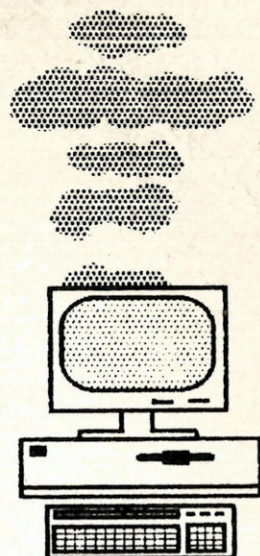
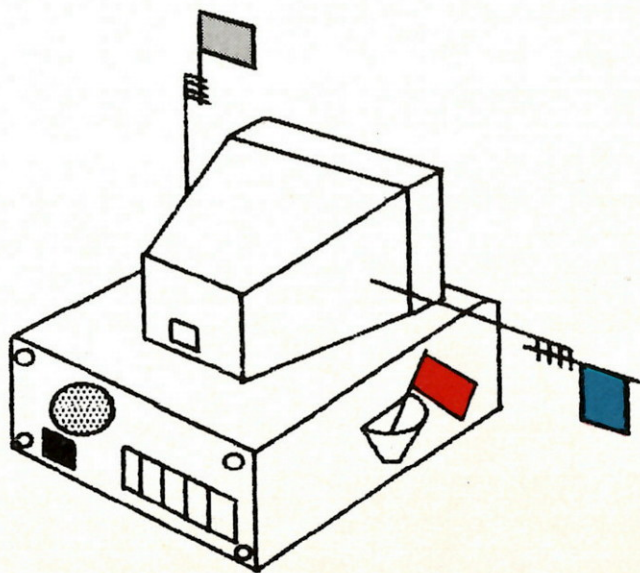


**CSÓRIÁN SÁNDOR**



# Számítógépes kommunikáció

**Hálózatokról  
felhasználóknak  
és döntéshozóknak**



**ALAPLAP KÖNYVEK 10**

**Csórián Sándor**

**Számítógépes kommunikáció**

**Hálózatokról felhasználóknak és döntéshozóknak**



Csórián Sándor

# **SZÁMÍTÓGÉPES KOMMUNIKÁCIÓ**

Hálózatokról felhasználóknak  
és döntéshozóknak



**CÉDRUS KIADÓ**

**Készült az SzKI támogatásával**

**ISSN 0866-434x**  
**ISBN 963 7429 48 4**

**© Csórián Sándor, 1993**



# Ajánlás

A számítástechnika és a kommunikáció eszközei, valamint szolgáltatásai mára már mindennapi életünk szerves részévé váltak.

Ahogy gazdasági rendszerünk a mindannyiunk által szorgalmazott átalakulás eredményeként egyre inkább közeledni próbál a fejlettebb országok színvonalához, úgy válik szerepük még jobban meghatározóvá, mégpedig társadalmi és gazdasági életünk szinte minden területén.

Az SzKI Rt. számítástechnikai és kommunikációs rendszerfejlesztői, rendszerintegrálói munkája során azt tapasztaltuk, hogy az előrelépéshez elengedhetetlenül szükséges új informatikai rendszerek és szolgáltatások bevezetése és befogadása elképzelhetetlen anélkül, hogy a döntéshozókat, valamint a majdani üzemeltetőket és felhasználókat e rendszerek adottságait és alkalmazási lehetőségeit pontosan körvonalazó információs háttérrel lássuk el.

A nemzetközi gyakorlatnak megfelelően ezt a célt legjobban egy olyan bevezető szakkönyv közreadásával érhetjük el, mely tömör és közérthető ismertetést ad a számítástechnika és a kommunikáció, valamint a hálózatok leggyakrabban használt fogalmairól, működési elveiről és eszközeiről, nem kevésbé azok alkalmazásairól is.

Úgy gondoljuk, hogy e bevezető szakkönyv használhatóságát jelentősen növelheti az a körülmény, hogy e könyvben az általános ismertetéseken túlmenően különös hangsúllyal térünk ki a magyarországi referenciaképes alkalmazások ismertetésére, valamint azokra a speciális infrastrukturális adottságokra, melyek a nemzetközileg szokásostól eltérő megoldásokat kívánnak. Ezért is ajánljuk ezt a könyvet mindazoknak a nem számítástechnikai vagy kommunikációs szakterületen működő vezetőknek, fejlesztési és üzemeltetési témában dolgozó szakértőknek, akik munkáik során számítógépes és kommunikációs hálózatok létrehozásával vagy működtetésével kerülnek kapcsolatba.

Budapest, 1993. április 7.

**Németh Pál**



# I. Információtárolás a számítógépben

## Miért digitális?

A számítógépek digitális eszközök. Mi az, hogy digitális?

A fogalom megértéséhez gondoljuk végig a következő elméleti feladatot. Adatokat kell továbbítanunk két város között, gyorsan, megbízhatóan és lehetőleg olcsón. A legnagyobb sebességet elektromos jelek továbbításával érhetjük el. Ha például el akarjuk küldeni mondjuk a 20-as számot, választhatunk analóg eljárást: 20 voltot kapcsolunk a vezetékre. Ezt a másik oldalon egy automata műszer megméri és rögzíti. Ha 20 000-t kell továbbítanunk, ekkora feszültséget a vezetékre adni már kissé kockázatos, célszerűnek látszik leosztani mondjuk 100-zal. Most azonban a kisebb számokkal leszünk bajban, a 48 például 0,48 voltként jelenik meg, amit pontosan mérni már nem könnyű, a berendezés igen drága lesz.

A fenti módszer az analóg információtovábbítás. Az eljárás korlátai nyilvánvalóak: a számok tartományát erősen korlátoznunk kell, különben az átvitel pontossága lesz gyenge. Az igények növekedésével a költségek meredeken emelkednek. Javíthatjuk a módszert, ha például több vezetékot használunk. Olcsó és megbízható megoldást azonban úgy kapunk, ha más oldalról közelítjük meg a problémát.

A vezetéken csak két, könnyen mérhető feszültséget engedünk meg, a 0 és az 5 voltot, a legtöbb számítógépben ezeket a szinteket használják. Így csak két-féle számot tudunk továbbítani, ezért a küldendő adatokat átírjuk kettes alapú (bináris) számrendszerbe, mivel ebben csak két számjegy van, a 0 és az 1, a 0 és az 5 voltot ezeknek feleltetjük meg. A bináris számot számjegyenként küldjük el, ha a jegy 1-es, akkor 5 voltot, ha 0, akkor 0 voltot – azaz a földet – kapcsolunk a vezetékre.

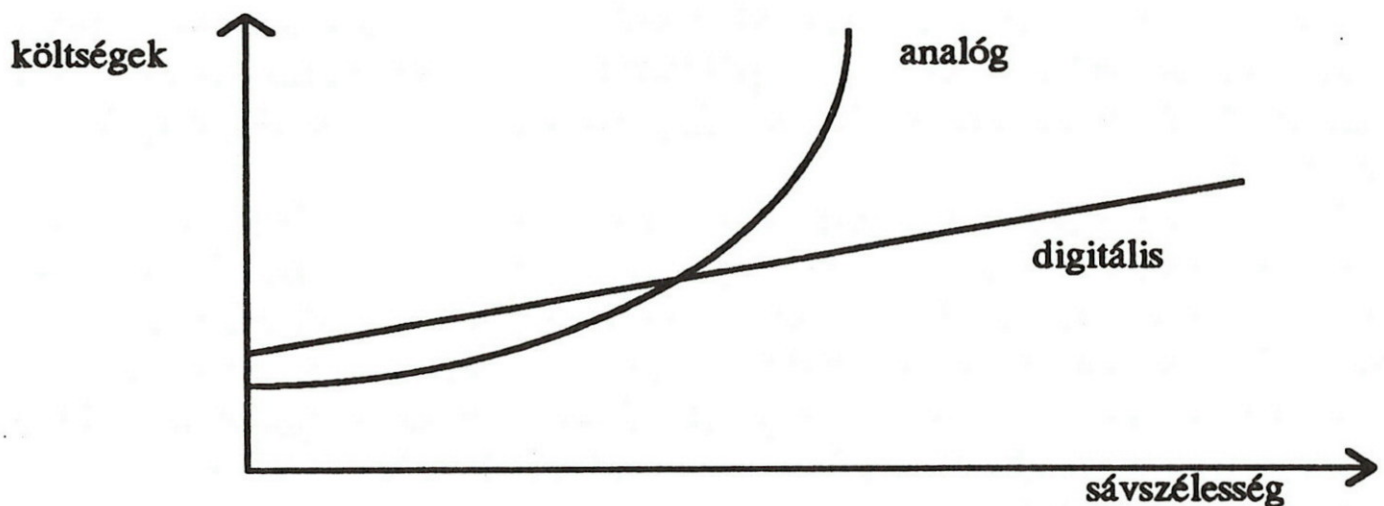
Ez az eljárás a digitális információtovábbítás. Egyetlen hátránya, hogy a binárisan felírt számok hosszabbak, több számjegyűek, továbbításuk több lépésben történik, hosszabb időt vesz igénybe. Előnyei azonban ezt bőven ellensúlyozzák. A számok nagyságát nem kell korlátoznunk, és a pontosság is tetszőleges, nincs szükség tized vagy század voltok mérésére.



<u>Decimális</u>	<u>Bináris</u>	<u>Decimális</u>	<u>Bináris</u>
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001
5	0101	10	1010
20 =		10100	
20 000 =		100111000100000	

1. táblázat Decimális és bináris számok

Az így ábrázolt adatokat nemcsak továbbítani, de feldolgozni – például két számot összeadni – is egyszerűbb. A módszert előnyei miatt nemcsak a számítógépekben, hanem a karórától kezdve az orvosi műszereken át a digitális hangrögzítésig – gondoljunk a közkedvelt CD-re – nagyon sok helyen alkalmazzák.



1. ábra Az analóg és a digitális rendszer költségei

- Az 1. ábrán egy diagram szemlélteti az analóg és a digitális eljárás költségviszonyait. Bizonyos minőségi követelmény felett a digitális módszer lényegesen olcsóbb, noha induló költségei jól láthatóan magasabbak. Miből adódik ez? A környezetből érkező analóg információkat – ez lehet egy gépkocsi sebessége vagy az olvasztókemence hőmérséklete – először megfelelő elektronikus áramkörökkel digitális formájúvá kell alakítani a továbbításhoz és feldolgozáshoz. Ez minőségi követelményektől függetlenül többletköltséget jelent.



## Adatformátumok

A számítógépben használt bináris számok egy számjegyét nevezik bitnek. Értéke 0 vagy 1 lehet, ami megfeleltethető egy kérdésre adott nem vagy igen válasznak, ezért szokták a legkisebb információegységnek is nevezni. Az előző példában szereplő vezetéken a feszültség kapcsolgatásával biteket továbbítunk egymás után. Mivel a számok nagysága és így a biteik száma is változó, honnan tudja a másik oldal, hogy az előző számot befejeztük, a következő bit pedig már egy új számhoz tartozik?

A számok között tarthatnánk hosszabb szünetet is, ez azonban feleslegesen lassítja az átvitelt. Célszerűbb a biteket csoportokra osztani: Egy ilyen 8 bitből álló csoport a bájt (byte). Minden számot egy, vagy ha 8 biten nem fér el, több bájton ábrázolunk, és előre megállapodunk abban, hogy hány bájtos számokat továbbítunk. Így nem kell szünetet tartanunk, csak a biteket számolni.

A bájt nemcsak az információtovábbításban, hanem a feldolgozásban is alapegység. A korábbi számítógépek ekkora adatot tudtak egy lépésben kezelni. A gépek teljesítményének jellemzéséhez ma is megadják, hogy egyszerre mekkora adatmennyiséget dolgoznak fel. Ezért beszélhetünk 8, 16, 32, sőt 64 bites számítógépekről.

Egyetlen bájton 0 (ha valamennyi bite 0) és 255 (ha valamennyi bite 1) közé eső pozitív egész számot lehet tárolni. Nagyobb vagy több jegyből álló törtszám ábrázolásához több, de mindig páros számú bájtot használnak.

Eddig információn csak számokat értettünk, természetesen azonban szükség van szövegek továbbítására és feldolgozására is. Ezért minden betűhöz egy 1 bájtos számot rendelünk, a kódját. A szöveg így számok sorozatává alakítható és a már ismert módon kezelhető. A kódot a nyomtatóra vagy a képernyőre küldve az kinyomtatja, illetve megjeleníti a neki megfelelő betűt.

A betűkhöz még az 1 bájtos méreten belül is sokféleképpen rendelhető szám, ezért fontos a kódolás szabványosítása, hogy egy kódon minden eszköz ugyanazt a betűt értse. A legelterjedtebb kódolást az ASCII (American Standard Code for Information Interchange, azaz Amerikai Kódszabvány Információcsere-rehez) szabvány írja le. Ennek karakterkészletébe a betűkön kívül beletartoznak az írásjelek, a számok, a műveleti jelek és néhány speciális jel, a százalék vagy a dollár jele. Eredetileg az ASCII csak az angol ábécé betűit tartalmazta, és ehhez elegendő volt 7 bit, a legnagyobb helyiértékű mindig 0 volt. Ennek a bitnek a felhasználásával később kibővítették, felvették bele a legnagyobb nemzeti nyelvek különleges betűit és néhány olyan jelet, amelynek segítségével egyszerű ábrák kirajzolhatók a képernyőn és a nyomtatón. Az eredeti, 7 bites kódkészlet a 2. táblázatban látható.



Kód	Karakter	Kód	Karakter	Kód	Karakter	Kód	Karakter
0	null	33	!	66	B	99	c
1	soh	34	"	67	C	100	d
2	stx	35	#	68	D	101	e
3	etx	36	\$	69	E	102	f
4	eot	37	%	70	F	103	g
5	enq	38	&	71	G	104	h
6	ack	39	'	72	H	105	i
7	bel	40	(	73	I	106	j
8	bs	41	)	74	J	107	k
9	ht	42	*	75	K	108	l
10	lf	43	+	76	L	109	m
11	vt	44	,	77	M	110	n
12	ff	45	-	78	N	111	o
13	cr	46	.	79	O	112	p
14	so	47	/	80	P	113	q
15	si	48	0	81	Q	114	r
16	dle	49	1	82	R	115	s
17	dc1	50	2	83	S	116	t
18	dc2	51	3	84	T	117	u
19	dc3	52	4	85	U	118	v
20	dc4	53	5	86	V	119	w
21	nak	54	6	87	W	120	x
22	syn	55	7	88	X	121	y
23	etb	56	8	89	Y	122	z
24	can	57	9	90	Z	123	{
25	em	58	:	91	[	124	
26	sub	59	;	92	\	125	}
27	esc	60	<	93	]	126	~
28	fs	61	=	94	^	127	
29	gs	62	>	95	¯		
30	rs	63	?	96	·		
31	us	64	@	97	a		
32	space	65	A	98	b		

2. táblázat Az eredeti ASCII kódkészlet



Az első, 0 és 31 közé eső csoport kódjai nem megjelenítendő betűket, hanem parancsokat jelölnek. Ilyen például a 10-es kódú LF (line feed = soremelés) karakter, amelyet ha elküldünk egy nyomtatónak, az emel egy sort a papíron. Ezek a vezérlőkaraktereknek nevezett kódok is rendelkeznek megjelenési formával, egyszerűen a felismerésük megkönnyítésére.

A 40-es kódú szóköztől kezdődnek az angol ábécé karakterei, míg 128-tól következnek a nemzeti és grafikus karakterek.

Az angol ábécé betűit közvetlenül beírhatjuk a számítógép, például egy PC billentyűzetén. A vezérlőkarakterek legtöbbjéhez nem tartozik külön billentyű, ezeket beüthetjük a CTRL billentyű lenyomva tartása mellett a neki megfelelő betűvel – a képernyőn megjelenő „ ^ ” (caret) jel a CTRL billentyűt jelöli – vagy az ALT billentyű lenyomva tartásával a jobb oldali külön numerikus billentyűzeten beírva a kódját. Így például a parancslezáró CR (carriage return) vezérlőkódot kiadhatjuk a CTRL-M leütéssel, az ALT-13 kód beírásával, vagy az Enter billentyűt lenyomva, mivel ennek a kódnak van saját billentyűje. A hatása természetesen mindhárom esetben ugyanaz.

A kiterjesztett tartomány karaktereit szintén az ALT lenyomva tartása mellett a kódjuk beírásával jeleníthetjük meg.

A magyar nem tartozik a nagy nemzeti nyelvek közé, így hosszú ékezetes betűink nem kaptak helyet a bővített készletben. Ezek beillesztése csak valamely karakterek helyettesítésével lehetséges. Sajnos nincs szabvány arra nézve, hogy milyen kódot kapjanak ékezetes betűink, így többféle elrendezést is használnak. A 3. táblázat a három leggyakoribb kód kiosztást mutatja. Egy szöveg számítógépen való begépelésekor el kell döntenünk, milyen kód kiosztást használunk a magyar ékezetes betűkre. Szövegünkben később más gépen csak ugyanilyen kód használata mellett fogjuk helyesen látni valamennyi ékezetes betűt.

### **Adat és utasítás**

A számítógépnek az adatok mellett utasításokra is szüksége van, amelyek meghatározzák a működését, lépésről-lépésre előírják az adatokkal végzendő műveleteket. Az utasításokat szintén bájtokra tagolt bináris számokként tárolja. Mivel a formátum azonos, hogyan különböztethető meg egymástól az adat és az utasítás? Pusztán ránézéssel ez nem lehetséges. A gépet működtető operációs rendszer és a futó program feladata annak pontos nyilvántartása, hogy mely bájtok tartalmazzanak adatot és melyek utasításokat. Ha egy adatot, pl. az A betű ASCII kódját utasításként próbál meg értelmezni, ez bizonyosan hibás működést eredményez.



Betű	CWI	IBM 852	Ventura Publ.
É	144	144	144
é	130	130	130
Á	143	181	199
á	160	160	160
Ó	149	224	209
ó	162	162	162
Ö	153	153	153
ö	148	148	148
Ő	167	138	221
ő	147	139	219
Ü	154	154	154
ü	129	129	129
Ű	152	235	222
ű	150	251	220
Ú	151	233	214
ú	163	163	163
Í	140	214	205
í	161	161	161

**3. táblázat** A magyar ékezetes betűk kódjai a három legelterjedtebb kiosztás alapján

Az utasításokat szokás gépi kódnak is nevezni. Ennek oka, hogy az utasítás formája az adott számítógéptípusra jellemző, más gépen ugyanezt az utasítást más bináris szám írja le. Míg egy ASCII kóddal megadott szöveg minden olyan számítógépen használható, amely alkalmazza ezt a kódolást, az utasítások nem vihetők át egy másik géptípusra. Így egy utasítások sorozatából álló program csak egyfajta gépen használható, más géptípusra újra el kell készíteni.

### Mi a fájl?

Az adatokat és utasításokat számítógépünk tehát bájtonként tárolja és továbbítja. A bájtot azonban meglehetősen kis egység, szükség van egy jobban kezelhető, nagyobb formátumra. Ez a formátum a fájl (file) vagy állomány. A fájl egysze-



Programutasításokat  
tartalmazó fájl bájtjai

	Jelentése
10111100	REG A
11100110	+
00110111	REG B
01100001	+
11100111	REG C

## Szöveges fájl bájtjai

	Jelentése
01010100	T
01001001	I
01010011	S
01011010	Z
01000001	A

2. ábra Utasítás és ASCII bájt sorozat

rűen bájtok sorozata. A fájlban lévő bájtok száma csak tőlünk függ. Állhat akár egyetlen bájtból, maximumát pedig a gép kapacitása korlátozza. Természetesen a fájl alkotó bájtok valamilyen szempontból összetartoznak. Így egy fájlban helyezhetjük el például a számítógépen írt levelet. Ekkor a levél betűinek – beleértve a szóközöket is – ASCII kódjait tartalmazza a fájl a megfelelő sorrendben. A levélhez készült mellékletet is tehetjük ebbe a fájlba, vagy nyithatunk számára külön egy másikat.

A csak szabványos karaktereket és a sorlezárás (CR = carriage return), soremelés (LF = line feed), lapdobás (FF = form feed) és vízszintes tabulálás (HT = horizontal tab) vezérlőkaraktert tartalmazó fájl nevezik szöveges vagy text fájlnek. Egyszerűségének előnye, hogy szinte minden géptípuson és szövegszerkesztő programmal feldolgozható. Amennyiben a fenti levélben mondjuk a cím alá szeretnénk húzni, akkor a betűk ASCII kódjain kívül azt az utasítást is el kell a fájlban helyeznünk, hogy a nyomtató kapcsolja be az aláhúzást, majd a cím végén kapcsolja ki.

Ennek az utasításnak a formája azonban már függ a nyomtató típusától, vagy az alkalmazott szövegszerkesztő programtól, ezért a szebb nyomtatás ára, hogy a levelünk már nem dolgozható fel bármelyik gép bármely programjával, hanem csak azzal, amellyikkel begépeltük. Az ily módon specializált fájlokat bináris adatfájlnek hívják. A fájlok tartalmazhatnak utasításokból álló programot is, ezek a program vagy végrehajtható fájlok.

Minden fájlra a nevével lehet hivatkozni, a nevet a fájl létrehozásakor kell kiválasztanunk. A fájl adatainak nyilvántartása az egyes géptípusokon némileg



különböző. A nevéen kívül általában feljegyzik a méretét (hány bájtól áll) és létrehozásának dátumát és idejét. A PC-ken legelterjedtebb DOS operációs rendszer például a névhez illesztett EXE vagy COM toldalékkal jelöli meg a programot tartalmazó fájlokat. Az ezekben a fájlokban tárolt programot elindíthatjuk egyszerűen a fájl nevének beírásával.

A DOS rendszer a fájlnevekben és a parancsokban a kis- és nagybetűket nem különbözteti meg egymástól, bármelyikkel gépelhetünk. A Unix operációs rendszer azonban különbséget tesz közöttük, a level.doc fájl nem azonos a LEVEL.DOC-kal.

Egy fájl tartalmát megnézni a legegyszerűbben úgy lehet, ha kiíratjuk a képernyőre. Minden operációs rendszerben van ilyen parancs. A szöveges fájl ASCII kódjait a képernyő betűkké, írásjelekké, számokká fordítja a megjelenítéskor. Ezzel a módszerrel programfájlok is kiírathatók ugyan, de mivel nem ASCII kódokat próbál meg értelmezni, az eredmény olvashatatlan, sőt mivel valószínűleg vezérlőkaraktereket is talál, meglepő formájú lesz.

### A számítógép modellje

Számítógépünk tehát működése során bináris formában tárolt utasításokat hajt végre, és szintén binárisan tárolt adatokat dolgoz fel. Azt az egységet, amely az utasításokat végrehajtja, processzornak vagy CPU-nak (Central Processing Unit) nevezik. A számítógép teljesítménye elsősorban a processzor teljesítményétől függ. Ennek jellemzésére két alapvető adatot adnak meg, az órajelfrekvenciát és a bitszélességet.

Az órajelfrekvenciát úgy képzelhetjük el, mint egy metronóm ketyegését, ennek ütemében dolgozik a CPU. A szaporább ütem nagyobb frekvenciát, gyorsabb működést jelent. A frekvenciát megahertz-ben (MHz) adják meg, ez egy-millió ütemmel egyenlő másodpercenként. A PC-kben használt processzorok szabályos frekvenciái: 8, 10, 12, 16, 20, 25, 33, 40, 50, sőt 66 MHz is lehet.

Bitszélességen annak az adatnak a hosszát értjük, amelyet a CPU egy lépésben (egy utasítással) kezelni tud. Eszerint beszélnek 8,16,32 és 64 bites processzorokról. A különböző típusokat egy típuszámmal jelölik meg, esetleg megadják a gyártó nevét is, pl. AMD 386.

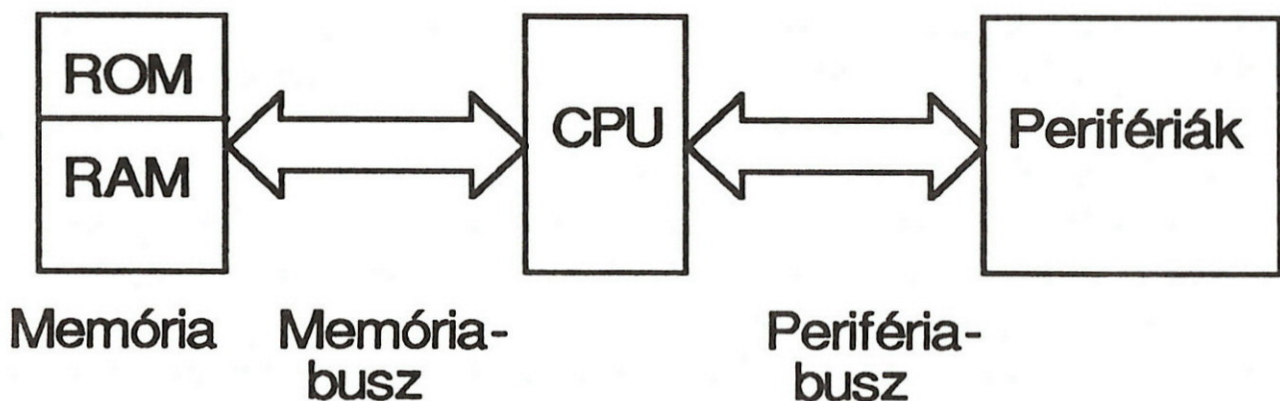
A típus egyértelműen azonosítja a bitszélességet is, különböző bitszélességű processzorok nyilván más felépítésűek, így ezt a hirdetésekben nem jelzik. Ugyanazt a típust azonban készíthetik különböző sebességgel, ezért az órajelfrekvenciát meg szokták adni.



Az utasítások és adatok tárolása a számítógépben a memória feladata (1. a 3. ábrát). A memória is bájtanként tárolja az információt, nagyságát – kapacitását – ezért bájtban mérik. Ennek nagyobb egysége a kilobájt (KB) és a megabájt (MB). Mivel a gép a kettes (bináris) számrendszert használja, ezért a kilo nem 1000 bájtot jelent – mint a kilométer = 1000 méter –, hanem a  $2^{10}$  hatványát, ami 1024. Ugyanígy a megabájt a  $2^{20}$  hatvány, azaz 1 048 576.

A memóriát angol rövidítéssel RAM-nak (Random Access Memory) nevezik, ami tetszőleges elérésű memóriát jelent. Az elnevezés arra utal, hogy a processzor a memória bármelyik bájtjának a tartalmát lekérheti (kiolvashatja) és bármelyik bájtjába küldhet tárolandó információt (írhatja). A RAM memória egyetlen komoly hátránya, hogy a számítógép kikapcsolásakor a tartalma elvész, törlődik. Ezért minden programot és adatot valamilyen mágneses elven működő tárolóeszközön kell rögzíteni.

A gép bekapcsolása után a RAM memória még üres, a CPU pedig csak a memóriából kezdhet el dolgozni. Ezért szükség van olyan memóriára is, amely megőrzi a tartalmát a gép kikapcsolt állapotában. Ez a ROM vagy EPROM memória, amelyet azonban csak olvasni lehet, erre utal a ROM (Read Only Memory) betűszó. Tartalmát a gyártáskor rögzítik, később nem, vagy csak a számítógépből kivéve, speciális eszközzel lehet megváltoztatni.



**3. ábra** A neumann-i számítógépmodell

A memóriát és a processzort egy vezetékkegység kapcsolja össze, amelyet sínnek vagy az angol neve után busznak (bus) hívnak. Ez továbbítja a két egység között az utasításokat és adatokat. Minden bit számára külön adatvezetéket tartalmaz, így ezek száma megegyezik a processzor bitszélességével.

A memórián és a CPU-n kívül a számítógépben használt minden más eszköz a perifériák közé tartozik. A perifériákat aszerint csoportosíthatjuk, hogy küldenek vagy kapnak adatot a számítógéptől. Az adatot beküldő vagy bemeneti (input) perifériákhoz tartozik például a billentyűzet. Az adatot fogadó vagy kime-



neti (output) perifériákra a képernyő a legjobb példa. A kétirányú adatforgalmat lebonyolító perifériák közé tartoznak a mágneses elven működő tárolók, mint a floppy vagy a harddisk-egység. Ezek csatlakoztatására külön vezetékek, a perifériabusz (sínrendszer) szolgálnak. A perifériák a memóriába – néhány kivételtől eltekintve – csak a CPU-n keresztül, annak irányításával küldhetnek vagy kérhetnek adatot, ezt tükrözi az 5. ábra elrendezése.

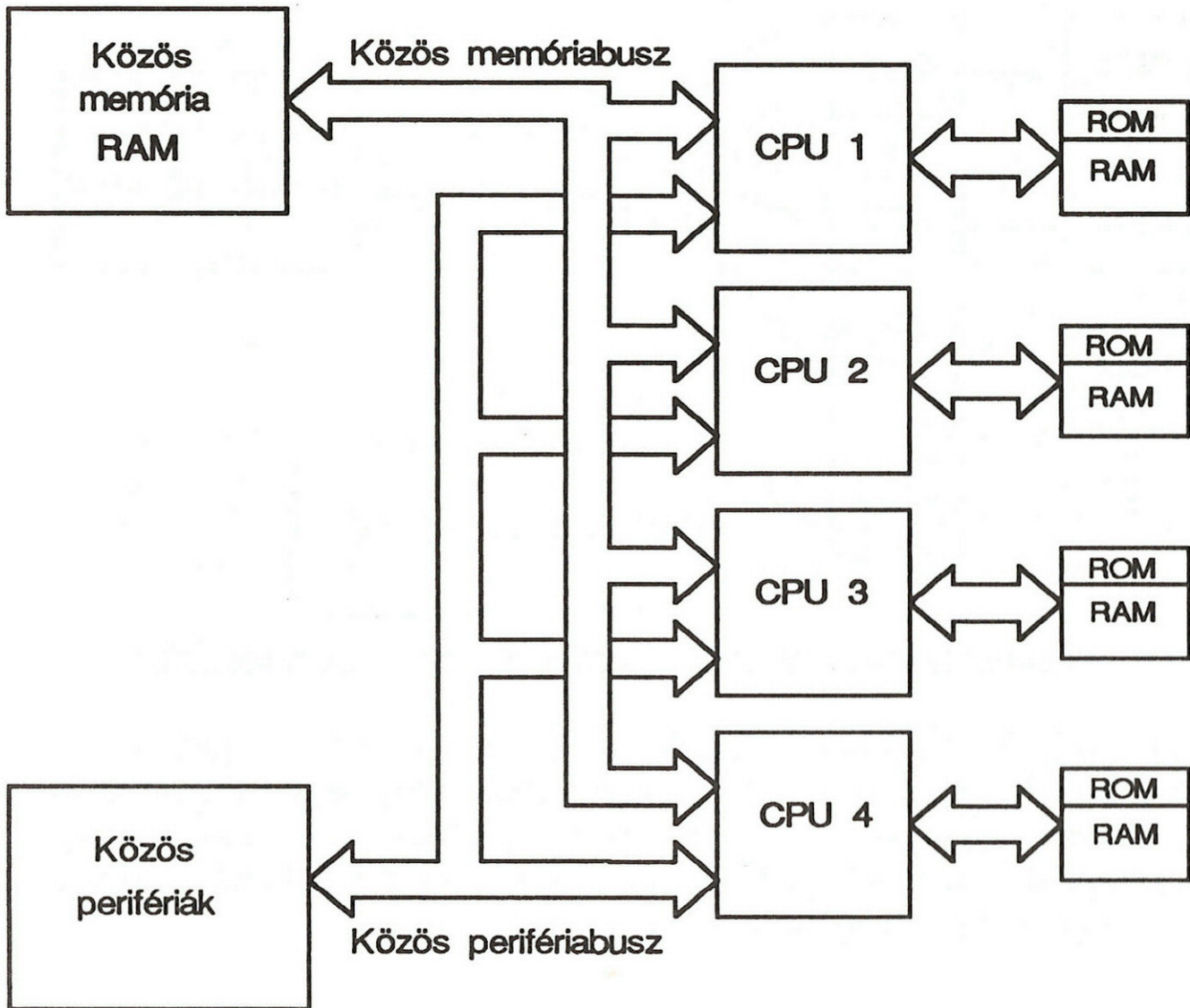
A számítógép működésének elméleti kidolgozásában nagy szerepe volt az 1940-es években a magyar származású Neumann Jánosnak, aki után az ábrán látható modellt klasszikus vagy neumann-i modellnek nevezik.

A számítógép teljesítménynövelésének egyik logikus módja, hogy több processzort alkalmazunk, amelyek megosztóznak a feladatokon, egymástól függetlenül, párhuzamosan dolgoznak. Ezeket a gépeket nevezik többprocesszoros vagy multiprocesszoros rendszereknek. Fontos kitétel, hogy a processzorok egyenrangúan vesznek részt a program végrehajtásában és nem csak valamilyen részfeladatot, pl. egy periféria kezelését látják el, ami a teljes rendszer teljesítményét nem növeli lényegesen.

A legfontosabb kérdés itt, hogy az egymástól függetlenül dolgozó CPU-k hogyan kommunikálnak egymással. Ennek kétféle megoldása szerint kétféle felépítést alkalmaznak a gyakorlatban. Az osztott memóriájú modellben (l. 4. ábra) a processzorok rendelkezhetnek saját memóriával, de egymással a közös memórián keresztül érintkeznek. Ez mint egy üzenethagyó postaláda működik, minden CPU-nak jut benne egy rész. Az egyes CPU-k a közös buszon át férhetnek a közös memóriához, ezt egyszerre csak egy processzor használhatja, a többieknek meg kell várniuk, míg befejezi a műveletet. A processzorok számának növelésével az egy CPU-ra eső várakozás egyre hosszabbra nyúlik, ezért újabbak beépítése már csak kisebb mértékben növeli a teljesítményt. A rendszer előnye viszonylag egyszerű programozhatósága.

A közös memória hátrányát küszöböli ki a közvetlen kapcsolatú modell, amelyben a processzorok közvetlenül kapcsolódnak egymáshoz, hálószerűen (l. az 5. ábrát). Kifejezetten erre az elrendezésre tervezte az INMOS cég transzputernek nevezett processzorait, amelyek a szokásos processzorfunkciókon kívül négy, gyors adatátvitelt lehetővé tevő vezetékkel tartalmazzák, ezeken át kapcsolódnak egymáshoz. Így gyors rendszerek alakíthatók ki, hátrányuk azonban a bonyolult programozhatóság.



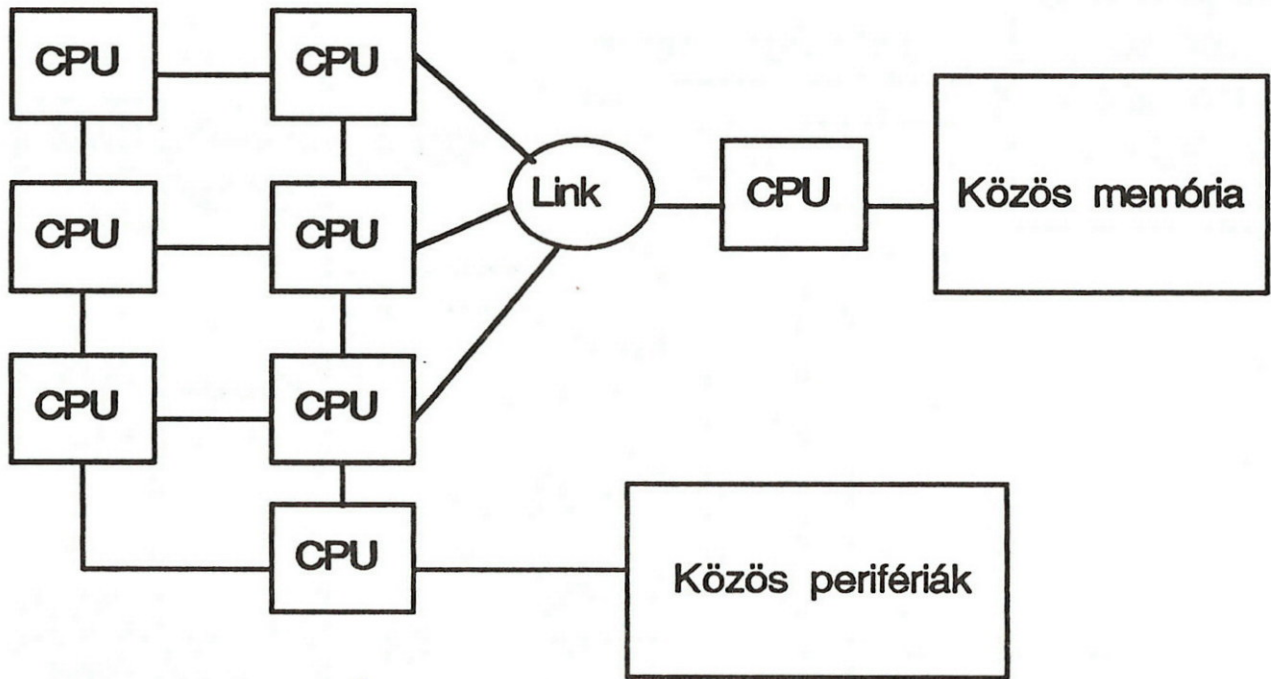


4. ábra Osztott memóriájú többprocesszoros rendszer modellje

#### Adattovábbítás a számítógépen belül

Mint a 3. ábrán láttuk, a processzor nem önmagában dolgozik, folyamatos kapcsolatot tart a memóriával, innen kapja – szaknyelven: hívja le – az utasításokat és az adatokat, ide írja ki a feldolgozás eredményét. A processzort és a memóriát összekötő buszt memória- vagy rendszerbusznak nevezik. Sebessége nagyban befolyásolja a rendszer teljesítményét.

A perifériabusz a külső eszközökkel való kommunikációt bonyolítja, sebessége nem ennyire döntő, bár bizonyos határokon túl itt is igaz, hogy a leggyengébb láncszem határozza meg az egész terhelhetőségét. A buszok – bármilyen busz – működése kétféle elven történhet, lehet szinkron vagy aszinkron. Szinkron esetben minden műveletet a processzor órajele ütemez.



5. ábra Közvetlen kapcsolatú processzorhálózat modellje

A PC-kben használatos 286-os, 386-os processzorok esetén például, mint a 6. ábra mutatja, az első ütemben a CPU kiválasztja, hogy a memória melyik bájtyát – vagy bájtyait, hiszen a CPU bitszélessége 286-os esetén 16, 386-osnál 32 bit – akarja olvasni, majd a második ütemben a memória elküldi az adatot. Egy ütem két processzor órajel hosszúságú.

1.ütem



2.ütem



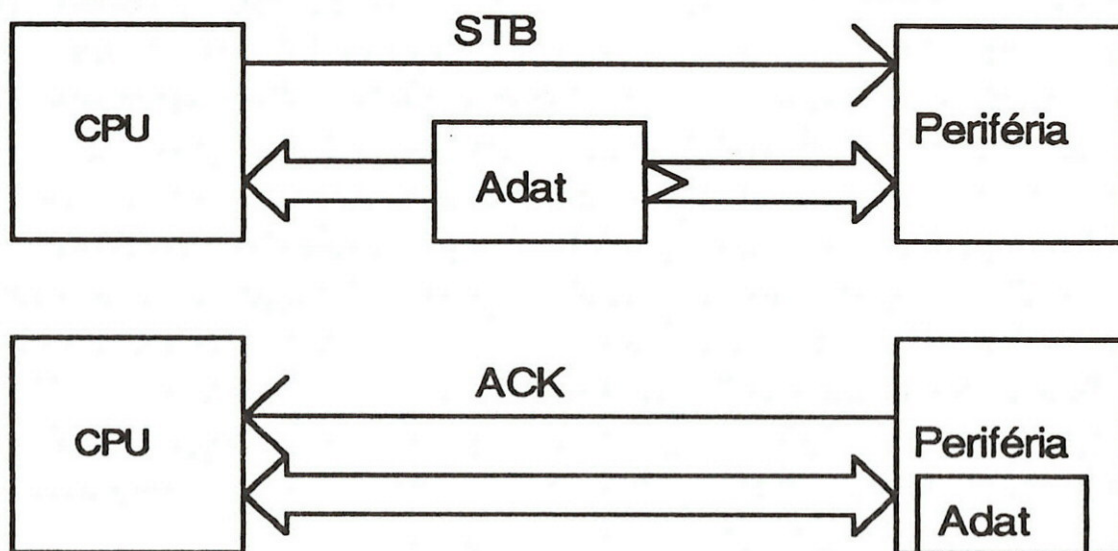
6. ábra Szinkron átvitel a processzor és a memória között



Szinkron esetben tehát az órajel üteme szabja meg az átvitel sebességét, ez azonban nem növelhető tetszőlegesen, mivel a memóriának időre van szüksége a kért bájttal előkereséséhez. Ha a memória a második ütemben még nem tudja küldeni az adatot, a CPU várakozó ütemet (WS = wait state) iktat be, ami tulajdonképpen holtidő. Hiába növeljük tehát az órajel frekvenciáját, az adott gépben belül csak a várakozó ütemek száma nő.

A memória az olvasáskérést általában gyorsabban tudja végrehajtani, mint az írást. Így a várakozó ütem arra használható, hogy az olvasáskor elérhető maximális sebesség mellett az íráshoz biztosítja a megfelelő időt.

Aszinkron esetben nem használnak órajelet, ezért segédjelekre van szükség. A 7. ábrán a processzor adatot küld egy perifériának és ezt egy külön vonalon – amit adatérvényesség jelzőnek vagy STB-nek (STB = strobe) nevezünk – tudatja vele. A periféria, ha sikeresen vette az adatot, a nyugtázó vonalon (ACK = acknowledge) szól vissza. A processzor ezután küldheti a következő adatot. Ha az adat iránya fordított, a CPU a periféria kiválasztásával szólít fel az adat beküldésére, amit ha a periféria megtett, az adatkész vagy READY vonalon jelzi.



7. ábra Aszinkron átvitel

Ezeket a segédjeleket kézfogás-jeleknek (handshake) is hívják, mivel adó-fogadó elrendezés szerint támogatják a biztonságos átvitelt. Az egyirányú adatforgalomhoz két – egy érvényesítő és egy nyugtázó – jelre van szükség, ha az átvitel a másik irányban is folyik, még egy jelpárt használnak.

Látszólag az aszinkron átvitel lehet a gyorsabb, hiszen nem kötődik a fix frekvenciájú órajelhez. A valóságban azonban a lassú eszközökkel kommunikál a CPU aszinkron módon. A perifériának bizonyos határok között van ideje reagálni. A memóriabuszt szinkron módon, a perifériabuszt pedig típustól függően szinkron vagy aszinkron módon használják a rendszerben.



Az átvitel gyorsaságát az egy másodperc alatt továbbított bájtok számával mérhetjük. Így például egy 33 Mhz-cel dolgozó 386-os CPU maximálisan 63 Mbájt/sec sebességgel használja a memóriabuszt. Összehasonlításképpen, a RISC munkaállomások 300-400 Mbájt/sec-ot is elérnek.

A perifériabusz órajele általában jóval alacsonyabb frekvenciájú, mint a processzor órajele, így sebessége mind szinkron, mind aszinkron módban lényegesen kisebb.

Az IBM PC típusú számítógépek elterjedésének egyik fontos oka volt, hogy tervezői a perifériabuszt nyitottan valósították meg. Aki már belenézett egy PC-be, láthatta a perifériabusznak az alaplapon sorakozó csatlakozóit. Ezekbe sokféle, különböző feladatot ellátó adapterkártya csatlakoztatható, egy mozdu-lattal a helyére illesztve.

A PC 1981-es megszületésekor a versenytársak gépei általában csak nehézkesen, és csak az eredeti gyártó – legtöbbször borsos árú – adaptereivel voltak bővíthetők.

A PC, majd az XT típusok perifériabusza a bennük alkalmazott 8088-as processzornak megfelelően 8 bit szélességű, szinkron busz volt. Az 1984-ben megjelent AT típus új processzort, a 286-ost tartalmazta, amely 16 bites, ezért a perifériabuszt is 16 bitesre bővítették. Ez a busz később az ISA (Industry Standard Architecture) nevet kapta és kvázi szabvánnyá vált. A 386-os processzor alkalmazására a PC eredeti gyártója, az IBM, 1987-ben új gépet tervezett PS/2 néven. Ebbe a 32 bites processzornak megfelelően egy 32 bites perifériabuszt épített, amelybe a korábbi, az ISA-buszhoz készült adapterkártyák nem használhatók. Ráadásul ezt a buszt, amelyet Mikrocsatornának (MC = Micro Channel) nevezett el, jogilag védetté tette, más gyártó csak licenrdíj fizetése ellenében alkalmazhatta saját gépeiben. Mindezek ellenére, mivel a PS/2 ugyanazokat a programokat futtathatja, mint a PC – szaknyelven szólva: szoftver szempontból kompatibilis –, ezért egyfajta PC-változatnak tekinthető.

Azok a PC-gyártók, akik nem éretek egyet a mikrocsatornával, vagy nem voltak hajlandók jogdíjat fizetni, 1989-ben létrehoztak egy EISA (Extended Industry Standard Architecture) névre keresztelt buszt, amely szintén 32 bites, és bárki szabadon alkalmazhatja. Az EISA-csatlakozókba a korábbi ISA-hoz készült kártyák is behelyezhetők, vagyis kompatibilisek vele.

A magasabb költségek miatt sem az EISA, sem a mikrocsatorna nem szorította ki az ISA-buszt, sok 386-os processzort alkalmazó PC-ben található meg.

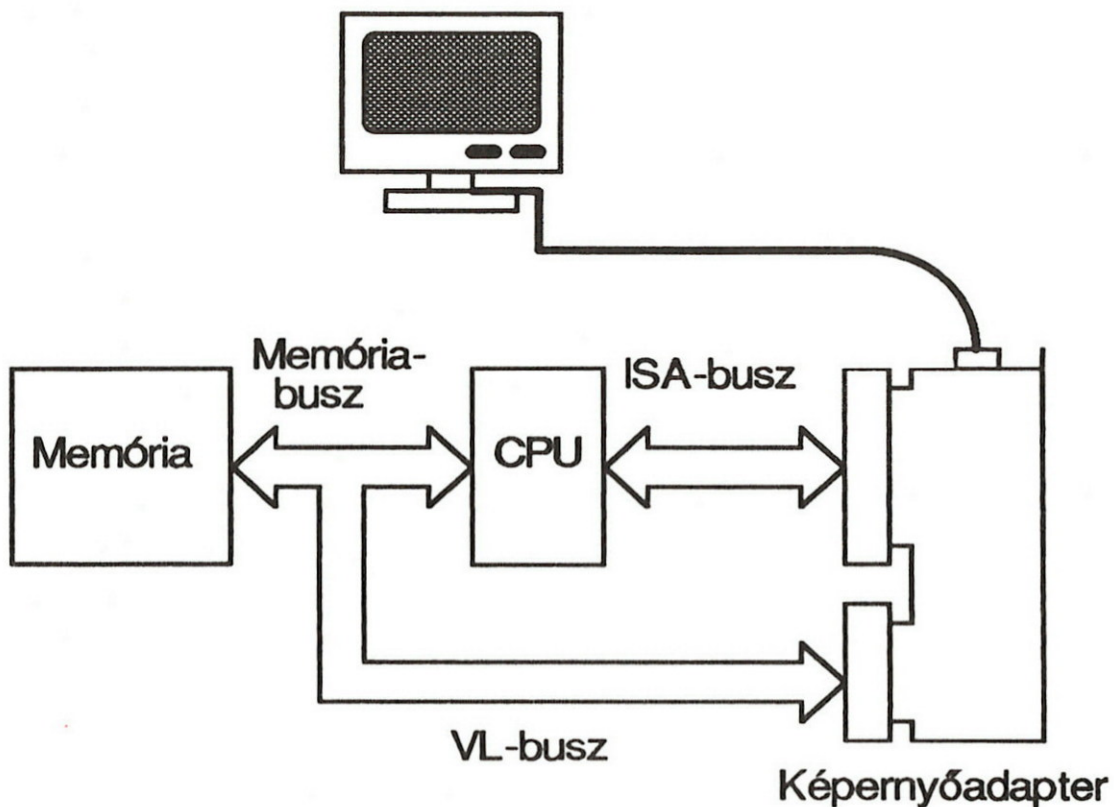
Az ISA órajele 8 MHz, ami szinkron módban sem tesz lehetővé nagy sebességű átvitelt, igaz erre sokáig nem is volt szükség. Az EISA- ill. Mikrocsatorna-buszok tervezésénél már gondoltak a gyorsabb perifériákra. A Mikrocsatorna aszinkron típusú busz, eredeti változata 20 Mbájt/sec maximális sebességet tett



lehetővé, ezt később 33 Mbájt/sec-ra növelték. Az EISA szintén 33 Mbájt/sec sebességgel működhet, de szinkron típusú.

A gyors képernyővezérlők és harddisk-illesztők megjelenésével felmerült egy gyors és lehetőleg olcsó perifériabusz igénye. A többféle elképzelés közül a több mint 120 gyártó által létrehozott VESA (Video Equipment Standards Association) nevű szervezet javasolta VL-busz került előtérbe. Ezt lokális busznak (local bus) is nevezik, 32 bit szélességű változata szinkron módban maximálisan 130 Mbájt/sec-os átvitelt tesz lehetővé. Könyvünk írásakor már kaphatók a VL-busszal épített PC-k, míg más elképzelések még csak tesztelési fázisban vannak.

A lokális buszok közvetlenül a processzorhoz csatlakoznak a memóriabuszon át. A VL-buszt használó adapter megtartja az ISA-buszra való kapcsolódást is, ahogy a 8. ábra mutatja.



8. ábra A VL-buszra csatlakozó képernyőadapter

### Az IBM PC-kompatibilis PROPER számítógépcsalád

Az első IBM-kompatibilis PC-t a PROPER család kifejlesztésével és gyártásba vitelével az SzKI hozta létre nem csupán magyarországi viszonylatban, hanem a világon is az elsők között. A PROPER sorozat révén a PC-technikában és technológiában jelentős hazai know-how bázis jött létre. Ez egyrészt a PC-k

komplex rendszerekben történő széles körű alkalmazási lehetőségeinek alapját teremtette meg, másrészt a korai PC-kultúra és -import befogadásának feltételrendszerét hozta létre.



## II. Adatátvitel számítógépek között

### Interfészek

A számítógépek közötti adatátvitel arra ad lehetőséget, hogy az egyszer már valamilyen módon, digitális formában gépre vitt információt egyszerűen és gyorsan megoszthatjuk másokkal.

Az előző fejezetben bemutatott számítógépmodellek zártak, abban az értelemben, hogy nem kapcsolhatók össze közvetlenül, például a perifériabuszon át. Ez modellszinten még elképzelhető lehetne, de a valódi, sokféle számítógéptípus között nem. Mivel a gépek közötti kapcsolat szükségessége nyilvánvaló, erre a célra különböző csatlakozási lehetőségeket hoztak létre, amiket átfogóan interfésznek (interface) neveznek. Ezek a processzor szempontjából egyfajta speciális perifériáknak tekinthetők.

Az interfész szót a számítástechnikában ennél szélesebb értelemben is használják, mint valamihez való hozzáférési lehetőséget.

A gépek közötti kapcsolatot létrehozó interfész tartalmazza a fizikai kapcsolatot megvalósító áramköröket (hardver interfész) és az ezek kezelését végző programot (szoftver interfész). A megbízható átvitelhez az adatokat mindkét oldalon, akár két teljesen eltérő felépítésű gépben is, azonosan kell kezelni. Ezért érdemes a különböző interfészeket szabványosítani. Ez jelentheti valamilyen szervezet tényleges, bejegyzett szabványát, vagy valamilyen, sok gyártó által elfogadott és betartott eljárást.

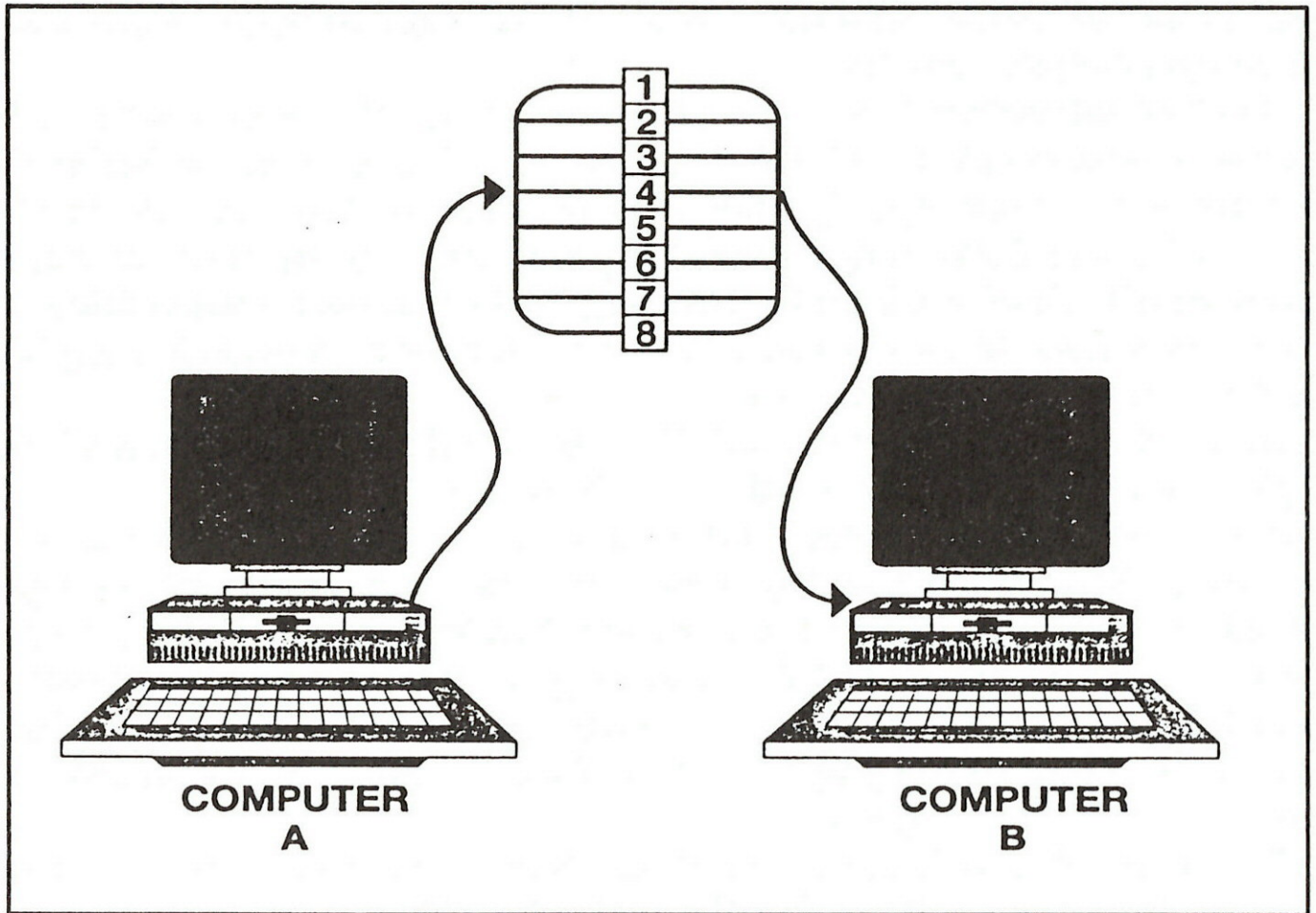
Az interfészek közül azokkal foglalkozunk, amelyek az IBM PC-ken is használhatók, mivel ezzel a típussal találkozunk a legtöbben.

### Párhuzamos átvitel és párhuzamos port

Mivel a számítógép bájtonként tárolja az információt, logikusnak tűnik a gépek között is bájtonként átvinni az adatokat. Ahhoz, hogy egy bájtot digitálisan, azaz bitenként egyszerre küldhessünk el, minden bitnek külön vezeték szükséges. A bitek egyidejű továbbítása miatt a módszert párhuzamos vagy parallel átvitelnek nevezik, a megvalósító áramköröket és magát a csatlakozót – tehát az interfész hardver részét – pedig párhuzamos portnak. Egy teljes bájtot továbbít egyszerre, így viszonylag gyors lehet, annak ellenére hogy aszinkron módon dolgozik, vagyis kézfogás-jeleket használ. Hátránya, hogy sok vezetéket igényel (8 adatbit + a kézfogás-jelek), ami nagyobb távolságon már költséges.



Bár vannak programok, mint pl. a LapLink, amely két PC között gyors adatátvitelt végez a párhuzamos porton egy megfelelő kábellel, a PC-ken a párhuzamos port tipikusan a nyomtató illesztésére szolgál. Mi is a nyomtató illesztés szemszögéből mutatjuk be a PC párhuzamos portját, de fontos tudni, hogy megfelelő szoftverrel sok más feladat is megoldható rajta keresztül.



9. ábra Párhuzamos adatátvitel sémája két számítógép között

Az IBM PC maximálisan három ilyen portot tartalmazhat, ezeket a DOS operációs rendszer LPT1, LPT2 és LPT3 néven azonosítja. Ha több ilyen csatlakozó van a gépünkben és nincs rajtuk jelölés, akkor csak kipróbálva dönthetjük el, melyik az LPT1, vagy az LPT2. A csatlakozó formáját mutatja a 9. ábra, 25 pólusú de nem használja valamennyit. A csatlakozón a vezetékek számozottak (1. 10. ábra), a következő ismertetésben mi is ezzel a számmal hivatkozunk rájuk. A port jelei három csoportra oszthatók:

Az 1. csoportot, a nyolc adatbitnek megfelelő nyolc adatvezeték alkotja, amelyek a 2-9 vezeték számot kapták.



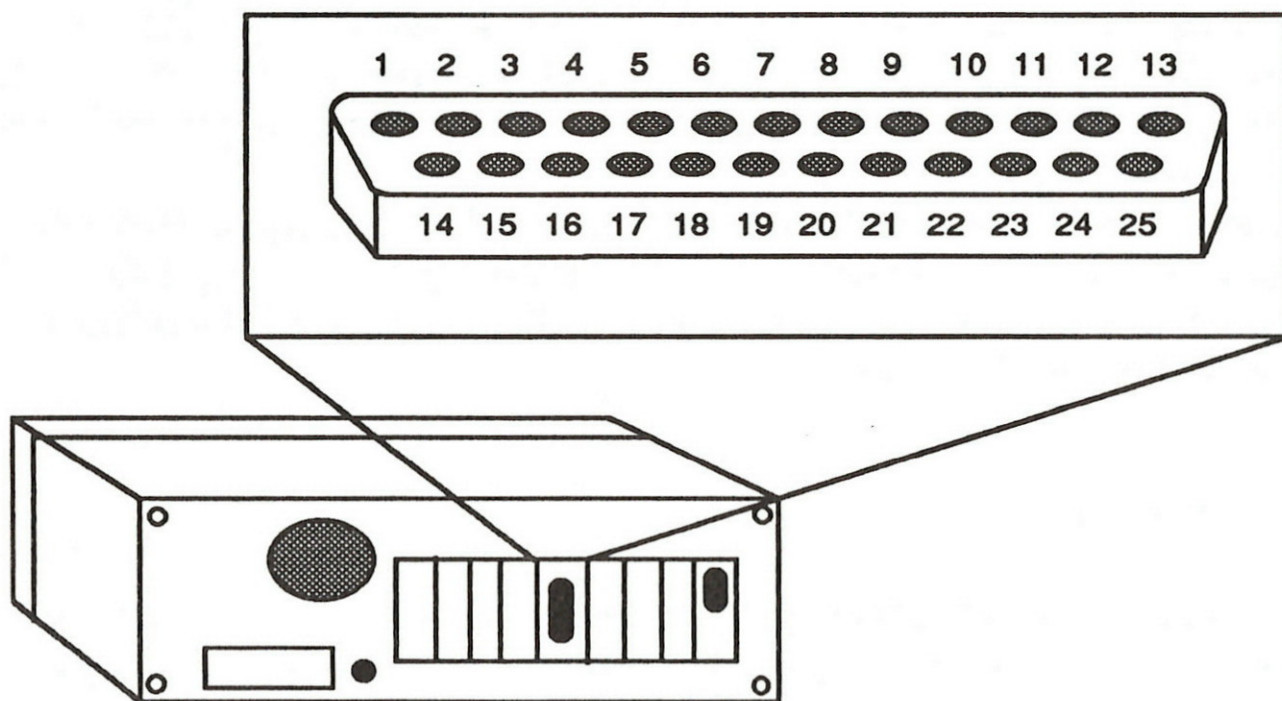
A 2. csoport, 4 vezérlővonal, amelyek a számítógéptől továbbítanak 4 bitet a printernek. Ezek adnak lehetőséget a printer vezérlésre. A jelek a következők:

1 STROBE (STB)

Ez a jel tudatja a printerrel, hogy az adatvezetékeken a számítógép adatot küld. Megfelel az előző fejezetben az adatérvényességet jelző kézfogás-jelnek.

14 AUTO FEED XT

Ezzel a jellel minden sor végén egy soremelésre készíthetjük a printert. Normál esetben egy szöveges fájlban a sorok végén egy CR (sorlezárás) és egy LF (soremelés) vezérlőkarakter van. Ha erre a vonalra jelet adunk, megspórolhatjuk az LF karaktert, a CR hatására a printer automatikusan sort is emel.



10. ábra A párhuzamos port csatlakozója a PC-n

16 INIT

Ennek a jelnek a hatása hasonló a számítógépen található Reset gomb hatásához. A printer minden műveletet félbehagy, és a bekapcsolása utáni helyzetbe áll.

17 SLCT IN

Ezzel a jellel tudatja a számítógép a printerrel, hogy az interfészen kiadott jel neki szól. Elvileg több printer kezelésére adna lehetőséget egy porton át, de nem használják.

A 3. csoport 5 állapotvonal, amelyeken át a printer jelezhet vissza a számítógépnek.



### 10 ACK

Ezen jelzi a printer, hogy megkapta az adatvonalakon küldött bájtot. Megfelel az előző fejezetben leírt nyugtázó kézfogás-jelnek.

### 11 BUSY

Foglalt jelzés, a printer ezzel tudatja, hogy most nem tud adatot fogadni.

### 12 PE

A printer akkor ad jelet ezen a vonalon, ha a papír kifogyott.

### 13 SLCT

A printer jelzi, hogy képes kapcsolatot tartani a számítógéppel.

### 15 ERROR

Hibajelzés a printertől, beszorult a papír vagy más hiba történt.

A párhuzamos portot, mivel a PC-ken szinte kizárólag a nyomtató illesztésére használják, gyakran nevezik printerportnak is. A jelek szintje a már korábban említett 0 és 5 volt. Ezekkel csak maximum néhány méteres távolság hidalható át megbízhatóan.

A nyomtató illesztésénél csak egyirányú átvitelre használjuk, de az adatvonalak kétirányú átvitelt is lehetővé tesznek. A szükséges kézfogás-jeleket pedig a 4 kimenő és 5 bejövő vonalon továbbíthatjuk, a bemutatott elrendezés csak a nyomtatók esetén érvényes.

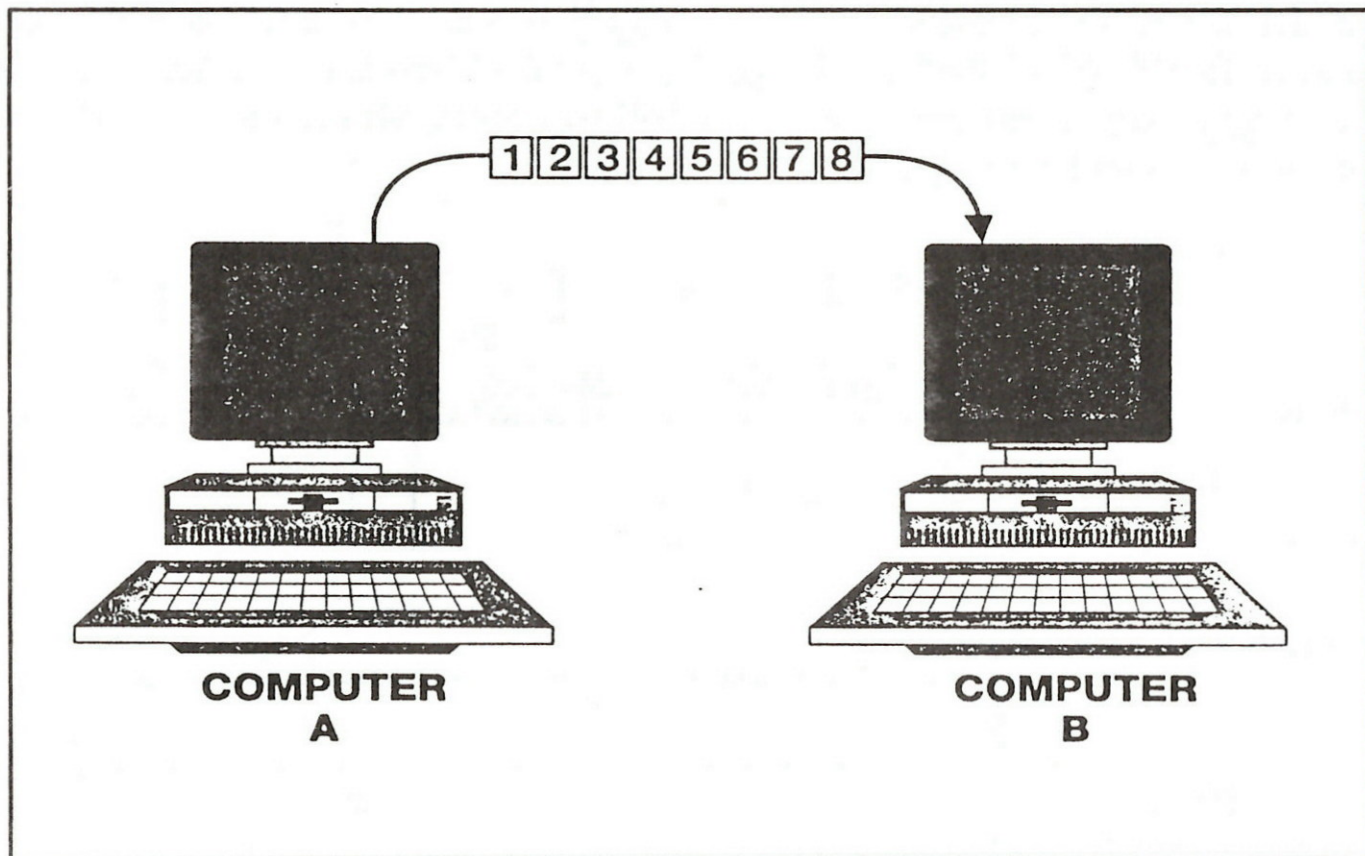
## Soros adatátvitel

Egy bájtvalemennyi bitjének egyidejű átvitele sok vezeték igényel, ezért nagyobb távolságokon a költségek miatt erről le kell mondanunk. Meg kell elégednünk egy bit egyidejű átvitelével, amihez csak két vezeték szükséges, a jelvezeték és a föld. Ez, mint később látni fogjuk, azzal az előnnyel jár, hogy a telefonhálózatot is felhasználhatjuk információ továbbításra.

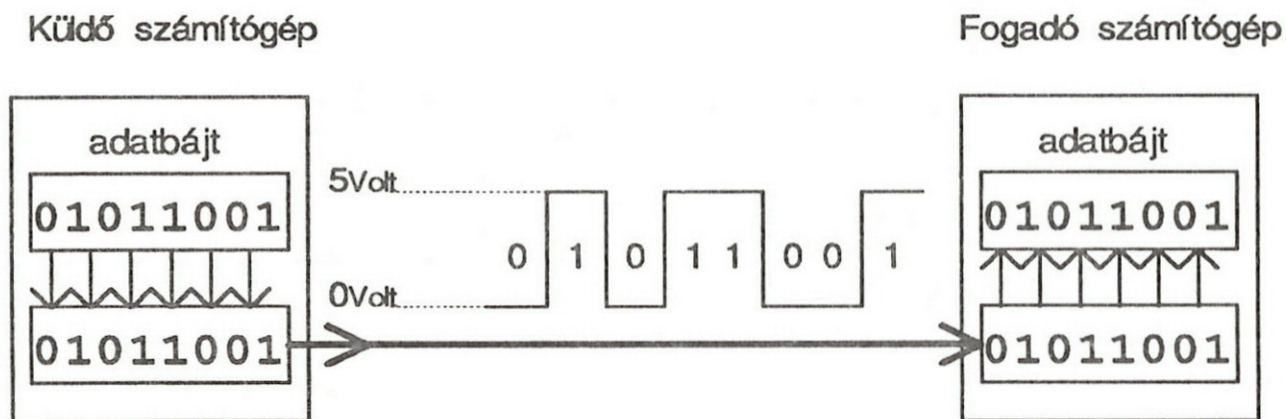
Két vezetéken való adattovábbításkor az egyes bájtok bitjeit sorban, egymás után küldjük el, ezért ezt a formát soros átvitelnek hívják (l. 11. ábra). Könyvünk elején, a digitális információküldés bemutatásakor is soros átvittel képzeltük el a bináris számok továbbítását. A soros adatküldés kétségtelenül lassúbb, mint a párhuzamos és több feladatot is ró az interfész áramkörökre, a bájtok bitjeit egyenként kell küldeni, a fogadó oldalon pedig a bitekből ismét összeállítani, ahogy a 12. ábra mutatja.

A biteket rárajzoltuk a vezetékekre, bejelölve a kétféle bithez tartozó feszültség-szinteket is. Ez meglehetősen egyszerűsített ábrázolás, hiszen nyilvánvaló hogy valamennyi bit nem utazhat egyszerre a vezetéken. Nézzük meg pontosabban hogyan is történik ez.





11. ábra A soros adatátvitel sémája

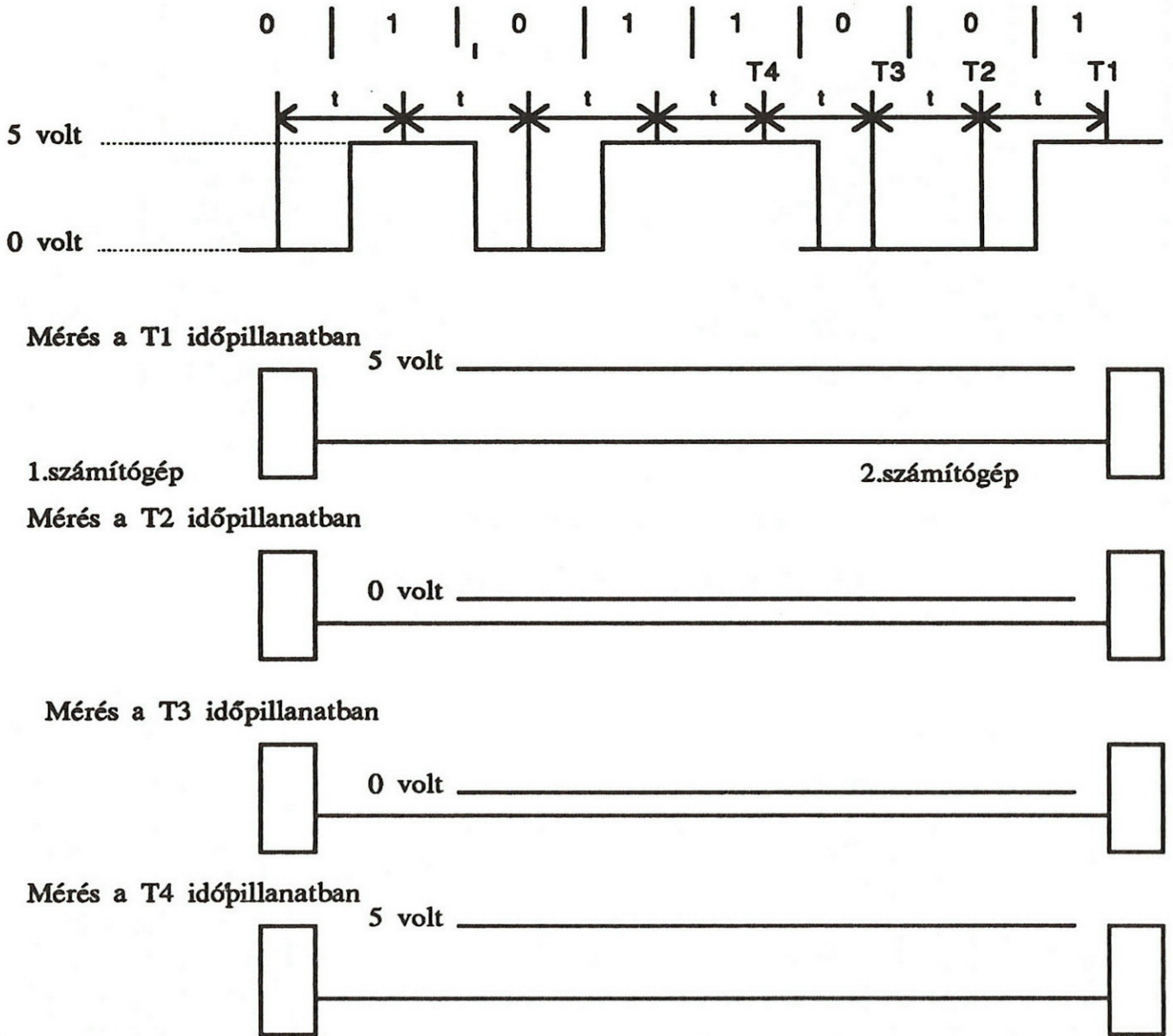


12. ábra A soros átvitel jelszintjei

Az egyszerűség kedvéért most tegyük fel, hogy a vezeték nem túl hosszú (max. néhány kilométer) és a biteket nem továbbítjuk nagyon gyorsan. Ez azt jelenti, hogy a vezetéken vagy 5 volt feszültség van – ha a küldött bit 1-es – vagy 0 volt, ha a küldött bit 0. Természetesen van olyan időpillanat is, miközben az egyik szintről a másikra vált, amikor a két érték között mérhetnénk valamit, de ez az idő a bitek időtartamához képest elhanyagolhatóan rövid.



A 13. ábrán T-vel jelöltük azokat az időpillanatotkat, amikor a fogadó oldalon mérjük a küldött biteket, alul pedig a mérések eredményét mutatja. Lát-szik, hogy pontosan abban az ütemben kell mérnünk, ahogy a biteket a küldő oldal a vezetékre kapcsolja.



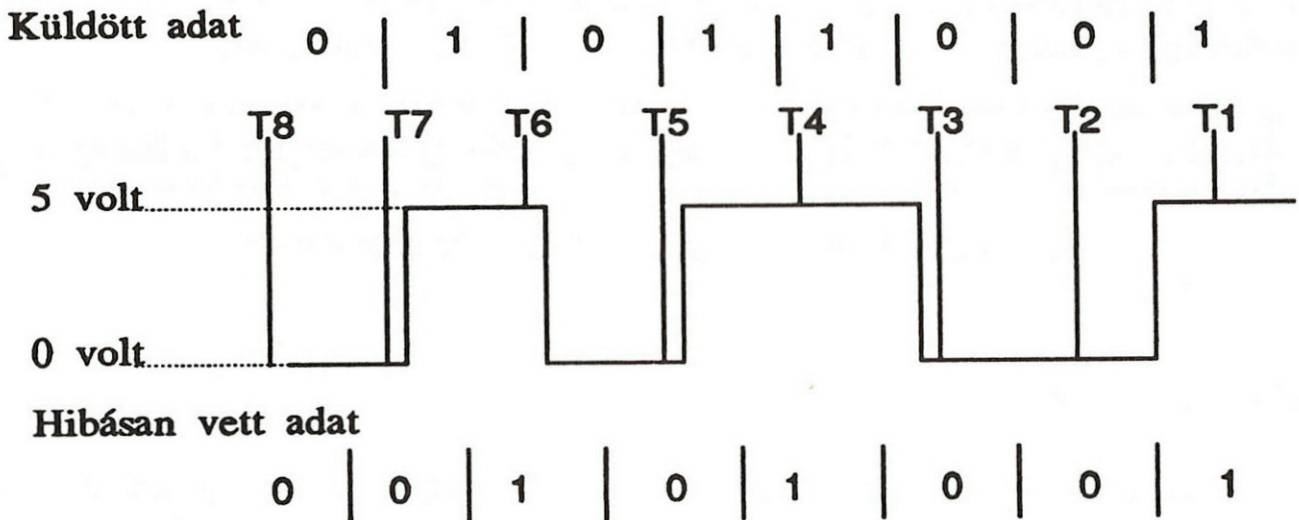
13. ábra A soros átvitel fizikai megvalósítása

Ha a mérések közötti idő csak egy kicsit is hosszabb vagy rövidebb ennél, előbb-utóbb lecsúszunk egy bitről – erre mutat példát a 14. ábra – vagy egy bitet kétszer mérünk meg. Mindkét esetben hibás lesz az átvitel.

Mivel egy sorozatban sokszor több ezer bitet továbbítunk, a két oldalnak nagyon pontosan kell együtt dolgoznia.

Gondolatmenetünk a soros adattovábbítás sarkalatos pontjára világított rá, ami a két oldal pontos szinkronizálása. A szinkronizálás megoldására kétféle





14. ábra Rossz időzítés miatt hibásan vett adat

módszert használnak, és ezek szerint különböztetnek meg szinkron és aszinkron soros átvitelt. Ezek a fogalmak a soros átvitelnél egészen mást jelentenek, mint amit a buszoknál láttunk.

### Szinkron átvitel

A megfelelő szinkronizálás mindig az adattovábbítás megkezdésekor okoz gondot. Ha hibás, az viszonylag korán kiderül – később látni fogjuk, hogyan – és újra lehet kezdeni az átvitelt, a vevő vagy az adó oldal kissé megváltoztatott paramétereivel. Annak esélye, hogy a szinkronitás hosszabb átvitel közben romlik el, ha kezdetben jó volt, elég kicsi. Ezért a legbiztosabb az lenne, ha a kapcsolat ideje alatt folyamatosan továbbítanánk adatokat, mert minden leállás és újraindítás hibalehetőséget jelent. Ez azonban nem mindig tartható, sokszor nincs folyamatosan küldendő adat, a szünet azonban nem olyan hosszú, hogy érdemes lenne megbontani a kapcsolatot. A szinkron és aszinkron átvitel alapvetően abban különbözik, hogy mit csinálunk akkor, ha éppen nincs küldendő adatunk.

Szinkron esetben ilyenkor is továbbítunk olyan bájtokat, amelyeknek nincs különösebb jelentésük, egyszerűen a szinkron fenntartására szolgálnak, ezért szinkronkarakter vagy szinkronbájt a nevük. Ezeket a küldő oldal szinkron soros portjának áramkörei automatikusan kiadják, ha nem kapnak továbbítandó adatot. A fogadó oldali szinkron port veszi őket, de a processzor felé nem továbbítja, egyszerűen lenyeli. Szinkronkarakternek nyilván olyan bájtot (vagy bájtokat) kell választani, amely az átvitt adatokban nem fordul elő, hiszen azokat a foga-



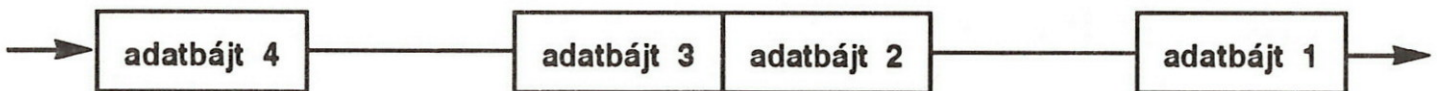
dó oldal lenyeli. Ezért nincs a szinkronkarakterre előírás, az adó és a vevő előre megállapodik abban, hogy mit tekintenek szinkronkarakternek.



15. ábra Szinkron soros átvitel bájtsorozata

### Aszinkron átvitel

Az aszinkron átvitel során ha éppen nincs küldendő adat, a vezeték üres, ami azt jelenti, hogy a következő adatbájt küldéséig 5 voltos feszültség szinten van. Amikor azonban szünet után megjelenik a vonalon az első bájt, a fogadó oldalnak egy kis időre van szüksége a készenlétbe álláshoz, ezért aszinkron átvitelnél az adat formája eltér a szinkron átvitel formátumától.



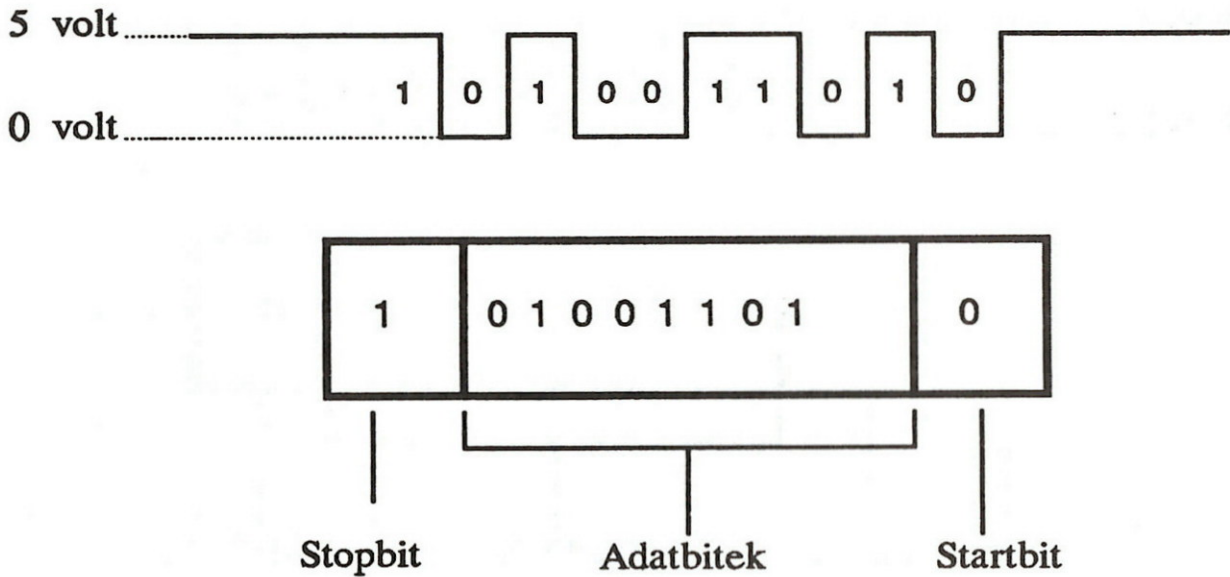
16. ábra Aszinkron soros átvitel bájtsorozata

### Aszinkron adatformátum

Mivel a fogadó oldalnak időre van szüksége a vételkésztséghez, az adatbájt első bitjét egy figyelmeztető bit, a startbit előzi meg. Az adatok közötti szünetben a vonal 5 voltos szinten van, ezért logikus, hogy a startbit mindig 0 szintű, azaz 0 (l. 17. ábra). Ezután következnek az adatbájt bitjei, ami 8 bitet jelent. Azonban ha nem továbbítunk programokat (vagyis gépi kódú utasításokat), csak angol nyelvű szöveges fájlokat vagy parancsokat, akkor az ASCII karakterkészletnek csak az első, standard részét használjuk, amelyek kódjaiban a legnagyobb helyiértékű bit mindig 0. Ezt nyugodtan elhagyhatjuk, vagyis az adatbitek száma csak 7 lesz az elküldött bájtban. A felesleges bit elhagyásával pedig gyorsul az átvitel.

Természetesen a fogadó oldalnak előre tudnia kell, hogy hány bites adatokat várjon, egyébként a beérkező bitekből rosszul állítja majd össze az adatokat. Az adat végét egy stopbit jelzi, ami mindig 1, vagyis 5 voltos szintű. Ennek akkor van jelentősége, ha az adat után nem szünet, hanem közvetlenül a következő bájt bitje érkezik. A stopbit segít a két bájt bitjeinek a szétválasztásában. A





17. ábra Aszinkron adatformátum

stopbitek száma nemcsak 1, hanem 2 is lehet, sőt másfél stopbitet is használnak, igaz ritkán. Ez természetesen nem a bit elfelezését jelenti (ez nehéz lenne), hanem csupán annyit, hogy a neki megfelelő 5 voltos szintet másfél bitnyi időre kapcsolják a vezetékre.

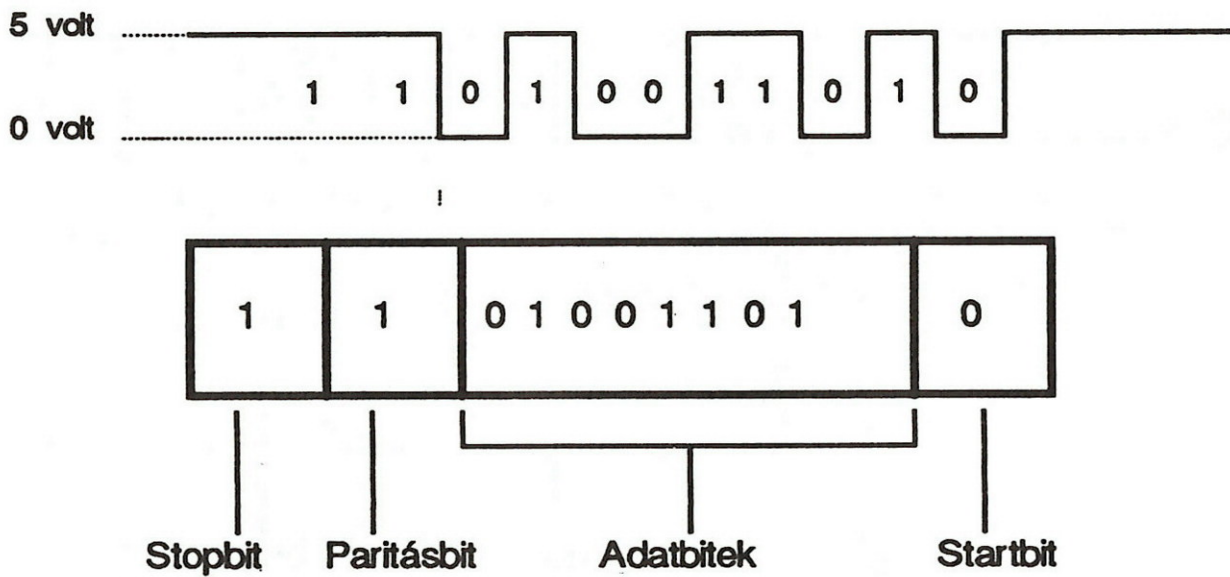
Korábban említettük, hogy a helytelen szinkronizálás, nem megfelelő adatformátum – amikor a vevő más formátumú adatot vár, például 7 adatbitet 8 helyett – vagy külső zavarok is hibát okozhatnak. A beérkező bitfolyamból a vevő az esetek többségében nem tudja észrevenni, hogy hiba történt, ezért szükség van hibaellenőrző mechanizmusokra. Ezek legegyszerűbb formája a paritásbit.

Az adó és vevő előre megegyeznek, hogy használják-e a paritásbitet, és ha igen, milyen értelemben, páros vagy páratlan. A küldendő adatban az egyes értékű bitek száma 0 és 8 között lehet. Amennyiben például páros paritást használnak az átvitel során, minden olyan bájthoz, amelyben az 1-es értékű bitek száma páratlan, – a 0-t páros számnak tekintik –, hozzáírnak még egy 1-es értékű bitet, ez a paritásbit. Ha az 1-esek száma páros, a paritásbit 0 lesz. Így tehát páros paritás esetén a paritásbit úgy egészíti ki az adatot, hogy az 1-es értékű bitek száma mindig páros legyen. A vevőoldal, megszámlálva az egy adaton belüli 1-eseket, ha páratlan számút talál, tudja hogy hiba történt.

A paritásbitet az utolsó adatbit és a stopbit közé helyezik el, mint azt a 18. ábra mutatja.

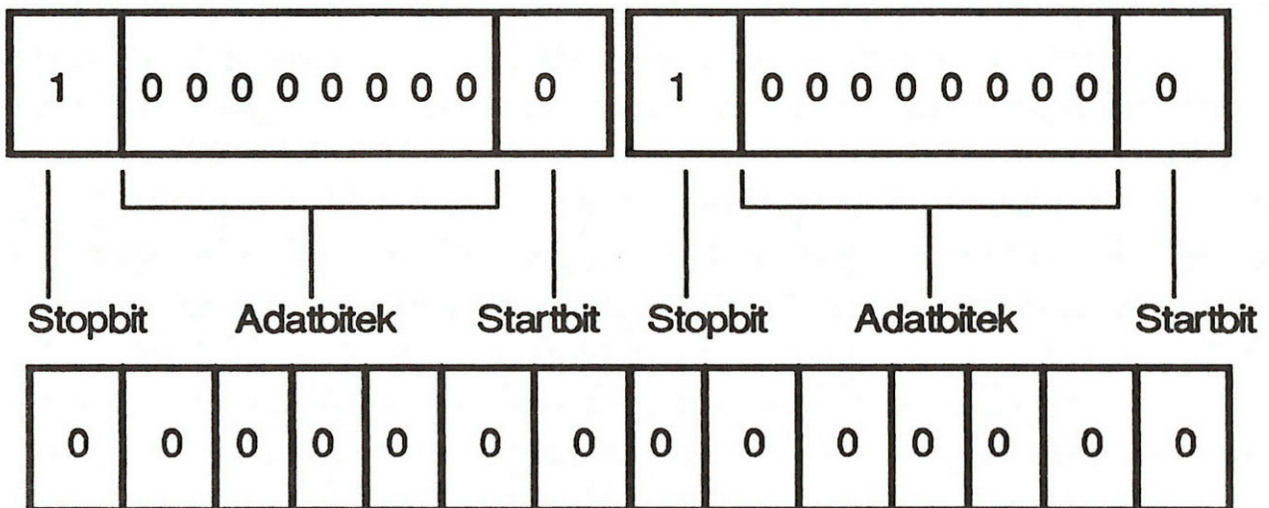
Minden továbbított adatban legalább egy 1-es értékű bit van, a stopbit. Tehát ha csupa 0-t tartalmazó karaktereket küldünk egymás után, akkor is lesznek a bitsorozatban 1-es bitek, a stopbitek. Adásszünetben a vezeték 5 voltos szinten van, így a csupa 0-ból álló bitsorozatot külön funkcióra lehet fenntartani. Ez a





18. ábra Aszinkron adatformátum páratlan paritással

break (törés) -jel, amely egyszerűen úgy jelenik meg, hogy több bájtnyi időre 0 feszültséget kapcsolunk a vezetékre. A break-jela körülményektől függően jelenthet figyelem kérést, pl. az egyik oldal jelezheti a másiknak, hogy kész fogadni az adatokat, vagy adás közben a fogadó oldal ezzel tudathatja, hogy hibát észlelt, az adást le kell állítani, majd megismételni.



19. ábra Csupa 0-t tartalmazó karakterek és a break-jel

Az aszinkron adatátvitelnél, mint láttuk, mindkét oldalnak ugyanazt az adatformát kell használnia, amiben előre megállapodtak. Ez jelenti az adatbitek és a stopbitek számát, ha használnak paritásbitet, hogy az páros vagy páratlan, és a bitek küldési gyakoriságát, hogy szinkronban legyenek. Ez tulajdonképpen az adatátvitel sebessége. A leggyakrabban használt aszinkron adatformátum a 8 adatbit, 1 stopbit, paritásbit nélkül.



A kétféle – szinkron és aszinkron – átvitel közül elméletileg – hiszen nem kell sem start-, sem stopbit – és a gyakorlatban is a szinkron átvittel érhető el nagyobb sebesség. Ennek ára azonban a magasabb költség, ami a drágább hardverigényből adódik.

### Az adatátvitel sebessége

Az adatátvitel sebességét az egy másodperc alatt továbbított bitek számával mérjük és bps-mal (bit per szekundum) jelöljük. Az adó és a vevő oldalnak az adatformátum rögzítése mellett az átviteli sebességben is meg kell állapodnia. Enélkül nem szinkronizálódhatnak össze megfelelően.

Az átviteli sebességet gyakran nem bps-ban, hanem baudban adják meg. Ez a mértékegység Frenchman J.M.E. Baudról kapta a nevét, aki a távírón és a telexen használt 5 bites kód kidolgozója volt.

Egy adatátviteli kapcsolat baudban és bps-ban kifejezett sebessége lehet ugyanaz a szám, de lehet különböző is, ez a körülményektől függ. A bps – mint láttuk – az egy másodperc alatt továbbított bitek száma, míg a baud az egy másodperc alatti jelváltások számával egyenlő. Eddigi példáinkban tisztán digitális átvitelt mutattunk be, amelyekben minden feszültség-szint-váltás (0 és 5 volt között) egy újabb bit továbbítását jelentette. Ekkor a baud és a bps értéke megegyezik. (Fordítva nem igaz a dolog, nem kell minden bithez feszültséget váltani, gondoljunk két egymást követő 0-ra vagy 1-re.)

Már említettük, hogy a soros átvitelnél használhatjuk a telefonhálózatot is az adatok továbbítására. Ennek előnye, hogy már egy meglehetősen jól kiépült rendszer áll a rendelkezésünkre, nem nekünk kell a kábeleket lefektetni, tehát olcsó. A telefont azonban beszéd és nem digitális jelek továbbítására tervezték, a 0-k és 1-esek sorozatát nem kapcsolhatjuk rá. Ehelyett mindkét bitnek megfeleltethetünk egy adott frekvenciájú hangot, pl. a 0-nak 1000 hertzes az 1-nek pedig 2000 hertzes hangot, és ezeket kapcsolgatjuk az éppen küldendő bitnek megfelelően. A jelváltások ekkor is egy új bitet jelentenek, vagyis a baud egyenlő a bps-mal. Az átvitel gyorsítására, ha a költségek még nem túl nagyok és a telefontársaság engedélyezi, más módszert is használhatunk. Nem két, hanem négy különböző hangot állítunk elő, és ehhez a biteket kettesével csoportosítjuk. A kettes bitsorozatoknak a következőképpen feleltetjük meg a hangokat:

00 = 1000 Hz

01 = 1500 Hz

10 = 2000 Hz

11 = 2500 Hz



Most egy hangváltás nem egy, hanem két újabb bitet jelent. Ha tehát egy másodperc alatt 1000-szer váltjuk a hangokat, az  $2 \times 1000$ , azaz 2000 bitet jelent. Az átvitel sebessége ekkor 1000 baud, bps-ban pedig 2000 bit/szekundum. A baud-érték tehát csak akkor ad információt az átvitel sebességéről, ha tudjuk, hogy hány bps-nak felel meg. A bauddal inkább az átvivő eszköz ill. berendezés – esetünkben a telefon – képességeit jelzik, hány jelváltást enged meg másodpercenként.

A legáltalánosabban használt szabványos átviteli sebességek: 300, 1200, 2400, 4800, 9600 és 19 200 bps.

### Hibaellenőrzés

Az átvitel során keletkezett adathibákat a vevő nem veszi automatikusan észre, erről külön kell gondoskodni. Legegyszerűbb módszer a már ismert paritásbit. Ennek hátránya, hogy csupán egyetlen bit hibás továbbítását észleli biztosan. Ha a hiba pl. két 1-es bitet állít 0-ra, a paritásbit ezt nem mutatja ki, hiszen az egyesek száma páros vagy páratlan maradt. A hibák ráadásul általában lokálisan jelentkeznek, azaz ritka hogy egy-egy bájtban csupán egy-egy bit romlik el.

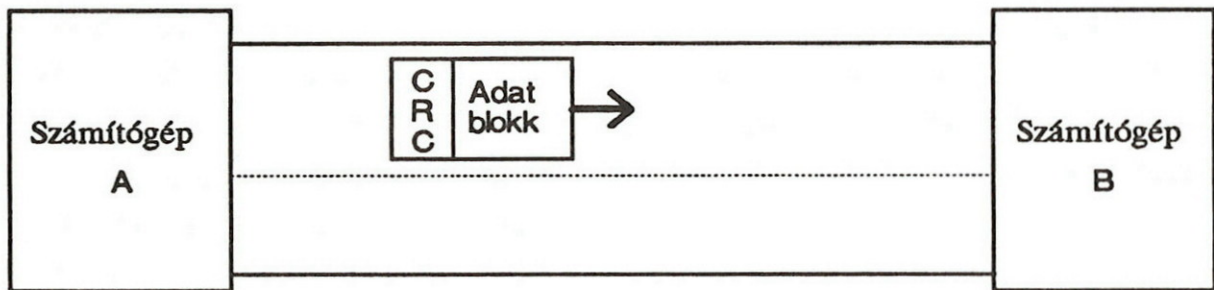
Az átviteli eljárásokat általánosan protokolloknak nevezik. A protokoll egyszerűen egy szabálygyűjtemény. A hibák észlelésére szolgáló módszerek szintén a protokollok részét alkotják. A legtöbb esetben az átvitelre szánt adatokat kisebb részekre – blokkokra vagy csomagokra – osztják és minden részhez hozzáfűznek egy hibaellenőrző kódot. Ennek hossza változó, de általában több bájt. Legelterjedtebb a CRC (cyclic redundancy check), vagyis ciklikus redundancia ellenőrző kód. A blokk bájtjaiból a küldő egy meghatározott módon kiszámítja a CRC kódot és hozzáfűzi a blokk végéhez. A vevő, a blokk bájtjainak beérkezése után ugyanazzal a módszerrel kiszámítja a CRC értékét és összeveti a beérkezett CRC kóddal. Amennyiben a kettő nem egyezik, akkor az átvitel során valamilyen hiba történt. A CRC kód a hiba javítására nem ad lehetőséget, csak megismételni lehet a blokk küldését. A CRC-t nemcsak a kommunikációban használják, a PC így ellenőrzi a mágneses adattárolókra (floppy és harddisk) írt adat helyességét. A protokollok nagy része minden blokk helyes vétele után egy nyugtázó üzenet küldését írja elő a vevő számára. Ha ez elmarad, az adó megismétli a küldést. Mindebből az is következik, hogy mind az adó, mind a fogadó oldalon ugyanazt a protokollt kell használni.



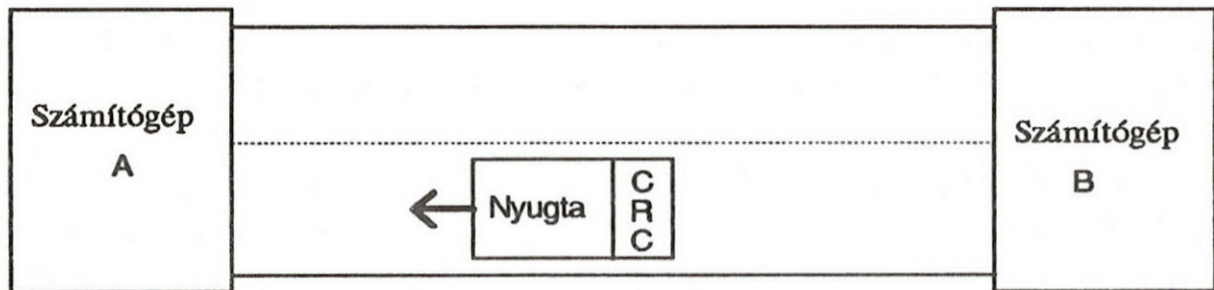
### Átviteli üzemmódok

Mindeddig az egyszerűség kedvéért a soros portot – akár szinkron, akár aszinkronról volt szó – úgy tekintettük, mint két szál vezeték, amelyen egyirányú átvitel folyik a küldő számítógéptől a fogadó felé. A valódi soros portok azonban a két gép között kétirányú átvitelt tesznek lehetővé, két adatutat vagy csatornát tartalmaznak, a két oldal egyszerre adhat és vehet adatokat. A soros adatátvitelt vezérlő protokollok kétféle módon használhatják az átviteli vonalat, fél-duplex és teljes-duplex üzemmódban.

#### 1. lépés: küldés



#### 2. lépés: nyugtázás



20. ábra A fél-duplex átvitel sémája

A fél-duplex módban a vonalon egyszerre mindig csak egyirányú átvitel folyik (l. 20. ábra). A küldő minden blokk továbbítása után megvárja a vevő nyugtázó üzenetét.

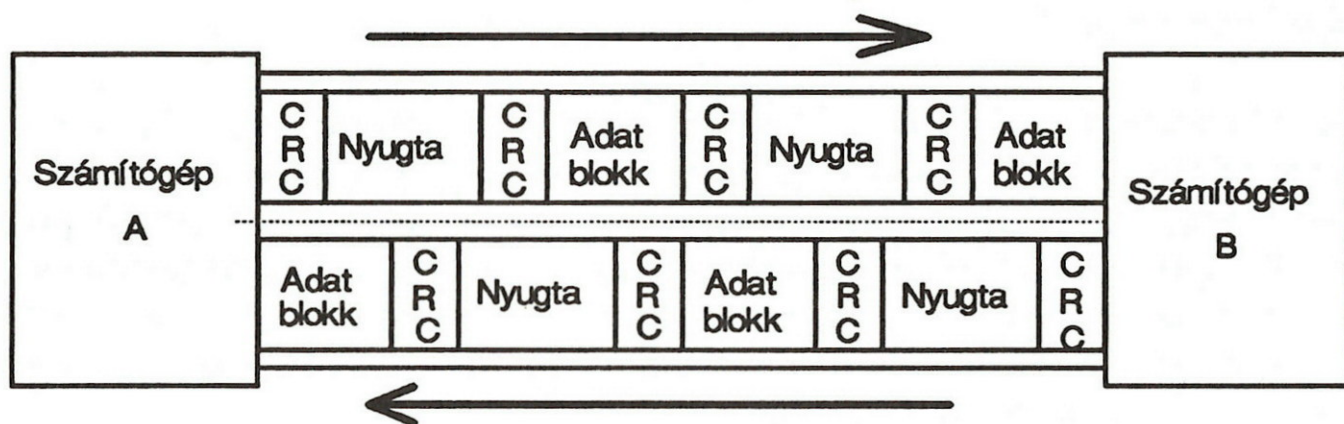
A teljes-duplex átvitelnél mindkét csatornát kihasználják, a küldött blokkok közé beszúrva a beérkezett csomagok nyugtázó üzenetét.

A teljes-duplex átvitel gazdaságosabb, mégis használják a fél-duplex átvitelt is, mert előfordulhat hogy az adott környezetben csak erre van lehetőség.

Az egyes protokollok az adatátvitel folyamatának szabályozásában is különböznek, aszerint hogy szoftver vagy hardver eszközöket használnak ehhez.

A PC-ken, ha hosszabb szöveget ír az operációs rendszer a képernyőre, a CTRL-S billentyű lenyomásával ez megállítható, majd bármely billentyű leütésé-





21. ábra A teljes-duplex átvitel sémája

re továbbmegy. Ez a legegyszerűbb példa a szoftveres folyamatvezérlésre, a CTRL-S egy vezérlőkarakter, amely felfüggeszti a képernyőre írást. Az adatátvitel során a fogadó oldal a CTRL-S (amely az előző fejezetben lévő ASCII kód-táblán DC1 néven szerepel) küldésével függeszti fel az adó oldal küldését, majd ha kész újabb adatokat fogadni, a CTRL-Q (DC3) kóddal kéri a folytatást. Szoftver vezérlésnél tehát vezérlő karaktereket használ a protokoll. A két vezérlőkarakterre elterjedt az XON (CTRL-Q, DC3) és az XOFF (CTRL-S, DC1) elnevezés is, az ezt használó protokollokat pedig XON/XOFF típusúnak nevezik.

A hardveres folyamatvezérlésnél a soros átvitel kézfogás-jeleit használják fel. Ezek általában a CTS/RTS jelek, és a következő részben foglalkozunk velük.

A szoftveres folyamatvezérlés egyszerűbb, de sok esetben azzal a megkötéssel jár, hogy maguk a vezérlő karakterek nem fordulhatnak elő az adatok között, mert ez érthetően megzavarná az átvitelt. Ez azt jelenti, hogy csak szöveges fájlokat továbbíthatunk, programfájlokat nem, hiszen azokban mindenféle karakter előfordul. Ennek kivédésére néhány protokollnál a vezérlőkaraktert mindig egy break-jel előzi meg, ami jelzi, hogy a következő bájtnak nem adat.

Azokat a protokollokat, amelyek a továbbítandó adatokkal szemben semmilyen megkötést nem tartalmaznak, átlátszóaknak nevezik.

## A soros port konfigurálása

Az átvitel jellemzőit – sebesség, adat- és stopbitek száma, paritás típusa – tehát mindkét oldalon azonosan kell beállítani, a két portot – idegen szóval – azonosan kell konfigurálni. A mai soros interfészek meglehetősen rugalmasan állíthatók be megfelelő programmal, nem kell a gép belsejébe nyúlni. Régebben használtak ilyen, hardver úton konfigurálható portokat is. A PC-kben maximá-



lisan négy aszinkron soros port lehet, COM1, COM2, COM3 és COM4 néven. Beállításukra használható a DOS operációs rendszer MODE parancsa, azonban a különböző kommunikációs programok általában ennél sokkal kényelmesebb beállítási lehetőséget nyújtanak.

### Az RS-232C interfész

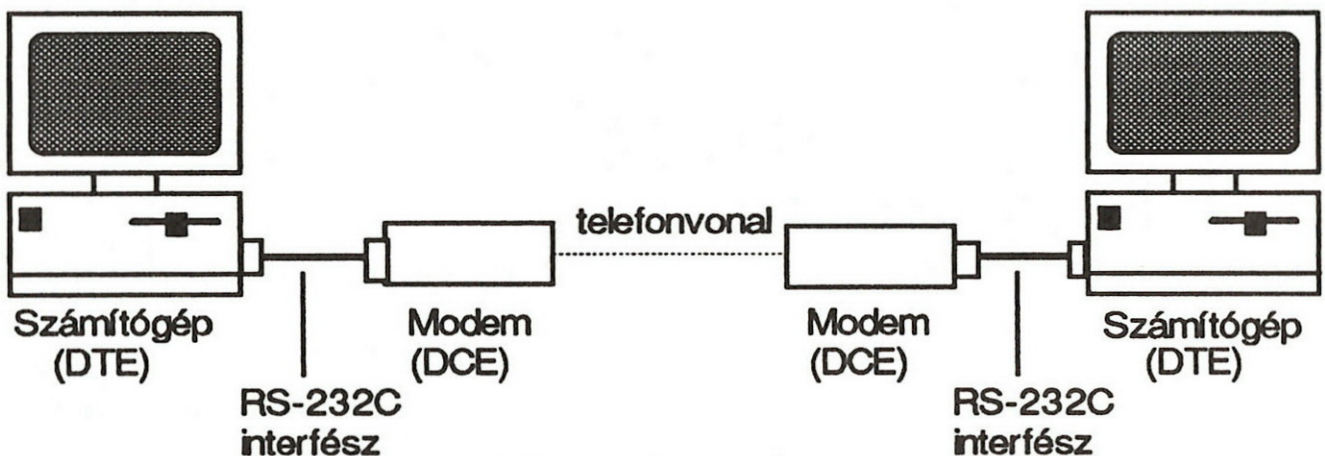
Az előző részben áttekintettük a soros kommunikáció alapfogalmait, ezután a legelterjedtebb soros interfészt mutatjuk be, az RS-232C-t. Ennek eredeti változatát az EIA (Electronics Industry Association) nevű szakmai szervezet 1971-ben dolgozta ki. Azóta kétszer vizsgálták felül, így kapta az RS-232C jelölést (Revised Standard 232). Nagyjából megegyezik a CCITT (Consultative Committee on International Telephone) nemzetközi szervezet V.24 sorszámú szabványával. A szabvány három ponton ad támogatást a soros aszinkron átvitelhez:

- pontosan megadja a csatlakozó szabványos méretét, egyes jelek bekötését,
- rögzíti az átvitelt támogató kézfogás-jelek funkcióját,
- leírja az interfészen át létrejövő kapcsolat egyes lépéseit.

A V.24 szabvány aszinkron átvitelt támogat, az RS-232C csatlakozóját és kézfogás-jeleit azonban szinkron átvitelhez is használják.

A PC-ben használhatunk szinkron átvitelt megvalósító adaptereket is (ilyen a BSC vagy az SDLC kártya), de az általánosan soros portnak nevezett interfész aszinkron, RS-232C átvitelt valósít meg.

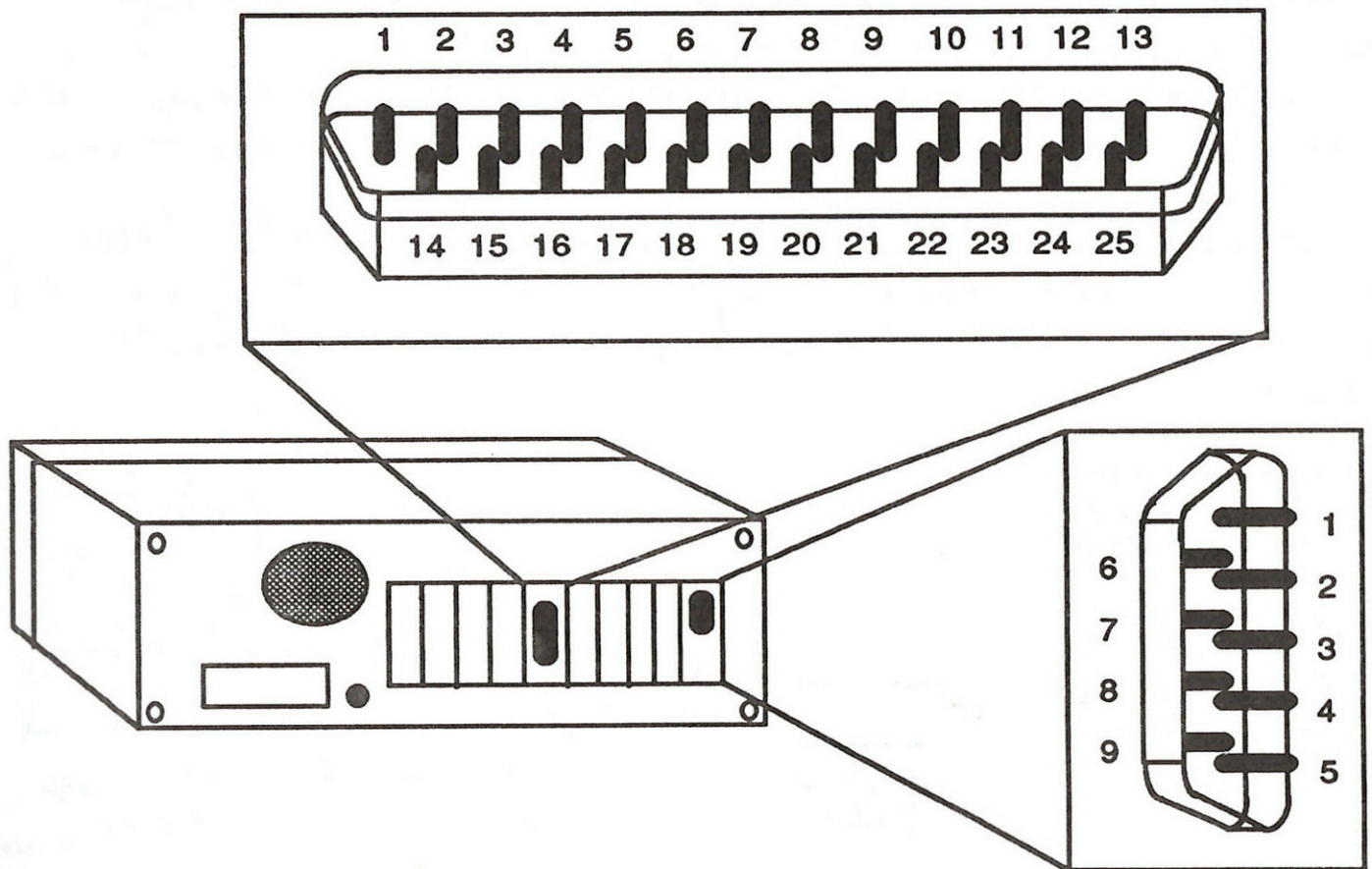
Bár az RS-232C két egyenrangú számítógép közvetlen összekapcsolását is lehetővé teszi, eredetileg nem erre szánták. Az egymástól nagyobb – akár több-száz kilométer – távolságra lévő gépek között kétféleképpen létesíthetünk kapcsolatot.



22. ábra Számítógépek összekapcsolása a telefonhálózaton keresztül



Építhetünk egy kifejezetten erre a célra szolgáló hálózatot, amelyen viszonylag nagy sebességet érhetünk el, kiépítése azonban meglehetősen költséges (a kapcsolat elképzelhető akár műholdon keresztül is), vagy a hagyományos, már kiépült telefonhálózatra támaszkodunk, ami kétségtelenül olcsóbb. A telefonhálózatot azonban nem számítógépes átvitelre, hanem emberi hang továbbítására tervezték, ezért csak kis sebesség érhető el rajta, és digitális adatokat nem kapcsolhatunk rá. Ezért szükség van – sok esetben a számítógépes hálózatonál is – egy átalakítóra, amely számítógépünk digitális adatait analóg formátumúvá alakítja és így továbbítja a telefonvonalon, vagy a hálózaton. A másik oldalon ugyanilyen berendezés veszi az adatokat és a digitális formára alakítás után átadja a másik számítógépnek. Ezeket az átalakítókat modemnek (a modulátor-demodulátor szóból) vagy a szabvány hivatalos szóhasználatával DCE-nek (Data Communications Equipment), adatátviteli berendezésnek nevezik, míg a számítógép szabványbeli neve DTE (Data Terminal Equipment), azaz adatvég berendezés. A szabvány a számítógép mint DTE és a modem mint DCE közötti kapcsolatot írja le, és két gép közvetlen összekötését, tehát a DTE-DTE kapcsolatot is lehetővé teszi.



23. ábra Az RS-232C csatlakozói a PC-n



Az RS-232C szabványos csatlakozója 25 pólusú, DB-25 néven, azonban nem használja valamennyi érintkezőt, így elterjedtek az olcsóbb, 9 pólusú csatlakozók is (l. 23. ábra).

A csatlakozó az adatvonalakon kívül a kézfogás-jeleket tartalmazza, amelyek a következők:

Érintkező száma a csatlakozón		Rövidítése	Teljes neve
25 pólusú	9 pólusú		
2	3	TD	Küldött adat (Transmitted Data)
3	2	RD	Vett adat (Recived Data)
4	7	RTS	Adáskérés (Request To Send)
5	8	CTS	Adásra kész (Clear To Send)
6	6	DSR	Adat kész (Data Set Ready)
7	5	GND	Jelföld (Ground)
8	1	CD	Vivőérzékelés (Carrier Detect)
20	4	DTR	Adatterminál kész (Data Terminal Ready)
22	9	RI	Csengetés jelző (Ring Indicator)

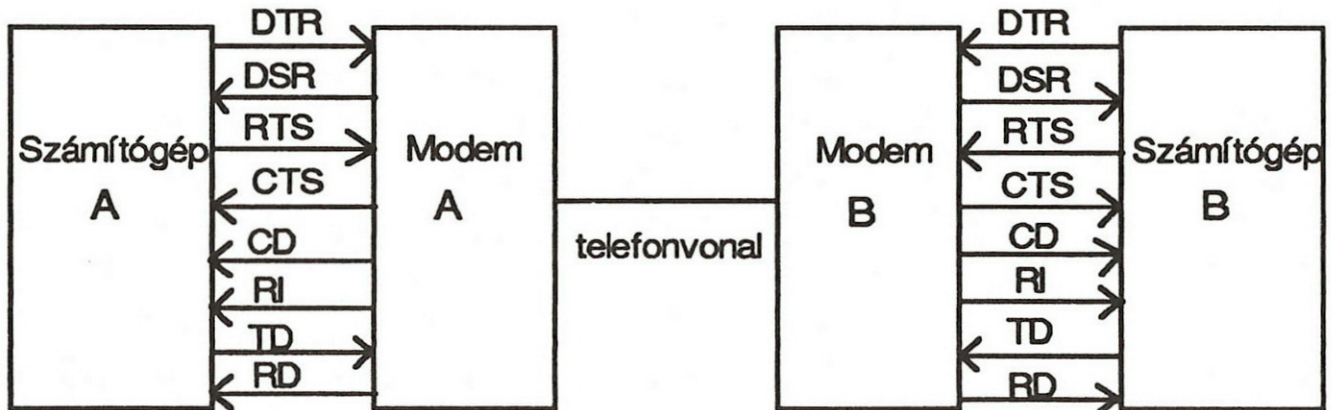
A szabvány a fentieknél több jelet tartalmaz, de ezek a legfontosabbak, és az aszinkron kommunikációhoz elegendők.

Az adattovábbításra kétféleképpen használhatjuk a telefontvonalat. Ha csak időnként kell adatokat küldenünk, ugyanúgy járunk el, mint normál beszélgetésnél, tárcsázás után ha kapcsolt, elindítjuk az átvitelt. A kapcsolat végén természetesen le kell tenni a kagylót. Mivel a telefon ugyanúgy kapcsol, mint a beszélgetéseknél, ezt kapcsolt vonalon való kommunikációnak nevezik. Amennyiben rendszeresen nagy mennyiségű adatot kell átvinnünk és esetleg jobb minőségű vonalat szeretnénk, bérelhetünk egyet a telefontársaságtól, amely már csak a mi rendelkezésünkre áll, így nem kell tárcsáznunk. Ez a bérelt vonalas átvitel. A kapcsolt esetben érdemes figyelembe venni, hogy ha a vonal rossz minőségű – zajos, áthallásos stb. –, a hibák miatt az átvitel egyes részeit többször ismételni kell, ami a telefonszámlán – az adatok mennyiségéhez képest – érezhető lesz. A bérelt vonalak esetén nemcsak két számítógépet kapcsolhatunk össze – úgynevezett pont-pont összeköttetést létesítve –, hanem, a konferenciahívásokhoz hasonlóan, egyszerre több gép is összekapcsolódhat. Ekkor az adatküldést megelőzi a címzett gép kiválasztása egy kóddal, de egyszerre is küldhetünk adatot minden gépnek.



## Adatátvitel modemen keresztül

A 24. ábra segítségével lépésenként végigkövetjük két számítógép kapcsolatfelvételét a telefonhálózaton át. Kapcsolt vonalat tételezünk fel, mivel ez a bérelt vonalas átvitelt is magában foglalja. A modemet használhatjuk önállóan vagy a telefontal közösen, ekkor, ha a modem nem képes rá, kézzel kell tárcsáznunk. Ilyenkor egy kapcsolóval állítható, hogy a modemet vagy a telefont kapcsoljuk a vonalra.



24. ábra Az RS-232C interfész jelei a modem és a számítógép között

1. Példánkban az A számítógép akar adatot továbbítani, és hívja fel a B gépet a telefonhálózaton át. Először a DTR (Adatterminál kész) jelet adja ki a modemnek. Kézi kapcsolásnál a kezelő felemeli a kagylót és tárcsáz. Az automata modem önműködően tárcsáz.

2. A kapcsolás után a B modem érzékeli a csöngetést és erről az RI (Csengés jelző) vonalon tájékoztatja a B számítógépet. Ha a B gép kész fogadni az adatokat, a DTR (Adatterminál kész) vonalon jelzi a modemnek. A B modem egy válaszhangot ad a vonalon. Kézi kapcsolás esetén ekkor a kezelő a vonalat átkapcsolja a modemre.

3. Az A modem meghallva a válaszhangot, a DSR (Adat kész) vonalon jelzi az A gépnek a kapcsolat létrejöttét.

4. Az A gép az RTS (Adás kérés) vonalon jelzi, hogy át akarja adni a küldendő adatot. Az A modem erre egy vivőhangot ad a vonalra.

5. A vivőhangot a B modem a CD (Vivőérzékelés) vonalon jelzi a B gépnek.

6. Az A modem a CTS (Adásra kész) jellel kéri az adatot, amelyet az A gép a TD (Küldött adat) vonalon ad át a modemnek. A modem a vivőhangot modulálva továbbítja az adatbajtót a telefonvonalon át a B modemnek, amely ezt visszaalakítva már digitális adatként továbbítja a B gépnek az RD (Vett adat) vonalon.

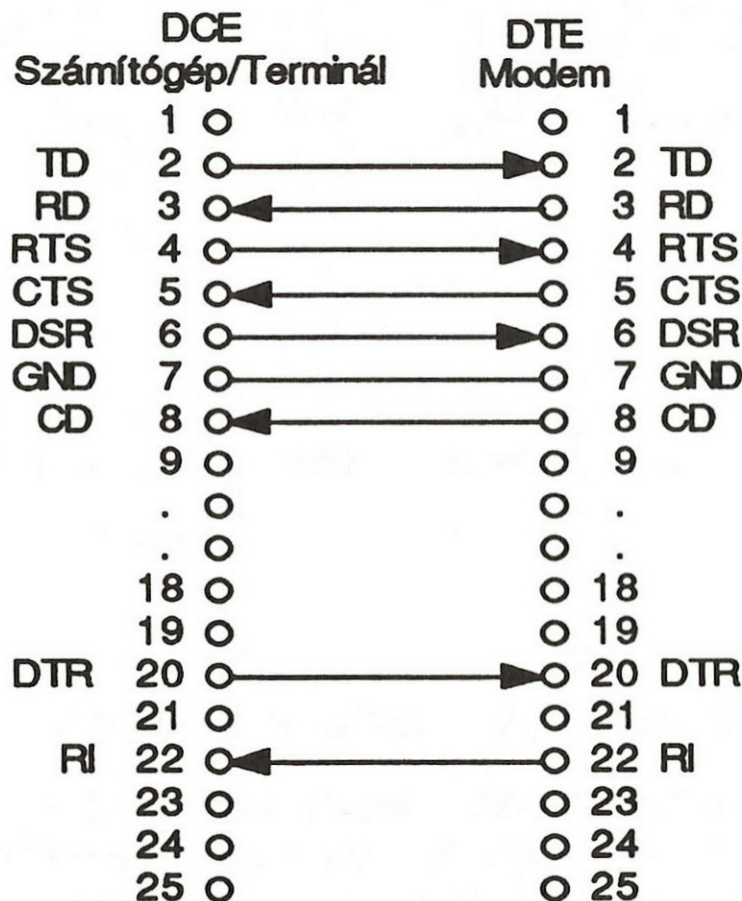


7. Ha az A gép befejezte a küldést, megszünteti az RTS (Adáskérés) jelet, amelyre a modem a CTS (Adásra kész) visszételével reagál és nem adja tovább a vivőhangot a telefonvonalra. A B modem a CD (Vivőérzékelés) jel visszavételével tudatja a B géppel, hogy már nem hallja a vivőhangot.

Ezután a két számítógép szerepet cserél és a B gép a 4. lépéstől kezdve átveszi a kezdeményező szerepet. Bérelt vonal esetén a tárcsázást leszámítva ugyanazt az eljárást követik. Mint látható ez az eljárás fél-duplex, a telefonhálózat sok helyen nem teszi lehetővé – műszaki okok miatt – a teljes-duplex átvitelt.

Az interfész tervezésekor még nem léteztek PC-k, sem más olcsó számítógépek, az adatátvitel elsősorban a nagyszámítógéphez kapcsolódó terminálokról történt. Ezért szerepel a jelek elnevezésében. A terminál nem önálló számítógép, hanem csak egy, a számítógéppel kapcsolatot tartó eszköz. Az interfész használata szempontjából azonban valójában lényegtelen, hogy milyen eszköz van a drót túlsó végén, csak kövesse a szabvány előírásait.

A modem szintén egy 25 pólusú csatlakozóval rendelkezik, a számítógéppel vagy terminállal való összekötése egy 9 vezetőket tartalmazó kábelt igényel, bekötését mutatja a 25. ábra.



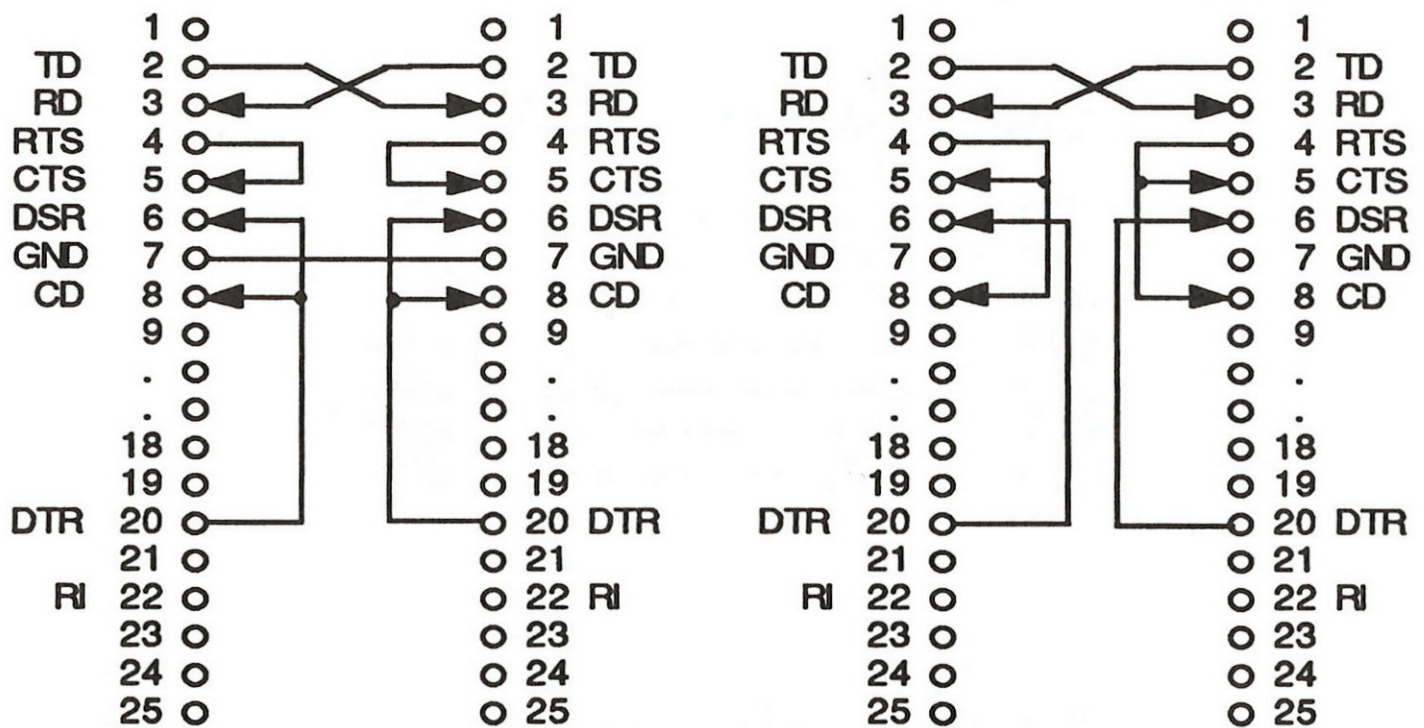
**25. ábra.** A számítógép és a modem összekapcsolása



**Közvetlen kapcsolat az RS-232C-n**

Az interfészen át közvetlenül – modemek nélkül – is összekapcsolhatunk két számítógépet. A szabvány max. 15 méteres távolságon 19 200 bps sebességet engedélyez, a gyakorlatban azonban ennél nagyobb távolságon is lényegesen nagyobb sebesség érhető el. Ez annak köszönhető, hogy az interfészen továbbított bitek és kézfogás-jelek feszültségszintje nem a már ismert 0 és 5 volt. A szabvány +3 és +15 volt közötti szintet 0-s bitnek, a -3 és -15 volt közötti szintet pedig 1-es bitnek tekinti. A PC-kben alkalmazott soros port +12 ill. - 12 voltal dolgozik. A ténylegesen használható sebesség a távolságon kívül függ a kábel minőségétől és a környezet zavaró hatásaitól. Ha az átvitel hibás, alacsonyabb sebességet kell választanunk.

Közvetlen kapcsolatnál a gépeket egy null-modemnek nevezett kábellel kötjük össze. Mivel a szabvány nem foglalkozik a közvetlen kapcsolattal, hanem csak a modemes kommunikációval, a null-modem kialakítása nem teljesen egységes; két megoldást mutat a 26. ábra.



**26. ábra** Kétféle null-modem bekötés

A null-modemen a TD és az RD vonalak csatlakoznak egymáshoz, ellentétben a 25. ábrán látható bekötéssel. A szabvány szerint a null-modem DTE-DTE kapcsolat, ezért kell keresztbe kötni az adás és vétel vonalat. A modemvezérlő jelek nem mennek át a kábelen, mindegyik kézfogás-jel saját válaszjelét állítja elő. Az egyetlen különbség a két bekötés között a Vivőérzékelés (CD) jel előállításában van.



Az adatátvitelhez természetesen nem elég a kábel, megfelelő kommunikációs szoftverre is szükség van mindkét gépen. A szoftverek különböző módon vezérelhetik az átvitelt. Használhatják a kézfogás-jeleket, a DTR/DSR vagy a RTS/CTS párt. Az DTR jel – mint láttuk – egyszerűen üzembesz állapokra szólítja fel a modemet, amely erre a DSR jellel válaszol. Bérelt vonal esetén – mert nem kell felépíteni a kapcsolatot a másik géppel, hiszen a vonal csak a miénk – a DSR jel állandóan aktív, ha a modem be van kapcsolva. Az RTS jellel a konkrét adatküldési szándékot jelzi a számítógép, amelyre a vivőhang kiadása után a modem a CTS-sel válaszol. Bármelyik jelpárt használja is a szoftver, a null-modemmel mindig megkapja a nyugtázást. A vivőérzékelés (CD) jel előálítható mindkét kézfogásjel-párból, a csengetésjelző RI-re pedig nincs szükség.

A szoftver használhatja az XON/XOFF vezérlőkaraktereket is az átvitelhez, ekkor tulajdonképpen nincs szükség a null-modemre, a két adatvezeték és a föld is elegendő.

A szoftverek harmadik csoportja vezérlőkaraktereket sem használ, a fájlokat egyszerűen kisebb részekre – csomagokra – osztja és hozzáfűzve a hibaellenőrző CRC kódot, fél-duplex módon átküldi a vonalon, megvárva a nyugtázást. Ez a módszer azonban csak fájlok átvitelére használható, párbeszédés átvitelre nem.

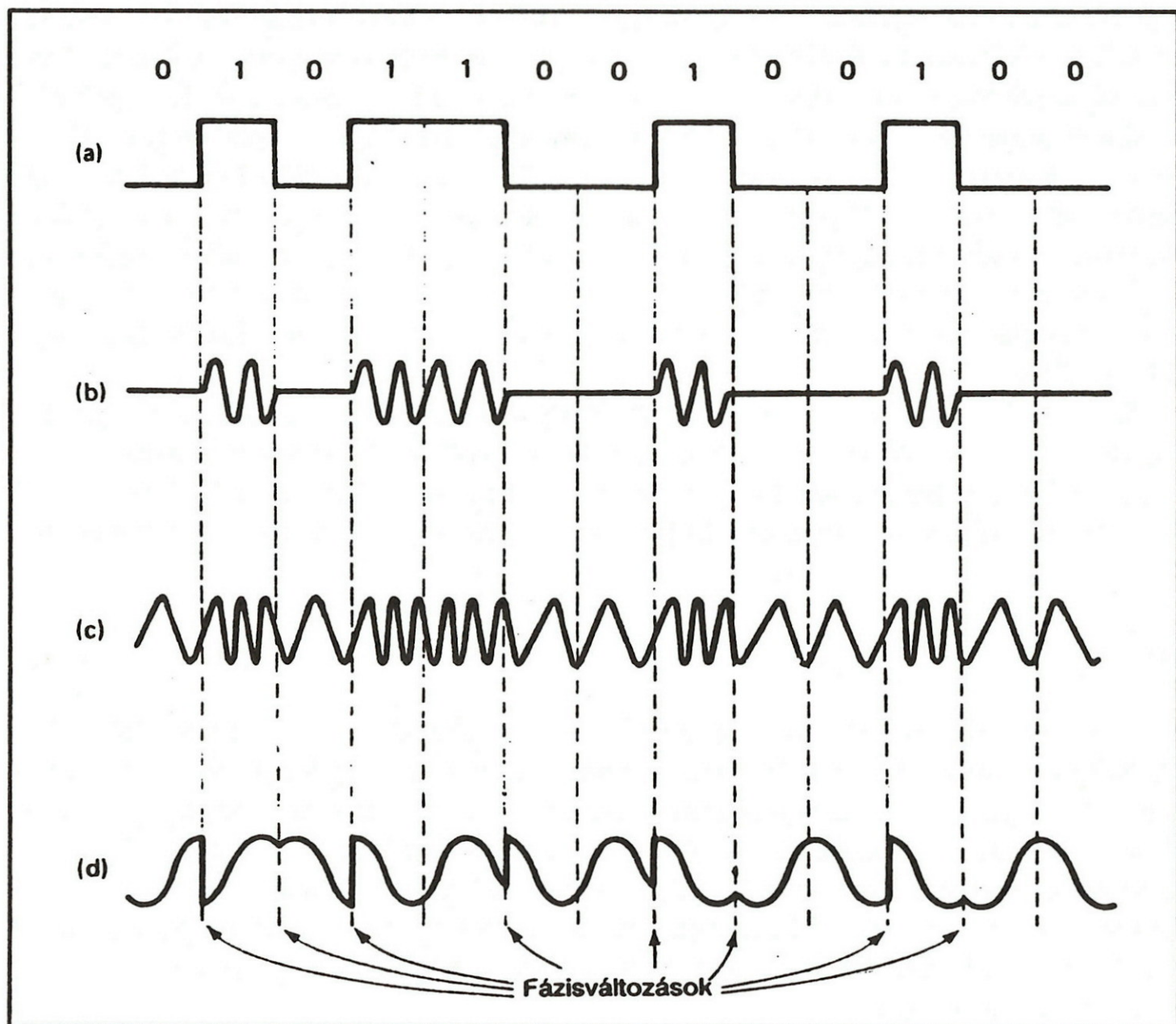
### Modemek

A modemre tehát azért van szükségünk, hogy a számítógép digitális adatait analóg jelként elküldjük a telefonvonalon vagy az erre a célra kiépített hálózaton. A másik oldalon szintén egy modem alakítja át az adatot ismét digitális formájává és adja át a számítógépnek. A kereskedelemben kapható modemek többsége aszinkron módban dolgozik, de léteznek szinkron modemek is, amelyeket bérelt vonalon, illetve számítógép-hálózaton használnak. Az előző fejezetben egy fél-duplex kapcsolat lépéseit követtük végig, de vannak teljes-duplex módot használó modemek is.

A telefonvonalon a továbbítható hangok frekvenciáját különböző szűrőkkel 30 és 3000 Hz között tartják. Ebben a tartományban közvetlenül – azaz digitális formában – átvinni a biteket csak kis távolságon és nagyon alacsony sebességgel lehetne. A modem egy meghatározott frekvenciájú hangot – a vivőhangot vagy egyszerűen csak vivőt (carrier) – használja az adatátvitelre. A vivőjel valamely fizikai jellemzőjét változtatva kódolja az adatbiteket. Ez a paraméter lehet a hang erőssége – pl. egy erősebb hangot feleltetünk meg az 1-es és egy gyengébbet a 0-s biteknek –, ezt hívják amplitúdómodulációnak. Választhatunk két kü-



lőnböző magasságú hangot a két bitnek (ez a frekvenciamoduláció) és változtatjuk a jelerősséget ugrásszerűen, amit fázismodulációnak neveznek (l. 27. ábra).



**27. ábra** A vivőjel modulációjának lehetőségei  
 a) digitális jel, b) amplitúdómoduláció, c) frekvenciamoduláció,  
 d) fázismoduláció

A mai modemek ezeket a lehetőségeket kombinálják, így egy jelváltozással több bitet is átvihetnek. A jelváltozások száma – azaz a sebesség baud-értéke – a fent említett korlátozás miatt nem haladhatja meg a 3000-t, ezért a sebesség növelése csak a kódolás fejlesztésével érhető el. Például egy másodpercenként



2400-szor jelet váltó modem, ha minden jelváltás 4 bit információnak felel meg – ezt a frekvencia- és fázismoduláció kombinálásával érik el –, 9600 bit/sec tényleges sebességgel továbbít.

Fél-duplex modemeknél egyszerre csak egy vivő – az éppen adó modemé – van a vezetéken, a teljes-duplex modemek két, különböző frekvenciájú vivőt használnak a két oldal egyidejű adatküldéséhez.

Mint korábban láttuk, az adatokhoz a kommunikációs szoftver hibellenőrző (CRC) kódot fűz, amely segít az átvitel közben keletkezett adatvesztés felismerésében. A modemek fejlesztése során ezt a funkciót beépítették a modemekbe. A következő fokozat a hibajavító ECC kódok (ECC = Error Correction Code) alkalmazása volt, amellyel kisebb átviteli hibákat a fogadó modem ki tud javítani, nem kell a küldést megismételni. A mai modemek a kapott adatokat tömöríteni is tudják, amivel ugyanolyan fizikai feltételek mellett nagyobb sebesség érhető el. Mindezekkel ma jó minőségű, kapcsolt vonalon 19200 bit/sec használható maximálisan.

A hibajavítást és az adattömörítést a kommunikációs szoftver is végrehajthatja, azonban ez a processzor terhelését növeli. Ráadásul ezek a feladatok sokféleképpen megoldhatók és egyáltalán nem biztos, hogy a tőlünk esetleg többszáz kilométerre lévő partnerünk programja ugyanolyan típusú tömörítést használ, mint a miénk. A funkciók modembe építésekor azonban a gyártók rákényszerültek a vonatkozó szabványok alkalmazására, enélkül ugyanis termékük nem lenne képes más gyártók készülékeivel együttműködni.

Az átvitel lehetőségeit a résztvevő modemek együtt határozzák meg. A fejlesztések igyekeztek tekintettel lenni a korábbi típusokra. A kapcsolat felvételekor a két modem azonosító információt cserél egymással és a többet tudó típus a másikhoz igazítja a paramétereket. Ami az jelenti, hogy hiába van nekünk egy 9600 bit/sec-os, tömörítést is végző típusunk, ha a partner készüléke csak 2400 bit/sec-ra képes, tömörítés nélkül, akkor eszerint kell az adatokat továbbítanunk, igaz modemünk ehhez automatikusan alkalmazkodik.

### **Modemtípusok**

A modemek az átvitelen kívül a következő funkciókat láthatják el:

- tárcsázás
- válaszadás („felveszi a kagylót”)
- kapcsolatbontás

A legelső, akusztikus modemek csak az átvitelt végezték, a fent említett feladatokat a kezelő látta el. Ezek a modemek mikrofonnal és hangszóróval tartot-



ták a kapcsolatot, a telefonkagylóra kellett őket illeszteni, innen a nevük. Ma már nem használatosak.

A közvetlenül, elektronikusan csatlakoztatott típusok normál és automata modemekre (smartmodem) oszthatók. A normál modemek válszolnak és lezárják a kapcsolatot, de tárcsázni nem tudnak. Ezért a telefonkészülékre is szükség van mellettük. Az automata modemek tárcsáznak is, így telefonkészülék nélkül is használhatók. Megjelentek a faxmodemek, amelyek a modemfunkción kívül faxok továbbítására és vételére is képesek. A szövegszerkesztővel elkészített szöveg közvetlenül elküldhető, nem kell kinyomtatni, mint a normál faxon való továbbításhoz.

A modemek beállítása, konfigurálása korábban hardver úton, kapcsolókkal történt. A mai típusok szoftver parancsokat fogadnak, és a kezelőprogramokkal kényelmesen beállíthatók még olyan paraméterek is, mint például, hogy hány csöngetés után vegye fel a kagylót. Ez persze képletes, hiszen az automata modemek mellett nincs szükség a telefonkészülékre. Az egységes kezelhetőség érdekében az automata modemek többsége ugyanazt az AT-nek vagy a bevezető gyártóról Hayesnek (Hayes Microcomputer Products Inc.) nevezett parancskészletet használja.

Az automata modem bekapcsolása után automatikusan parancs üzemmódba áll. Ez azt jelenti, hogy a számítógéptől az RS-232C interfészen át kapott bájtot nem továbbítandó adatnak, hanem neki szóló parancsnak tekinti. Az AT parancskészlet parancsai egyszerűek, az „ATD1234567” parancs például az 1234567-es szám tárcsázására utasítja a modemet. A kapcsolat létrejöttkor a modem automatikusan adattovábbító módba vált át, ekkor az interfészen keresztül kapott adatokat továbbítja a vonalon. Működés közben – feltéve, hogy éppen nem adatokat fogad – egy vezérlő kóddal bármikor átkapcsolhatunk a parancs és az adattovábbító mód között.

Parancs módban tehát minden kapott bájtot utasításnak, adattovábbító módban pedig elküldendő adatnak tekint. Korábban láttuk, hogy az univerzálisan használható átvitelnek átlátszónak kell lennie, vagyis valamenyi, a programokban vagy adatokban előforduló lehetséges karaktert továbbítania kell. Milyen kóddal tudunk akkor átkapcsolni a parancs és adatmód között, amely biztosan nem fordul elő adatként? Az AT parancskészletben ezt egy másodperces szünetet (csendet) követő három + karakter, majd ismét egy másodperces szünet végzi. Ez a sorozat biztosan nem fordul elő fájl továbbításakor, ha egy másodpercig nem jön adat, akkor valamilyen hiba lépett fel. Előfordulása akkor is nagyon valószínűtlen, ha a billentyűzeten begépelte szöveget visszük át közvetlenül, tehát a gépelés ütemében. A sorozat hátránya, hogy viszonylag sok időt igényel.



Az automata modemek a parancskészlettel együtt használhatják az RS-232C interfész már ismert modemvezérlő jeleit, de az AT parancskészlethez elvileg nem szükségesek, a két adatvezeték és a föld is elegendő a kommunikációhoz.

Az automata modemek automatikus válaszadásra is beállíthatóak. Ebben a módban a számítógépet kezelő utasításai nélkül is válaszolnak a bejövő hívásra, veszik és tárolják az adatokat, majd lezárják a kapcsolatot (leteszik a kagylót).

Parancs módban a kapott parancsokat automatikusan visszaküldik a számítógépnek, ebből az esetleg arra következtethet, hogy a kapcsolat már létrejött, és nem saját parancsait, hanem a vett adatokat kapja. Ilyen esetekben le lehet tiltani a modemben a parancsok visszaküldését. Az IBM PC-hez használatos modemek lehetnek önálló egységek, amelyeket ténylegesen a PC soros portjára kell csatlakoztatni, vagy lehetnek egy adapterkártyára építve, amelyet be kell helyezni a gépbe. Ezek a belső modemek. Kezelésük – a fizikai csatlakoztatást kivéve – teljesen azonos.

### Szabványok

Megemlítjük a modemekre vonatkozó legfontosabb nemzetközi szabványokat, mivel sok esetben a modem jellemzésére csak azt adják meg, hogy milyen szabványoknak tesz eleget. A szabványok követik a modemek fejlődését, egyre újabbak jelennek meg. Az európai szabványokat a CCITT V. sorozata, míg az amerikaiakat a Bell Laboratories jegyzi.

CCITT V.21. Közhasználatú, kapcsolt vonalon alkalmazott, duplex, 300 bit/sec-os modem.

CCITT V.22. Közhasználatú, kapcsolt vonalon alkalmazott duplex, 1200 bit/sec-os modem.

CCITT V.22.bis. Közhasználatú, kapcsolt vonalon alkalmazott duplex, 2400 bit/sec-os modem, amely érzékelve a vonal gyenge minőségét, automatikusan 1200 bit/sec-ra csökkenti a sebességet. Azonos a Bell 224 szabvánnyal.

CCITT V.27. Bérelt, négyhuzalos vonalon alkalmazott modem 4800 bit/sec-os sebességgel.

CCITT V.27.ter Közhasználatú, kapcsolt vonalon alkalmazott 4800 vagy 2400 bit/sec-os modem.

CCITT V.29. Bérelt négyhuzalos vonalon alkalmazott, szinkron duplex 9600 bit/sec-os modem, amely rossz minőségű vonalon automatikusan 7200 bit/sec-ra, vagy 4800 bit/sec-ra csökkenti a sebességet.

CCITT V.32. Megegyezik a V.29 szabvánnyal, de kapcsolt vonalon használt modemekre vonatkozik, aszinkron módot is lehetővé tesz és a kódolása is különböző.



CCITT V.42. Hibajavító eljárás, megfelel az MNP 4. osztályának (l. alább).

CCITT V42.bis Adattömörítési eljárás, max. 4:1 arányú tömörítést valósíthat meg, az adatok szerkezetétől függően.

A különböző hibajavítási és adattömörítési eljárások közül a Microcom cég MNP (Microcom Network Protocol) nevű módszere vált a legnépszerűbbé, egy részét a CCITT szabványként elfogadta. Az MNP a modemek típusától függően 10 osztályba sorolta a hibajavítási és adattömörítési lehetőségeket. Ha két MNP modem kapcsolatba lép, akkor a mindkettőjük által ismert legmagasabb osztályú eljárást használják. Az MNP modem letiltja a hibajavítást és a tömörítést, ha a másik oldalon lévő készülék nem ismeri ezeket.

### A PC-adatkommunikáció néhány hazai eredménye

A PROPER számítógépcsald önálló alkalmazása mellett rövid időn belül felmerült az igény a kommunikációra, információcserére szolgáló rendszerekbe való beépítése és az ehhez szükséges eszközök létrehozása iránt. A hardver elemek – szinkron és aszinkron adapterkártyák – mellett a különböző kommunikációs szoftverek egy-egy meghatározó eleme készült el az SzKI-ban. Így a BSC és SDLC kártya, különböző terminálemulációk (BSC 3270, BSC 3780), SNA terminálszoftver stb. Ezek alkalmazásával a PC-k a klasszikus nagyszámítógép távfeldolgozási rendszereinek részévé váltak és ugyanakkor megalapozták a hazai PC bázisú kommunikációs kultúrát. Néhány jellegzetes alkalmazás:

– Az IBM 3270/3780, csoportos interaktív és kötegetelt feldolgozásra egyaránt alkalmas nagyszámítógépek környezetében alkalmazhatóvá váltak a PC-k.

– Az X.25 átjáró (gateway) lehetővé tette egyedi és helyi hálózatba kötött PC-k csomagkapcsolt hálózati alkalmazását.

– Lehetőség van a PC-k között műholdon keresztüli adattovábbításra.

– A PC-fax adapterkártya és a kezelőszoftver (SzKI FaxSmile) lehetőséget teremt a hagyományos fax berendezések leváltására, segít a papírmentes iroda kialakításában.

– Az analóg modemek használata mellett megkezdődött a digitális, nagysebességű (64, 128 Kbit/sec) PC-kártyák tervezése és alkalmazása. Ilyen az SzKI és az Albacomp vállalkozásában, a MATÁV támogatásával készülő TAVIS PILOT rendszer.



# III. Számítógép-hálózatok

## Miért kell a hálózat?

Az információ megosztása céljából, két gép összekapcsolása után logikus gondolat újabb gépek csatlakoztatásával a számítógép-hálózat kialakítása. A gépek pusztán összekapcsolása azonban még nem hálózat. Hálózattá azok a szoftverek teszik, amelyek működtetik a rendszert, gondoskodnak arról, hogy a hálózat egyik számítógépét használó minél kényelmesebben érje el a távoli gépen tárolt adatokat. Sokszor nem is tudja, hogy az általa kért adat melyik gépen található, vagy a programja a hálózat melyik gépén fut. Az ilyen mértékben integrált hálózatokat nevezik elosztott rendszereknek.

A számítógép-hálózat a ma ismert leghatékonyabb információ-továbbítási módszer, hatása a múlt században a vasút, századunk első felében pedig a gépkocsi hatásához mérhető.

Elterjedését a számítógépek számának az elmúlt évtizedben bekövetkezett robbanásszerű növekedésének köszönheti. A számítástechnika alkalmazását egészen a 70-es évekig az úgynevezett nagyszámítógépek (mainframe) uralták. Ezek légkondicionált helységet és külön kezelőszemélyzetet igényeltek. Áruk és üzemeltetési költségeik olyan magasak voltak, hogy a legtöbb cég csak egyetlen géppel dolgozott. A felhasználó legtöbbször nem is találkozott a géppel. Papíron leadta az adatokat vagy a programját – ennek gépre vitelét általában a kezelőszemélyzet végezte –, az eredményt pedig szintén papírra nyomtatva kapta meg. Ezt a fajta géphasználatot nevezik kötegelt (batch) feldolgozásnak. Ilyen körülmények között hálózatra egyszerűen nem volt szükség, a gépek közötti információcserét az adathordozók (leginkább mágnesszalagok) szállításával oldották meg. A nagyszámítógépek vezető gyártója a kezdetektől napjainkig az IBM (International Business Machines).

A 70-es évek elején a technológia fejlődése kisebb és olcsóbb – igaz, teljesítményben is gyengébb – számítógépek építését is lehetővé tette. Ezeknek a minigépeknek nevezett számítógépeknek az üzemeltetése lényegesen kisebb költséggel járt, kisebb vállalatok is alkalmazhatták őket. Ezek a gépek is támogatták a számítógépek új típusú használatát, amelyben a felhasználók egy monitorból és billentyűzetből álló terminálon – a már ismert soros interfészen át – közvetlen kapcsolatban állnak a számítógéppel. Ezt az interaktívnak is nevezett kapcsolatot ma már természetesnek vesszük, akkor azonban a kötegelt feldolgozáshoz képest nagy előrelépés volt. A minigépek legnevesebb gyártója és fejlesztője a DEC (Digital Equipment Corporation) volt.



Amint egy cégnél több számítógépet kezdtek el használni, felmerült az összekapcsolásuk igénye. Ennek kielégítésére a nagy gyártók saját gépeik támogatására létrehozott hálózatokat, amelyek azonban más típusú gépekhez csak nehézkesen voltak alkalmazhatók. A vevőknek ugyanattól a cégtől kellett vásárolnia a hálózatot is, akitől a gépeket. A 70-es években jelent meg a mikroprocesszor, amellyel az évtized végére még kisebb méretű, asztalra helyezhető gépek építése vált lehetségessé. Ezt a típust, amelyben kezdetben az Apple cég volt az úttörő, mikroszámítógépnek nevezték el. Később, a 80-as évek elején, az IBM megjelent saját gépével ebben a kategóriában, az IBM PC-vel. Ez a kategória azután inkább a sajátosságait jobban kifejező személyi számítógép nevet kapta.

A nagygép, minigép, személyi számítógép hármass felosztás, ismét a technológia fejlődésének köszönhető teljesítménynövekedés miatt, a 80-as évek végén kezdett felborulni. A személyi számítógépek az egyre újabb processzorok – különösen az évtized második felében megjelent RISC processzorok – segítségével elérték, sőt meghaladták a fél évtizeddel korábbi minigépek teljesítményét. Ezért egyre inkább asztali (desktop) gépeknek nevezik őket. A teljesítménynövekedéssel a minigépek a korábbi nagyszámítógépeket haladták meg, lényegesen kisebb költséggel, aminek eredményeként a hagyományos értelemben vett nagyszámítógépek a 90-es években fokozatosan visszaszorulnak.

A költségek csökkenésével a vállalatoknál a számítógép a géptermekekből átköltözött az íróasztalokra. Ezek természetesen külön-külön nem tárolnak minden, a céghez tartozó adatot, mint a korábbi egyetlen nagygép, de erre nincs is szükség. Hálózatba való összekapcsolásukkal az arra jogosultak minden, más gépen tárolt adathoz hozzáférhetnek.

Az árak csökkenésével az otthonokban is egyre több számítógépet használnak, ami lehetőséget adott az általános információkat szolgáltató nyilvános hálózatok megjelenésére.

### **Hálózattípusok**

Egy hálózat által nyújtható szolgáltatások elsősorban a fizikai kapcsolatokon át elérhető adattovábbítási sebességtől függenek. Ha ehhez képest túl sokat kívánunk tőle, a várakozási idő elfogadhatatlanul hosszúvá nyúlik.

A sebességet pedig – ésszerű költségeken belül – a távolság határozza meg. Rövid távolságokon – max. néhány kilométer – megabit/sec nagyságrendű sebesség érhető el, míg nagyobb, több száz kilométeres távon csak kilobit/sec-ban mérhetjük a sebességet.



A hálózattal szemben támasztott igények a távolság szerint általában két csoportra oszthatók. Egy cég egy vagy több épületében lévő gépek összekapcsolása a néhány kilométeres kategóriába tartozik, míg a különböző részlegek, bankok, egyetemek, utazási irodák stb. közötti kapcsolat megteremtése már a száz kilométer feletti távolságot jelenti. Ennek megfelelően két fő hálózati technológia alakult ki, a rövid távút helyi hálózatnak vagy LAN-nak (Local Area Network), míg a nagyobb távút távolsági hálózatnak vagy WAN-nak (Wide Area Network) nevezik. A két kategória között az eszközök fejlődésével új kategória is megjelent. A városi hálózatnak (MAN = Metropolitan Area Network) nevezett típus a LAN-ok sebességével azokhoz hasonló szolgáltatásokat nyújt, de az áthidalható távolság az optikai szálnak köszönhetően 50-100 km.

#### Távolsági hálózatok

A távolsági hálózatok két csoportra oszthatók:

##### 1. Nyilvános hálózatok.

Előfizetési díj fejében bárki által számítógéppel elérhető nyilvános adatszolgáltató hálózatok.

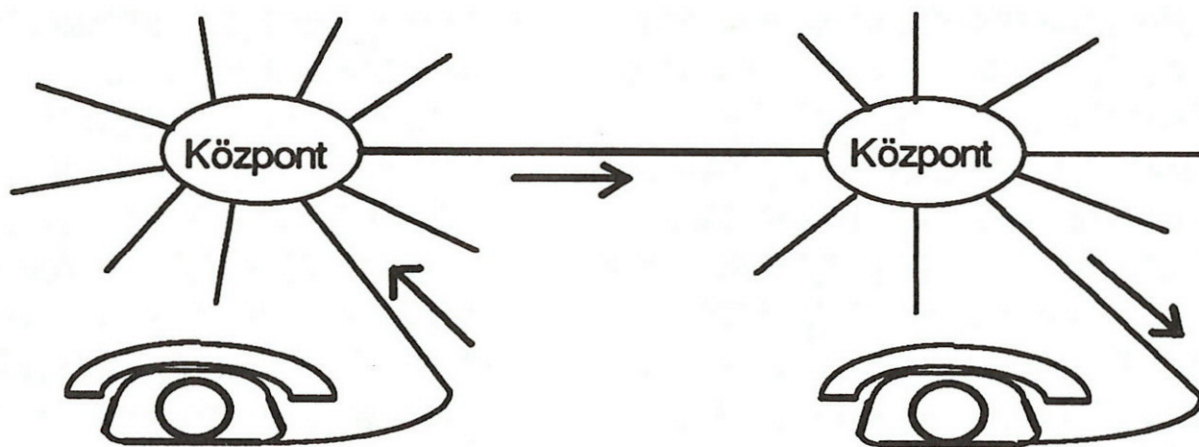
##### 2. Privát hálózatok.

Különböző cégek, bankok üzemeltetik, hogy saját, egymástól távol lévő részlegeik közötti információs rendszert hozzanak létre. A hálózathoz a tulajdonos – vagy a bérlő – vállalat alkalmazottai férhetnek hozzá. Magyarországon ilyen például az IBISZ hálózat, amely az IBUSZ irodáiban lévő számítógépeket kapcsolja össze.

Mindkét hálózattípusban felhasználhatják a nyilvános telefonhálózatot kapcsolt vagy bérelt vonalak formájában. Vannak ugyan szinte kizárólag bérelt vonalakra épülő nyilvános hálózatok, azonban a telefonvonalon elérhető kis sebesség miatt általában nem támaszkodnak csak a telefonhálózatra, hanem a saját, kifejezetten számítógépes átvitelre tervezett hálózatokra is. Ezek működésének megértéséhez gondoljuk végig, hogyan szerveződhet egy ilyen rendszer. Mivel a feladat sok szempontból hasonló, induljunk ki a telefonhálózatból (l. 28. ábra).

Készülékek csatlakoztatásához minden előfizető egy külön vezetékpárt (előfizetői hurkot) kap a hozzá legközelebb eső központból. Ha a kért állomás szintén ehhez a központhoz csatlakozik, a központ közvetlenül kapcsol. Távolabbi célállomásnál egy gyors átviteli vonalon értesíti a hívott készülék központját, és azon keresztül éri el a másik állomást. A beszélgetés ideje alatt a két készülék a vezetékeken és a központok áramkörein keresztül fizikai kapcsolatban van.

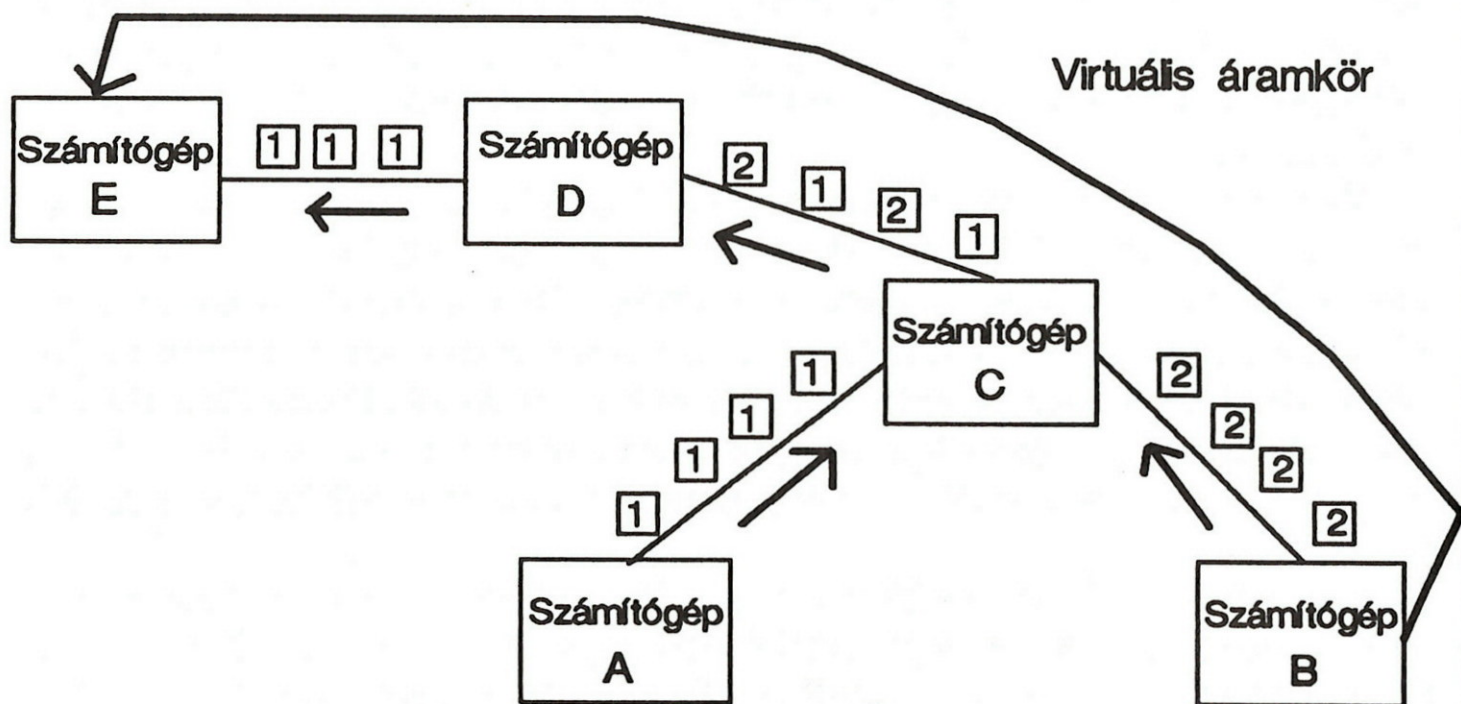




28. ábra A telefonhálózat modellje

Ezért nevezik vonalkapcsoltnak az ilyen hálózatot. A adatátvitel előtt a kapcsolatot létre kell hozni, ami, ha több központot is érint, 10-15 másodpercet is igénybe vehet, és ez a számítógép számára hosszú idő. További hátránya, hogy ha az éppen kommunikáló – küldő vagy fogadó – gépet más is hívja, foglalt jelzést kap mindaddig, míg az előző kapcsolat le nem zárult. Amennyiben a két központ közötti vonal, amely sok beszélgetés egyidejű lebonyolítására képes, „megtelik” és nem tud újabb átvitelre lehetőséget adni, szintén várnunk kell.

A hátrányok miatt a számítógép-hálózat kialakításához más eljárást használnak, amelyet csomagkapcsolásnak neveznek. Megértéséhez nézzük a 29. ábrát.

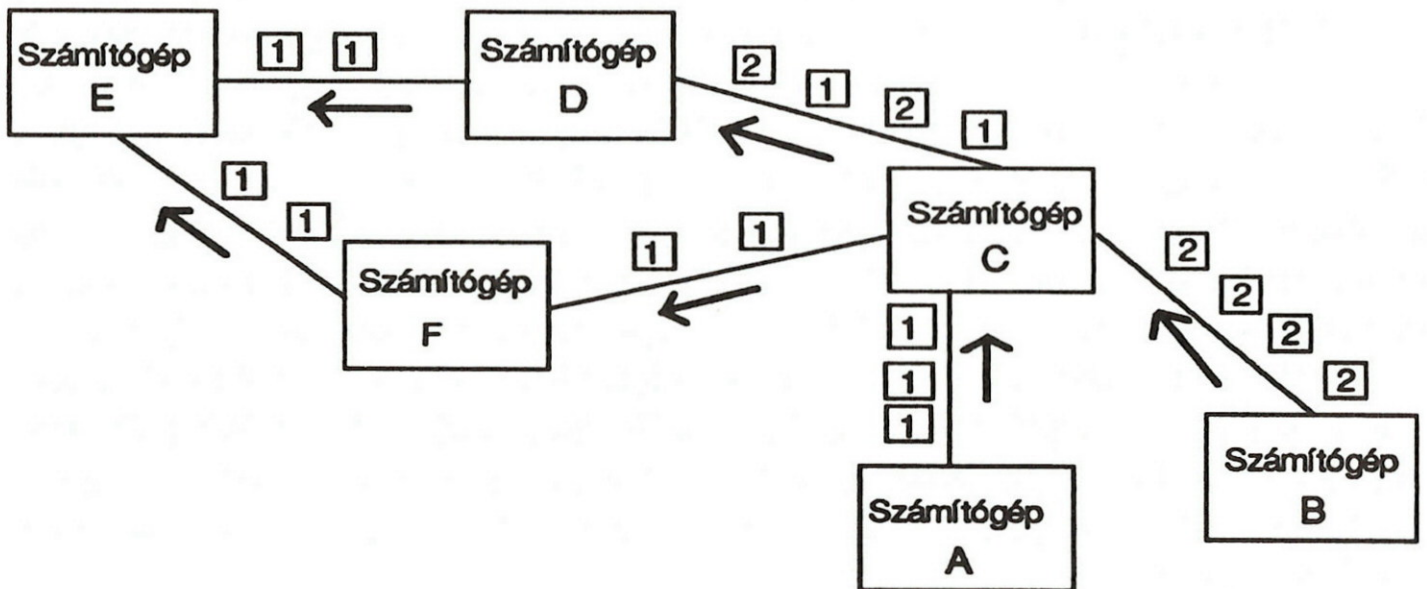


29. ábra Csomagkapcsolt hálózat modellje



Példánkban az A gép az E-vel, a B gép pedig a D jelűvel szeretne kapcsolatba lépni. A küldendő információt kisebb adagokra, csomagokra bontják, és így küldik el a C gépnek, amely váltakozva továbbítja mindkettőt. Ez azzal az előnnyel jár, hogy ha a B gép mondjuk előbb kezdte el a küldést, az A akkor is szóhoz jut, nem kap foglalt jelzést, mintha ugyanez vonalkapcsolt módon történné. A B jelű gép csak azt tapasztalhatja, hogy csomagjai lassabban jutnak el a címzetthez, mintha csak ő küldene a C-nek. Az E gép a neki szóló csomagokat feldolgozza, a D-nek szólókat pedig továbbküldi. A csomagokat a címzettek összeállítják és egy nyugtázó üzenetet küldenek a feladóknak. A nyugták ugyanezt az útvonalat járják be, csak visszafelé.

A csomagkapcsolt módszernél tehát minden gép csak a szomszédaival tart fizikai kapcsolatot, továbbítandó adatait ezeknek adja át. A továbbítónak – esetünkben a C jelű gép – tárolnia is kell a csomagokat, ha a célgép felé menő vonal túlterhelt, ezért nevezik tárolva továbbításnak is az eljárást. Ha a csomagok méretét korlátozzuk – terjedelmes adataival senki ne foglalhassa le sokáig a vonalat –, az összeköttetés párbeszédese, tehát az ember számára elég gyors kapcsolatot tesz lehetővé, még nagy távolságokon is. A csomagoknak az adatokon kívül nyilvánvalóan tartalmazniuk kell a címzett és a küldő állomás azonosítóját – ez egy szám és/vagy egy név –, hibaellenőrző és egyéb információkat. Ezek részben az adatok elé, részben mögé kerülnek, mintegy keretszerűen kerülveszik, ezért gyakran kereteknek hívják őket. Amennyiben minden csomag ugyanazt az útvonalat járja végig, ezt a fizikai áramkör mintájára virtuális áramkörnek nevezik.



30. ábra Virtuális áramkör nélküli kapcsolat sémája

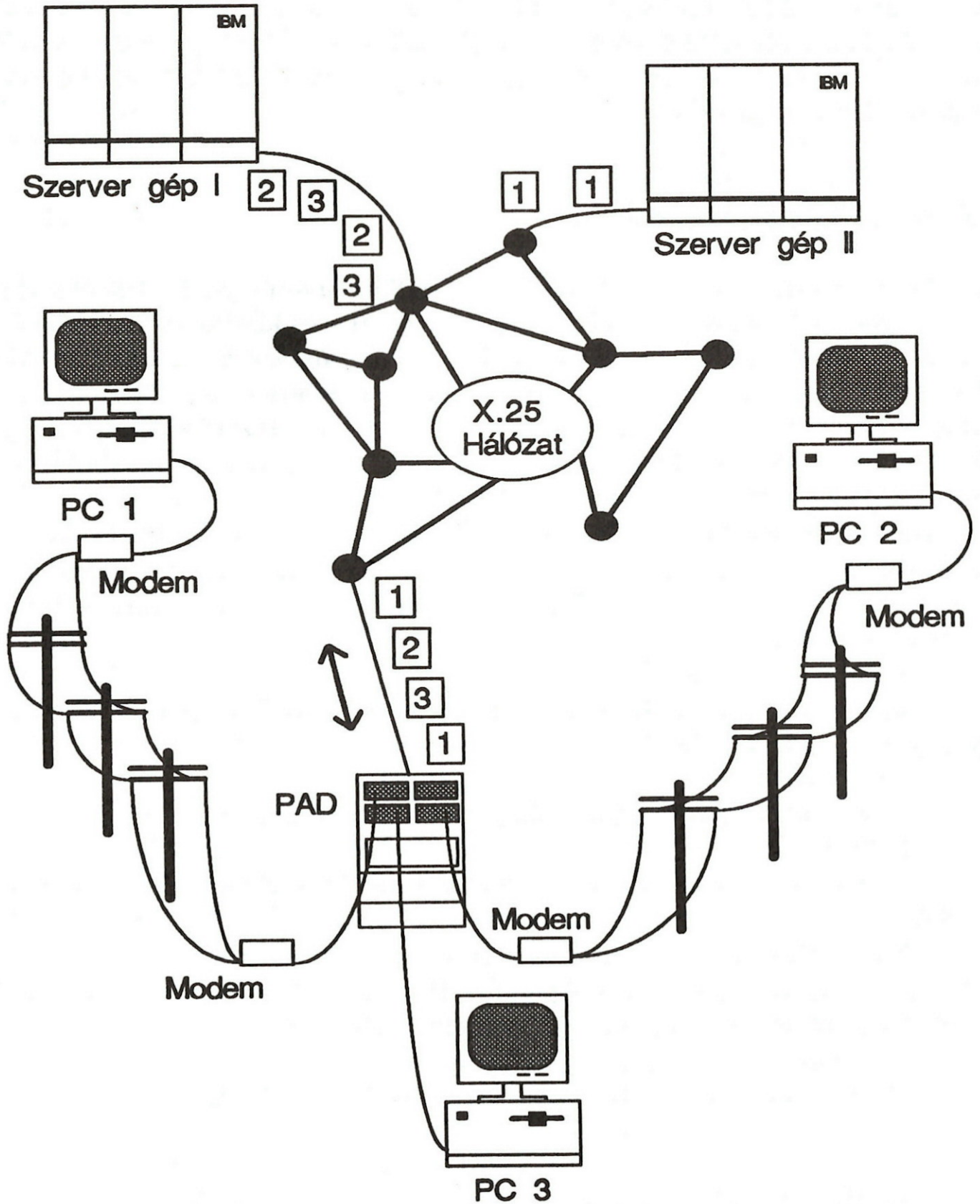


A 30. ábrán látható elrendezésben a C állomás a D gép felé menő vonal túlterheltsége miatt a csomagok egy részét más útvonalon továbbítja, ezért nem rendelhető virtuális áramkör a kapcsolathoz. Ez javítja a hálózat kihasználtságát, de azzal a következménnyel jár, hogy a csomagok az eltérő útvonal miatt esetleg nem a feladás sorrendjében érkeznek meg. A célállomáson kell a helyes sorrendben összeállítani őket. A két, hálózaton át kapcsolatot tartó gépnek nem kell ismernie a csomagok útvonalát, számukra úgy tűnik, mintha csak ketten vennének részt a kapcsolatban. Ezért nevezik pont-pont összeköttetésnek. A csomagkapcsolt eljárás legfontosabb jellemzőit a CCITT X.25 szabvány rögzíti. A korábban említett V sorozat a telefonhálózatokra, az X sorozat a közhasználatú adathálózatokra vonatkozik. A csomagkapcsolást használja minden nyilvános távolsági hálózat. A magánhálózatok megoldásai elég változatosak, a körülményektől függően. Egy nyilvános távolsági hálózathoz való csatlakozásra mutat példát a 31. ábra.

Ahhoz, hogy egy nyilvános, X.25 protokoll szerint működő hálózathoz hozzáférhessünk például egy IBM PC-vel, szükségünk van egy hozzáférési pontra, amely tulajdonképpen a hálózat egyik állomása. Két okból van rá szükség, egyrészt PC-nk nem része a hálózatnak, másrészt nincs az X.25-nek megfelelő interfésze. A hozzáférést biztosító eszközt, amely maga is egy számítógép, PAD-nek (Packet Assembler Disassembler – csomagösszeállító-szétbontó) nevezik. Feladata, hogy a hálózat szolgáltatásait igénybe vevő gépek soros, aszinkron módon érkező adataiból összeállítsa a csomagokat és szinkron módon továbbítsa a hálózatra, a válaszokat pedig soros, aszinkron adatként küldje vissza. A 31. ábrán a PC1 és PC2 gépek a telefonhálózaton át felhívják a PAD-et. Adataikat az RS-232 csatlakozón továbbítják a modemnek, amely analóg jelként küldi át a telefonvonalon. A túloldalon szintén egy modem alakítja vissza digitális jellé és adja át a PAD-nek. A PC3-as gép elég közel – egy épületben – van a PAD-hez, így nincs szüksége modemre, az RS-232-es interfészével közvetlenül csatlakozik hozzá. A PAD az egyes gépek adataiból összeállított csomagokat továbbítja a hálózaton. A PC1 által kért adatokat a II-es szolgáltató – vagy szerver – gép tárolja, míg a PC2 és PC3 igénye az I-es szerverhez fut be. A szerverek a hálózaton át a PAD-hez továbbítják a kért információt. Ez szétbontja és rendezi a csomagokat és a megfelelő PC-hez továbbítja. A PC előtt ülő felhasználó nem tudja, hogy az adatai milyen útvonalon mentek végig a hálózaton, sőt esetleg még azt sem, hogy melyik városban van az a szerver, amellyel kapcsolatba lépett.

Erre nincs is szüksége, csak azt a telefonszámot kell ismernie, amelyen felhívhatja a PAD-et. A legtöbb nyilvános hálózatot előfizetési díj fejében használhatjuk csak, ezért kapunk egy azonosítót, amit a bejelentkezéskor meg kell ad-





31. ábra Hozzáférés egy nyilvános hálózathoz



nunk. Ennek hiányában nem áll szóba velünk. A PAD feladatait a CCITT X.3-as szabványa, a soros aszinkron interfész (RS-232) és a PAD kapcsolatát az X.28, végül a PAD és a csomagkapcsolt hálózat vagy egy másik PAD kapcsolatát az X.29 szabvány rögzíti.

### **Nyilvános hálózati szolgáltatások**

A számtalan nyilvános hálózat közül a CompuServe-t mutatjuk be, mivel szolgáltatásai a tervek szerint 1993 júniusától budapesti – és később vidéki – telefonszámokon is elérhetőek lesznek. A hálózathoz az amerikai számokat tárcsázva korábban is hozzá lehetett férni itthonról, de a távhívás magas költsége és az előfizetés nehézségei miatt kevesen használhatták. A hálózat üzletpolitikáját jellemzi, hogy sokféle modem vásárlásakor egy hónapi ingyenes használati lehetőséget kapunk, csak a telefonszámlát kell fizetni.

A hazai üzemeltetők egyelőre egyszerűen átveszik az amerikai információkat, később, bizonyos számú előfizetés elérése után tervezik magyar vonatkozású adatokat tartalmazó szekció indítását. A témák sokrétűségét csak címszavakban mutathatjuk be.

#### **1. Hírek, sport, időjárás.**

Az Associated Press óránként frissített hírszolgálat. Sport, üzleti hírek, időjárásjelentés (az USA-ban).

#### **2. Üzleti információk.**

Tőzsdei, értékpapír és valutaárfolyamok. Elemzések, előrejelzések.

#### **3. Kidványok.**

A 21 kötetes, negyedévenként frissített Gorlier Enciklopédia és több száz szakújság.

#### **4. Elektronikus posta, minden előfizetőnek.**

5. Utazási információk, repülőjegy és szállodai szobafoglalás, autókölcsönzés.

6. Számítógépes fórum, nagyon sokféle témában.

#### **7. Szórakozás és játékok.**

Mozi- és videofilm-ismertetések, számítógépes játékok.

### **Magánhálózati szolgáltatások**

Ma Magyarországon az üzleti életben csak az IBUSZ üzemeltet országos kiterjedésű, saját, X.25 alapú hálózatot, az IBISZ-t (IBUSZ Országos Információs Rendszer). Az IBUSZ és az SzKI együttműködésében kidolgozott rendszer 1991



júniusa óta működik. Az IBISZ közel 100 irodában lévő 400 PC-t kapcsol össze. Ezek nemcsak egyedi gépek, az irodákban kb. 65 helyi hálózatot is használnak, így az IBISZ vegyes (távolsági és helyi hálózatot is magában foglaló) rendszernek tekinthető. Tetszőleges irányú adatátvitelt biztosít az adatbázisok hozzáférésehez és karbantartásához, amelyek hierarchikusan épülnek fel (ORACLE). A rendszer mind az utazási, mind a banki tevékenységhez szükséges gyors hozzáféréseket támogatja.

#### Helyi hálózatok

Mint láttuk, a távolsági hálózatokat általában információk, adatfájlok elérésére használják. A helyi (LAN) hálózatokon a lényegesen nagyobb átviteli sebesség a szolgáltatások sokkal szélesebb választékának megvalósítását teszi lehetővé. A helyi – vagy lokális – hálózat alkalmazásával elérhető előnyök közül a leggyakrabban előfordulókat tekintjük át.

Az első ilyen cél a hardvereszközök jobb, hatékonyabb kihasználása. Egy irodában például, ha mindenki nagyjából ugyanazokat a programokat használja a saját gépén, akkor ezek minden gépen helyet foglalnak a tárolókon. Hálózattal összekapcsolva őket, elég egy gépen egy példányban tárolni a szoftvereket, ha valaki használni akarja valamelyiket, egyszerűen elkéri a hálózaton át. Ez csak néhány másodperc többletidőt jelent a saját gépen való tároláshoz képest. Hasonló módon, a hálózatba kapcsolt géphez csatlakoztatott nyomtatót is mindenki használhatja a saját gépéről. A hardvereszközök jobb kihasználása volt korábban a helyi hálózatok terjedésének egyik fő oka.

Az eszközök árának drasztikus csökkenésével később más előnyök kerültek előtérbe. A hálózat jó alapot teremt a kollektív munkára, mivel a felhasználók egymás gépeihez is hozzáférhetnek. Az egy központi raktárral, de több eladóhellyel működő kereskedő a hálózattal egyszerűen elkerülheti annak a veszélyét, hogy a raktárban lévő árut egyidejűleg többször is eladják. Az adatok biztonságos tárolásában is segít a hálózat, a fontos információkat párhuzamosan egyszerre két tárolón őrzik, így az egyik meghibásodása esetén a másik átveszi a funkcióját, és ezt csak a hálózatot kezelő személy veszi észre, a felhasználók nem. A hálózat lehetőséget ad az adattitkosítás megoldására. A felhasználók különböző hozzáférési jogokkal ruházhatók fel, a hálózatot használók közül nem tekinthet be mindenki bármelyik fájlba, csak ha engedélye van rá. Ugyanígy korlátozható az írás (adatfelvitel) és a programvégrehajtás joga is.

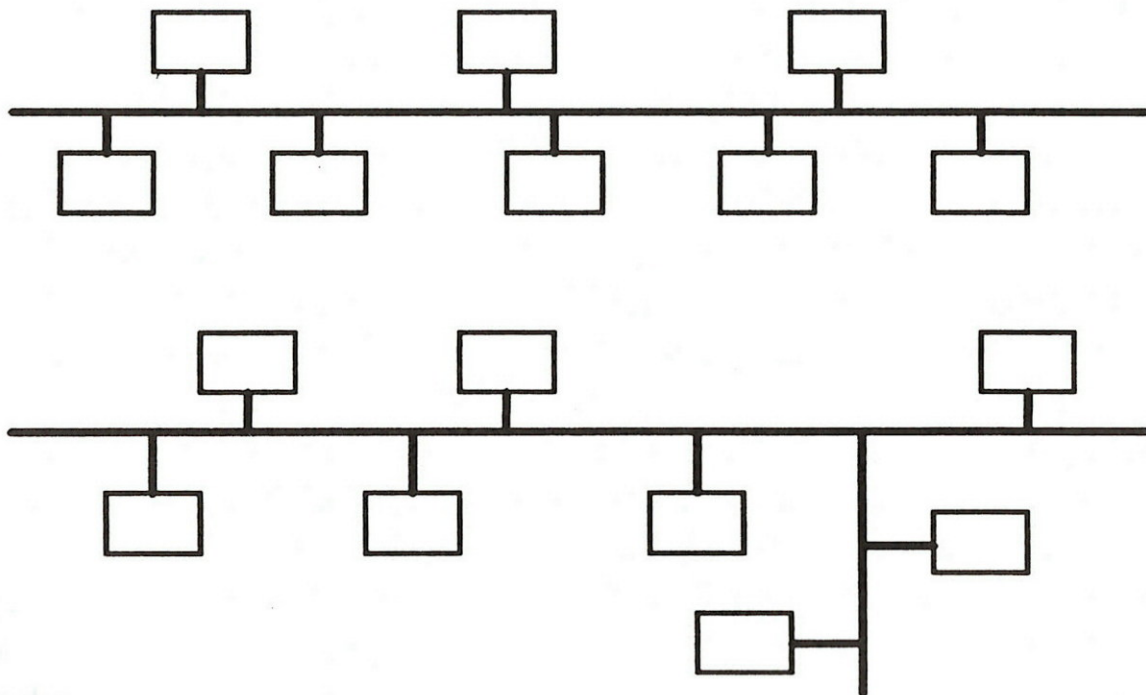
Mindezek természetesen nem gépek pusztá összekapcsolásával válnak elérhetővé, a szolgáltatások elsősorban a hálózatot kezelő operációs rendszertől függenek.



## Helyi hálózatok topológiái

A helyi hálózatok nemcsak az átviteli sebességben különböznek a távolsági hálózatoktól, működésük alapelvei is eltérők. A távolsági hálózatoknál az egyes állomások összekapcsolása sokféleképpen történhet, a forgalom nagyságától és fő irányaitól függően (l. 31. ábra). A helyi hálózatokban sokkal szigorúbban kötött az állomások elhelyezkedése egymáshoz képest, a hálózat topológiája csak háromféle lehet. Ennek oka az állomások eltérő kommunikációja. A távolsági hálózatoknál minden állomás csak a szomszédaival tart fenn fizikai összeköttetést, és a kapott csomagok közül a nem neki szólókat – esetleg átmeneti tárolás után – csak továbbítja. Így a hálózat különböző ágaiban eltérő forgalom zajlik, és a két kapcsolatot tartó állomás pont-pont összeköttetésben van – noha adataik több közöttük lévő állomáson haladnak át, amelyek egyszerűen csak kézbesítő szerepet játszanak.

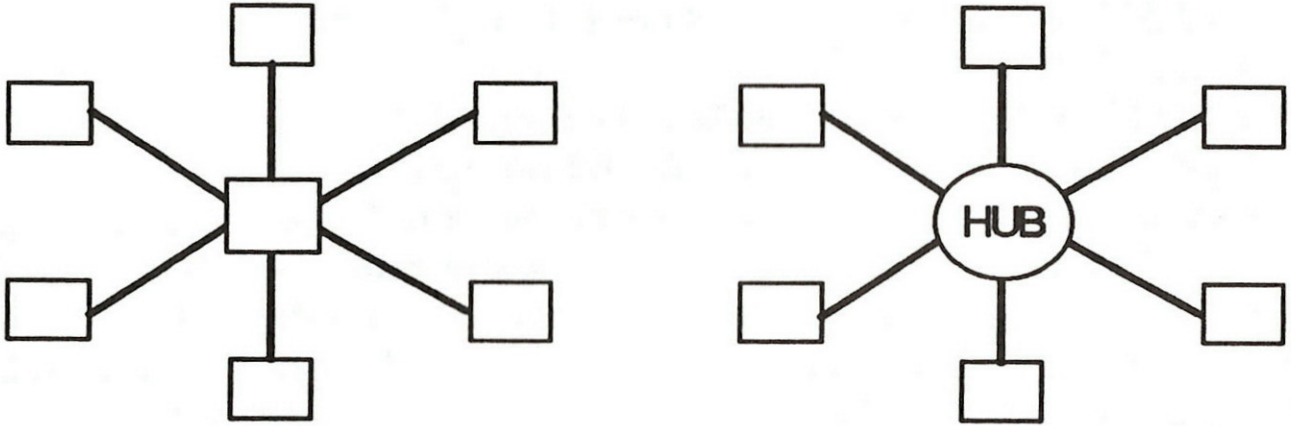
A helyi hálózatoknál – egy esettől eltekintve – az átvivő közegre, a vezetékre valamennyi résztvevő egyszerre csatlakozik. Az éppen adatot küldő állomás jeleit valamennyi gép „hallja”, ezért a pont-pont összeköttetéssel szemben gyakran üzenetszórásosnak is nevezik. Egyszerre tehát csak egy gép továbbíthat a hálózaton át, ellentétben a távolsági hálózattal. A fizikai működés egyik fő kérdése annak a szabályozása, hogy az egyes állomások hogyan, milyen sorrendben kapják meg a közös vezeték használatának a jogát. Az erre vonatkozó szabályokat közeg-hozzáférési protokollnak nevezik. Ezek a protokollok szorosan kötődnek a fizikai elrendezéshez, ezért kell a három alaptopológiával megismerkednünk.



32. ábra A busz- és a fa-topológia



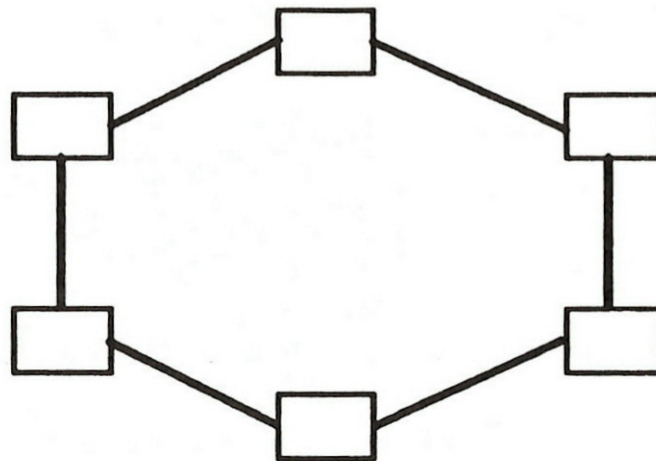
A leggyakrabban használt elrendezés látható a 32. ábrán, amikor valamennyi gépet egyetlen vezeték köt össze, ez a busz-topológia. Lehetnek leágazásai, ekkor szokás fa-topológiának nevezni. A csillag alakú topológiát mutatja a 33. ábra. A csillag központja lehet egy számítógép, amely ekkor nyilván kitüntetett helyzetű.



33. ábra Csillag-topológia

A többiek csak rajta keresztül kommunikálhatnak, és nem a vezetéken, hanem a központi gépen osztoznak. A csillag elrendezésben lehet egy HUB-nak nevezett kapcsolóegység is középen, amely egyszerűen összeköti a vezetékeket. Ekkor tulajdonságai a busz-topológiához hasonlóak.

A gyűrű-topológia látható a 34. ábrán. A küldő adatai a többi gépen át érik el a célállomást.



34. ábra Gyűrű-topológia



## Hálózati kábelek

A helyi (LAN) hálózat sebességét alapvetően befolyásolja a használt kábelek típusa és minősége. A kábelezés a hálózat kialakításának meglehetősen költséges része, később egy más kábelre való esetleges áttérés gyakorlatilag a hálózat újraépítését jelenti.

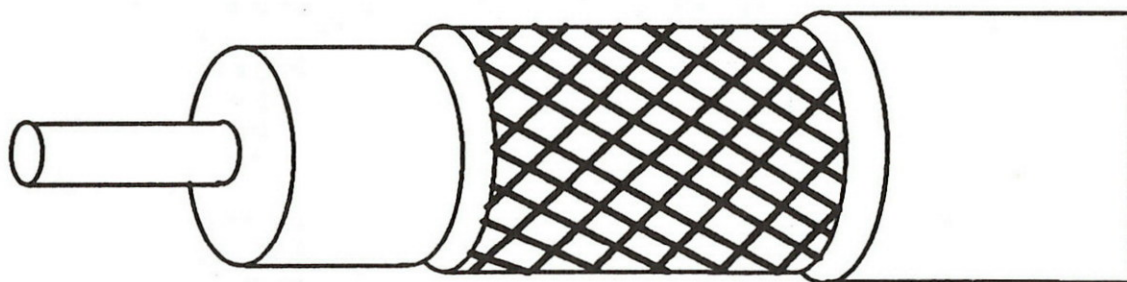
Ma a LAN hálózatokban a következő kábelfajták használatosak:

1. Sodrott érpár
2. Koaxiális kábel, alapsávú és szélessávú változatban
3. Optikai szál, egymódusú és többmódusú típusai

A sodrott érpár két, műanyag szigetelővel bevont, általában 1 mm vastag rézhuzal, amelyeket összesodornak. Ez a külső zavarok hatását csökkenti, mivel két párhuzamosan futó huzal antennaként viselkedik, a sodrott érpár azonban nem. A használt kábelek közül a legolcsóbb, de a legalacsonyabb sebességet biztosító típus. A csatlakozók bekötése, a kábel esetleges meghosszabbítása egyszerű. Sodrott érpárt használnak a telefonok bekötéséhez is. Ezt a vezetéktypust sokszor már az építkezés idején elhelyezik az irodaházakban, ami csökkenti a későbbi hálózatkiállítás költségeit.

A koaxiális kábel egy vezetőhuzalból és a tőle műanyag szigeteléssel elválasztott sűrűn font árnyékoló harisnyából áll. Kívülről szintén műanyag védőréteg borítja. Az árnyékolásnak köszönhetően a külső zavarokkal szemben viszonylag érzéketlen. Bár a sodrott érpárnál drágább, nagyobb sebességű és biztonságosabb működést tesz lehetővé, ezért ma a hálózatok kialakításánál ezt használják a leggyakrabban.

rézhuzal      szigetelés      árnyékoló vezető      külső szigetelés



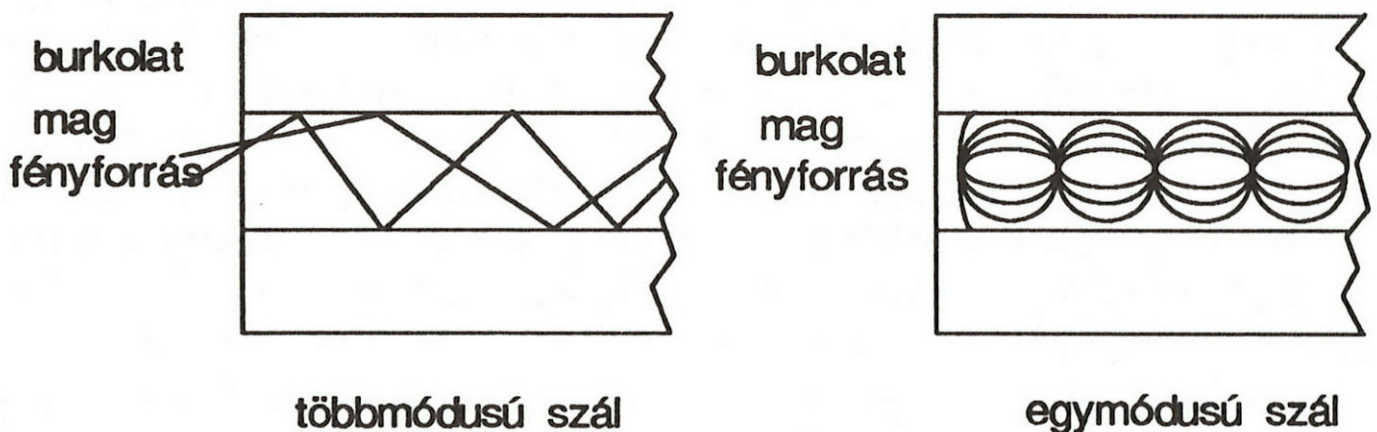
35. ábra Koaxiális kábel

Két változata van, az alapsávú, amely 50 ohmos és a szélessávú, amely 75 ohmos. Az alapsávú működésnél a kábelben a biteket közvetlenül 0 és 5 voltos feszültségszintekkel továbbítjuk, míg a szélessávú kábelben egy nagyfrekvenciás vivőjelet használunk, ennek a modulációja hordozza a biteket. Az alapsávú mű-



ködésnél egyszerre csak egyetlen adat továbbítható – szaknyelven egycsatornás –, míg a szélessávú esetben különböző frekvenciájú vivőjeleket használva egyidejűleg több adat haladhat a vezetéken. Ennek ára azonban a lényegesen nagyobb költség, ráadásul a szélessávú esetben a vezeték csak egy irányban továbbíthat, a kétirányú átvitelhez két kábel kell. A szélessávú átvitelhez nagyon hasonló a kábeltelevíziós rendszerek átvitele. A szélessávú kábelen egy vivőjellel – azaz egy csatornán – továbbítható adatok sebessége lényegesen kisebb, mint az alapsávú kábelé. Ezért és a kisebb költség miatt az alapsávú hálózatok sokkal elterjedtebbek.

Az optikai kábel egy nagyon tiszta kvarcüvegből vagy műanyagból igen vékonyra húzott szál. Nem elektromos jeleket, hanem fényimpulzusokat továbbít nagyon gyorsan. Vezető magból és a külső védőburokból áll. A többmódusú szálnál a külső burkolatról verődik vissza a magban haladó fény – (l. 36. ábra) –, így a különböző fényimpulzusok különböző utat futnak be a szálban, innen a többmódusú elnevezés. A többmódusú szál magjának átmérője 50-100 mikron (a milliméter ezredrésze), ha a szál átmérőjét a fény hullámhosszára – 5-10 mikron – csökkentjük, a fény folyamatosan a magban halad, nem verődik vissza.



36. ábra Optikai kábelek

Ez kisebb veszteséget jelent, az egymódusú szál nagyobb távolságot hidalhat át a fény erősítése nélkül. Fényforrásként LED diódát – amit a különböző elektromos készülékek kis színes jelzőlámpájaként mindenki látott már – vagy pedig lézerefényt kibocsátó lézervedióát használnak. Ilyen lézervedióát alkalmaznak a CD lejátszóknak vagy a lézerprinterekben is. A LED olcsóbb, de gyengébb fényt ad. Az optikai szálakat a telefonhálózatban kezdték alkalmazni a központok összekapcsolására (l. 28. ábra). A nagy sebesség mellett fontos szempont, hogy a szál átmérője csak töredéke a hagyományos fémkábelének, így azonos körülmények között több kapcsolat kialakítását teszi lehetővé. A telefonhálózatban



a nagy távolságok miatt egymódusú szálakat és lézerefényt használnak. A műanyag szálak egyszerűbben kezelhetők, de jobban gyengítik a fényt, ezért kisebb távolságon használhatóak erősítés nélkül.

Az árak lassan csökkennek ugyan, de LAN hálózatokban, mivel a távolság kisebb, mint néhány kilométer, inkább az olcsóbb, többmódusú szálakat és a LED diódákat alkalmazzák. Az optikai hálózat költségei azonban még így is magasak ahhoz, hogy közvetlen gépösszekapcsolásra alkalmazzák, ehelyett inkább több LAN hálózat egyesítésére, úgynevezett gerincvezeték kialakítására használják. A költségeket az is növeli, hogy a kábelen a jelek útja nyilvánvalóan egyirányú, a szál egyik végén a fényforrás, a másikon az érzékelő helyezkedik el. Kétirányú átvitelhez két külön szál szükséges.

Mindezek ellenére az optikai szál rohamosan terjed, számos előnyének köszönhetően, amelyeket a 4. táblázat foglal össze. A táblázat adatai az elméletileg elérhető maximális értékeket mutatják, a konkrét gyakorlati megvalósítások ennél sokszor gyengébb paraméterekkel rendelkeznek. A jellemzők erősen függenek a kábel hosszától, a táblázat 1 km-es szakaszra érvényes értékeket mutat, amelyek rövidebb távon javulnak, hosszabb távon romlanak.

A tesztek szerint a fémkábeleken átlagosan minden 1 000 000-modik bit érkezik hibásan, az optikai szál hibája ennek 1000-ed része. Az adatbiztonság sokszor az egyik legfontosabb oka az optikai kábelek alkalmazásának. A fémvezetőkön ilyen sebességek esetén az adatok gyakorlatilag rádióhullámként terjednek, és energiájuk egy részét antennaként kisugározzák. Ez sokféle módon csökkenthető, de a hálózat viszonylag egyszerűen lehallgatható. Az optikai kábelnek nincs kisugárzása, és észrevétlen megcsapolása gyakorlatilag lehetetlen. Mivel nem elektromos jelekkel dolgozik, biztonságosan használható robbanásveszélyes vagy villámcsapásnak, erős elektromos zavarásnak kitett környezetben.

Jellemző	Sodrott érpár	Koaxiális kábel	Optikai szál
Sebesség (megabit/sec)	16	500	1000 felett
Jelkisugárzás	igen	igen	nem
Hibás bitek aránya	1:1 millió	1:1 millió	1:1000 millió
Lehallgathatóság	könnyű	könnyű	nehéz
Szikraveszély	igen	igen	nem

#### 4. táblázat Hálózati kábelek összehasonlítása

A különböző hálózati kábelek és az előző fejezetben áttekintett topológiák nem párosíthatóak tetszőlegesen. Például az optikai szál nehéz lehallgathatósá-



ga azt is jelenti, hogy nem tudjuk egyszerűen elágaztatni a busz- vagy fa-topológia kialakításához. Az optikai szálakat tipikusan gyűrű-topológiában használják két pont összekötésére. A topológiák és a kábelek lehetséges párosítását mutatja az 5. táblázat.

Kábelek	Busz	Fa	Gyűrű	Csillag
Sodrott érpár	*	–	*	*
Alapsávú koax.	*	*	*	*
Szélessávú koax.	*	*	–	–
Optikai szál	–	–	*	–

5. táblázat Topológiák és kábelek párosítása

### Hozzáférés a kábelhez

A helyi hálózatoknál tehát egy időben csak egy állomás használhatja a kábelt, és a tervezés egyik fő kérdése, hogy a gépek milyen módszer szerint osztoznak rajta, hogyan kapják meg a hálózathoz való hozzáférés jogát. A legegyszerűbb csillag-topológiánál, ha egy gép a központ (l. 33. ábra), nincs ilyen gond, hiszen minden állomás külön kábelen csatlakozik a központi szerver géphez. Nyilván a központi gép dönti el, hogy melyik állomással foglalkozik éppen. Választhatja azt a módszert, hogy azt az állomást szolgálja ki először, amelyik előbb jelentkezett be a többinél, és utána veszi sorra a következőket. Ezzel persze feltételeztük, hogy az állomások bármikor fordulhatnak a szerverhez. A másik megoldás, hogy az állomásoknak nincs önálló megszólalási joguk, a központi gép rendszeresen körbekérdezi őket, akarnak-e vele kommunikálni. Ez egy egyszerű hálózat, de ha a központi gép meghibásodik, az egész megbénul. Logikus lépés, hogy több szervert kapcsoljunk be, ezek különböző feladatokat láthatnak el, vagy az egyik hibája esetén a másik átveszi a feladatait. Ezzel azonban ismét ugyanazzal a problémával találkozunk, ki mikor használhatja a kábelt. Az ezt meghatározó módszereket összefoglalóan közeg-hozzáférési protokolloknak nevezik, és két fő alapgondolatukat már megismertük.

Amennyiben a közös kábelre busz-, fa- vagy egyesített csillag-topológiában csatlakoznak a gépek és önálló megszólalási joguk van, az a gép, amely üzeni akar a hálózat egy másik állomásának, figyelni kezdi a kábelt, használja-e éppen valaki a hálózatot. Ha igen, vár, amíg a folyamatban lévő művelet befejeződik. Amikor úgy tapasztalja, hogy a kábel éppen szabad, elkezdni továbbítani a saját



adatait, amelyben meg kell adnia annak a gépnek az azonosítóját – címét –, amelynek az üzenet szól, hiszen adását valamennyi gép hallja. Hogy senki se foglalhassa le a hálózatot hosszú ideig, az egyszerre elküldhető adatok hossza korlátozott, még ha maradt is továbbítandó információja, az állomásnak be kell fejeznie, ha elérte ezt a korlátot, és a többit majd csak a következő alkalommal továbbíthatja. A címzett állomás egy rövid nyugtázó üzenettel válaszol.

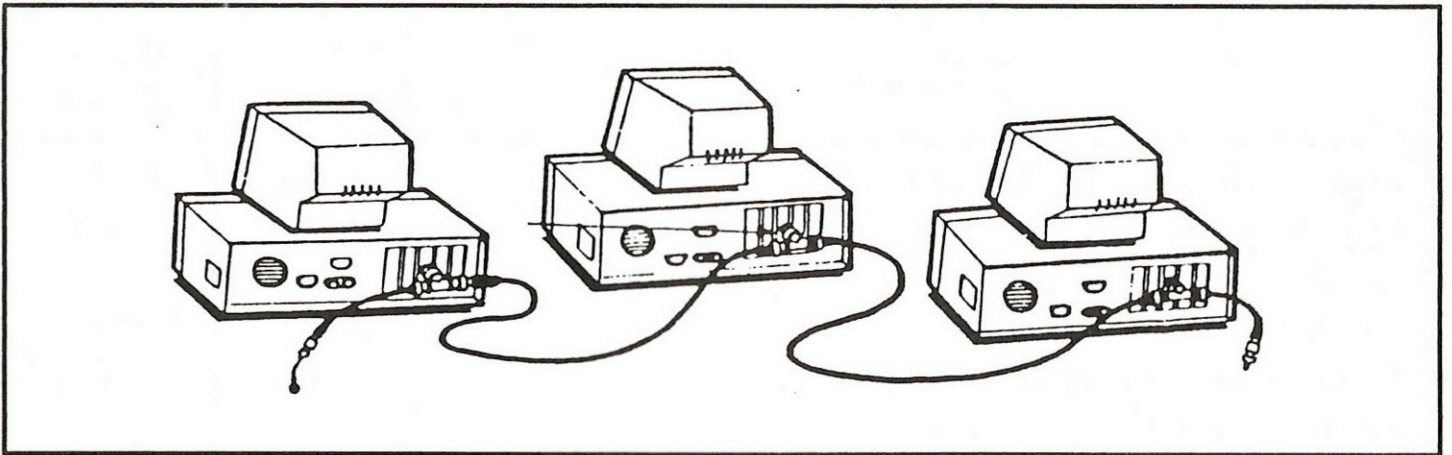
A küldő ekkor újra figyelni kezdi a kábelt, dolgozik-e valaki. Ha nem, elküldi az információk következő adagját. Ez a módszer tehát az elsőnek megszólaló számára ad használati jogot.

Mi történik azonban akkor, ha egyszerre két gép figyeli a kábelt, és úgy tapasztalva, hogy az éppen szabad, egyszerre kezdi el a továbbítást? Ekkor az üzeneteik összekeverednek és jóvátehetetlenül tönkremennek. Az állomások azonban adásuk közben is figyelik a vezetékét, és miután mindketten észlelték az ütközést, azonnal abbahagyják az adást. Bizonyos ideig várnak, majd figyelés után újra próbálkoznak. Ha a szünetidő mindkét gépen azonos volna, nagy valószínűséggel a második próbálkozás is ütközéshez vezetne. Ezért az ütközések utáni kötelező várakozási időt – itt milliszekundumokról, azaz a másodperc ezredrészéről van szó – bizonyos határok között véletlenszerűen állapítják meg. A leírt módszer szerint működő protokollokat csatornafigyelő és ütközésérzékelő protokollnak vagy az angol rövidítéssel CSMA/CD-nek (Carrier Sense Multiple Access with Collision Detection) nevezik. Sok változata közül az Ethernet nevű a legismertebb és a legelterjedtebb. Ezt eredetileg a Xerox cég építette meg 1976-ban, és a sikeres kipróbálás után a Xerox, a DEC és az Intel cégek létrehozták az Ethernet szabványt, amely később az IEEE (Institute of Electrical and Engineers) szervezet által elfogadott szabvány alapja lett. Sokszor tévesen Ethernetnek hívnak minden CSMA/CD elven működő protokollt.

Az IEEE számos hálózati szabványt alkotott, a helyi hálózatokra vonatkozóak 802 sorozatszámokkal szerepelnek, az Ethernet a 802.3-as. Ez 50 ohmos koaxiális kábelen 10 Mbit/sec sebességű hálózatra ad javaslatot. Az eredeti szabvány kétféle kábel használatát támogatja. A vékony Ethernet olcsóbb – ezért néha Cheapernet-nek is nevezik –, de kisebb távolságú állomásokat köthet össze, míg a vastag Ethernet drágább, de nagyobb távolságon használható.

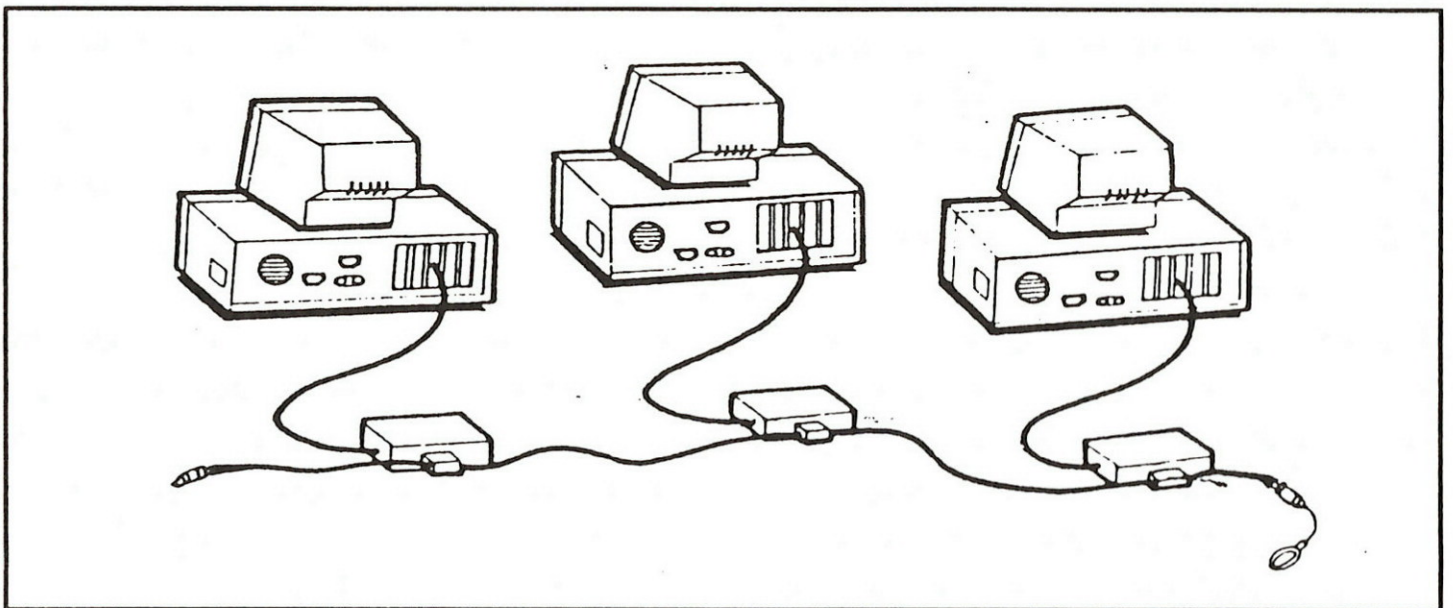
A hálózatkezelő áramköröket, amelyek ahhoz szükségesek, hogy a számítógép csatlakozhasson a hálózatra, néha eleve beépítik a gépbe, ilyenek a már említett munkaállomások. Sokszor azonban – és ez az IBM PC-knél is így van – külön kell megvásárolni egy adapterkártyát, amelyet a gépbe kell helyezni. A vastag és a vékony Ethernet abban is különbözik egymástól, hogy míg a vékony kábel közvetlenül az adapterkártyákra csatlakozik – l. 37. ábra –, a vastag Ether-





**37. ábra** A vékony Ethernet kábel bekötése

netnél szükség van egy adó-vevőre, erre csatlakoznak egy külön – nem koaxiális – kábellel a gépek, ahogy a 38. ábra mutatja.



**38. ábra** A vastag Ethernet kábel bekötése

A kétféle kábellel kialakítható hálózat adatait foglalja össze a 6. táblázat. Szegmensnek nevezik a hálózatban a két szomszédos gép közötti kábelszakaszt. Hosszú kábeleknél szükség van a jelek erősítésére, amelyet az ismétlőnek (repeater) nevezett eszköz végez, nem vizsgálja a kapott adatokat, csak megismételve továbbítja őket.

A vastag Ethernet kábelen a csatlakozási pontok csak 2.5 méterre vagy ennek többszörösére lehetnek egymástól. A csatlakoztatható állomások számának elméleti maximuma 1024. Az Ethernet hálózat szabványa koaxiális kábelt ír elő, de az olcsóbb megvalósítás miatt használják sodrott érpáron is, ennek a specifi-



Jellemzők	Vékony kábel	Vastag kábel
Egy szegmens max. hossza	185 m	500 m
Egy szegmensre csatlakozó állomások max. száma	30	100
A hálózat teljes max. hossza 5 szegmens 4 ismétlővel	925 m	2500 m
Minimális távolság két állomás között	0,5 m	2,5 m
Adó-vevő kábel max. hossza	–	50 m

### 6. táblázat A vékony és vastag Ethernet adatai

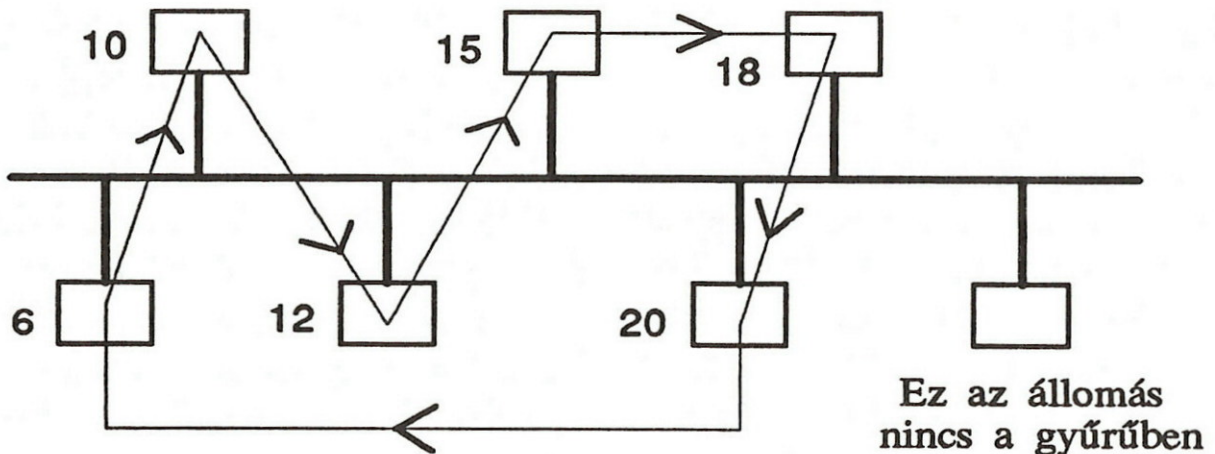
kációja a 10BASE-T. Sebessége szintén 10 Mbit/sec. A különböző hozzáférési protokollok között a CSMA/CD előnye, hogy nem igényel külön kezelést a gépek ki- és belépése a hálózatba. Amikor bekapcsolunk egy hálózatba kötött gépet, az automatikusan elkezd figyelni a kábelt, és ha küldendő adata van, bekapcsolódik a forgalomba, illetve jelzi a többieknek a fogadókésztségét.

A hálózat hatékonysága, amíg viszonylag kevés gép dolgozik, jó, alig történik ütközés, szinte minden gép azonnal szóhoz jut. A forgalom növekedésével az ütközések száma is nő, ami a forgalom szempontjából kiesett idő. Normális terhelés alatt ütközések és a forgalmi idő aránya beáll egy állandó értékre. Azonban ha a hálózatot – például túl sok állomás bekapcsolásával – túlterheljük, az ütközések számát semmi sem korlátozza, a hatékonysága drasztikusan romlik. A CSMA/CD alaphelyzetben minden állomás számára azonos esélyt ad a vezetékhöz való hozzáférésre. Azonban ha az ütközések utáni várakozási idő tartományát egy gépen a többieknél rövidebbre állítjuk be, akkor ez a gép előnybe kerül, prioritása lesz a többiekkel szemben. Ez az előny azonban csak valószínűségi, egyáltalán nem biztos, hogy mindig előbb szerzi meg a hozzáférést. Az Ethernetnek és más ütközéses eljárásnak egyik fő hiányossága, hogy nem tudunk olyan időkorlátot megadni, amelyen belül egy állomás biztosan szóhoz jut, még a legnagyobb forgalom esetén sem. Ez az irodai alkalmazásoknál nem követelmény, de az ipari, gyártásautomatizálási feladatoknál fontos.

Ezért olyan protokollokat is kidolgoztak, amelyek ezt szavatolják. Az ütközések okozzák az említett bizonytalanságot, ezért ebben a protokollban az állomásoknak általában nincs önálló megszólalási joga. Csak akkor használhatják a kábelt, ha erre engedélyt kaptak. Az engedély egy vezérlőjel, a hálózaton át továbbított speciális adat, ezért ezt az eljárást vezérlőjeles busz-protokollnak nevezik. A vezérlőjel angol neve (token) miatt gyakran nevezik token busznak is. Az állomások busz- vagy fa-topológiába rendezettek, és a hálózat működése során



egymásnak adják az engedélyt. A topológia révén minden állomás minden forgalmat, így a vezérlőjeleket is hallja, de csak azt veszi figyelembe, ami neki szól, az ő számát tartalmazza. Ha megkapta az engedélyt, a hálózaton egy másik állomásnak címezve elküldheti az adatait, amelyeket a címzett a sikeres vétel után automatikusan nyugtáz. Az egyszerre küldhető adatok hossza itt is korlátozott, és a vezérlőjelet birtokló állomásnak egy sikeres küldés után a vezérlőjelet tovább kell adnia, még akkor is, ha maradt továbbítandó adata. A vezérlőjelet az utána következő kisebb sorszámú állomásnak címezve adja tovább. Amennyiben egy állomásnak éppen nincs küldendő adata, a kapott vezérlőjelet azonnal továbbadja a következőnek.



A vezérlőjel körbejárása a gyűrűben

39. ábra A vezérlőjeles busz

Így a vezérlőjel az állomások között körbejár, a körbejárás ideje a bekapcsolt állomások számától és a forgalomtól függ, de előbb-utóbb biztosan mindenki megkapja. A vezérlőjel továbbítási sorrendje nem azonos a gépek fizikai sorrendjével, de ez nem is fontos, hiszen a topológia miatt egyszerre hallják egymást. A vezérlőjel útját gyakran logikai gyűrűnek nevezik (l. 39. ábra).

A protokoll bonyolultságát az adja, hogy az állomásoknak ismerniük kell a gyűrűben előttük lévő és az utánuk következő szomszédjuk számát a vezérlőjel továbbításához. Ebből következően egy állomás a bekapcsolásával nem kerül be automatikusan a hálózatba – mint az ütközéses protokollnál –, amíg a többiek tudomást nem szereznek róla.

Külön eljárást igényel az állomások kiléptetése is, vagy az esetleg meghibásodott állomások kizárása a gyűrűből, például ha éppen a vezérlőjelet birtokló állomás hibásodik meg. A protokoll ezeket természetesen megoldja, de ennek további bonyolódás az ára.

A vezérlőjeles busz-protokollt az IEEE 802.4 sorszámmal fogadta el szabványként. Hatékonysága kis forgalom esetén nem túl jó. A tétlen állomások egymás-



nak adogatják a vezérlőjelet, és annak, aki küldeni akar, meg kell várnia, míg hozzá ér. A forgalom növekedésével a „felesleges” vezérlőjel-továbbítások száma csökken, a határfok javul.

A protokoll 75 ohmos koaxiális kábelt használ, több különböző gyakorlati megvalósítása létezik. Közülük a Datapoint cég által kifejlesztett ARCnet a legelterjedtebb, amely 2.5 Mbit/sec-os sebességgel dolgozik, ami nem túl sok, de a PC-kbe illesztendő hálózati adapterkártyák ebben a kategóriában a legolcsóbbak. Van busz- és csillag-topológiájú változata is.

A 34. ábrán látható gyűrű-topológiánál nyilván nem alkalmazható sem az ütközéses, sem a vezérlőjeles busz-protokoll. Alapvető eltérése a busz- vagy fa-topológiától, hogy nem üzenetszórásos, azaz az állomások nem hallják egyszerre egymást, hanem pont-pont összeköttetésben vannak, mint a távolsági hálózatoknál. A gyűrű-topológiánál leggyakrabban a vezérlőjeles gyűrű-protokollt használják a hálózathoz való hozzáférés ütemezésére. Ezt az IEEE 802.5 számmal fogadta el szabványként, és megegyezik az IBM 1985-ben kidolgozott Token Ring hálózatával. Mint a vezérlőjeles busznál, az egyes állomásoknak itt sincs önálló megszólalási joga. Az adásengedélyt jelentő vezérlőjel (token) a gyűrűben körbejár. Az adatokat továbbítani akaró állomásnak meg kell várnia, míg a vezérlőjel a gyűrűn hozzá ér. Ekkor ezt leveszi a gyűrűről és az adatokat kezdi el továbbítani, megadva a címzett állomás, valamint a saját számát is, egyéb információk (például hibaellenőrző kód) mellett. Az utána következő állomás a címből látja, hogy az nem neki szól, ezért változtatás nélkül továbbítja. Amikor a címzett állomáshoz ér, az bemásolja az adatokat a memóriájába, de nem veszi le a gyűrűről, hanem egy nyugtázójelet hozzáfűzve továbbküldi. Így az végül visszaérkezik a feladójához. A feladó megismeri a saját csomagját, és a nyugtázásból látja, hogy a címzett állomás megkapta. Ha a nyugta hiányzik, a címzett nem kapta meg az adatokat, de a feladó tudja, hogy ez nem a hálózat hibájából történt, hiszen a csomag épségben körbejárt, hanem mert a címzett állomás valószínűleg nincs bekapcsolva, vagy meghibásodott. A feladó ezután kivonja az adatokat a gyűrűről, és visszateszi a vezérlőjelet. Ezt a gyűrűben következő állomás kapja meg, amely, ha éppen nem akar kommunikálni, azonnal továbbítja a következőnek.

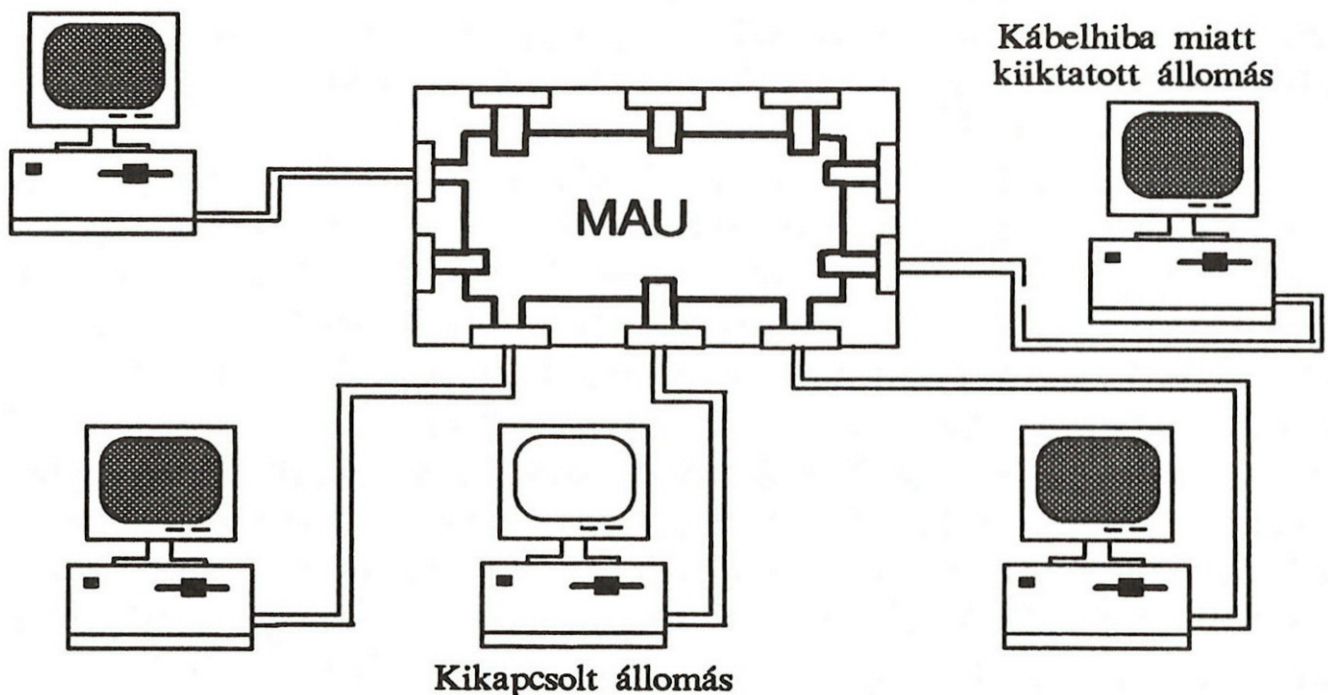
A 34. ábra csak az elrendezést szemlélteti, a gyakorlatban így nem használható, hiszen csak akkor működne, ha minden állomás be volna kapcsolva, és bármelyik meghibásodása esetén a hálózat megbénulna. Ezért a valóságban az állomások nem közvetlenül csatlakoznak egymáshoz, hanem egy kábelközponton keresztül, amelyet az IBM MAU-nak (Media Access Unit) nevez. A MAU, mint a 40. ábra mutatja, minden állomás számára biztosít egy csatlakozót, amely két vonalat tartalmaz, az egyiket az állomás felé, a másikon az állomástól haladnak



az adatok. Így a hálózat kívülről nézve csillag alakú, amely a MAU-n belül kapcsolódik gyűrűvé.

A MAU alkalmazásának számos előnye van. Új állomás hálózatba kötésekor csak egyetlen kábelt kell a kábelközponttól a számítógéphez vinni, míg ha a gépeket közvetlenül kötnénk össze, két vezeték kellene. A MAU csatlakozójának elektronikája képes érzékelni, hogy a csatlakoztatott gép kikapcsolt vagy bekapcsolt állapotban van, és ennek megfelelően kapcsolja, vagy a csatlakozó összezárásával kiiktatja a gépet a gyűrűből. Ezzel egyszerűen megoldja a gépek hálózatba való ki- és beléptetésének gondját, amelyre a vezérlőjeles busznál egy bonyolult eljárást kell alkalmazni.

Bár a gyűrűben az állomások egyenrangúak, szükség van egy vezetőállomásra, amely felügyeli a gyűrű forgalmát, és beavatkozik, ha szükséges. Ilyen eset például, ha a vezérlőjel elvész, mert valamelyik állomás a hibája miatt „lenyelte”, nem adta tovább. Ekkor a vezetőállomás egy új vezérlőjelet bocsát ki. Az is előfordulhat, hogy az éppen aktív állomás az adatok elküldése után hibásodik meg, és a körbeért csomagját már nem veszi le a gyűrűről. Ez keringve akadályozza a forgalmat. A vezetőállomás észreveszi, hogy ugyanaz a csomag már többször körbejárt, leveszi a gyűrűről és kiadja a vezérlőjelet.



40. ábra Token Ring hálózat

Megtörténhet, hogy egy állomás úgy hibásodik meg, hogy azért a hálózatra kapcsolódását megtartja, azaz a MAU-ban a csatlakozó elektronikája nem érzékeli a hibát. Ekkor a vezetőállomás észreveszi a hálózat elakadását, hiszen hiába adja ki újra és újra a vezérlőjelet, az mindig elvész. Ha a MAU elég intelligens



– például maga is egy számítógép –, és össze van kapcsolva a vezetőállomással, akkor az teljesen automatikusan végrehajt egy diagnosztikaprogramot. Ebben arra utasítja a MAU-t, hogy sorra zárja rövidre az egyes gépek csatlakozóját, miközben ő újra és újra kiadva a vezérlőjelet próbálja elindítani a hálózat forgalmát. Amelyik gép csatlakozójának zárása után ez sikerül, az a hibás, vagy a csatlakozó kábele sérült. Bár a hálózat forgalma átmenetileg elakad, tulajdonképpen emberi beavatkozás nélkül lehet ismét elindítani, a rossz gépet, illetve kábelt kivéve.

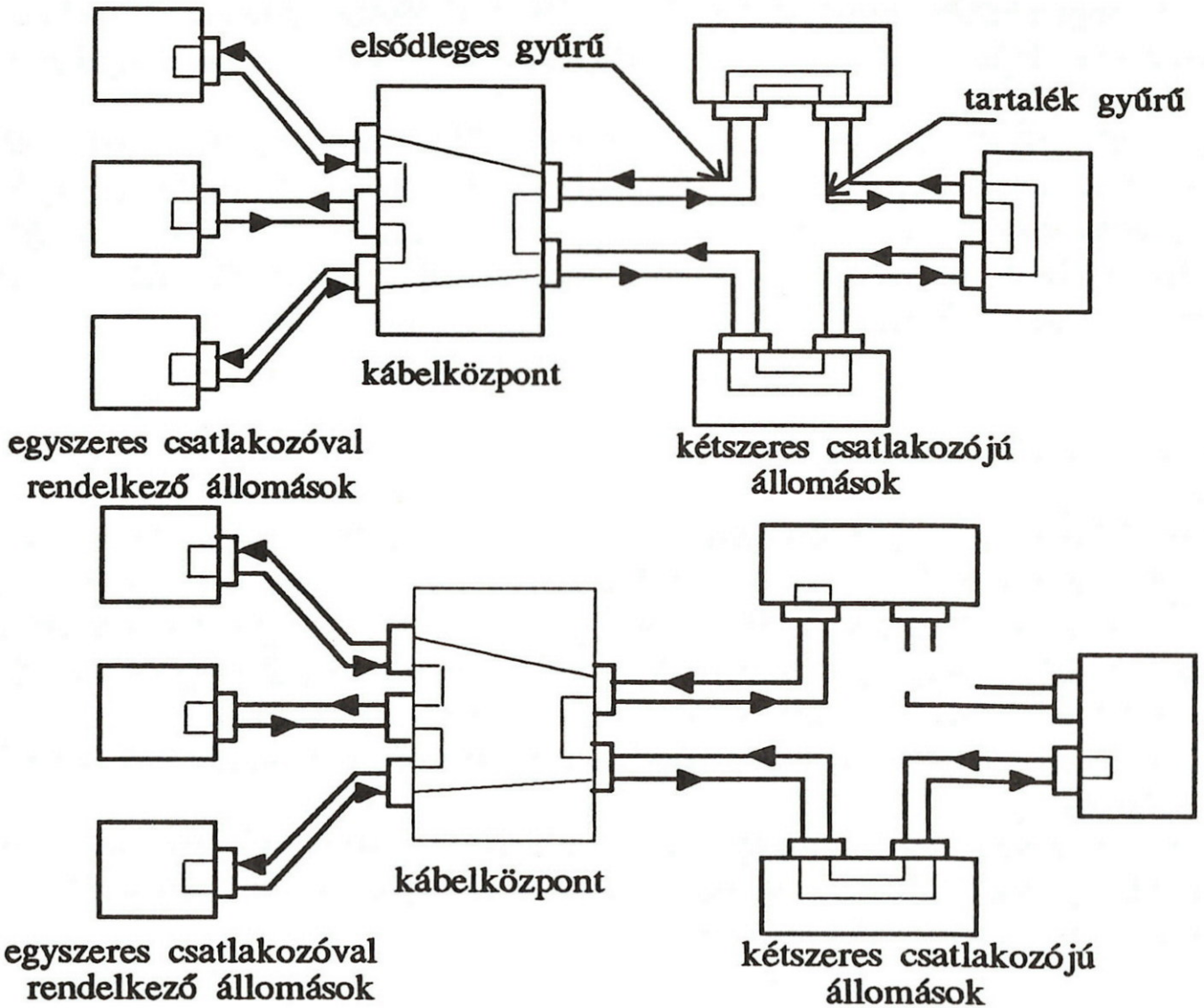
Természetesen a vezérlőjeles gyűrű esetén is a bekapcsolt gépek számától függően megadható az a maximális idő, ami után minden állomás megkapja a kábelt. A hálózat hatékonysága hasonló a vezérlőjeles buszéhoz. Kis forgalomnál várni kell, amíg a vezérlőjel odaér, a forgalom növekedésével a kihasználtság javul, megközelíti a 100%-ot, akárcsak a vezérlőjeles busznál.

Az IBM Token Ring-jének első változata 4 Mbit/sec sebességű volt, ma 16 Mbit/sec-mal működik. A kábelközponttól (a MAU-tól) az állomások maximális távolsága 300 méter. Több kábelközpont is összekapcsolható – ugyanúgy, mint ha állomások volnának –, így a hálózat jóval kiterjedtebb lehet, mint az Ethernet vagy az ARCnet, maximálisan 260 állomás csatlakozhat rá. A kábel lehet sodrott érpár vagy koax, sőt mivel pont-pont kapcsolatokat kell megvalósítani és csak egyirányú a kábeleken az átvitel, egyre gyakrabban használnak optikai szálakat, amelyek a másik két topológiában nem alkalmazhatók.

Ezeket a protokollokat fémkábelre tervezték – bár a vezérlőjeles gyűrűn használható az optikai szál is –, nyilvánvaló volt, hogy az optikai szál nagy sebességének a kihasználásához új protokollra van szükség. Az USA szabványügyi hivatala, az ANSI (American National Standard Institute) 1986-ban összeállított egy munkacsoportot egy nagy sebességű, optikai szálra épülő hálózati szabvány létrehozására. Ez lett az FDDI (Fiber Distributed Data Interface), az optikai szálas elosztott adatinterfész. Az FDDI vezérlőjeles gyűrűhálózatot valósít meg, elveiben hasonlít a Token Ringhez, de sok ponton lényegesen eltér tőle. Átviteli sebessége 100 Mbit/sec, két állomás maximális távolsága 2 km, míg a két legtávolabbi állomás 100 km-re lehet egymástól. A csatlakoztatható állomások száma 500. Az FDDI-n kétféle állomástípus használható, az egyiknek egyszeres hálózati csatlakozója van – egy bemenet és egy kimenet –, a másiknak pedig kettő, két kimenet és két bemenet. Az FDDI nem egy, hanem két gyűrűt használ, a kettős csatlakozóval rendelkező állomások mindkét gyűrűre csatlakoznak. A két gyűrűben az adatok haladási iránya ellentétes, de a második csak tartalék. Az első meghibásodása esetén átveszi a forgalmat. Még ha a második gyűrű is megsérül ugyanott, akkor sem kell egyetlen állomást sem kiiktatni, ahogy a 41. ábra mutatja. Az egyszeres csatlakozóval rendelkező állomások egy kábelközponton át



köthetőek a gyűrűbe, de ebben az esetben a kábelhiba az adott állomás kizárásával jár.



41. ábra Az FDDI gyűrű és átkonfigurálása kábelhiba esetén

### Olcsó hálózatok

Az eddig megismert helyi hálózatok a legelterjedtebbek, de vannak más, olcsóbb megoldások, amelyek egy adott feladatra szintén jól alkalmazhatóak. Ilyen például az Apple cég Machintos gépeinek hálózata, az AppleTalk. Busz elrendezésű, egyszerre max. 32 eszköz (számítógép, nyomtató, modem stb.) csatlakozhat rá. Pont-pont összeköttetést valósít meg a kábelon mindkét irányban. A hálózatra a párhuzamos porton át csatlakoznak az állomások, ami minden eszközbe eleve beépített, nem igényel külön adaptereket, és ez olcsóvá teszi. Adap-



terekkel természetesen a Token Ring vagy Ethernet hálózat is kialakítható a Macintosh gépekből. Az AppleTalk-re megfelelő adapterrel és szoftverrel IBM PC is csatlakoztatható.

Az Apple hálózata nem kínál olyan széles szolgáltatásokat és nem foghat át akkora távolságot, mint a korábban tárgyalt LAN-ok, de kisebb igényekre jól megfelel.

A PC-kre is léteznek olyan szoftverek, amelyekkel külön hálózati adapter nélkül, egyszerűen a soros porton át kapcsolhatunk össze gépeket, egy szerény hálózatot hozhatunk létre. Mivel a soros port sebessége csak töredéke a valódi hálózatokénak, csak nagyon egyszerű szolgáltatásokat várhatunk el tőle, és csak néhány gépet érdemes összekötni.

### **Amikor nem kell hálózat**

Bár a hálózatok a számítástechnikában tapasztalható általános árcsökkenésnek megfelelően szintén egyre olcsóbbak lesznek, mindig érdemes megfontolni, hogy az adott feladathoz valóban szükség van-e rá. Például pusztán egy nyomtató két, közös helyiségben lévő számítógépről való használatához nem szükséges hálózat. Erre tökéletesen megfelel egy átkapcsoló is, ami lehet egyszerű mechanikus szerkezet, vagy memóriával ellátott, akár több gépet is kiszolgáló intelligens egység.

Amennyiben két közeli gép között csak ritkán – mondjuk naponta – kell adatokat átvinni, ehhez csak egy null-modem kábel és egy kommunikációs szoftver kell, nincs szükség hálózatra.

### **Operációs rendszerek**

Mindeddig csak a hálózat fizikai – vagy hardver – kialakításáról esett szó. Egy állomásnak a kábelhasználat jogának megszerzésén kívül még jó néhány feladatot meg kell oldania a szolgáltatások biztosításához, ha például átmásolunk egy programot a hálózat egy másik gépéről a sajátunkra, vagy küldünk egy névre szóló üzenetet valamelyik kollégánk gépére. Mindezeket a gépen futó szoftvernek kell elvégeznie. A számítógép alapvető működését az operációs rendszernek nevezett program vezérli. A gép bekapcsolása utáni első feladata az operációs rendszer memóriába töltése valamilyen tárolóeszközzel. Amíg ez nem történik meg, gépünkkel nem tudunk dolgozni. Minden más, valamilyen célfeladatot, például szövegszerkesztést végző programot az operációs rendszer



indít el, és ha a munkát befejeztük vele, törli a memóriából. Az operációs rendszereket tulajdonságaik alapján csoportosítjuk, bár ezeket természetesen maguk a gépek is befolyásolják, nem használható minden operációs rendszer bármely típusú gépen.

A legegyszerűbb operációs rendszer egyszerre csak egyetlen parancsot tud végrehajtani, illetve csak egy programmal dolgozhatunk alatta. Új parancs kiadásához meg kell várnunk az előző befejezését, újabb program betöltéséhez ki kell lépnünk az előzőből. Mindebből következik az is, hogy egyszerre csak egy ember használhatja a számítógépet. Ezt a típust egyfelhasználós, nem multiprogramozható operációs rendszernek nevezik, és közéjük tartozik az IBM PC-ken leggyakrabban használt MS-DOS vagy IBM DOS.

A következő fokozatot az egyidejűleg több parancs, illetve program végrehajtására képes, egyfelhasználós, multiprogramozható rendszerek jelentik. Mivel csak egy embert szolgálnak ki, valódi előnyük nem is igazán az, hogy egyszerre több programot futtatnak – hiszen az ember általában egy dolgot csinál egyszerre –, hanem hogy az egyszerre futó programok között megoldják az információátadást. Ilyen operációs rendszer például a PC-ken az OS/2 vagy a DOS alól induló MS-Windows.

Egy gépen többen is dolgozhatnak, ha az operációs rendszer ezt lehetővé teszi. A többfelhasználós operációs rendszer értelemszerűen multiprogramozható is, hiszen nem mindenki akar ugyanazzal a programmal dolgozni. Legismertebbek közöttük a Unix különféle változatai vagy a VMS operációs rendszer.

A felsorolt kategóriák mindegyike egyetlen gépen fut. Távolsági hálózaton való kommunikációjukhoz megfelelő szoftver szükséges, amelynek ismernie kell a hálózaton alkalmazott protokollt. A helyi hálózatok lényegesen nagyobb sebessége miatt és sokrétű szolgáltatásainak biztosításához folyamatos kapcsolattartás szükséges az állomások között, ezért külön kategóriát alkotnak a LAN operációs rendszerek.

Az operációs rendszer kiválasztása szabja meg a helyi hálózat használatának módját. A hálózatba kötött gépek lehetnek egyenrangúak, ami azt jelenti, hogy minden gépről elérhető bármely másik. Használhatja a többi gép tárolóeszközzeit, nyomtatóját stb., és engedélyezheti, hogy a többiek is elérjék a saját tárolóit, nyomtatóját. Az egyenrangú állomásokat definiáló LAN operációs rendszert peer-to-peer rendszernek is nevezik, ilyen például a LANtastic vagy a LANSmart operációs rendszer. Más igényeknek felel meg az állomásokat kiszolgáló (server) és kliens (client, ügyfél) kategóriára felosztó client-server elrendezés. A kiszolgáló, szerver gépek tárolják a hálózaton megosztandó információt, programokat, adatfájlokat. A kliens gépek csak hozzájuk fordul-



hatnak közvetlenül, egymás eszközeit nem használhatják, de üzenetet küldhetnek egymásnak a szerveren át. Ebben a kategóriában a Novell cég NetWare nevű operációs rendszere a legnépszerűbb.

Az egyenrangú állomások a bekapcsolásuk után a DOS operációs rendszert töltik be, majd ezután az hálózatkezelő programot, ezért nevezik néha DOS alapú hálózatnak is őket. A felhasználó a DOS-sal a szokott módon dolgozhat, nem kell tudnia, hogy az éppen használt tároló vagy nyomtató melyik állomáson van. A szerveres hálózatoknál a kliens gépek szintén a DOS alatt futnak, és egy program betöltése után férnek hozzá a kiszolgáló gépen tárolt fájlokhoz. A szerver gép azonban nem a DOS-t használja, hanem magát az önálló, hálózati operációs rendszert. A DOS támogatja ugyan a hálózatokat, de igen sok korlátja van, hiszen eredetileg nem erre tervezték. Ezért az önálló, hálózati operációs rendszert használó, szerveres hálózatok lényegesen jobb teljesítményt nyújtanak, több állomás kiszolgálására képesek, mint a DOS alapú hálózatok. Ezekben a szerverek csak a hálózatra kapcsolódó állomásokat szolgálják ki, dolgozni általában nem lehet rajtuk.

A különböző hálózati operációs rendszerek általában többféle hozzáférési protokollt, és így hálózati adapterkártyát használhatnak. Sajnos az azonos protokollt alkalmazó hálózati adapterek nem teljesen egyformák, különösen Ethernet kártya vásárlásakor érdemes érdeklődni, támogatja-e a kiválasztott operációs rendszert.

Több felhasználó egyidejű munkája egy közös fájlrendszerben nemcsak jó csapatmunkára, az adatok megosztott használatára ad lehetőséget, de különböző biztonsági rendszabályok alkalmazását is szükségessé teszi. Nemcsak a véletlen vagy szándékos ütközéseket kell megakadályozni a felhasználók között, de az adatbiztonság is megköveteli, hogy ne férhessen hozzá bárki, bármilyen információhoz a hálózatban.

A védelmi rendszer a legtöbb hálózati operációs rendszernél többfokozatú. A felhasználónak, ha be akar kapcsolódni a hálózatba, meg kell adnia a nevét és a titkos jelszavát, mert a rendszer ezek alapján tudja őt azonosítani. Minden egyes felhasználóra beállíthatók különböző jogok, amelyek megszabják, hogy mit csinálhat a hálózatban. Ilyen jog szükséges egy fájl vagy fájlcsoporthoz olvasáshoz vagy írásához. Logikus például, hogy a raktárkészletet tartalmazó fájlt mindenki elolvashatja, de beleírni csak a raktárosnak van joga. Több tucat felhasználónál a jogokat egyenként beállítani kényelmetlen és nehezen áttekinthető. Ezért a felhasználók – mondjuk az egy osztályon dolgozók – csoportba szervezhetők és közös csoportjogokkal ruházhatók fel, az egyéni jogaik mellett.

Mindezen feladatok elvégzése a különleges jogokkal felruházott rendszergazdára hárul. Ő látja el a hálózat felügyeletét. Külön rendszergazdára általában



csak a sok felhasználóval rendelkező kliens-szerver hálózatoknál van szükség. A kevesebb, egyenrangú állomást tartalmazó DOS alapú hálózatoknál az egyes gépen dolgozó döntheti el, hogy kit és milyen jogokkal enged a saját gépéhez hozzáférni.

#### **Az OSI modell**

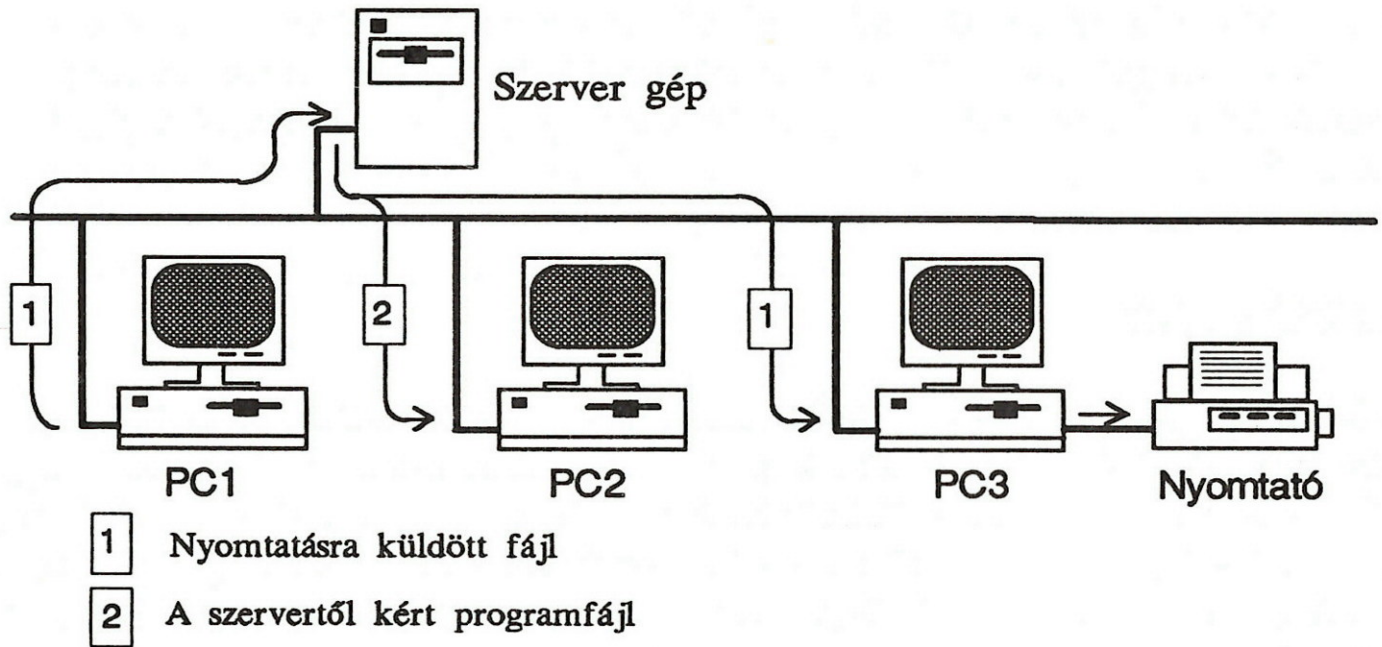
A hálózati operációs rendszer működteti a hálózatot, futtatja a különböző programokat. Az egyfelhasználós DOS operációs rendszer alá készült szoftverek egy része minden változtatás nélkül futtatható a hálózaton, mások más beállítást (installálást) igényelnek a hálózati használathoz, és végül vannak olyanok, amelyeknek külön hálózatra készült változata is van, az egyfelhasználós verzió nem működik hálózatban.

Ezek a hálózati operációs rendszerek alapvetően kétféle szolgáltatást nyújtanak. Egyrészt lehetővé teszik a szerver gépen tárolt állományok elérését (ezért nevezik néha fájl szervernek is), másrészt minden állomás használhatja a hálózat bármely gépéhez csatlakoztatott nyomtatót. Ezt a szolgáltatást távoli nyomtatásnak is hívják (l. 42. ábra). A fájlok elérhetősége azt jelenti, hogy ha a felhasználó egy programot akar futtatni, amely a szerveren vagy egyenrangú állomásoknál egy másik gépen tárolt, akkor az operációs rendszer elküldi a programfájlt a hálózaton át a felhasználó gépére és annak a memóriájában indítja el. Ugyanígy küldi el az adatfájlokat is, amelyeket ha a felhasználó megváltoztatott, vissza is ír a hálózaton át. Az egyenrangú hálózatoknál a szerver funkció nincs rögzítve az egyes gépekhez, mindig az a szerver, akitől a fájlt éppen elkérik.

A két alapfunkción túl a hálózati operációs rendszerek sok más szolgáltatást is lehetővé tesznek, ilyen például az elektronikus posta. Ezeket vagy maga az operációs rendszer tartalmazza, vagy külön vásárolható szoftverekkel valósíthatók meg.

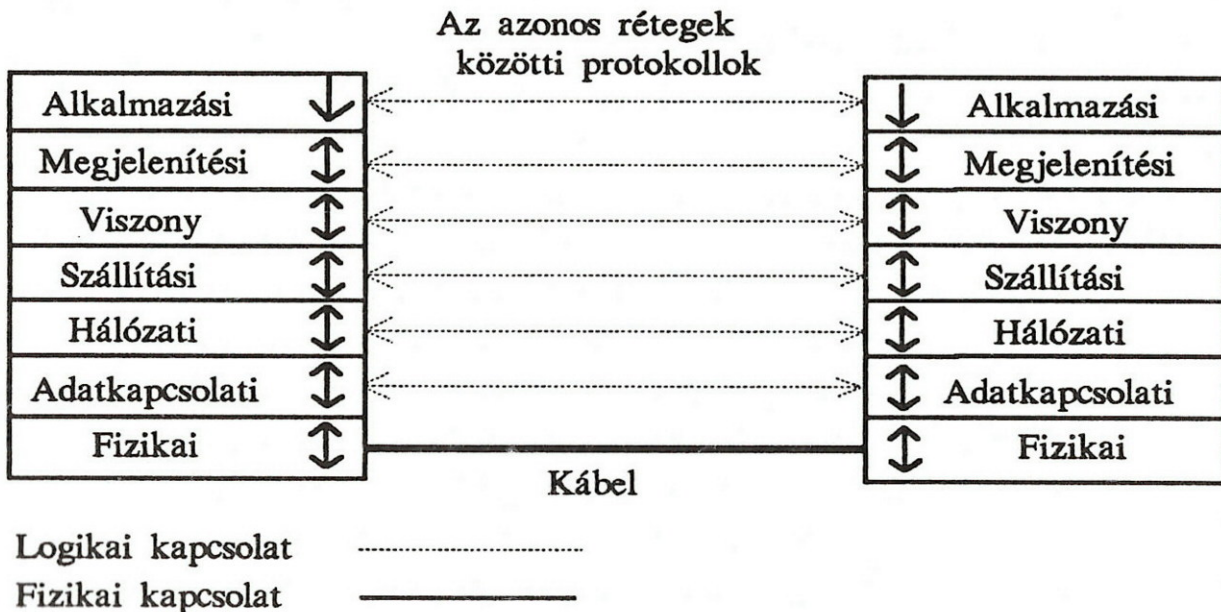
A hálózaton futó programok nyilván nem maguk végeznek minden feladatot, hanem az operációs rendszerre támaszkodnak, amihez viszont tudniuk kell, hogy mit várhatnak tőle. Általában egy hálózat konkrét megvalósítása nagyon sokféle lehet, szükség volt valamilyen szabványos megoldásra, amely lehetővé teszi, hogy a különböző hardvergyártók és szoftverkészítők termékei jól használhatóan illeszkedjenek egymáshoz. Ezért dolgozott ki a Nemzetközi Szabványügyi Szervezet, az ISO (International Standards Organization) egy ajánlást OSI néven (Open System Interconnection) a nyílt rendszerek összekapcsolására. Az ajánlás, amely nemcsak a helyi, de a távolsági hálózatokra is érvényes, a hálózat működését hét, rétegnek (layer) nevezett részre bontja, ezért gyakran nevezik





42. ábra A LAN hálózatok tipikus szolgáltatásai

hétrétegű modellnek is. Ezek a részek a hálózati kábelre adott feszültségtől kezdődően az állomásokon elérhető szolgáltatásokig fogják át a hálózat funkcióit. Minden réteg meghatározott feladatot lát el, az adatokat meghatározott szabályok – azaz protokollok – szerint adja és kapja a felette lévő rétegtől, és a réteg funkciójának megfelelő átalakítás után szintén valamilyen protokoll szerint cseréli ki az alatta lévővel. Minden réteg csak a két szomszédjával cserél adatot, a többiről nem tud. Két, a modellnek megfelelő hálózati állomáson a rétegeken belülről úgy tűnik, mintha a másik állomás azonos funkciójú rétegével állnánk kapcsolatban (l. 43. ábra).



43. ábra Az OSI modell rétegei



A rétegekhez tartozó protokollok nem részei a modellnek, ezek önálló szabványokként jelentek meg. A modell csak az egyes rétegek feladatait tartalmazza, az ezek végrehajtására szolgáló eljárások a rendszerek tervezőitől, programozóitól függenek.

Vegyük sorra a rétegeket, a hálózat fizikai működésétől haladva a bonyolultabb szolgáltatások felé.

1. szint. A fizikai réteg  
(Physical layer)

A fizikai réteg a modell legalsó rétege, meghatározza a használt kábelek, csatlakozók jellemzőit, az átvitelhez használt feszültség szinteket, a bitek milyen időzítéssel haladnak a kábelen stb.

2. szint. Az adatkapcsolati réteg  
(Data Link layer)

Az adatkapcsolati réteget két alrétegre bonthatjuk, az alsó, közeghozzáférést vezérlő réteg (Media Access Control) feladata, hogy meghatározza az egyes állomásoknak a közös kábelhez való hozzáférését. A második alréteg szabja meg a továbbított keretek formáját, azonosítja a küldő és a fogadó állomást, valamilyen CRC kóddal ellenőrzi a kereteket. Ehhez a réteghez tartoznak a korábban megismert Ethernet, Token Ring, vezérlőjeles busz-szabványok. A hálózati adapterkártyák általában a modell alsó két rétegének feladatait látják el.

3. szint A hálózati réteg  
(Network layer)

Ez a réteg olyan hálózatokban fontos, ahol a nagy távolság vagy több hálózat bonyolult összekapcsolása miatt két állomás többféle útvonalon is kapcsolatba léphet egymással. A 30. ábra olyan távolsági hálózatot mutat, ahol két állomás közötti kapcsolat csomagjai külön útvonalakon haladnak a hálózat egyenletesebb terhelése érdekében. Az adott terhelés melletti optimális útvonal meghatározása a hálózati réteg feladata. Amennyiben minden csomag ugyanazon az útvonalon, egy virtuális áramkörön (l. 29. ábra) halad át, ennek kiépítése is a hálózati rétegre hárul. A korábban említett CCITT X.25 szabvány az alsó három réteg számára definiál protokollokat. Üzenetszórásos, helyi hálózatokban csak egyféle útvonal van, így ott ez a réteg sokszor hiányzik, nem kap funkciót.

4. szint A szállítási réteg  
(Transport layer)

A szállítási réteg tördeli a felette lévő szintről folyamatosan kapott információt a hálózaton át továbbítható csomagokra. A kapott csomagok fizikai, érkezési sorrendje különböző útvonalak esetén nem mindig azonos a logikai, feladási sorrenddel. Az érkező csomagok helyes sorrendben való összeillesztése is a szállítási réteg feladata. A szállítási réteg az alatta lévő rétegekre támaszkodva már képes egy komplett kapcsolat kiépítésére két állomás között.



### 5. szint A viszonyréteg

(Session layer)

A viszonyréteg a szállítási réteg által biztosított kapcsolatokon magasabb szintű szolgáltatásokat szervez. Így kapcsolatot tud tartani párbeszédés formában két állomás között.

### 6. szint A megjelenítési réteg

(Presentation Layer)

Ez a réteg a továbbított adatok megfelelő formájúvá alakítását végzi. Ide tartozik az esetleges titkosítás, tömörítés, számformátumok konvertálása stb.

### 7. szint Az alkalmazási réteg

(Application layer)

A felhasználói programok részére olyan magas szintű szolgáltatásokat kínál, mint fájlok továbbítása két állomás között vagy az elektronikus posta.

Az OSI modell követése nem garantálja automatikusan a hálózati hardverek és szoftverek együttműködését, csak lehetővé teszi. Az egyes feladatokat – mint a kábelhozzáférés – többféle szabvány szerint is meg lehet oldani, és csak az azonos protokollt használó szoftverek és hardverek illeszthetők egymáshoz. Sok olyan hálózatot is használnak, amelyek felépítése nem követi az OSI modellt, ilyen például az IBM saját fejlesztésű, nagygépes hálózata, a SNA (System Network Architecture).

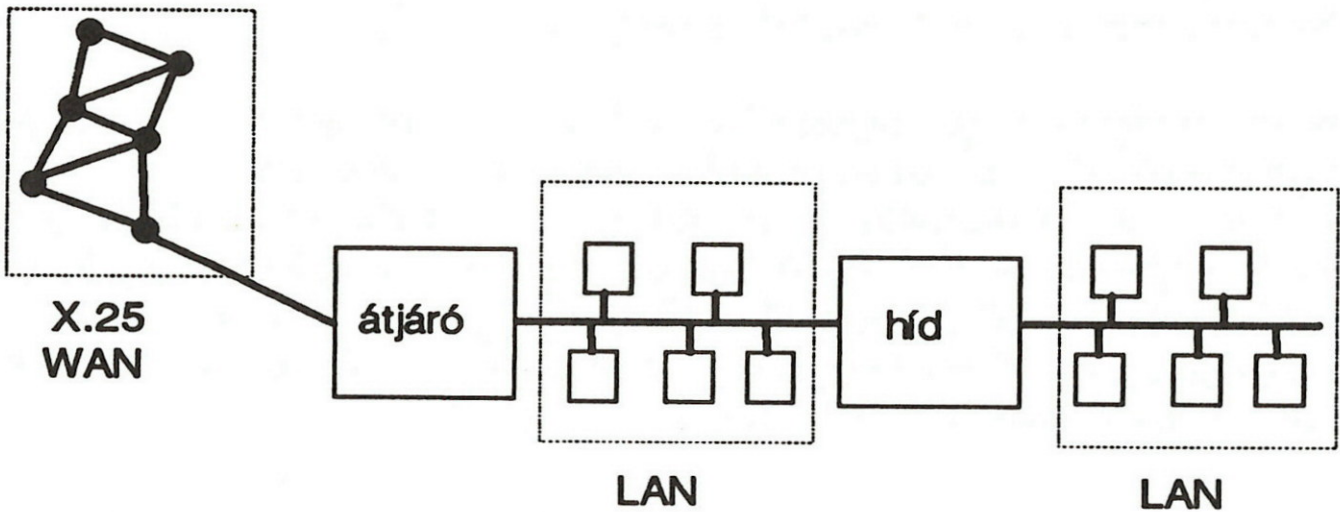
## Hálózatok összekapcsolása

A meglehetősen különböző – és a LAN-ok esetében távolságban is korlátozott – hálózatokat nyilván nem lehet egyszerűen a kábelek összekötésével összekapcsolni. Azt, hogy mi szükséges az összekapcsoláshoz, a két hálózat típusa dönti el.

Legegyszerűbb esetben csak a távolság növeléséről van szó. Mint a 6. táblázat mutatja, az Ethernet hálózat esetén a folyamatos kábelhossz 500 méter lehet. Jelerősítőkkel azonban öt ilyen szakasz kapcsolható össze. Az erősítőt ismétlőnek (repeater) hívják, mivel a kapott biteket egyszerűen megismétli a másik oldalon, mindkét irányban. Az OSI modell szerint így az ismétlő a fizikai rétegben kapcsolja össze a két szakaszt, ezért nem is szokás külön hálózatnak tekinteni őket.

A helyi hálózat típusától függő hosszt – az Ethernet esetén ez 2.5 km – azonban az ismétlőkkel sem haladhatjuk meg, két külön hálózatot kell építeni. Ezeket, ha azonos kábel-hozzáférési protokollt használnak – például mindkettő Ethernet –, híddal (bridge) kapcsolhatjuk össze. A híd mindkét hálózatra csatla-





44. ábra Hálózatok összekapcsolása

kozik, mindkettő forgalmát figyeli. A két oldalon vett kereteknél megvizsgálja a címezést, ha a címzett állomás a másik hálózaton található, ellenőrzés után a keretet – általában változatlan formában – kiadja arra a hálózatra. Mivel a két hálózat forgalma egymástól független, előfordulhat, hogy az egyik oldalról kapott keretet nem tudja azonnal továbbítani a másikra, mert a kábel éppen foglalt. Ekkor a kereteket tárolnia is kell, amíg meg nem kapja a hálózat használati jogát. A híd tehát – ellentétben az ismétlővel – önálló számítógép, és az OSI modell szerint az adatkapcsolati rétegben valósítja meg az összeköttetést.

Amennyiben a két hálózat nem azonos kábel-hozzáférési protokollt használ – például az egyik Ethernet a másik pedig Token Ring –, akkor a kereteket át is kell alakítani a továbbítás előtt. Ezt csak az adatkapcsolati réteg felett lévő szinten, a hálózati rétegben tehetjük meg, átjárók (gateway) alkalmazásával. Az átjáró feladatát végző számítógépnek nem elegendő csupán a címzett állomás azonosító számát ismernie (mint a hídnak), mert az állomások a kétféle hálózattípusban eltérő módon számozottak, maguknak a hálózatoknak is rendelkezniük kell címmel, azaz azonosító számmal. Ebből tudja az átjáró, hogy melyik hálózat melyik állomása a címzett. Az átjáró lényegesen több munkát végez, mint a híd, ezért az általa továbbított információ mennyisége is kisebb. A 44. ábrán két azonos típusú helyi hálózatot híd, míg a távolsági hálózattal egy átjáró köti össze.

Két lényegesen eltérő szerkezetű, például nem az OSI modellt követő hálózatot egy magasabb szinten forgalomirányító (router) számítógéppel kapcsolhatunk össze.



### **Távinformatikai rendszerek hazai lehetőségei**

Az SzKI kidolgozott egy olyan PC-bázisú hardver- és szoftver-technológiát, amelynek révén digitális térinformatikai rendszer építhető fel. A rendszerben ROUTER és NODE PC működik. A ROUTER PC a felhasználói platformokat (NOVELL, UNIX, MS-DOS/WINDOWS) postai előfizetői bérelt vonalakon keresztül köti össze a NODE PC-kkel. A NODE PC-k egymás között 2 Mbit/s sebességű digitális vonalon kapcsolódnak. 1993 második felében a TÁVIS PILOT nevű rendszer kísérleti üzemeltetésére kerül sor.



# IV. Nyílt rendszerek kommunikációja

## A Unix operációs rendszer

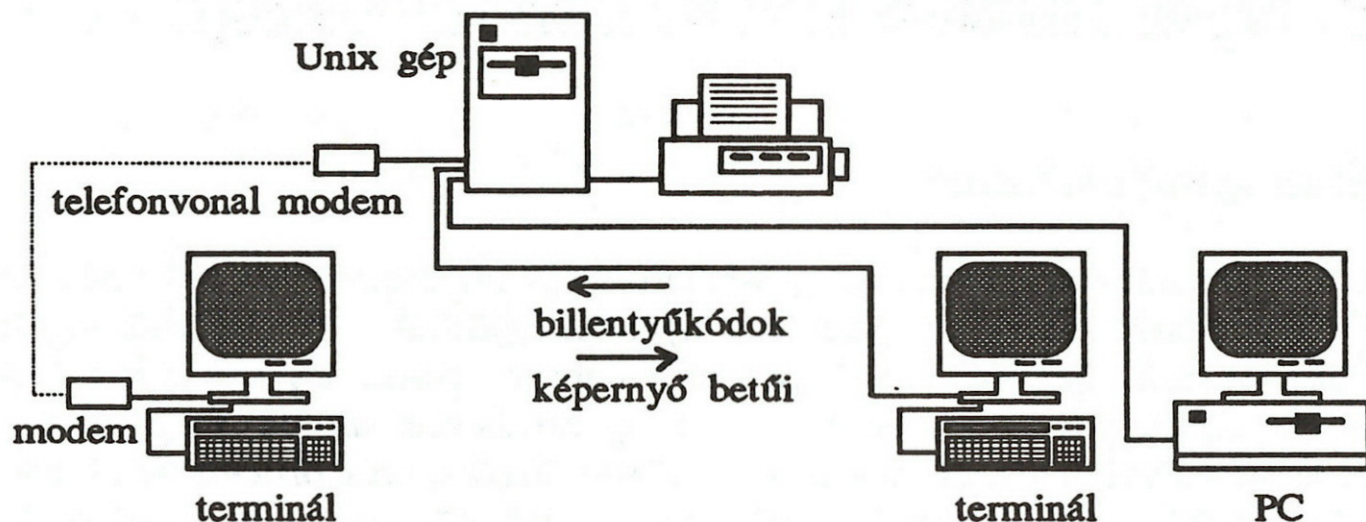
A Unix többfelhasználós, multiprogramozható operációs rendszer első változata 1969-ben készült el, a C programozási nyelvvel együtt. A hetvenes évek elején sokat javítottak rajta, tulajdonképpen ez a mai rendszerek alapja. Az első részben láttuk, hogy a programokat – és az operációs rendszer is egy program, hiszen utasításokat tartalmaz – minden, különböző típusú processzorral működő számítógéphez el kell készíteni. A Unix egyik előnye – a számos többi között –, hogy egy új géptípushoz viszonylag egyszerűen elkészíthető. A legelterjedtebb többfelhasználós operációs rendszer, eredetileg minigépekre tervezték, de mivel jól kihasználható vele a gépek növekvő teljesítménye, valamennyi, az elmúlt években megjelent RISC processzoros gép a Unix valamelyik változata alatt fut.

Az IBM PC-k teljesítményének növekedésével a Unix többféle változatát is elkészítették rájuk.

A gépen dolgozók egy billentyűzetből és monitorból álló terminállal használják a gépet. A terminálok a már ismert RS-232 interfészen át kapcsolódnak a számítógéphez. Így modemek segítségével, a telefonvonalon át akár nagyobb távolságról is hozzá lehet férni a számítógéphez. Sokféle terminált gyártanak, ezek billentyűzetelrendezése, de főként a képernyőkezelése némileg eltér egymástól, ezért a számítógépen be kell állítani a terminálok típusát. A legelterjedtebb a VT100-as termináltípus, ezért az ANSI 3.64-es számon ezt szabványosnak fogadta el és gyakran csak ANSI terminálként emlegetik. Egy PC is használható terminálként. Ehhez a null-modem kábelén kívül csak egy terminálemulációs szoftver futtatása szükséges. A legtöbb kommunikációs szoftvernek van terminálemulációs üzemmódja, legtöbbször a terminál típusa is kiválasztható. Be kell állítani a soros interfész paramétereit – start-, stop-, adatbitek száma és a sebesség –, és bejelentkezhünk a Unix gépre (l. 45. ábra).

Az elrendezés első pillantásra hasonlít egy helyi hálózathoz, de itt másfajta működésről van szó. A helyi hálózaton általában teljes fájlok továbbítása történik, a szerver elküldi a kért programot, és azt az állomás a saját memóriájában, saját processzorával hajtja végre. Itt minden programot a Unix gép hajt végre, a kábelén csak a felhasználó billentyűleütései és a képernyőre íródó információk haladnak át. A program és a megjelenítés, illetve a parancsok kiadása tehát szétválik.





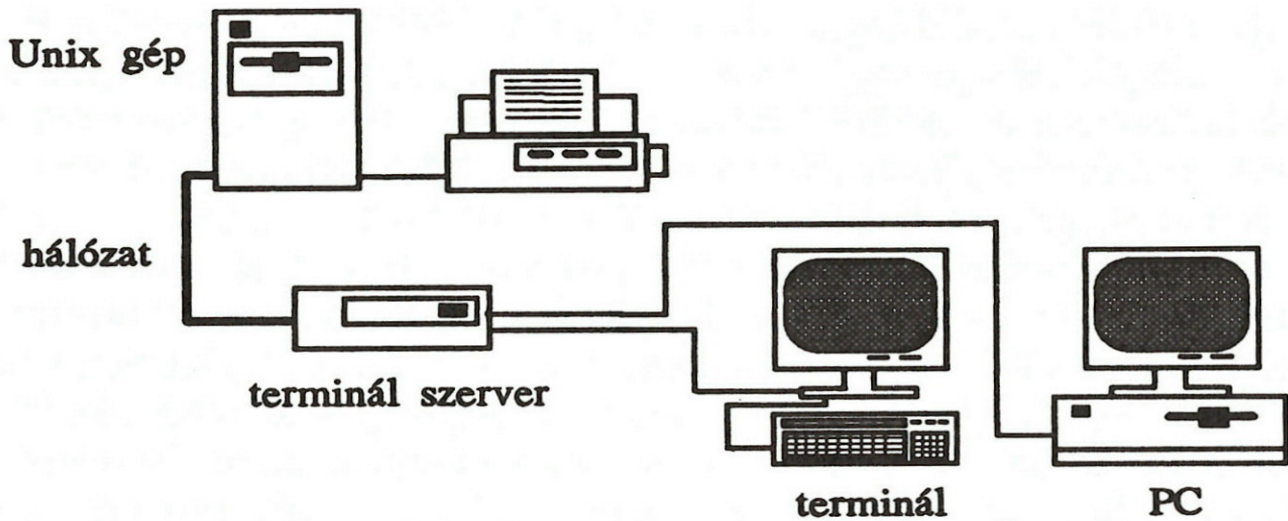
45. ábra Unix rendszer sémája

A terminálemulációs szoftver feladata ennek a beállítása a PC-n, még akkor is, ha az képes lenne a saját memóriájában végrehajtani a programot. A legtöbb esetben a Unix gép is csak egy processzorral rendelkezik, és egy processzor egyszerre csak egy feladatot tud végrehajtani. Ezért ezt úgy oldja meg, hogy minden programmal foglalkozik egy rövid ideig – milliszekundumokról van szó –, aztán átkapcsol a következőre. Amikor a sor végére ér, kezdi előlről. Ezt olyan gyorsan végzi, hogy a felhasználók nem veszik észre, mindenki úgy érzi, csak ő használja a gépet. A CPU az idejét osztja meg a feladatok között, ezért szokás az ilyen rendszert időosztásosnak (time sharing) nevezni. Mivel itt is többen dolgoznak egy gépen, érvényesek a LAN hálózatoknál látott biztonsági rendszabályok, jelszavas bejelentkezés, használati jogok stb.

A Unix gépbe legtöbbször nem helyezhetünk el annyi soros portot, ahány terminált használni akarunk, ráadásul mivel ezeket a processzor egyenként kezeli, a számuk növekedésével a rendszer jelentősen lassul. Ezért alkalmazzák a 46. ábrán látható terminál szervereket. Ezek olyan számítógépek, amelyeknek egyetlen feladata a terminálok és a Unix gép közötti kommunikáció támogatása. Gyors hálózattal csatlakoznak a számítógéphez, és soros portokat biztosítanak a termináloknak. Sokszor az RS-232-es szabvány által előírtnál jóval nagyobb sebességet – max. 115 Kbit/sec – is használhatunk velük.

A hagyományos Unix rendszereknél a gép a képernyőre ASCII karakterkódokat küld, a billentyűzetről pedig billentyűkódokat fogad. Ez csak szöveges információk megjelenítését és kezelését teszi lehetővé. A grafikus, képi információk számítógépes feldolgozásához grafikus terminálok szükségesek, amelyek a képet pontokból állítják össze, és ez új problémát vet fel. Már a karakteres – vagy más néven alfanumerikus – termináloknál is gond volt a típusok sokfélesége, de mivel alapvető működésük azonos – mindegyik ASCII kódokat hasz-





46. ábra Terminál szerver alkalmazása

nál –, ez viszonylag könnyen megoldható. A grafikus termináloknál a bonyolultabb feladat a megoldások sokkal szélesebb választékát is jelenti. Fennállt a veszély, hogy sokféle, az együttműködésre képtelen eszköz születik. Szükség volt egy szabványra, amely egységessé teszi a grafikus Unix terminálokat. Ez a szabvány az X Windows.

### Az X Windows

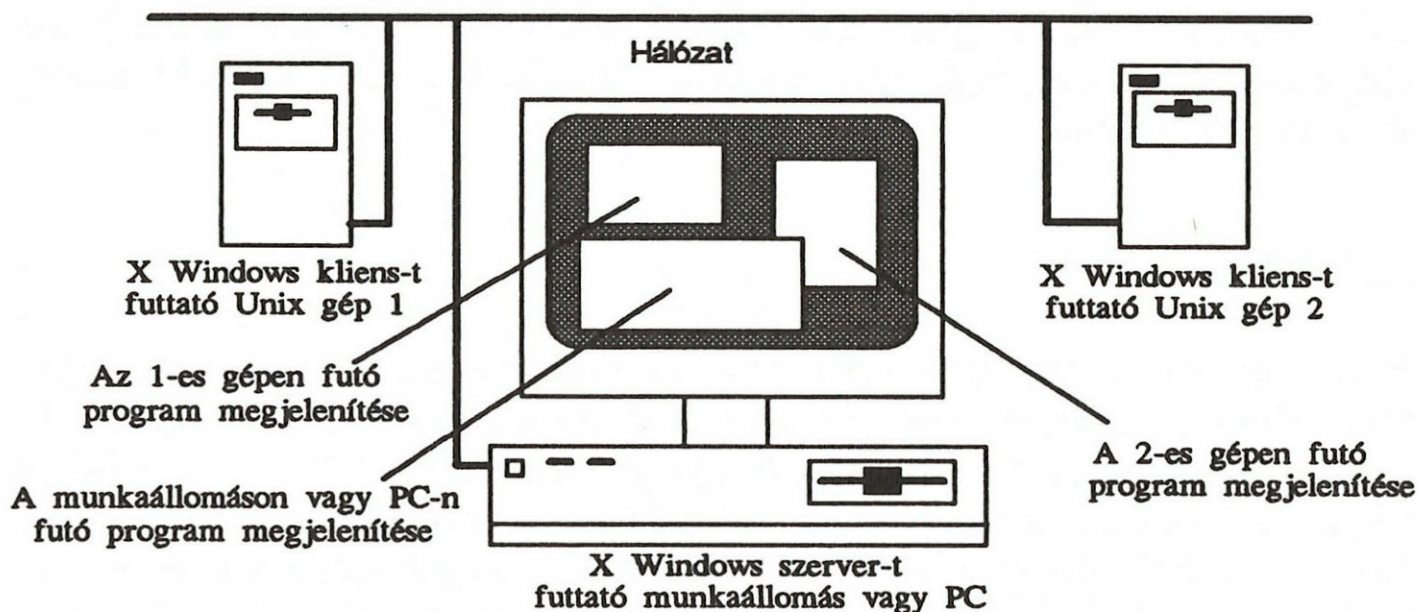
Az X Windows szabványt a MIT-en (Massachusetts Institute of Technology, a Massachusetts-i Műszaki Egyetem) dolgozták ki és egy a gyártóktól független bizottság folyamatosan felülvizsgálja, és igény szerint továbbfejleszti. A szabvány célja, hogy grafikus megjelenítést és adatbevitelt – ebben benne van az egér (mouse) kezelése is – tegyen lehetővé az időosztásos operációs rendszerrel, függetlenül a hardvertől, azaz a számítógépek és a használt adatátvitel típusától.

Az X Windows két program együttműködésén alapul. A terminálon az X szerver fut és tart kapcsolatot a Unix gépen végrehajtott X klienssel. A szerver-kliens elrendezés itt tehát fordított, mint a LAN hálózatoknál. Az X-szerver nem a teljes képernyőn – bár ez is beállítható –, hanem annak csak egy kijelölt, ablaknak nevezett (ablak = windows, innen a szabvány neve) területén jeleníti meg a billentyűleütéseket és a Unix gépről kapott információt. Egyszerre több ablakot is kezel, így egyidejűleg akár több, különböző Unix gépen futó programmal is dolgozhatunk, ahogyan a 47. ábra mutatja. Ez magyarázza a kliens-szerver elrendezés megfordítását a helyi hálózatokhoz képest. A Unix gépen minden bejelentkezett és X Windows-t alkalmazó felhasználóhoz egy X kliens tartozik, míg a terminálon az X szerver annyi X klienssel tart kapcsolatot,



ahány gépre a felhasználó bejelentkezett. Így több különböző gépen futó program eredményét láthatjuk egy képernyőn és az ablakok között egyszerű az adatok átmásolása is. A terminál lehet kifejezetten erre a célra gyártott grafikus X terminál, amelybe beépített az X szerver, vagy megfelelő programmal egy munkaállomás vagy gyors PC is használható X terminálként.

Az X Windows tehát nagyon kényelmes és hatékony módszer, de ennek ára van. Az alfanumerikus terminálok jól működnek az általában 9600 bit/sec sebességű soros vonalon. Ez az adatformától függően – hány stopbit, van-e paritásbit stb. – 800-900 bájt/sec-nak felel meg. A képernyőn legtöbbször 25 sor van, soronként 80 karakter, ami 2000 karakter összesen. Mindegyikhez tartozik egy kiegészítő bájt, ami színes képernyőnél a színeket, monokróm esetben a villogást, aláhúzást, intenzív megjelenítést stb. határozza meg. Így a teljes képernyő 4000 bájt, amit a fenti sebességgel 4-5 másodperc alatt lehet teleírni, ez elfogadható sebesség.



47. ábra Az X Windows működésének sémája

Grafikus esetben a képernyő pontokból rajzolja ki a betűket és az ábrákat. A PC-ken használatos VGA képernyő legtöbbször 480 sort és soronként 640 pontot jelenít meg. Ez monokróm képernyőnél, ahol egy képpont egy bitnek felel meg, 38 400 bájt. A fenti sebesség mellett a képernyő kitöltése 40-50 másodpercet vesz igénybe, színes képernyőnél ennek a többszörösét. Ezért, bár az X Windows valóban hardverfüggetlen, és működik akár a soros vonalon át is, a gyakorlatban valamilyen hálózatot igényel, a legtöbbször Ethernettel használják. Az X terminálok lényegesen drágábbak az alfanumerikus típusoknál, ha pedig PC-t használunk valamilyen programmal X terminálként, a leggyorsabbak



közül kell választani, 4-8 Mbájt memóriával. A PC-kkel szemben további hátránya, hogy a DOS programok közül csak a karakteres megjelenítést használók futtathatók X Windows alatt, grafikus, Microsoft Windows vagy OS/2 alá írt programok nem.

Mindezek ellenére az X Windows sikeres, mivel sokféle gép és Unix változat szinte korlátozás nélküli összekapcsolását teszi lehetővé. Az X Windows így tulajdonképpen egy szoftver interfésznek tekinthető. Ahogy az RS-232 mint hardver interfész minden olyan készülék – terminál, PC, modem, PAD – összekapcsolását lehetővé teszi, amely követi az előírásait, az X Windows is számtalan program és gép összekapcsolására ad lehetőséget. A Unix operációs rendszer sokféle változata miatt környezetében több ilyen szoftver interfészt is kidolgoztak, ami megkönnyíti az együttműködésüket. Ezért a Unix rendszereket gyakran nyílt vagy nyitott rendszereknek is nevezik.

Az X Windows szabvány szerint működő grafikus felhasználói interfész (GUI = Graphical User Interface) a Motif és az Open Look. Érdemes megemlíteni a Quarterdeck cég DESQview/X szoftverjét, amellyel a DOS operációs rendszeréből indítva a PC-t X terminálként használhatjuk. Az eredeti X Windows szabvánnyal szemben Microsoft Windows programok futtatását is lehetővé teszi.

### ONC/NFS

Az X Windows mellett, ami tényleges szabvány, meg kell említeni a szabványként még nem bejegyzett, de szinte minden jelentős gyártó által támogatott ONC/NFS (Open Network Connection/Network File Server) interfészt. Ezt 1984-ben a Sun Microsystems fejlesztette ki. Tulajdonképpen nem egy eljárás, hanem több, egymást kiegészítő protokoll, amelyek közül az NFS a legismertebb. A protokollok valamilyen hálózaton keresztül elérhető szolgáltatásokat rögzítenek. A gyártók szinte minden számítógéptípushoz és operációs rendszerhez elkészítették az ezeket a szolgáltatásokat nyújtó szoftvereket, amelyek az előírások szerint kommunikálnak egymással a hálózaton át. Így sokféle gép és operációs rendszer működhet együtt. Az eljárások – amelyek részletes ismertetése meghaladná könyvünk kereteit – lehetővé teszik mind a helyi operációs rendszereknél látott hálózatoknál szokásos fájlátvitelt, mind a Unix rendszereknél megismert távoli gépen való programvégrehajtást. Az NFS szolgáltatások, akár csak az X Windows, már PC-ről, a DOS operációs rendszer alól is elérhetők.



# Tartalom

<b>Ajánlás</b> .....	5
<b>I. Információtárolás a számítógépben</b> .....	7
Miért digitális? .....	7
Adatformátumok .....	9
Adat és utasítás .....	11
Mi a fájl? .....	12
A számítógép modellje .....	14
Adattovábbítás a számítógépen belül .....	17
Az IBM PC-kompatibilis PROPER számítógépcsald .....	21
<b>II. Adatátvitel számítógépek között</b> .....	23
Interfészek .....	23
Párhuzamos átvitel és párhuzamos port .....	23
Soros adatátvitel .....	26
Szinkron átvitel .....	29
Aszinkron átvitel .....	30
<b>Aszinkron adatformátum</b> .....	30
Az adatátvitel sebessége .....	33
Hibaellenőrzés .....	34
Átviteli üzemmódok .....	35
A soros port konfigurálása .....	36
Az RS-232C interfész .....	37
Adatátvitel modemem keresztül .....	40
Közvetlen kapcsolat az RS-232C-n .....	42
Modemek .....	43
Modemtípusok .....	45
Szabványok .....	47
A PC-adatkommunikáció néhány hazai eredménye .....	48
<b>III. Számítógép-hálózatok</b> .....	49
Miért kell a hálózat? .....	49
Hálózattípusok .....	50
Távolsági hálózatok .....	51
Nyilvános hálózati szolgáltatások .....	56
Magánhálózati szolgáltatások .....	56
Helyi hálózatok .....	57
Helyi hálózatok topológiái .....	58



Hálózati kábelek .....	60
Olcsó hálózatok .....	71
Amikor nem kell hálózat .....	72
Operációs rendszerek .....	72
Az OSI modell .....	75
Hálózatok összekapcsolása .....	78
Távinformatikai rendszerek hazai lehetőségei .....	80
<b>IV. Nyílt rendszerek kommunikációja .....</b>	<b>81</b>
A Unix operációs rendszer .....	81
Az X Windows .....	83
ONC/NFS .....	85



**CÉDRUS Kiadó Kft**

**Felelős kiadó Sebestyén Ilona ügyvezető igazgató**

**Felelős szerkesztő Szántó Zsuzsanna**

**Tipográfia Székelyhidi Ilona**

**Borítóterv Faklen Pál**

**Nyomdai előkészítés GRAF-ICA BT**

**Nyomta és kötötte az Alföldi Nyomda**

**A nyomdai megrendelés törzsszáma: 8671.66-13-2**

**Készült Debrecenben, az 1993. évben**

**Felelős vezető: György Géza**



356,- Ft



# SZÁMÍTÁSTECHNIKA KOMMUNIKÁCIÓ – IGÉNYESEKNEK!



## SZÁMÍTÁSTECHNIKA, KOMMUNIKÁCIÓ ÉS INNOVATIKA RT.

1011 Budapest I., Iskola utca 10.

Levélcím: 1251 Budapest, Pf. 19.

Telefon: 201-8122, 201-8535, 201-8080

Fax: 201-7773

---

SZAKBOLT: 1011 Budapest I., Fő utca 31.

Telefon: 201-4409 Fax: 201-4322