchikus névrendszer: kezelhető, fel kell osztani a névteret kisebb egységekre
- Név alakja: local.site[.site]..., geográfikus felosztás, funkcionális felosztás
- Top domain nevek: COM, EDU, GOV, MIL, NET, ORG, ARPA, INT, (ISO) ország kódok: pl. HU, DE, IT, stb.
- Név-cím összerendelés DNS (Domain Name Server)-ek (névszerverek, más néven névszolgáltatók) végzik
- Direkt névfeloldás: domain név ismert, keressük az IP címet
- Reverz névfeloldás: IP cím ismert, keressük a domain nevet
- A név-szerverek egymás között adatot cserélnek, ha az egyik nem ismeri a címet, feljebb adja a kérést
- Name caching: (név tárolása) hatékonyság növelésére, a hoszt emlékszik, de csak egy darabig (a névszerver mondja meg meddig)
- Minden hálózatban CÉLSZERŰ egy névszerver üzemeltetése, mert ez jelentősen gyorsíthatja az ismeretlen nevek feloldását és egyszerűsítheti a névszerver adminisztrációt. Ugyanakkor minden kiosztott Internet címhez KÖTELEZŐ egy elsődleges - primary - és legalább egy másodlagos - secondary - névszerver üzemeltetése, amelyek NEM LEHETNEK egyazon helyen (hiba esetén ne álljanak le egyszerre!).



IP ÚTVÁLASZTÁS (ROUTING)

A csomagkapcsolt rendszerekben az útválasztás (routing) azt a folyamatot jelöli, amivel kiválasztjuk az útvonalat (path), amin a csomagot továbbküldjük és a útvonal választó (router) az a számítógép (IMP), amely ezt végrehajtja.



Az útválasztó olyan eszköz, amelynek több 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.

Két fajtáját különböztetjük meg a routolásnak : direkt módút, illetve indirekt módút.

A 127. ábrán látható négy helyi hálózat három routeren keresztül van összekötve.

A **direkt** módúban a datagram átvitele egyik gépről a másik felé fizikailag is megtörténik.

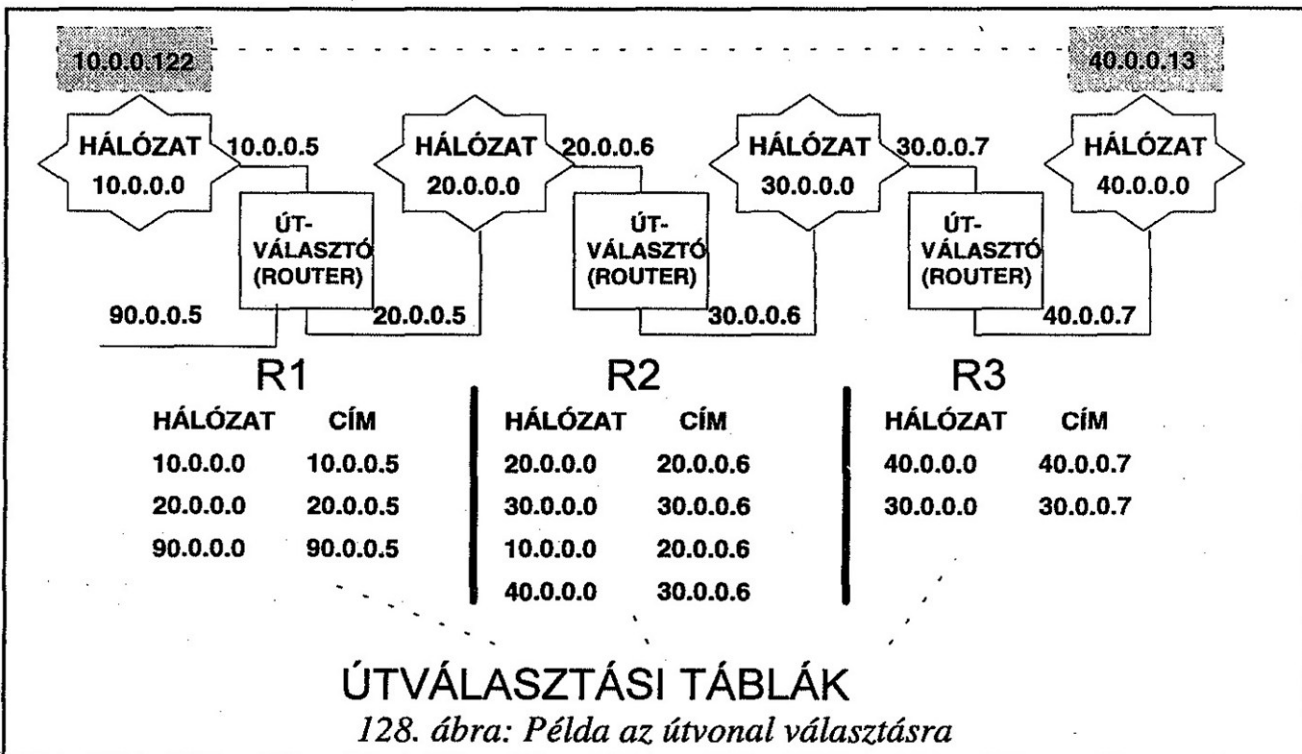
Az útválasztási táblák tárolják az információt az elérhető csomópontokról és azok elérési útvonalairól. Tipikusan a táblázat (N,G) párokból áll, ahol N az elérhető hálózat IP-címe és G annak a útválasztónak a címe, amelyiken keresztül ez a hálózat elérhető.

Indirekt átvitel van abban az esetben, amikor nem egy fizikailag rácsatlakozott hálózat a cél gép, Ilyenkor a küldő az útválasztónak adja át a csomagot, hogy az kézbesítse.

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



Ha két gép egyazon lokális hálózaton van, akkor útvonal-kiválasztás nélkül **közvetlen (direkt)** összeköttetés létesíthető közöttük. Különböző hálózatok közötti **közvetett (indirekt)** útvonal-

kiválasztásnál először a feladónak meg kell adnia azt a útválasztót, amihez a csomagot (datagramot) küldi. Majd a útválasztó fogja (esetleg újabb útvonal választókon keresztül) a cél-hálózatra továbbítani a datagramot.

Ezért például, ha a **40.0.0.25** című gép kommunikálni akar a **30.0.0.16** című géppel, akkor a következő történik:

A küldő gépen az IP protokoll megállapítja, hogy a csomag egy másik hálózatnak szól, ezért a router **40.0.0.7** címére küldi el. Az útválasztó megállapítja a címzett hálózati címét (**30.0.0.0**), és mivel az útválasztó táblájában szerepel ez a hálózat, a hozzátartozó csatolókártján (IP címe: **30.0.0.7**) keresztül küldi el a csomagot.

az IP címből a 255.0.0.0 hálózati cím maszk alkalmazásával állapítjuk meg a címzett hálózat címét:

30.0.0.16 bitenkénti ÉS kapcsolata 255.0.0.0 – val adja a hálózati címet: 30.0.0.0

Ha nem talál táblabejegyzést az adott hálózatra vonatkozólag, akkor három dolog történhet:

1. Az útválasztó hibaüzenetet küld a feladónak („Nem ismerek ilyen hálózatot”) és a csomagot eldobja.
2. Felvesszük a nem szerelő hálózatot az útvonalon lévő útválasztók táblázataiba, hogy a routolás megvalósulhasson.
3. Elküldjük a csomagot egy ún. alapértelmezett hálózati címre, ott talán majd ismerik ezt a címet.

Tételezzük fel, hogy a **10.0.0.0** hálózati című hálózatban lévő egyik gép (hoszt címe: **122**, így IP címe: **10.0.0.122**) kommunikálni akar a **40.0.0.0** hálózat **40.0.0.13** című gépével!

Az előbbieket szerint a **10.0.0.122** című gép a **10.0.0.5** című 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 lehet megoldani:

Az útválasztó táblákba fel kell venni még bejegyzést, hogy az adott hálózatnak szóló csomagot merre küldjük tovább. Esetünkre a megoldás:

az R1 táblázatába felvesszük a

20.0.0.0 **20.0.0.5** bejegyzést, (ez már szerepel az ábrán!)

az R2 táblázatába pedig a

40.0.0.0 **30.0.0.6** bejegyzést. (ez is már szerepel az ábrán!)

Természetesen ha a **40.0.0.0** hálózatból akarunk küldeni a **10.0.0.0** című hálózatba, akkor

R3: **10.0.0.0** **30.0.0.6**

R2: **10.0.0.0** **20.0.0.5**

bejegyzéseket kell szerepeltetni.

Számos útválasztóban megadható egy. ún. alapértelmezés szerinti átjáró (**default router**).

Ha nincs a táblázatban létező útvonal, akkor a szoftver az alapértelmezett útválasztónak küldi el a datagramot.

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

Ekkor, ha olyan hálózatba küldünk csomagot, amelyhez nem tartozik bejegyzés, akkor ezt mindig az egyik hozzá csatlakozó hálózatban lévő útválasztónak (default router) küldi el.

Az előző példánál maradva nem kell R1 és R3 táblázatát módosítani, ha R1 default routere **20.0.0.6** 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.

Az egész útképzés-döntés két tesztből áll : egy a helyi hálózatra, a másik, amelyik az alapértelmezett egyetlen lehetséges útválasztóra mutat. Még ha a hálózat néhány másik hálózatot is magában foglal, az útképzés egyszerű, mivel néhány tesztből áll a helyi hálózatokra, és az alapértelmezett az összes más célhoz.

A routolás nem csak hálózati, hanem hoszt szinten is megvalósítható: A hálózati menedzser bizonyos hosztokat kitüntethet, amiket az útválasztó **hoszt-specifikus** táblázatában helyez el. Ez sok esetben biztonsági, tesztelési célokat szolgál.

AZ ÚTVONAL KISZÁMÍTÁS ALGORITMUSA

A fentieket összefoglalva létrehozhatjuk az útvonal-kiválasztás algoritmusát. Két paraméter szükséges: a címeket (és adatokat)

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 hoszt-specifikus 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 alapértelmezési útválasztónak

egyébként útvonal-kiválasztási hiba deklaráció

Ilyenkor a tábla első oszlopa a hoszt teljes IP címét tartalmazza, a második oszlop azt a címet, amelyen keresztül az adott hoszt elérhető.



Az IP útvonal-kiválasztás nem könnyű, mert a hálózati pontok között többszörös útvonalak létezhetnek. Sok esetben az útvonal-kiválasztó 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. fejezetben erről, mint mértékről írtunk.)

A legtöbb internet útvonal-kiválasztó sokkal egyszerűbb és a választást egyszerűen legrövidebb útvonal alapján hajtja végre.

tartalmazó IP datagram, és router-táblázat.

STATIKUS ÉS DINAMIKUS ÚTVÁLASZTÁS

Az útválasztó 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 útválasztással 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 kommunikálni kezdenek egymással, és az így szerzett ismeretek alapján megkezdik táblázataik frissítését.

Az útválasztási információk cseréjére speciális kimondottan erre a célra kifejlesztett 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 nagy méretűvé növekedett, akkor a forgalomirányító táblák kicserélésével előálló forgalom már olyan nagy volt, hogy akadályozta a normál forgalmat is.

Ezért a megváltoztatott jelenlegi algoritmusban minden egyes IMP belsőleg fenntart egy adatbázist, amely az egyes vonalakon való késleltetéseket tartalmazza. Erre az adatbázisra alapozva minden IMP kiszámolja a közte és az összes többi IMP 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 IMP 10 másodperces átlagolási idővel méri vonalain a késleltetést. E mérések eredményét egy aktuális sorszámmal ellátva minden IMP megkapja. Az információ köröztetéséhez az IMP-k az elárasztásos algoritmust használják.

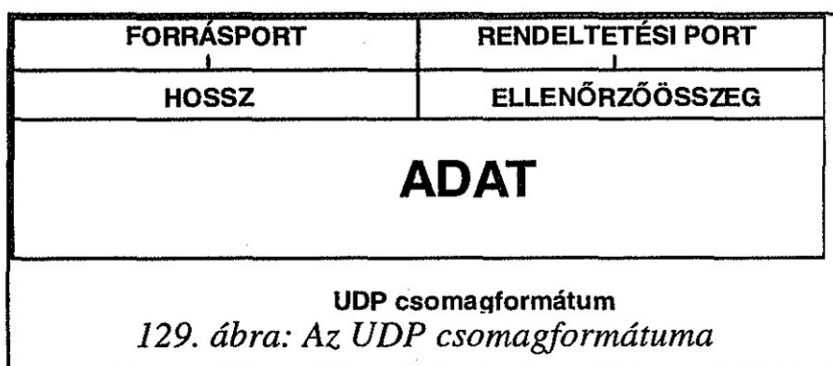
ÖSSZEKÖTTETÉS-MENTES SZÁLLÍTÁSI PROTOKOLL: AZ UDP.

Sok alkalmazás használ üzeneteket, amelyek elérnek egyetlen datagramban. 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 fordítani IP címre. Ehhez a név-szolgáltatóhoz kell a kérést tartalmazó üzenetet eljuttatni. A kérés annyira rövid, hogy biztosan elér 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.

Az UDP egy összeköttetésmentes 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 használatos, ahol ha a kérdés vagy a válasz elvész, a hiba egyszerű újrakérdezéssel megoldható.

A legtöbbet használt ilyen protokoll az **UDP (User Datagram Protocol felhasználói datagram protokoll)**. Hasonlóképpen illeszkedik a rendszerbe, mint a TCP. A hálózati szoftver az adatok

elejére ráilleszti az UDP fejléct 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.



AZ INTERNET VEZÉRLÉSE: AZ ICMP PROTOKOLL

Az Internet működését az IMP-k és az átjárók 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 protokoll az Internet tesztelésére is használható [1].

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 csomag" 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

Lehetővé teszi, hogy a routerek hiba- és vezérlő üzeneteket küldjenek más routereknek és hosztoknak

Az ICMP üzenetek IP csomagok adatmezőjébe csomagolva utaznak

Az ICMP üzeneteket CSAK a hibát kiváltó csomag eredeti feladója kapja meg.

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.

SOURCE QUENCH (forráslefojtás) üzenet a túl sok csomagot küldő hosztok megfékezésére használható. Amikor egy hoszt egy ilyen üzenetet vesz, akkor adási sebességét csökkentenie kell.

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.

HÁLÓZAT ELÉRÉSI RÉTEG — ARP

Jelenleg a legtöbb hálózat fizikai és adatkapcsolati szinten Ethernet kártyákat használ. Mivel az Ethernet keretnek saját fejléce van, saját egyedi, 48 bites címmel rendelkezik, ezért az IP csomagokat ilyen hálózaton közvetlenül nem lehet átvinni, be kell csomagolni. Minden Ethernet keretnek egy 14 oktetes fejléce van, amely a forrás- és a cél gép Ethernet címét, valamint egy típuskódot tartalmaz. A hálózaton lévő gépek csak az olyan kereteket figyelik, amelyek célmezőjében a saját Ethernet címüket, vagy a mindenkinek szóló körözvénycímet találnak. Minden számítógépnek van egy táblázata, amelyben felsorolja, hogy milyen Ethernet cím milyen Internet címnek felel meg. Ennek a táblázatnak a karbantartását a rendszer egy protokoll, az ARP (**Address Resolution Protocol** — címfeloldási protokoll) segítségével végzi.

Egy alhálózatban, amelyik Ethernet összeköttetést használ, tegyük fel, hogy a 193.18.24.196 IP című hosztról a 193.18.24.75 hoszttal szeretnénk kapcsolatba lépni. A kezdeményező 193.18.24.196 című hoszt megnézi, hogy szerepel-e a saját ARP táblázatában a 193.18.24.75 címhez tartozó Ethernet cím bejegyzés. Ha igen, akkor a datagramhoz egy Ethernet fejléccet csatol, és elküldi. Ha azonban ilyen

Ezekon az üzeneteken kívül van még négy másik, amelyek az internet címmel foglalkoznak, és lehetőséget biztosítanak a hosztok számára, hogy azonosítsák saját hálózat-számukat, felfedezzék a címzési hibákat: például kezelni tudják azt az esetet, amikor egyetlen IP-címet több LAN birtokol.

Hálózati szinten IP címet használnak, de az adatkapcsolati és fizikai rétegben fizikai (kártya)címet kell használni!

Ez megfelel az OSI modell fizikai és adatkapcsolati rétegének.



Egy hoszt megtalálhatja a megcímzendő másik hoszt fizikai címét, csupán annak IP címét ismerve.

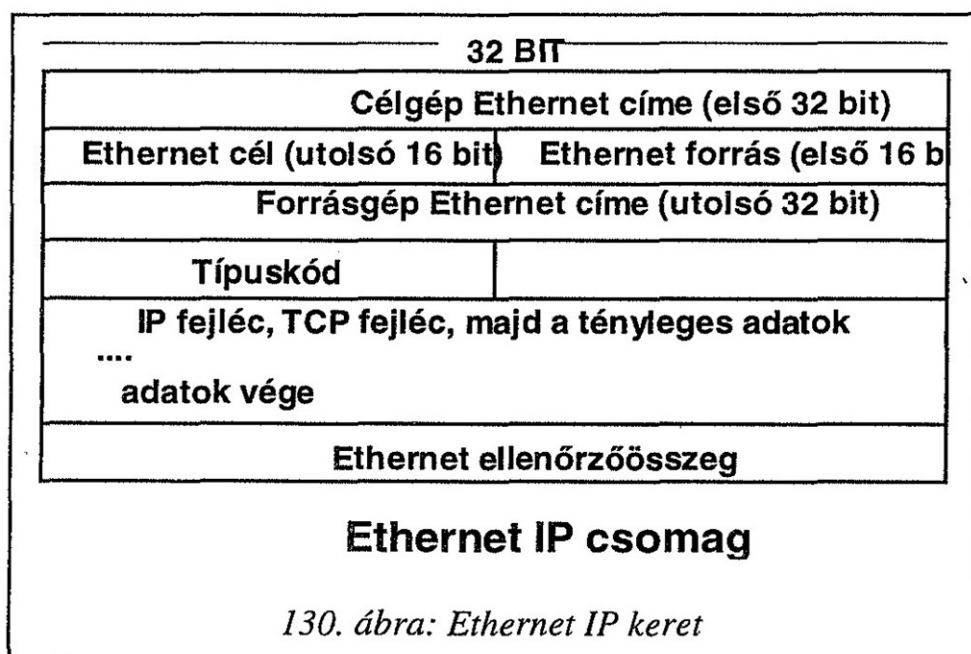
7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

bejegyzés az ARP táblázatban nincsen, akkor a csomagot nem lehet elküldeni, hiszen nincs meg az Ethernet cím.

Ekkor lép működésbe az ARP protokoll. A 193.18.24.196 hoszt egy „a 193.18.24.75 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 193.18.24.75 hallja a kérést, és egy ARP üzenetet küld válaszul a kérdezőnek, amelyben megadja a 193.18.24.75 IP című gépben lévő kártya Ethernet címét, pl.: 12:3:44:12:52:11. A kérést adó rendszer a kapott információt bejegyzi az ARP táblázatába.

Abban az esetben, ha a kért IP cím nincs a közös Ethernet hálózatra 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. Az üzenetszórást jelző IP című csomagokat (pl. 255.255.255.255 vagy 193.18.24.255) is csupa egyes bitből álló Ethernet címre kell küldeni.

A címek mellett a fejlécben szerepel még egy a használt protokollt azonosító típuskód is. Ennek segítségével ugyanazon a hálózaton többfajta protokollkészlet használata is lehetséges: TCP/IP, DECnet, Xerox, NS stb... A protokollok mindegyike különböző értéket helyez a típus mezőbe. A csomag végén az ellenőrzőösszeg található, amely az egész csomagra vonatkozik. Az Ethernet keret tehát így néz ki:

Az ilyen módon „burkolt” (encapsulated) keretek megérkezése után az egyes fejléceket leszedi a megfelelő protokoll. Az Ethernet interfész az Ethernet fejléctet 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.

Létezik az ARP protokoll fordítottja, a RARP (Reverse Address Resolution Protocol) amely olyan táblázattal dolgozik, amelyben az van felsorolva, hogy milyen Ethernet cím milyen IP címnek felel meg.

A hoszt nem tudja saját IP címét, de ismeri saját fizikai címét. RARP célja, hogy a hoszt megszerezze saját IP címét a hálózatban olyan gépektől, akik RARP szerverként működnek.

Foglaljuk össze a lényeget még egyszer!

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 Protocol) az egyedi datagramok továbbításáért felelős.

Például ha egy adathalmazt akarunk a hálózaton átvinni:

XX

a TCP ezt datagramokká darabolja:

xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx xxx

a TCP minden datagram elejére egy fejléctet 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 fejléctet képez (I=FEJ(IP)) :

ITxxx ITxxx ITxxx ITxxx ITxxx ITxxx ITxxx



7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

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

A fogadó oldal ezeket sorban egymásután leszedi, ha IP típusú, akkor az IP-nek adja tovább, ha TCP típusú, akkor a TCP-nek, ami a sorozatszám alapján visszaállítja az eredeti adatfolyamot.

RÖVIDEN A UNIX HÁLÓZATKEZELÉSÉRŐL

A TCP/IP-t protokollt egy primitívhalmazon keresztül lehet elérni, amelyeket rendszerhívásokként valósítottak meg (implementáltak). Ezen keresztül érheti el a felhasználó a szállítási szolgáltatásokat. A főbb rendszerhívásokat a következő táblázatban soroltuk fel [1]:

Mivel az Internetben sok gépen a UNIX-ot használják operációs rendszerként, ezért érdemes röviden összefoglalni, hogy ez az operációs rendszer hogyan támogatja a hálózati lehetőségeket.

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 kész-e olvasásra vagy írásra

A szolgálatinterfész központi jelentőségű fogalma a socket (foglat), 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 összeköttetés kérést 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 nem-blokkolósna 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.

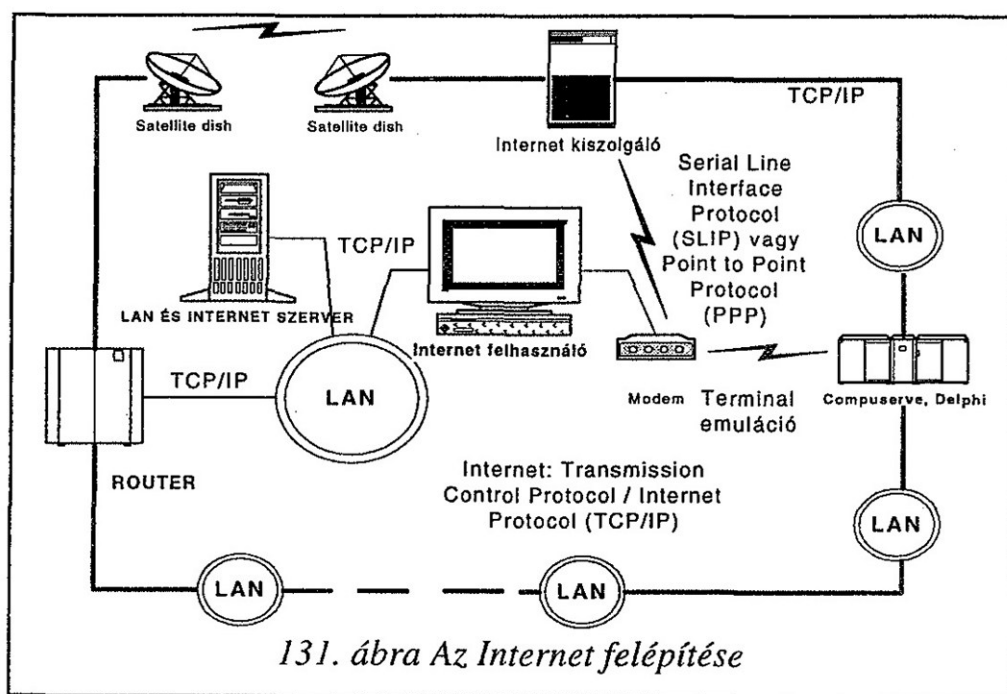
A **send** és **recv** hívások üzenetek küldésére és vételére használhatók. Ezen alaphívásoknak több változata is létezik.

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.

PÉLDA AZ ALKALMAZÁSI RÉTEGRE: AZ INTERNET SZOLGÁLTATÁSOK

A következőkben az alkalmazási réteg funkcióinak jobb megértésére az Internet alkalmazási rétegét mutatjuk be.



KAPCSOLÓDÁS AZ INTERNETRE

Az Internet felépítést a **131. 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 egy 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. 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. 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 kapcsolódó gépen fut az a program, amelyet a telefonvonalon keresztül a számítógépet terminálként használva kezelünk.

INTERNET-ELÉRÉS TELEFONVONALON KERESZTÜL

Az egyszerű felhasználók Internetre való kapcsolódásának legegyszerűbb módja modemmel telefonvonalon keresztül valósítható meg. Ez természetesen feltételez a másik oldalon a felhasználót modemes kapcsolattal fogadó valamilyen Internet-szolgáltatót (provider-t), amely már nagy sávszélességű vonalon kapcsolódik az internetre. Mivel a kapcsolatban IP csomagok átvitele történik, ezeket a telefonvonalon átvitt keretbe 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 szabvány.

SERIAL LINE INTERNET PROTOCOL — SLIP

Az RFC 1055 mint nem szabványt 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 a 0xDB és a 0xDC kétbájtos szekvenciával.

A SLIP nem tesz lehetővé címezést, csomagtípus- (packet type) azonosítást, hibaellenőrzést és -javítást, s az alap SLIP protokoll-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, hibaellenőrző és -javító kóddal (HDLC Frame Check Sequence — FCS) mindegyik keretben. A teljes PPP keret nyolc oktettet használ az egyes keretek becsomagolására, de lehetőséget ad ennek két oktetre történő rövidítésére is (nagy sebességű hardver esetén mód van a 32 bites korlátozások betartására), továbbá escape mechanizmust biztosít kontrolladatok (pl. XON/XOFF) átvitelére.

Az Interneten, mivel eltérő felépítésű hálózatokat kötnék össze, szükséges az Interneten folyó kommunikáció közös szabványainak kidolgozása, amelyeket az RFC (Request for Comments) dokumentumok tartalmazzák, amelyekről a fejezet végén adunk egy rövid összefoglalót



A teljes PPP csomag keretformája:

delimiter 0x7E	cím 0xFF	ctrl 0x0 3	protoco l 0xXXXX	adatcsoma g	FCS	delimiter 0x7E
--------------------------	--------------------	-------------------------	-------------------------------	-----------------------	------------	--------------------------

A PPP a datagramok becsomagolásának leírásán kívül tartalmaz egy ún. **Link Control Protocolt (LCP)** a pont-pont közti kapcsolatok felépítésére, konfigurálására, a kapcsolat 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, egyeztetni 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.

INTERNET SZOLGÁLTATÁSOK

A felhasználót általában nem ez, hanem az elérhető szolgáltatások érdeklik. A szolgáltatások alapvetően két csoportba sorolhatók: közvetlen hálózati kapcsolatot nem igénylő (off-line) szolgáltatás — ilyen a levelezés — , és azt igénylő (on-line) szolgáltatások. Ennek megfelelően is több megoldás lehetséges:

A legegyszerűbb szolgáltatás a **levelezés**: ez lényegében hálózati kapcsolatot nem igényel. 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;

Shell-számlát nyitunk: terminálként (vagy a szolgáltató speciális szoftverén keresztül) bejelentkezünk a szolgáltató gépére, és arról böngésszük a hálózatot;

SLIP vagy PPP számlát nyitunk, amelyen keresztül gyakorlatilag minden böngésző, levelező és kommunikációs Internet-alkalmazást futtathatunk;

Ha megvan a lehetőség rá, **beköthetjük helyi hálózatunkat az Internetbe**. TCP/IP-t és az Internet-segédprogramokat telepítünk a hálózaton, majd a LAN-t valamilyen hálózati kapcsolattal (X.25, ISDN-, nagy sebességű bérelt telefonvonal) routeren keresztül rácsatlakoztatjuk az Internetre.



SZÁMÍTÓGÉP - HÁLÓZATOK

Az Interneten, mivel eltérő felépítésű hálózatokat kötnek össze, szükséges az Interneten folyó kommunikáció közös szabványainak kidolgozása, amelyeket az RFC (Request for Comments) dokumentumok tartalmazzák, amelyekről a fejezet végén adunk egy rövid összefoglalót. A szabványok közös alapjául a UNIX operációs rendszerben megvalósított megoldások szolgáltak, mivel elsőként ilyen operációs rendszerű gépeket kötöttek össze, és **jelenleg is az Internetben lévő gépek többségén a UNIX valamelyik változat fut.** Az Internet lényegesebb alkalmazási protokolljai a következők:

A WINDOWS NT-ben is sok UNIX vonás van, a LINUX-ról nem is beszélve!

- **SMTP Simple Mail Transfer Protocol** egy alkalmazási protokoll, amely a hálózati felhasználók egymással való kommunikációját teszi lehetővé. Leveletet tud küldeni és fogadni.
- **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, de az összefüggések jobb megértése érdekében lesznek olyan dolgok, amelyeket megismételünk.

E-MAIL (ELECTRONIC MAIL)

A legalapvetőbb szolgáltatás, a legelső, amit az Interneten használtak, az elektronikus levelezés. Egy levelezőprogram (mail) segítségével szöveges állományt küldhetünk az Internet bármelyik felhasználójának

Egy ilyen levél – a hagyományoshoz hasonlóan – egy, a címzettet a feladót és néhány kiegészítő információt tartalmazó fejlécből, és a tartalmat (a levél szövegét tartalmazó) törzsből 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

A felhasználói_név egy rövid azonosító, ami nem tartalmazhat speciális karaktereket. A @ (kukac) jel a felhasználói nevet választja el a gépet leíró utána lévő résztől. A cím hierarchikus felépítésű, a legutolsó jelöli a legmagasabb szintet, és így szűkítve a kört. Ha ezt értelmezni akarjuk, akkor célszerű hátulról kezdeni.

Az utolsó egység az ország(intézmény)azonosító országra, vagy az intézmény jellegére utal. Egy rövid lista azokról, amikkel a legtöbbet lehet találkozni:

EDU	Oktatási intézmény az Egyesült Államokban
COM	Kereskedelmi cégek az Egyesült Államokban
GOV	Amerikai kormány szervezet
NET	Nagyobb Internet szolgáltató
MIL	Amerikai katonai szervezet
ORG	Amerikai vállalatok, amelyek más kategóriába nem fértek be...
HU	Magyarország
CA	Kanada
DE	Németország

...és a többi ország (kétbetűs) kódja

A cím további részeinél pontok választják el a részeket egymástól. A domain_név általában annak a környezetnek az azonosítója, amiben a gép található, az intézmény vagy egy lokális hálózat neve, pl. .kando: a Kandó Főiskola domain neve.

A subdomain_név akkor létezik, ha szükség van a domén további tagolására, pl. .obuda.kando.hu: a Kandó óbudai részlege. A gépnév



Az RFC 822 szabvány az ilyen leveleknek csak a fejlécét definiálta, magát a tartalom összeállítását a felhasználóra bízta.

Általánosan fogalmazva egy felhasználói név (username) és egy cím (domain) részből áll, a kettő között a @ jel található. Ez a "kukac" az angol "at" szót jelenti, vagyis arra utal, hogy ez a felhasználó HOL (melyik gépen) található meg.

Magyarországi felsőoktatási intézmények domain nevei pl.: kando, bmf, bme, bke, elte, pote, sote, stb.

értelemszerűen azt a gépet specifikálja a doménon belül, amelyen a felhasználó géphasználati joga (accountja) található. Pl. az én címem: *konya@novserv.obuda.kando.hu*.

Bár az elektronikus levelezés a legalapvetőbb a szolgáltatások között, mégis a felhasználó szempontjából van két nagyon kedvező tulajdonsága:

Ha csak levelezni szeretnénk, a vonal adatátviteli sebességének nem kell nagyok 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 leveleket 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 tartalomjegyzék-szerű listázása a levél témája (subject) mezőket mutatva,
- válasz adott levélre (reply),
- levél továbbküldése (forward),
- levél tárolása különböző irattartókba (folderekbe),
- levél törlése (delete).

Ha egy levél érkezik, akkor általában nemcsak maga az üzenet jön meg, hanem egy pár soros "fejléc" is. Ez a legtöbb esetben nem tartalmaz az átlagos felhasználó számára lényeges információt. Vegyünk egy példát:

Létezik már Mo.-n ingyenes levelezési lehetőség (freemail.c3.hu).

Így bárki egy modem, telefonvonal és persze számítógép+(ingyenes) program segítségével levelezhet.

...és hogy most írom a könyvet, már két szolgáltató is kínál INGYENES Internet elérést:

www.freestart.hu

www.kiwwi.hu

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

```
Received: from MAIL by KKM_F_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
```

A „Received”: sorokból azoknak az állomásoknak a neveit, és a használt protokollokat tudhatjuk meg, amelyeken a levél keresztülment. Minden levél esetében van legalább egy ilyen sor. Ha a levél nagyon „kavargott” a világban, akkor 10 fölé is mehet ezeknek a soroknak 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 egyedülálló 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.

UUENCODE/UUDECODE

Fontos megjegyezni, 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, elvesz. Így közvetlenül bináris fájlok átvitele nem lehetséges.

Több megoldás létezik erre a problémára, a legelterjedtebb program az UUENCODE/ UUDECODE.

Az UUENCODE a fájlt alkotó bináris bájt sorozatot 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:

```
AFH, 01H, 65H = 1010 1111 0000 0001 0110 0101 = 101011 110000 000101 100101
```

Mivel a speciális ill. vezérlőkérekek ASCII kódjai 0-31-ig terjednek, a normál szöveg kialakítása érdekében mind a négy így kapott bitsoporthoz hozzáad 32-t (00100000).

101011 110000 000101 100101 => 1001011 1010000 0100101 1000101 = 4BH 50H 25H 45H =

Így kaptunk 3 bájtából 4, a képernyőn is olvasható karaktert, amit aztán már elküldhetünk. Az UUDECODE program az így keletkezett fájlt kódolja vissza a fogadó oldalon. A kódolt fájl természetesen hosszabb lesz, mint az eredeti, mert a módszere semmilyen más változtatást (pl. tömörítés) nem végez.

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 formátumot, de az üzenetek, levelek törzsének is létrehoz egy struktúrát, valamint definiálja a nem ASCII tartalmú üzenetek kódolását elvégző eljárásokat is.

Mivel nincs eltérés az RFC 822-höz képest, a MIME levelek minden probléma nélkül továbbíthatók a már létező levelező programokkal és protokollokkal. Csak a feldolgozásuk igényel néhány — programokban jelentkező — változtatást.

Fejléc	Jelentés
MIME-Version:	azonosítja a MIME verzióját
Content-Description:	az üzenet tartalma
Content-Id:	egyedi azonosító
Content-Transfer-Encoding:	a törzs átviteli kódolásának formátuma
Content-Type:	az üzenet jellege

Content-Description: ez a fejléc egy ASCII szöveg, amely megadja az üzenet tartalmát.

Content-Id: a standard üzenet azonosítójának formátumával megegyező egyedi azonosító.

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 MIME öt új üzenet fejléccet definiál (ezek láthatók a táblázatban). Ezek közül az első csak egy jelölés ahhoz, hogy a fogadó program felismerje: milyen MIME verziójú üzenettel van dolga.

Bármely levél, amely nem tartalmazza a MIME fejléccet, alapértelmezés szerint egy sima, angol nyelvű üzenetként dolgozódik fel.

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

- 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).

A == és = jelek azt fogják jelenteni, hogy az utolsó egység csak 8 vagy 16 bitet tartalmaz. A kocsni vissza és soremelés karakterek figyelmen kívül maradnak és szabadon beszűrhetők a sorok megfelelő rövidegének érdekében. Ezzel a módszerrel tetszőleges bináris szöveg biztonságosan elküldhető.

Bináris adatokat a következő két kódolást felhasználva kell elküldeni

Az egyik módszer a **base64 kódolás** használatos. 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. Ez így folytatódik, majd következnek 26 db kisbetű, 10 számjegy 0-tól 9-ig, és végül + és / a 62-nek és 63-nak felel meg.

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: Az utolsó fejléc eléggé összetett, felépítését a következő táblázat mutatja, amely az egyes igényeknek megfelelően folyamatosan bővül. Lényegében az üzenet típusát ill. altípusát tartalmazza. amelyeket egy / jel választja el egymástól.

Tekintsük például a következőt:

Content-Type: image/gif

Az altípust pontosan meg kell adni, mivel nincs alapértelmezett beállítása:

Típus	Altípus	Leírás
Text	Plain	Formázatlan szöveg
	Richtext	Egyszerű formázási utasításokat tartalmazó szöveg
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 dokumentum

SZÁMÍTÓGÉP - HÁLÓZATOK

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

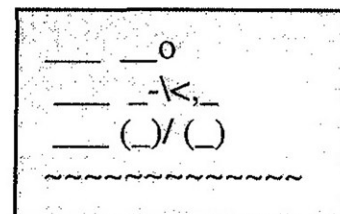
SMILEYS (MOSOLYGÓK)

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.

FTP (FILE TRANSFER PROTOCOL)

Az ftp protokoll a hálózatban lévő gépeken megtalálható fájlok átvitelére használható. Használata az Email-el szemben már folyamatos hálózati kapcsolatot igényel.

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:

Jelenleg már ezeket a parancsokat a windowsos grafikus FTP programok elrejtik, de nem árt ismerni a működését...

Adatátviteli sebesség igénye is jelentősebb, hiszen elfogadható időn belül kell átvinnünk esetleg több száz kilobájtnyi adatot. Néhány kbájt/s-os átviteli sebesség már elfogadható.

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

- **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.

Az ascii és binary üzemmódok közt az **asc** illetve **bin** paranccsal lehet váltani. Vannak mindenki számára elérhető ún. 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.

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 a 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.

TELNET

Egy távoli gépre úgy lehet belépni, mintha egy terminálja előtt ülnénk. Azaz a Telnet a gépek közti távoli bejelentkezést lehetővé tevő protokoll neve. Ez is folyamatos (on-line) hálózati kapcsolatot igényel, és sebességigénye hasonló az ftp-hez, (persze, csak ha azt szeretnénk, hogy egy leütött billentyű ne 10 másodperc múlva jelenjen meg...). Telnettel csak akkor tudunk egy másik gépre belépni, ha azon a gépen is van accountunk.

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 felhasználói általában akkor tud egy távoli gépről/gépre másolni, ha a távoli gépen is rendelkezik felhasználói jogosultsággal (account-tal).



Anonymous FTP

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.

Ezen lehetőség a hálózati gépek biztonságának egy sebezhető pontja. Ha ugyanis egy távoli gépre rendszer-adminisztrátori jogokkal tudunk belépni (felhasználói név: root, a jelszót automatikus próbálkozási módszerrel „kitaláljuk”), akkor a géppel mindent megtehetünk.

Az ilyen behatolás módot nyújt arra is, hogy a távoli gépet felhasználva (a Telnetet ott elindítva) lépjunk be egy „kényesebb” gépre. Ez utóbbi behatolás felderítésekor a behatoló címe az erre használt gép címe, és ha az oda történő behatolás nyomait eltüntetjük, akkor nem lehet kideríteni a kényesebb gépre behatolót.

ARCHIE

A Telnet protokoll egy nyilvánosan használható változata. 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..

Használatához egy archie szerverhez kell kapcsolódnunk, és az általa különféle ftp szerverekről összegyűjtött és folyamatosan frissített adatbázisban kereshetünk. A bécsi egyetem archie-szerverére a

telnetarchie.univie.ac.at

paranccsal kapcsolódhatunk. Az archie elérhető levelezéssel is. Ilyenkor az

archie@archie.univie.ac.at

címre kell levelet küldeni, levél törzsébe a find kulcsszó után kell a keresett részt beírni. Az eredménylistát saját Email címünkre fogjuk megkapni.

FINGER

Ha a finger szó után megadunk egy Email-cím szerkezetű címet (rámutatunk), információkat tudhatunk meg a megjelölt felhasználóról. Ilyenek például: A felhasználói login és saját neve, home könyvtára, alapértelmezett shell-je, utolsó bejelentkezésének időpontja, illetve, ha éppen be van jelentkezve, akkor mióta, és honnan.

Ezen kívül még olyan információk olvashatók, amiket az illető maga adott meg. Többnyire minden rendszerben van egy erre kijelölt fájl, amibe bármit beírhatunk, amit közölni szeretnénk a minket finger paranccsal megkereső személlyel.

Unix rendszerben a home könyvtárunkban lévő. plan nevű fájlban található információ olvasható el a finger paranccsal.

LEVELEZÉSI LISTÁK ÉS A USENET

Az olyan levelezési fórumokat, amelyek hasonló témájú információcserére alakultak levelezési listáknak 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 való foglalkozásra alakult ún. hírcsoport-ot tartalmaz. 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 többiekben csak egy-egy kiválasztott részük található.

A kezdők bekapcsolódását, kérdéseket és rá a válaszokat tartalmazó dokumentumok, az ún. FAQ-ok (magyarul: GYIK = Gyakorta Ismétlődő Kérdések) segítik.

INFORMÁCIÓK SZERVEZÉSE A HÁLÓZATON.

Az információk összegyűjtése, rendezése és megkeresése, a dinamikus változó hálózaton nem egyszerű dolog. Hogyan szervezhetjük meg az információk közötti kapcsolatokat? Alapvetően két megoldás kínálkozik:

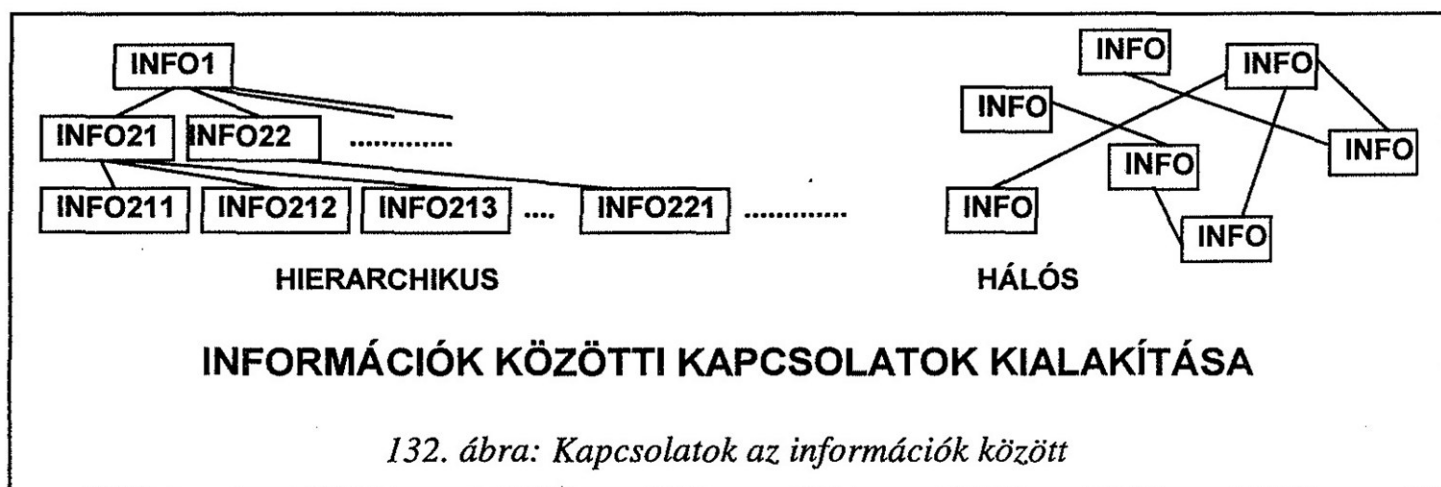
- Az egyik a **hierarchikus**: ez azt jelenti, hogy az információk közötti összefüggéseket egy szintekből álló rendszerbe szervezzük; elindulunk a legmagasabb szintről, és minden szinten kijelöljük, amihez tartozó alatta lévő szinten folytatjuk a keresést. Ez hasonló a számítástechnikában megszokott menürendszerhez: egy menüpontot kiválasztva megnyílik egy almenü, majd abból kiválasztva egy újabb, és így tovább.

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.

Technikailag a Usenet egy speciális telnet parancs, amit akkor tudunk használni, ha gépünkön, vagy ahova be vagyunk jelentkezve, található megfelelő kliens program. PL. Unixban a tin, VMS alatt a news.

Jó példa erre a könyvtárakban használt Egyetemes Tízes 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.



- 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. kereszt-hivatkozások (jelölése: ->, jelentése: lásd).

A lapszámon leírt okok miatt 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 keresésre jó példa a Gopher. Tipikus gopher példa egy könyvtárban való keresés, pl. adott könyv címe alapján. A gopher szerverek többnyire könnyen kezelhető menürendszerrel rendelkeznek, onnan lehet menüsinteken keresztül az információt megkeresni.

Előfordulhat, hogy a gopher adatbázisa több gépen helyezkedik el, ekkor a gopher program automatikusan kapcsolja azt a gépet, amely a kért információt tartalmazza.

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 böngésző-programok (Netscape, Explorer) 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

Ehhez hasonlít a mobil telefonos WAP...

Jelenleg a leggyorsabban terjedő, legnépszerűbb szolgáltatás az Interneten a Világméretű Háló, a WWW.

Sikerének oka, hogy látványos dokumentumok nézhetők vele, amik tele vannak kereszt-hivatkozásokkal (ez a hypertext), és képekkel, olyan, mint egy képes lexikon.

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

figyeli külön az egymás után beérkező igényeket, mindet új kérésként kezel, 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.

Ezekre 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

A dokumentumok logikai struktúráját a HTML (Hyper Text Markup Language) jelölései segítségével lehet szabályozni [10]. A HTML arra készült, hogy segítségével a dokumentumok szokásos, sorban egymás utáni olvasása helyett, a szövegben elhelyezett kapcsolatok alapján 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 hypertextnek 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 például a Microsoft Windows, illetve a Windows alatt futó programok súgói. Amennyiben a szöveg mellett más médiaobjektum is megjelenik, akkor hipermédiáról beszélünk.

A hálózaton az objektumok, illetve ezek részei közötti kapcsolatok magába a szövegbe épülnek be megjelölt szavak és grafikus elemek formájában. Amikor egy ilyenre a felhasználó az egérrel rákattint, a rendszer automatikusan létrehozza a kapcsolatot, és a kapcsolt objektumot megjeleníti a képernyőn (vagy ha hang, lejátszsa...)

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.

Egyetemes leírás, amellyel a különböző forrásokra lehet hivatkozni.



A dokumentum fogalmát itt általánosabban kell értelmeznünk: ezek objektumok, amelyek lehetnek: szöveg, kép(grafika), hang (zene), de akár mozgókép (film) is.

A kapcsolatok eredményeként adott információ a hálózat bármely pontjáról megszerezhető, illetve ugyanahhoz az információhoz több úton is el lehet jutni a különböző kapcsolatokon keresztül.

Az előbbieken leírtak illusztrálására egy mintaoldal forrásnyelvi alakja a következőképpen néz ki:

```
<HTML>
<HEAD>
<TITLE>Mintaoldal</TITLE>
</HEAD>
<BODY BACKGROUND="bg_hatter.jpg">
<H2>Valaki mintaoldala</H2>
<CENTER>
<IMG SRC = "kep1.gif" BORDER=3><BR>
</CENTER>
<HR> Ez az oldal a legegyszerűbb utasítások felhasználásával
készült. P>
<PRE>
Ez egy rövid előre formázott szöveg csak az illusztráció miatt.
</PRE><P>
Két dolgot választható <BR>
<UL>
<LI><A HREF = "http://alpha1.obuda.kando.hu/~konya/">Ugrás a
szerző Honlapjára...</A>
<LI><A HREF = "start.html">Tartalomjegyzék...</A>
</UL>
Itt egy kép, <A HREF = "start.html"> <IMG SRC = "up.gif"></A>
amivel szintén visszaléphet egy oldalt.
</BODY>
</HTML>
```

A legtöbbször minden egyes link hivatkozás egy másik HTML oldalra (azért ez alól van kivétel.) ami a Világ bármely pontján lehet.

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://alpha1.obuda.kando.hu:8080/WEB/ai/leiras.html>

A kapcsolt (a kapcsolatban hivatkozott) állomány a leiras.html nevet viseli, a WEB/ai könyvtárban található az alpha1.obuda.kando.hu gépen, amely a Web-szolgáltatásokhoz az alapértelmezésként szereplő

Az URL-ek a HTML-dokumentumba beépített szabályos szerkezetű sorok, segítségével hozható létre az a logikai szerkezet és dokumentum kapcsolat, ami a WWW hypertext lényege.

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

80-as port helyett a 8080-at használja. A kiszolgáló a HTTP-protokollal érhető el. A kiszolgáló Internet-neve helyett IP-címe is használható:

<http://193.224.41.1:8000/>

Az URL a következő információkat tartalmazza:

- **a protokollt, amelyet az adott forrás eléréséhez használunk** (ftp, http, gopher stb.); Az URL első tagja azt az adott forrás eléréséhez használandó protokollt adja meg. Az URL segítségével az Interneten használt legtöbb információforrás elérhető.
- **annak a kiszolgálónak az Internet-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 megjelenítő-program 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 kettősponttal (:) 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.

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

Azonos könyvtárban lévő dokumentumok esetén elég csak először a teljes URL-t megadni, utána már elég a többi fájlnak csak a nevét megadni.

<http://alpha1.obuda.kando.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. Ezekre a könyvtárakra való hivatkozás a ~ karakterrel kezdődik, és a könyvtári hivatkozás a felhasználó neve. A ~ karakter azt jelzi a kiszolgáló számára, hogy ez nem egy szokásos alkönyvtár, hanem az adott felhasználó alkönyvtárában kell az állományokat keresni. Például a konya felhasználói névhez tartozó személyes dokumentumok a

`http://alpha1.obuda.kando.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, otthlap, 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 ... Image (binary)
DIR ...	directory

Ha engem pl. 'konya'-nak hívnak, es 'golem' a jelszavam, akkor egy nem anonymous FTP-URL :

`ftp://konya:golem@alpha1.obuda.kando.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.

A CGI

Ahogy azt már leírtuk, a HTTP-protokollt a WWW ügyfél 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

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



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 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 dokumentum formájában továbbítja a kérést küldő megjelenítő-programnak.

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 ezeket a kéréseket egymástól teljesen függetlenül kezeli, és minden dokumentum elküldése után le is zárja a kapcsolatot.

Ez az állapotmentesség biztosítja, hogy a kiszolgáló mindenki számára egyformán elérhető és gyors.

A HTTP-kapcsolat négy lépése:[10]

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-fejlécből 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.



Ez az eljárás azt jelenti, hogy a kapcsolat során csak egy dokumentumot lehet átadni, illetve egyetlen feldolgozás megy végbe.

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.

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 kapcsolat (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 kapcsolatkerés eredményére utaló kód;
- az elküldött dokumentum hossza.

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 többi egyenként a grafikáknak, illetve képeknek.



COOKIE

Felmerül az az igény is, hogy a szerver a kérés küldése mellett egyéb információt is kapjon a klienstől. A cookie egy olyan azonosító, amelyet a kliens-oldali böngésző program (pl. Netscape Navigator, Internet Explorer, stb.) eltárol, és minden egyes, a megfelelő szerver felé irányuló kérés esetén a szervernek elküld.

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 lejáratí határideje, ekkor a kliens csak a megadott ideig használja azt, annak lejárta után "elfelejti". Ha nincs megadva lejáratí határidő, akkor a cookie csak a browserből való kilépésig él.

PORTÁLOK

Egyre jobban terjednek az olyan WEB helyek, amelyek számos szolgáltatással „csábítják” oda az internetezőket. Ezeket hívjuk **portál**-oknak. Ezek lényegében keresők köré szerveződnek, és több szolgáltatást gyűjtöttek egybe:

- Hírszolgáltatás
- Csevegő csatorna (chat)
- Sportrovat,
- Internetes vásárlóhely,
- Ingyenes e-mail
- Szaknévsor,
- Időjárás,
- térkép

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 „anyagtalán személytelenség” miatt, etikai kérdésekről is érdemes szót ejteni. Stílszerűen ezek egy szabályrendszerben, az. ún. Netiquette-ben (hálózati etikett) szerepelnek. Igaz, hogy az Internet a nagy szabadság egyik

A cookie-k nem biztosítanak semmiféle információszerzési vagy visszaélésre alkalmas lehetőséget. Nem lehet velük a kliensről vagy a futtató gépről információkat szerezni. Egyszerűen csak a kommunikáció egyszerűsítésére szolgálnak.



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 háttérünket, 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.

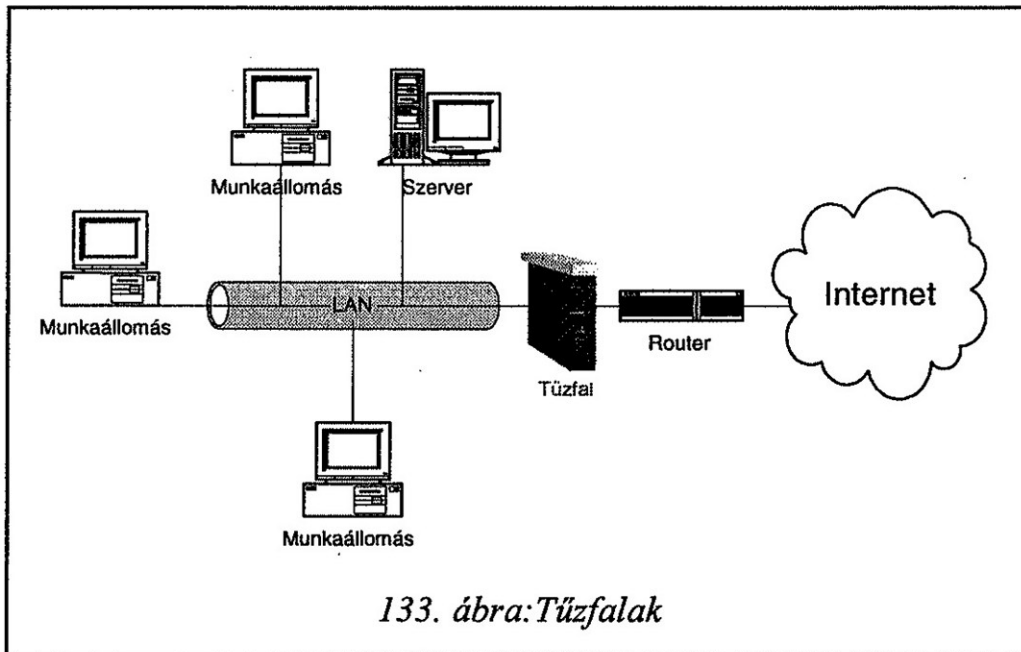
Ne éljünk vissza azzal, hogy a vitapartner fizikailag úgysem tud visszavágni. Mert amit élőben nem mernénk szemébe mondani, leírhatjuk levélben, hiszen nem kell senkivel szembe néznünk. Az ilyen magatartás gyakran vezet kilátástalan vitákhoz az egyes fórumokon.

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, pl. ne fűzzünk a leveleinkhez szép nagy ASCII grafikát aláírásunkként, mert sok felhasználó olyan vonalat használ amelynek kicsi az adatátviteli sebessége.

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 szolgáltatók, mennyire marad nonprofit a Hálózat, de reméljük, a gerinchálózatok, az Internet túlnyomó része legalább megmarad nonprofit szervezésűnek, vagy legalábbis a jövőben az Internet használata nem kerül többé egy átlagpolgárnak, mint ma egy TV előfizetési díj.

Ez konkrétan általában nincs megtiltva, a felhasználó józan belátásában célszerű reménykedni.

TÚZFALAK ÉS PROXY SZERVEREK



133. ábra: Tűzfal

A **tűzfal** (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 döntenek el, hogy az továbbítható-e vagy sem, ill. milyen egyéb intézkedéseket kell tenni. Azt is mondhatjuk, hogy ezek az eszközök a hálózati rétegben működnek

(bár az egyes IP csomagoknál felhasználják a szállítási réteg információit is, például a TCP portszámot), mivel az általuk kezelt objektumok a harmadik szint protokollemei (PDU-k).

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. melyikre szeretne távozni

A ma forgalomban levő router rendelkezik alapvető tűzfal funkcióval (pl. Cisco router-ek esetén ún. ACL-ek segítségével).

Alapvetően olyan helyen alkalmazhatók, ahol pontosan meghúzható két vagy több (al)hálózat határvonala mind topológiai mind funkcionális szempontból.

Tipikus példa erre egy intézmény Internet kapcsolata.

- egyéb protokollspecifikus információk: pl. TCP SYN, ACK, RST flag-ek, IP töredékek (fragments). Például a TCP SYN bitek figyelésével lehetőség nyílik kapcsolatfelépítési kérések blokkolására, így bár az IP csomagokat mindkét irányban átengedjük, kapcsolatot csak az egyik irányból lehet kezdeményezni.

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 alkalmazásszintű 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 tűzfaloknak hívják)

Ezeket a tűzfalakat **csomagszűrő tűzfal**oknak (packet filter firewall) nevezzük.

Ezzel szemben a **proxy**-k olyan eszközök, ahol a szűrést magasabb rétegben valósították meg, tipikusan az alkalmazási szinten (application layer). Így lehetőség nyílik a kapcsolatok 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

A két módszer előnyeit próbálja meg egyesíteni az utóbbi években kifejlesztett technológia, mely a ma kereskedelemben kapható számos tűzfal alapját képezi. Ezeket a tűzfalakat ún. **Stateful Packet Filter Firewall**-oknak nevezik, melyek a vizsgálandó csomagok által tartalmazott információon túl képesek figyelembe venni az addig érkezett csomagokban lévő információt is. Tulajdonképpen permanens 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.

Tehát egy cégnél a proxy szerveren beállítják, hogy pl. a *.nudegirl.com ne legyen látható, akkor senki nem fog munkaidőben leányzókat nézegetni (hacsak nem talál mást :-)

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

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 részben átfedő két 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 alapján 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.

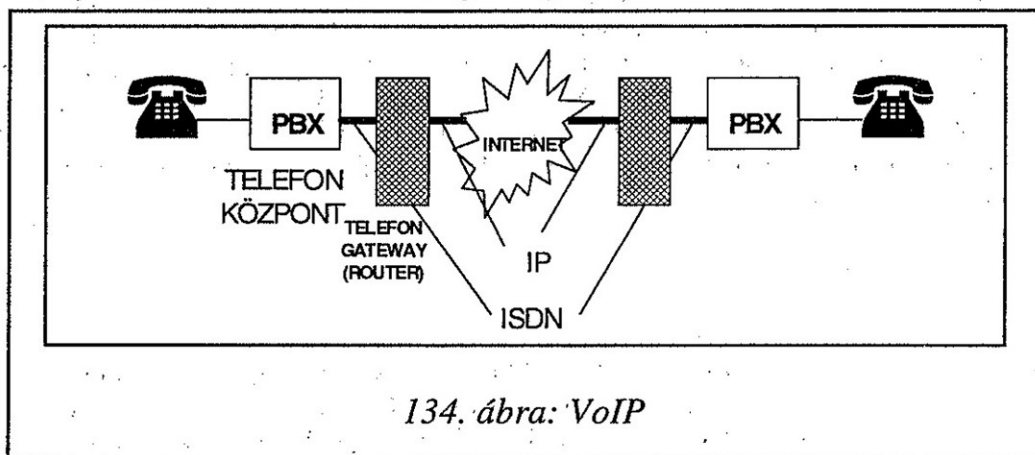
Ezt szeretik a nagyobb szolgáltató cégek használni, mert egészen jelentős (8-15 százalékos) sávszélesség-megtakarítást eredményez.

VALÓS IDEJŰ HANGTOVÁBBÍTÁS INTERNETEN (VOIP=VOICE OVER IP)

A bevezetőben említett távközlés-számítástechnika közeledésének példája az interneten keresztül történő telefonálás. A megvalósításhoz az IP telefon gateway szükséges. Az alapfunkcióját könnyű definiálni: ez az az eszköz amely átalakítja a „tradicionális” analóg hang adatot digitális adattá és az IP hálózaton keresztül, mint pl. az Internet elküldi egy másik gateway-nek amely visszaalakítja analóg adattá és tovább küldi egy telefonra.

A tényleges megvalósítás a **QSIG** protokollal történik.

A QSIG egy jelzésrendszer, amely telefonközpontok között, vagy a telefonközpont és a telefonközpont között használatos. Lényegében egy változata az ISDN D-csatorna jelzésrendszerének. Felhasználva a QSIG lehetőségeit, egyszerű módon lehetséges összekötni a



134. ábra: VoIP

hagyományos telefonhálózatot, és a csomagkapcsolt hálózatokat. Az alközpontok a telefonhálózat irányába a hagyományos telefon jelzéseket használja, míg a másik irányba a QSIG jelzéseket, ahogy ezt az **134. ábrán** is látható. A telefonhálózatról érkező hívás egy Primer ISDN-n keresztül csatlakozik a vonalkezelői számítógéphez, amely a kommunikációt, az **jogosultságot (autentikációt)**, a hívószám-felismerést végzi. A vonalkezelő továbbítja a hívás kérelmet az útválasztónak (routernek), aki a hívott szám alapján továbbítja hívást. Természetesen ez csak az egyik műszaki megvalósítási lehetőség.

A LEGFONTOSABB RFC DOKUMENTUMOK

Az itt szereplő részlet egy Interneten elérhető anyagból származik, aminek címe: Bevezetés a TCP/IP protokollba. Az eredeti dokumentáció:

Original Document Copyright © 1987, Charles L. Hedrick
Computer Science Facilities Group
Rutgers
New Jersey State University

Hungarian translation © 1996, Vincze Tamás.

Az Internet szabványokat RFC-knek hívják, ami a Request For Comments (Hozzászólásra, megvitatásra készített anyag) kifejezés rövidítése. Ha megszületik egy szabványtervezet, akkor azt először ajánlasként teszik közzé, és kap egy RFC számot. Ha végül az ajánlást elfogadják, akkor Hivatalos Internet Protokoll (Official Internet Protocols) válik belőle, de továbbra is az RFC számmal hivatkoznak rá. Megállapodás szerint minden RFC új számot kap, ha átdolgozzák. Két fontos RFC, az "Internet Számok" (RFC 1166) és a "Hivatalos Internet Protokollok" (RFC 1011) a tartalma miatt nagyon gyakran változik. A legutóbbi verzió száma az rfc-index.txt-ben található meg. A TCP/IP iránt érdeklődőknek javasolt az IP-t leíró RFC 791 tanulmányozása. Az RFC 1812, 1716 és 1009 szintén hasznos lehet. Ezekben az NSFnet által használt átjárók specifikációja, valamint az útvonal-választás szerepel.

Mint ilyen, rengeteg, TCP/IP technológiával kapcsolatos részt tartalmaz. Érdemes áttanulmányozni legalább egy alkalmazói protokollt, hogy érezzük a dolog gyakorlati részét is. Erre talán a legjobb a levelezés leírása (RFC 821/822). A TCP (RFC 793) persze alapműnek számít. A specifikáció eléggé összetett, így ennek tanulmányozása csak akkor javasolt, ha elég idő és türelem áll rendelkezésünkre a figyelmes olvasáshoz. Szerencsére Jon Postel, a főbb RFC-k szerzője, nagyon jól ír. A TCP RFC-t sokkal könnyebb olvasni, mint ahogy azt a tartalma alapján gondolnánk. Idővel a többi RFC-t is bátran nézegessük.

Következzen tehát a felsorolás:

rfc-index.txt	az összes RFC listája
rfc1122/3	Követelmények az Internet hosztok felé. Több protokoll áttekintése. A protokollok gyengéinek, a gyártók által elfogadott konvencióknak, a gyakorlatban fellépő problémáknak, a problémák megoldásainak a listája. Egy adott protokoll tanulmányozása során ne felejtsük el figyelmesen átnézni, mert a protokollokat leíró rfc-k ezeket az információkat nem tartalmazzák. Ugyanez vonatkozik az rfc1009-re is.
rfc1012	az RFC-k teljesebb listája

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

rfc1011	Hivatalos Protokollok. Hasznos az átböngészése, hiszen itt olvasható, hogy milyen feladatot látnak el az egyes protokollok. Leírja továbbá, hogy melyik RFC vált szabvánnyá.
rfc1010	Kiosztott Számok. Az Internet-tel dolgozva gyakran lehet erre referenciaként szükség. Olvasni nem olyan izgalmas. A hivatalosan definiált jól-ismert számokat és egyebeket listázza. A legutóbbi változata az rfc1700 Internet Számok nevet viseli.
rfc1009	Követelmények az Internet Átjárók felé. Jól használható bevezetést nyújt az IP útvonal-választáshoz és az átjárókhoz. (Lásd még: rfc1716, rfc1812.)
rfc1001/2	netBIOS: hálózattervezés PC-vel
rfc973	tartományok aktualizálása. Ezen a téren sok új információ jelent meg. Az rfc1034 és rfc1035 újabb verziót jelölnek. Ezek aktualizálása az rfc1101, rfc1876 és az rfc1348, rfc1637, rfc1706.
rfc959	FTP (állományátvitel)
rfc950	alhálózatok
rfc937	POP2: levelek olvasása PC-n
rfc894	IP továbbítása Ethernet-en, lásd az rfc826-t is
rfc882/3	tartományok ('hosztnév IP cím' megfeleltetés, UUCP). Lásd még: rfc973.
rfc854/5	telnet — a távoli bejelentkezés protokollja
rfc826	ARP — Ethernet címek leképezési protokollja (IP címre)
rfc821/2	levelezés — ennek legutóbbi verziója az rfc1495. (Lásd még: rfc987, rfc1148, rfc1327 és rfc1026, rfc1138.)
rfc814	nevek és port-ok — általában az ismertebb port-okról
rfc793	TCP
rfc792	ICMP
rfc791	IP
rfc768	UDP
rip.doc	a legjobban elterjedt útvonal-választási protokoll részletei (—> RFC 1058)
ien-116	régebbi névkiszolgáló (pár rendszer még használja)
ien-48	Catenet modell, a TCP/IP mögötti filozófia általános ismertetése

A következő dokumentumok egy-egy szűkebb területre specializálódtak:

rfc813	TCP ablak, és nyugtázási stratégiák
rfc815	datagram összerakási technikák
rfc816	hibakizárési és -feloldási módszerek
rfc817	modularitás és hatékonyság az implementációkban
rfc879	a TCP maximális szegmensméret opciója
rfc896	torlódásszabályozás
rfc827	rfc888,904,975,985 EGP (Exterior Gateway Protocol) és azzal kapcsolatos témák
rfc968	A 'Twas the Night Before Start-up című szellemes verset olvashatjuk, melyben a szerző a hálózatok telepítésekor felbukkanó problémákat ecseteli.

A legfontosabb RFC-k három kötetes gyűjteménye a DDN Protocol Handbook (DDN Protokoll Kézikönyv, 1985; ~12 cm vastag), amely a

DDN Network Information Center, SRI International, 333 Ravenswood Avenue, Menlo Park, California 94025, USA (telefon: ++1-800-235-3155)

címen rendelhető. Az RFC-k anonim FTP-vel is elérhetők a NIC.DDN.MIL címen. A dokumentumok nevei:

RFC:

/rfc/rfc-index.txt

/rfc/rfcN.txt, ahol N a kért RFC száma

Ajánlott még az InterNIC Directory and Database Services, ds.internic.net kiszolgáló anonim FTP elérése. A keresett RFC dokumentumok az rfc/rfc####.txt vagy rfc/rfc###.ps nevek alatt találhatóak, ahol a #### a kért RFC száma (kezdő nullák nincsenek benne). Ugyanezen kiszolgálótól levélben is kérhető a szolgáltatás. A mailserv@ds.internic.net címre az alábbi üzenetet kell küldeni:

document-by-name rfcNNNN

Itt az NNNN a kért rfc száma. Amennyiben postscript formátumban kell a szöveg, akkor a

document-by-name rfcNNNN.ps

üzenetet kell küldeni. Több RFC esetén azokat vesszővel válasszuk el, vagy minden kérést új sorba írjunk. Pl.:

document-by-name rfc1791, rfc1792

vagy

document-by-name rfc1791

document-by-name rfc1792

Magyarországon az

ftp://sunserv.kfki.hu/pub/documents/rfc/

címen érhetők el a különböző rfc dokumentumok.

ELLENŐRZŐ KÉRDÉSEK



1. Milyen rétegei vannak az Internet hálózatnak, és ez hogyan viszonyul az OSI modellhez? Mi az egyes rétegek feladata?
2. Mi a TCP protokoll feladata? Milyen információkat tartalmaz a TCP csomag?
3. Mi az IP protokoll feladata? Milyen információkat tartalmaz a IP csomag?
4. Ismertesse az Internet címzési rendszerét! Mik azok az A, B, C osztályú címek?
5. Magyarázza el a domén nevek rendszerét!
6. Milyen célt szolgál az ICMP protokoll?
7. Milyen célt szolgál az ARP protokoll? Mit jelent egy csomag „beburkolása”?
8. Mi az a socket?
9. Mutassa be az Email legfontosabb jellemzőit! Mi az SMTP? Milyen funkciókat kell megvalósítani egy levelező programnak?
10. Mi az UUENCODE/UUDECODE eljárás?
11. Mi az a mail szerver?
12. Mik azok a smiley-ik?
13. Mutassa be az FTP alkalmazás legfontosabb jellemzőit! Mi az anonymous FTP?
14. Mi az a TELNET? Milyen biztonsági problémákat okozhat?

7. A TCP/IP PROTOKOLL ÉS AZ INTERNET

15. Mi a finger? Mire használható?
16. Mik azok a levelezési listák? Mi a digest és a FAQ (GYIK) ?
17. Hogyan lehet az információkat szervezni a hálózaton? Mi a GOPHER?
18. Mi a WWW? Mi a HTML, URL?
19. Mi a hipertext és mi a hipermedia?
20. Mit jelent, és milyen célt szolgál a CGI?
21. Mi az a HTTP protokoll? Hogyan működik?
22. Mi az a cookie?
23. Mi a proxy szerver?
24. Sorolja fel a hálózati etikett (netikett) néhány alapszabályát!
25. Mik azok az RFC dokumentumok?

8. HÁLÓZATI FELÜGYELET

HÁLÓZATI FELÜGYELET (MENEDZSMENT)

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.

Erre együtt nincs magyar szó, azért használjuk **hálózati menedzsment** kifejezést.

A hálózati menedzsment több tevékenységből tevődik össze.

- **Hiba menedzsment** alatt a hálózati működési zavarainak érzékelését, behatárolását és kijavítását értjük.
- A **teljesítmény menedzsment** feladata a kiszámítható és hatékony szolgáltatás biztosítása és a jelenlegi kapacitások tervezését és tesztelését, valamint jövőbeli bővítését értjük.
- A **konfigurációs menedzsment** célja a rendszeresen jelentkező változások (ahogy a rendszer növekszik, gépek cserélődnek, stb.) és az alkotóelemek nevének és címeinek helyi adminisztrációjának kezelése.
- 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.

Ezeket a tevékenységeket egységesítették, szabványosították, és illusztrációként egy jellemző a TCP/IP hálózatokban elterjedt szabványos protokollt mutatunk be.

SNMP

A kidolgozott protokoll neve: **Simple Network Management Protocol – SNMP**, magyarul: egyszerű hálózatfelügyeleti protokoll. A protokoll alapjainak a leírását az RFC 1155-1158 és az RFC1213-as dokumentumok tartalmazzák. Az SNMP lényegében egy a hálózatfelügyelet alapjait magába foglaló hálózati kommunikációs meghatározásgyűjtemény. A hálózati környezet karbantartását, megfigyelését, módosítását (egyszóval: menedzselését) írja le. Olyan

8. HÁLÓZATI FELÜGYELET

módon tervezték, hogy a meglévő hálózatot a használatához ne kelljen nagyon megváltoztatni. Ezt úgy érik el, hogy a cserélt üzenetekben a hálózat adatokat küldünk.

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 a TCP/IP protokollt használja, és 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-beavatkozás.

- **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.

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 ún. csapda esemény küldésével jelzi saját felügyelőjének a

kommunikációs kísérletet. Egy ügynök három összetevőből áll:

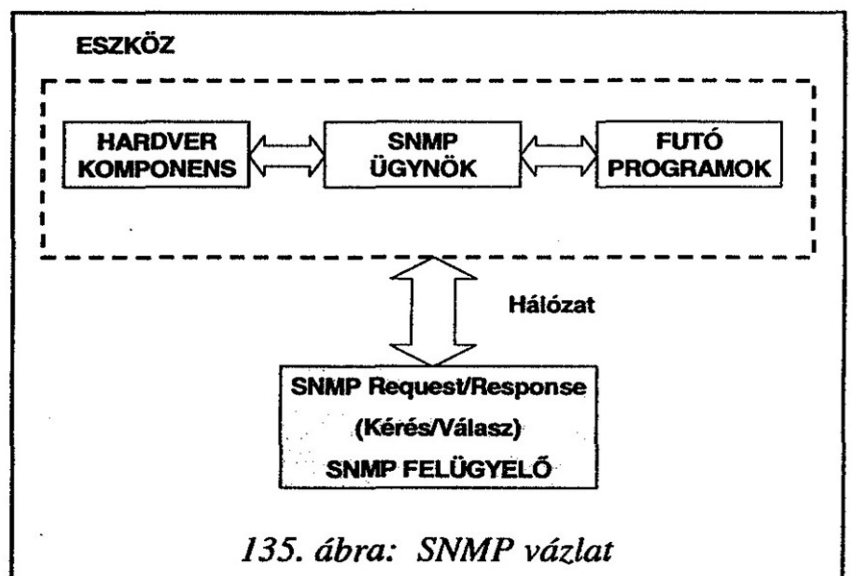
A hálózati menedzsment protokoll architektúrája két részből áll. Az első a kommunikációt kezeli, míg a második a vezérelhető adatokkal foglalkozik. Két szabvány van használatban: a

Simple Network Management Protocol (SNMP)

és a

Common Management Information Services (CMIP)
a TCP felett.

Mindkettő a **Management Information Base (MIB)** adatbázist használja



135. ábra: SNMP vázlat

<u>MIB kategória</u>	<u>Miről tartalmaz információt</u>
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 inernet TCP szoftverről
udp	UDP szoftverről
egp	EGP szoftverről

HÁLÓZATI MANAGE- MENT- MIB

A kategóriákhoz tartozó változókra néhány példája:

<u>MIB változó</u>	<u>Kategória</u>	<u>Jelentés</u>
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.

136. ábra: MIB adatok

- A protokollt kezelő és megvalósító részből,
- a figyeléshez szükséges komponensekből,
- valamint a gyűjtött adatokat tartalmazó **menedzsmen** **információs adatbázisból** (Management Information Base=**MIB**).

Minden információ elérés az ügynökön keresztül az információs adatbázis elérést 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 kapcsolattartást az ügynökökkel.

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.

8. HÁLÓZATI FELÜGYELET

A MIB a rendszerinformációk leírásához nyolc kategóriát definiált (136. ábra), és a kategóriákhoz szabványos elnevezésű változókat.

Ma már egyre több, még olcsóbb árkategóriájú hálózati eszközbe építik be ezt a menedzselési lehetőséget.

ELLENŐRZŐ KÉRDÉSEK

1. Mi a feledata a hálózati menedzsmentnek, és milyen részekből áll?
2. Milyen célt szolgál az SNMP ?
3. Mi az és az SNMP menedzser feladata, és hogyan működnek együtt?
4. Milyen szerepekben használják az SNMP ügynököket? Mi az a csapda esemény?
5. Milyen részekből áll egy ügynök?
6. Mit jelent, és hogyan épül fel a MIB?
7. Milyen információkat tárolunk a megfigyelt objektumról a MIB-ben?



A könyv olvasásakor és a tanulásakor mindig törekedjünk a legfontosabb dolgok megértésére, a lényegkiemelés-re...

A végére értünk...

VÉGE

Az emberek rang, hatalom és vagyon után sóvárognak, de ezek a dolgok a testhez képest jelentéktelenek. Ezért a bölcs csak annyit eszik, hogy az üresség megteljen és az energia el ne vesszen; úgy öltözik, hogy a testét eltakarja, s meg ne fázzon. Eredeti állapotához igazodik, és minden mást elutasít; szerzésre nem vágyik és nem gyűjt magának túl sokat.



Lao-Ce mondta:

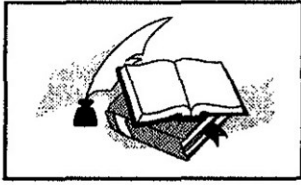
A boldogságnak nincs köze ranghoz és vagyonhoz: egyszerűen csak harmónia dolga.

Kezdődhet a tanulás...



VÉGE

ZSOLA



IRODALOMJEGYZÉK

- [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 foglalkozik 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 ülénk.
- [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 terjedős 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] Jutasi István-Mazgon Sándor: Adatkommunikációs hálózatok
Rádiótechnika Évkönyve, 1994. 16. - 24. oldal
Jó összefoglalása az adatkommunikációs hálózatokkal kapcsolatos alapismereteknek. A terjedelmi korlátok miatt a részletekre nem tér ki.
- [12] Hargittai Péter-Kaszanyiczki László: Internet haladóknak
LSI Oktatóközpont Budapest, 1996. ISBN 963 577 165 7
A könyv első részében az Internet alaposabb megismeréséhez nyújt segítséget, majd a PC-TCP programcsomag programjait mutatja be. Nagyon jó a könyvhöz adott CD !
- [13] Kis Balázs: Winternet
Szak Kiadó, 1997. ISBN 85564 5 5
- [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.

TÁRGYMUTATÓ

A,Á

adapterkártya, 173
 adási ablak, 126
 adatátviteli sebesség, 31
 adatkapcsolati réteg, 23
 ADSL, 97
 ADSL modem, 97
 Aktív illesztő, 44
 alapsávú koaxiális kábel, 40
 aldomén, 212
 alkalmazási réteg, 24
 ALOHA, 111
 AMI - Alternate Mark Inversion, 70
 anonymous ftp, 235
 áramhurok, 84
 arbitráció, 129
 archie, 236
 ARP, 220
 ARPA, 203
 ARPANET, 203
 ASCII karakter, 72
 ATM, 95
 ATM Adaptation Layer, 96
 ATM Adaptation Layer 5, 97
 ATMS, 95
 átvilágítás, 201
 AUI, 175
 autentikáció, 250

B

base64 kódolás, 233
 Bázis Adó-Vevő, 54
 befulladás, 146
 bithibák, 120
 bitorientált átvitel, 118
 bitorientált eljárás, 68
 BNC, 174
 bone, 225
 bridge, 195

C

CBR - Constant Bit Rate, 94
 CDP - Conditional Diphas, 71
 Ceasar-abc, 161
 cellacsoport, 51
 CGI, 243
 collision, 109
 cookie, 245
 CRC, 121

Cs

csillapítás, 30, 43
 CSMA/CD, 111
 csomag, 15
 csomagszűrő tűzfal, 248
 csoportcímezés, 16
 csővonal, 127

D

datagram, 222
 DCE (Data Circuit-Terminating Equipment), 79
 de-facto, 20
 default router, 216
 de-jure szabványok, 20
 demultiplexer, 33
 DES, 162
 dial-up-line, 50
 digitális bitcső (digital bit pipe), 89
 digitális multiplexelés, 35
 dinamikus útválasztás, 218
 DNS, 212
 domén (domain) nevek, 211
 doménnév, 212
 DTE (Data Terminal Equipment), 79
 Duplex átvitel, 19

E,É

EBCDIC, 158
 ECC, 120
 Egyetemes Tizes Osztályozás, 237
 egymódusú üvegszál, 44
 együttműködési réteg, 24
 előfizetői hurkok, 49
 Email, 229
 entitás, 25

F

FAQ, 237
 FDDI, 193
 FDM, 33
 Fél duplex, 18
 felhasználónév, 214
 felügyeleti adatbázis, 255
 finger, 236
 fizikai réteg, 22
 flow control, 19, 152
 fordított tanulás módszere, 144

forgalomirányítás, 140
 forgóablak, 126
 főállomás, 131
 FQDN, 212
 Frame Relay, 152
 FSK, 59
 FTP, 234
 ftpmail, 235
 funkcionális elem, 25

G

gateway, 195
 gopher, 238
 GPRS, 57
 Grafikus karakterek, 72
 GSM, 51

Gy

GYIK, 237

H

hálózatfelügyelő, 254
 hálózati kapcsoló pontok, 15
 hálózati réteg, 23
 Hamming távolság, 120
 handover, 51
 háromutas kézfogás, 156
 HDB3 - High Density Bipolar 3, 71
 HDLC, 131, 150, 152
 helyi központ, 49
 hipermedial, 239
 hipertext, 238
 holtpon, 148
 host, 212
 host table, 212
 hostcím, 212
 hoszt, 14
 hoszt-specifikus, 217
 HTML, 239
 HTTP, 243
 hullámhossz multiplexelést, 45

I,Í

ICMP, 219
 időosztásos kapcsoló, 92
 időreścserélő, 92
 IEC 625, 98
 IMP, 15

IRODALOMJEGYZÉK ÉS TÁRGYMUTATÓ

<p>INTERNIC, 212 IP, 207 IrDA, 46 ISDN, 87</p> <p style="text-align: center;">J</p> <p>jel-zaj viszony, 31</p> <p style="text-align: center;">K</p> <p>karakterorientált átvitel, 118 karakterorientált átviteli eljárás, 68 készletelési idő, 40 kezelői jogok, 200 kliens-szerver, 196 kombinált állomás, 131 kommunikációs alhálózatok, 14 könyvtár örökölt jogmaszk, 200 kriptológia, 161 kulcsfájl, 166</p> <p style="text-align: center;">L</p> <p>LAPB, 131 LCP, 227 Lefojtó-csomagok, 147 levelezési listák, 237 LLC, 170 LLC PDU, 171 login, 214</p> <p style="text-align: center;">M</p> <p>MAC, 169 Mail-szerver, 230 MAP, 194 MARK, 81 MAU, 175 megjelenítési réteg, 24 mellékállomás, 131, 137 mérték, 140 MIB, 256 migration, 201 mikrovezérlő, 98 MIME, 232 MNP, 63 modem, 58 multidrop, 131 multi-drop, 86 multimode, 43, 44</p> <p style="text-align: center;">N</p> <p>Name Server, 212 NCP, 227 NDS, 200 négyvezetékes átvitel., 86 névszolgáltató, 212</p>	<p>N-RESET, 155 NRZI - Non Return to Zero Invertive, 70 NT1 (Network Termination 1), 89 null-modem, 83</p> <p style="text-align: center;">Ny</p> <p>nyilvános hálózat, 17</p> <p style="text-align: center;">O,Ó</p> <p>oktet (octet), 68 OSI-modell, 22 OSPF, 218</p> <p style="text-align: center;">P</p> <p>PAD, 149 párhuzamos átvitel, 78 passzív illesztő, 44 passzív sínkábel, 89 PBX (Private Branch eXchange), 89 PCM (Pulse Code Modulation), 34 PE - Phase Encode, 71 peer-to-peer, 197 Permanent Virtual Circuits=PVC, 95 PGP, 164 piggy-back, 126 point-to-point, 15 PPP, 226 primitívek, 27 protocol stack, 194 protocol suite, 194 protokoll, 17 protokollkészlet, 194 protokollok, 228 protokoll-verem, 194 proxy, 248 PVC, 149</p> <p style="text-align: center;">Q</p> <p>QSIG proto-kol, 249 quoted-printable, 233</p> <p style="text-align: center;">R</p> <p>RARP, 222 recieving window, 127 repeater, 176 rés, 112 réteg, 17 réteginterfész, 17 RFC, 226, 228 RIP, 218 RLL, 159 roaming, 56</p>	<p>routing táblák, 139 RS-232-C, 79 RS-422, 86 RS-423-A, 85 RS-449, 85 RS-485, 86 RZ - Return to Zero, 70</p> <p style="text-align: center;">S</p> <p>SAP (Sevice Access Point), 25 SAP cím, 171 sávkorlátozás, 30 sávszélesség, 31 sending window, 126 signal to noise ratio, 31 SIM-kártyák, 53 SLIP, 226 SMS, 57 SMSC, 57 SNMP, 254 SNMP menedzser, 255 SNMP ügynök, 255 socket, 223 sodrott érpár, 38 Sorozathossz kódolás, 159 SPACE, 81 splitter, 97 START-STOP átvitel, 69 statikus útválasztás, 218 Statisztikai kódolás, 159 store-and-forward, 15 STP, 38 suballocation, 201 Subband kódolás, 160 subnet, 212 Switched Virtual Circuits=SVC, 95</p> <p style="text-align: center;">Sz</p> <p>szállítási réteg, 23 számítógép-hálózatok, 13 szelektív ismétlés, 127 szélessávú koaxiális kábel, 40 Szimplex, 18 szinkron átviteli módszer, 68 szint, 17 szolgáltatok, 25</p> <p style="text-align: center;">T</p> <p>T csatoló, 174 távhívó központok, 49 TCP, 205 token, 114, 115, 159, 184, 187 token ring, 186 TOP, 194 topológia, 17 torlódás, 146</p>
--	---	---

SZÁMÍTÓGÉP - HÁLÓZATOK

többszörös üvegszálnak, 43, 44
transparent proxy, 248
transzponder, 46
trörk, 49
tűzfal, 247

U,Ú

UDP, 219
Unicode, 77
URL, 239, 240
UTP, 38
UUENCODE/UUDECODE, 231

Ü,Ú

ügyfél-kiszolgáló modell, 196
ütközés, 109

V

vámpír csatlakozó, 175
vámprcsatlakozó, 41
VBR - Variable Bit Rate, 94
vételi ablak, 127
vezérjeles gyűrű, 186
vezérjeles sín, 184
vezérlőállomás, 130
vezérlőkarakterek, 72
virtuális áramkör, 137

virtuális hívás, 149
visszaverődés, 43
voice-modem, 65
VoIP=Voice over IP, 249
vonalkapcsolás, 32, 37, 107, 136
VSAT, 47

W

WAP, 57
WWW, 238

X

X.25 DTE-DCE, 150



GÁBOR DÉNES FŐISKOLA

MŰKÖDÉSI HELYEI

Budapest Informatikai Rendszerek Intézete:

Vezető: *Dr. Kovács Magda*
1037 Budapest, Bécsi út 324.
Információ

Tel.: 06-1-436-65-19; Fax: 06-1-436-65-28
Honlap: www.gdf.hu

Informatikai Alkalmazások Intézete:

Vezető: *Dr. Zárda Sarolta*
1115 Budapest, Etele út 68.
1139 Budapest, Frangepán u. 56.
Tanulmányi Osztály

Tel.: 06-1-203-02-83; 06-1-203-03-04/8900
E-mail: tanoszt@gdf.hu Honlap: www.gdf.hu

Baja

Vezető: *Búcsú Lajos*
6500 Baja, Oltványi u. 14.

Tel./Fax: 06-79-426-427; E-mail: oktatas@fit.hu
Honlap: www.fit.hu

Balatonboglár

Vezető: *Lakos István*
8630 Balatonboglár, Szabadság u. 41.

Tel.: 06-85-351-633; Tel./Fax: 06-85-351-316
E-mail: lakist@mathiasz.sulinet.hu
Honlap: www.extra.hu/balatonboglár

Békéscsaba

Vezető: *Kovács Zoltánné*
5600 Békéscsaba, Andrássy u. 73/1.

Tel.: 06-66-448-385; 06-30-3552-039;
Fax: 06-66-448-385; E-mail: gdf-bkk@nap-szam.hu
Honlap: www.gdf-bkk-03.nap-szam.hu

Cegléd

Vezető: *Sági Ferenc*
2700 Cegléd, Kossuth F.u. 32.

Tel.: 06-20-9575-958; 06-53-312-568;
Tel./Fax: 06-53-311-695; E-mail: sagif@infotars.hu
Honlap: www.extra.hu/gdf-cegled

Debrecen

Vezető: *Kovács Jánosné*
4029 Debrecen, Maróthy Gy. u. 5-7.

Tel.: 06-52-418-660; Fax: 06-52-425-761;
E-mail: gdf@cts.hu

Dunaújváros Szervezése Székesfehérváron

Vezető: *Róth Péter*
8000 Székesfehérvár, Mátyás király krt. 5.

Tel./Fax: 06-22-348-542; E-mail: roker@mail.alba.hu

Eger

Vezető: *Csathó Csaba*
3300 Eger, Széchenyi u. 58.

Tel.: 06-36-411-811; 06-30-9581-822;
Fax: 06-36-421-822; E-mail: sprinter@mail.agria.hu
Honlap: www.sprinter.agria.hu

Esztergom

Vezető: *Fűrész Tiborné*
2500 Esztergom, Bánomi út 8.

Tel./Fax: 06-1-230-43-76; E-mail: fureszt@axelero.hu
Honlap: www.egominfo.hu/GDF

Érd

Vezető: *Budai Csaba*
2030 Érd, Széchenyi tér 1.

Tel.: 06-20-9344-770; Fax: 06-1-283-76-84;
E-mail: budacsa@mail.datanet.hu

Gyöngyös

Vezető: *Gerják István*
3200 Gyöngyös Körösi Csoma S. u. 9.

Tel./Fax: 06-37-303-683;
E-mail: gerjak@mail.datanet.hu, trivium@tvnetwork.hu

Győr

Vezető: *Domonkos Gyuláné*
9023 Győr, Szabolcska M. u. 1/b.

Tel.: 06-96-526-900/113; Fax: 06-96-523-367
E-mail: gyulane.domonkos@gyor.szuv.hu

Hódmezővásárhely

Vezető: *Gál József*
6800 Hódmezővásárhely, Szántó K. J. u. 64.

Tel.: 06-62-535-536; 06-30-3135-873;
Fax: 06-62-535-530; E-mail: imsuli@delfin.hu
Honlap: www.gdf.vasarhely.hu

Isaszeg

Vezető: *Keresztúri Juszti*
2117 Isaszeg, Madách u. 1/a

Tel.: 06-28-496-206; 06-28-494-559;
Tel./Fax: 06-28-496-205 E-mail: gdfisa@mail.sziszi.hu

Kaposvár

Vezető: *dr. Paál Jenő*
7400 Kaposvár, Guba S. u. 36-40.

Tel.: 06-82-314-155/147; Fax: 06-82-320-757;
E-mail: posane@atk.kaposvar.pate.hu

Kecskemét

Vezető: *Pósfayné Bakota Éva*
6000 Kecskemét, Hunyadi János tér 2.

Tel.: 06-76-411-494; 06-20-9762-100; 06-20-9935-735;
Fax: 06-76-411-041; E-mail: gdfkecsk@axelero.hu

Keszthely

Vezető: *dr. Vargáné Dugonics Rita*
8360 Keszthely, Deák F. u. 57.

Tel.: 06-83-312-330/262; 06-83-312-330/290;
Fax: 06-83-314-334; E-mail: vdr@georgikon.hu

Mátészalka

Vezető: *Tóth Józsefné*
4700 Mátészalka, Seregély u. 17/A.

Tel.: 06-44-417-889; 06-20-9743-125;
Tel./Fax: 06-44-417-889;

E-mail: fic@okito.hu, tothne@okito.hu
Honlap: www.okito.hu

Miskolc

Vezető: *Dr. Czap László*
3535 Miskolc, Árpád u. 2.

Tel./Fax: 06-46-414-429;
E-mail: czap@mazsola.iit.uni-miskolc.hu
Honlap: www.gdfm.hq.hu

Nagykanizsa Vezető: *Németh Zoltán*
8800 Nagykanizsa, Ady Endre u. 74/a
Tel.: 06-93-312-383; 06-93-313-010/730; 06-30-9370-605;
Fax: 06-93-310-107; E-mail: nemethz.gdf@chello.hu

Nyíregyháza Vezető: *Méhész János*
4400 Nyíregyháza, Vasvári Pál u. 1.
Tel.: 06-42-406-844; 06-42-406-850; 06-20-3559-390;
Fax: 06-42-406-848; E-mail: Janos.Mehesz@chello.hu
Honlap: www.freeweb.hu/gdf-nyhz

Pápa Vezető: *Gerics Erzsébet*
8500 Pápa, Budai Nagy Antal u. 1.
Tel.: 06-89-311-297, 06-20-9583-058;
Tel/Fax: 06-89-311-297; E-mail: gerics@geqe-comp.hu

Pécs Nyílt nap helye:
Zipernowsky Károly
Műszaki Szakközép isk. 48-as tér 2.
Vezető: *Kedves Vera*
7624 Pécs, Tiborc u. 38/c
Tel.: 06-72-213-412; Fax: 06-72-310-259;
E-mail: gdfpecs@axelero.hu

Pilisvörösvár Vezető: *Fűrész Tiborné*
2085 Pilisvörösvár, Szabadság u. 21.
Tel./Fax: 06-1-230-4376; E-mail: fureszt@axelero.hu
Honlap: www.extra.hu/gdfpilisvorosvar

Salgótarján Vezető: *dr. Agócs József*
3100 Salgótarján, Kossuth u. 8.
Tel.: 06-32-416-833; Fax: 06-32-317-420;
E-mail: jozsef.agocs@ncsszi.hu

Sátoraljaújhely Szervezése Mátészalkán
Vezető: *Tóth Józsefné*
4700 Mátészalka, Seregély u. 17/A.
Tel.: 06-47-326-583; 06-20-9743-125;
Tel./Fax: 06-44-417-889; E-mail: zbeata@freemail.hu

Siófok Szervezése Székesfehérváron
Vezető: *Róth Péter*
8000 Székesfehérvár, Mátyás király krt. 5.
Tel./Fax: 06-22-348-542; E-mail: roker@mail.alba.hu

Sopron Vezető: *dr. Molnár László*
9400 Sopron, Bajcsi-Zsilinszky u. 4.
Tel.: 06-99-518-182; 06-30-4112-941;
Fax: 06-99-518-259; E-mail: molnarl@fmk.nyne.hu

Szeged Vezető: *Malincsa János*
6720 Szeged, Kígyó u. 4.
Tel./Fax: 06-62-423-258,
E-mail: malincsa@mailbox.hu, gdfszeged@tvnetwork.hu
Honlap: www.gdfszeged.hu

Szekszárd Vezető: *Doszkocs László*
7100 Szekszárd, Rákóczi u. 70.
Tel.: 06-74-413-435; Fax: 06-74-419-541;
E-mail: gdf@kvantum.hu
Honlap: <http://gdf.kvantum.hu/>

Székesfehérvár Vezető: *Róth Péter*
8000 Székesfehérvár, Mátyás király krt. 5.
Tel./Fax: 06-22-348-542; E-mail: roker@mail.alba.hu

Szolnok Vezető: *Nyáry László*
5000 Solnok, Arany J. u. 4.
Tel.: 06-56-375-122; Fax: 06-56-511-262;
E-mail: titszol@externet.hu

Szombathely Nyílt nap helye:
POTE Szhelyi Képzési Kp. Jókai Mór u. 14.
Vezető: *Suri Judit*
9700 Szombathely, Kisfaludy u. 51.
Tel.: 06-94-501-894; Fax: 06-94-501-899;
E-mail: gdf_okt@flagnet.hu Honlap: www.flagnet.hu

Tatabánya Vezető: *Gimesi Lászlóné*
2800 Tatabánya, Vértanúk tere 13.
Tel./Fax: 06-34-510-460; Tel.: 06-34-510-461;
E-mail: gnkft@axelero.hu Honlap: www.gnkft.hu

Vác Vezető: *dr. Molnár Lajos*
2601 Vác, Konstantin tér 6.
Tel.: 06-27-317-077; Fax: 06-27-315-093;
E-mail: boronkay@vac.hu

Veszprém Vezető: *Nagné László Mária*
8201 Veszprém, Egyetem u. 10.
Tel.: 06-88-422-022/4191; Fax: 06-88-429-671;
E-mail: laszlo@almos.vein.hu

Zalaegerszeg Vezető: *Dr. Sárvarnyé Kiss Katalin*
8900 Zalaegerszeg, Rákóczi u. 4-8.
Tel.: 06-92-311-229; Fax: 06-92-510-590;
E-mail: zeggdf@szam.hu

Határon túli központok:

Románia Erdélyi központok szervezése Kolozsváron (Cluj-Napoca) Vezető: *Dr. Selinger Sándor*
3400 Románia Kolozsvár (Cluj-Napoca),
str.: I. Budai Deleanu 64. Levélcím: OFP 5. CP 737
Tel./Fax: 00-40/2-64-431-841; E-mail: gdf@gdf.ro
Honlap: www.gdf.ro

Csíkszereda (Miercurea Ciuc)
Sepsiszentgyörgy (Sfântu-Gheorghe)
Székelyudvarhely (Odorheiu-Secuiesc)
Marosvásárhely (Târgu-Mureş)
Nagyvárad (Oradea)
Szatmárnémeti (Satu-Mare)

Jugoszlávia Szabadka Vezető: *Bóni László*
24000 Subotica Mosa Pijade 74. Jugoszlavia
Tel.: 00-381-24-546-067; 00-381-24-546-463;
Fax: 00-381-24-546-021; E-mail: imc@magnetron.co.yu

Szlovákia

Kassa (Košice) Vezető: *Samuelis László*
04001 Košice, Moyzesova 58., Slovenská Republika,
Tel.: 00-421-55-622-8126, Fax: 00-421-55-622-1043;
E-mail: samuelis@tuke.sk, jsamueli@kosice.upjs.sk
Honlap: www.teledom.sk

Diószeg Sládkovičovo Vezető: *Ladislav Spalek*
92521 Sládkovičovo, Fučíkova ul. 269. Slovenská Republika
Tel.: 00-421-31-788-1733; 00-421-31-788-1712;
Fax: 00-421-31-788-1710
E-mail: ladislav.spalek@ivcsr.sk, takacsatti@hotmail.com



A szerző dr. Kónya László, a Budapesti Műszaki Egyetem Villamosmérnöki Karán végzett 1976-ban, docensként aktív oktatási tevékenységet folytat két főiskolán:

- A Budapesti Műszaki Főiskola Kandó Kálmán Villamosmérnöki Karának Automatizálási Intézetében mikrokontrollerek alkalmazástechnikáját tanítja,*
- míg a Gábor Dénes Főiskolán a Számítógép-hálózatok tárgy vezető tanára.*

Kutatási területe a mikrokontrollerek, számítógépes-hálózatok oktatása, a korszerű oktatási módszerek bevezetése, és elterjesztése, a mikrokontrollerek alkalmazástechnikájának aktív, gyakorlati művelése. További részletek a szerző honlapján olvashatók:

alpha1.obuda.kando.hu/~konya

ISBN 963-577-222-X



9 789635 772223