Hegedűs Géza
Herdon Miklós
Kovács György
Némedi János

Számítógép-hálózatok

Számalk Kiadó
Budapest, 2002

Impresszum

Előszó

Tisztelt Tanuló/Hallgató!

Üdvözöljük hallgatóink körében! Köszönjük bizalmát, hogy segítségünkkel kívánja ismereteit megszerezni.
A tudásbővítés és készségfejlesztés megvalósításához sok sikert kívánunk!

Ön további tanulmánya során egyre több alkalommal fog találkozni olyan tananyagokkal, amelyek az önálló, irányított tanulást segítik. Ezek között is megkülönböztethetők a hagyományos önálló tanulást segítő (nyomtatott) taneszközök, továbbá egyre gyakrabban találkozhatunk elektronikus tananyagokkal (e-book).

Az utóbbi kategóriába sorolt taneszközök definiálása még nem egységes. **Elektronikus tananyag** kifejezést alkalmaznak a webre felhelyezett (elektronikusan megjelentetett) jegyzetekre, példatárakra, könyvekre, CD-formátumú, tanulási célra készített multimédia anyagokra éppúgy, mint olyan on-line vagy off-line tananyagokra, melyek előre tervezett elektronikus médiumok (a tartalom és az ismeretelsajátítás módját figyelembe véve), meghatározott képzés adott tananyagához fejlesztve. Az e-bookok fejlesztése és népszerűsége a számítógépet használók körének növekedésével is összefügg.

Szakmai közösségünk egy e-book sorozat elkészítésére vállalkozott, amelyből Ön most egy informatikai tantárgyhoz kapcsolódó tananyagot kapott kézhez.

Kérjük, hogy pár percig még tartson ki! Ne hagyja abba ennek a bevezetőnek az olvasását, mert sohasem fogja megtudni, hogy segítségünkkel mire vállalkozik!

Tisztelt Tanuló/Hallgató!

Tananyagaink tartalmilag azonos szerkezetben és formai megoldással készültek. Minden anyag strukturált, fejezetekre, alfejezetekre osztott. A fejezetek bevezetőjében eligazítást kap arról, hogy mit tárgyal az adott tananyagrész. A fejezeteket összefoglalók zárják, melyekben kiemeljük a legfontosabb, „hangsúlyos” részeket, a definíciókat megismételjük, a korában tanultakra is hivatkozva szintetizáljuk az ismereteket. Minden fejezetet önellenőrző kérdések, feladatok zárnak, melyekre természetesen a válaszokat (megoldásokat, indoklásokat) is megtalálhatja az anyagban.

Ha a tananyag gyakorlati jellegű, akkor az egyes fejezetek közül az első mindig ismeretátadás vagy ismétlés, a második feladatokat és megoldásokat tartalmaz, amely a megértést segíti, majd a harmadik fejezet ez esetben mindig az alkalmazási készség fejlesztését célozza.

A tananyagokban alkalmazott jelölések, ikonok:

Az elektronikus tananyagban általában ötféle ikont alkalmazunk, ezek jelentése valamennyi általunk fejlesztett tananyagban azonos, a következők szerint:

Kiegészítő ismeretanyag, amely a törzsanyagon túl az adott tárgykör, témakör magas szintű elsajátításához járul hozzá.

Nagyon fontos, a tárgykör (témakör, készség) elsajátításának elengedhetetlen feltételeként meghatározott tudás- vagy készségtartalom.

Terminológiai szótárra való hivatkozás. Amely kifejezések, szavak mellett ott találja ezt a jelzést, azokól egy ún. belső link vezet a Terminológiai szótár meghatározott részéhez, ahol megtalálja a szó vagy kifejezés magyarázatát, jelentését.

A nyitottkönyv-ikon a nyomtatott szakirodalomra való hivatkozást jelenti. Ahol ilyen jelet lát, ott megtalálhatók a további szakirodalmi ajánlások is.

E jelöléssel mindig arra utalunk, hogy ellenőrző, önellenőrző feladatsor vagy tesztek következnek.

Az ikonok egyértelműen megmutatják Önnek, hogy minden tananyag törzs és kiegészítő anyagra bontható. A kiegészítő anyag szorosan kapcsolódik ugyan a témához, de adott iskolatípusban a szakmai követelmények nem írják elő kötelezően. A törzsanyag ismerete vagy az abban foglalt készségek elsajátítása azonban feltétlenül szükséges.

A Terminológiai szótárban megtalálja azokat a szavakat, amelyek ebben az anyagrészben újnak számítottak, illetve az idegen szavak, kifejezések magyar megfelelőjét.

A tananyagfejlesztők által kijelölt hangsúlyos (!) részeket feltétlenül tanulja meg, illetve gyakorolja.

Tisztelt Tanuló/Hallgató!

Engedje meg, hogy tanulását még négy kérdés és az arra adott válaszok erejéig segítsük!

Hol tanuljak?
Bárhol, ahol a feltételek adottak. Iskolában, otthon, buszon esetleg vonaton. Természetesen az elektronikus tananyag is nyomtatható, így „papír" formában magával is viheti a tanuló bárhová. Ugyanakkor meg kell mondjuk, a leghatékonyabb, ha a képernyő előtt tanul, mert a „beépített" navigációs utat betartva haladhat leginkább a tanulásban (a nyomtatott tananyag nem tartalmaz linkeket, beépített segítségeket, pl. a válaszokat a kérdésekre, illetve kiegészítő anyagokat).

Mikor tanuljak?
Természetesen akkor, amikor erre ideje van, illetve akkor, amikor szükséges. Erre a kérdésre azonban az igazi jó válasz, hogy ütemterv (Tanítás-tanulási útmutató) szerint.
Ön közvetlenül felelős azért, hogy tanulási folyamata sikeres legyen.

A tananyagfejlesztők és tanárai ebben segítenek Önnek. Elkészítették ezt a tananyagot, megfogalmazták a tanulás és készségfejlesztés sorrendjét, a tananyag-feldolgozás és értékelés menetét, így részt vállaltak az Ön tanulási folyamatában.

Hogyan tanuljak?
Amit feltétlenül ajánlani lehet: úgy tanuljon, ahogyan azt megtervezték az Ön számára. Fogadja el a javaslatokat, ugyanakkor az Ön jól bevált tanulási szokásait sem kell elfelejtenie. Ezek jól hasznosíthatók, ha ütemterv szerint halad és sikerül teljesítenie azt. Amennyiben nem boldogul a tananyaggal egyedül, kérhet tanácsot tanáraitól. Ne feledje: Ön a tanulás főszereplője, mi valamennyien segítjük Önt, ha kéri.

Végül, de nem utolsó sorban, tegye fel önmagának a következő kérdést: ***Miért tanuljak?*** Erre a kérdésre bizonyára Ön tudja is a választ. Nyilván mielőtt jelentkezett, átgondolta, milyen előnyökkel járhat, ha elvégzi ezt a képzést. Biztosan számos dolog motiválta Önt, amikor elkezdett tanulni, és lesznek időszakok, amikor nagyon elkeseredett lesz, vagy már nem tartja annyira fontosnak azt, amire vállalkozott. Ne adja fel! A tanulás a lehető legjobb dolog, amire vállalkozhatunk, gazdagabbak leszünk általa, és értékesebb munkát végezhetünk.

Jó tanulást és sok sikert kívánok a szerzők és tananyagfejlesztők nevében:

Dr. Sediviné Balassa Ildikó
főszerkesztő

Bevezetés

A számítástechnikai ipar fejlődése napjainkra lehetővé tette az elérhető árú számítógépek beszerzését, melynek következtében teljesnek mondható a vállalatok számítógépes ellátottsága, és egyre több család rendelkezik számítógéppel. E hatalmas számítógép-állomány egy új problémát vetett fel a vállalatok életében: a hálózatok kialakításának, működtetésének problémáját. Ma már nélkülözhetetlen a vállalati hálózat, és egyre jobban előtérbe kerül a hálózatok egyik legismertebb megvalósítása, az Internet elérésének lehetősége a munkahelyekről vagy akár otthonról is. Ennek következtében az alapvető hálózati ismeretek elsajátítása szükségessé vált a mindennapi munkához, és nélkülözhetetlen az internetezéshez is. Természetesen más mélységű ismeretre van szüksége egy orvosnak, aki pl. az Internet segítségével szeretne információhoz jutni, vagy egy rendszergazdának, akinek a feladata a teljes hálózat menedzselése, vagy egy informatikai szakembernek, akinek a rendszer működésével kell tisztában lennie.

E jegyzet célja, hogy olyan ismereteket adjon át, melyekre egy informatikai szakembernek szüksége van a mindennapi munkájához, vagy akár egy rendszergazdának is kellő alapokat biztosítson a hálózati ismeretekben.
E tantárgy jelentősége azért is kiemelkedő, mert megalapoz számos más tantárgyat, pl. a Hálózati felhasználói ismereteket és a Linux gyakorlati alkalmazását is.

I. A hálózatok célja, alkalmazása, alapfogalmak

Bevezető

Ebben a fejezetben olyan alapfogalmakat tisztázunk, melyeket a következő részekben sűrűn fogunk használni. Nagyon fontos a szakszerű szóhasználat, melyet ez a fejezet mélyít el.
Tulajdonképpen az egész fejezet egy nagy bevezetőnek is tekinthető, mivel a jegyzet többi egysége az itt leírtakon alapul, illetve az itt leírtakat bontja ki részletesebben.
A legfontosabb most tárgyalásra kerülő rész az ISO OSI-modell, ugyanis az itt meghatározott ajánlások alapján működik a hálózatos világ.

A számítógépek és a távközlés közeledése – sőt napjainkra már sok területen egybeolvadása – alapvetően befolyásolta a számítógépes rendszerek szervezését. A régi modellt, melyben egy intézmény teljes számítástechnikai igényét egyetlen számítógép szolgálta ki, felváltotta a sok-sok különálló, de egymással összekapcsolt számítógép.

Ha **számítógép-hálózatok**ról beszélünk, a fogalom alatt önálló számítógépek összekapcsolt rendszerét értjük. Két számítógép akkor összekapcsolt, ha információcserére képesek. Az összekapcsolás többféle módon történhet, pl. rézvezeték, lézersugár, infravörös fény, mikrohullám vagy akár távközlési műhold segítségével is.

A számítógépek hálózatba kapcsolásának előnyei:

- **Erőforrásmegosztás**, azaz a külön-külön meglévő erőforrások (berendezések, perifériák, programok, adatok) elérhetőek a felhasználók számára, fizikai elhelyezkedésétől függetlenül. Az erőforrások igénybevételéhez természetesen megfelelő jogosultság szükséges.

- **Költségmegtakarítás**, **egyenletesebb teljesítménymegosztás**: a hálózat lehetővé teszi, hogy a rendszerben a drága, nagy teljesítményű perifériákat, nyomtatókat, háttértárakat, egyéb erőforrásokat felhasználók mindegyike elérheti fizikai elhelyezkedésétől függetlenül, így elég kevesebb példányban megvásárolni és üzemeltetni azokat.

  A rendszer az eszközök teljesítményének egyenletesebb megosztására is alkalmat ad. A kis számítógépek jobb ár/teljesítmény aránnyal rendelkeznek, mint a nagyobb gépek. Igaz ugyan, hogy az ún. erőforrásgépek egy nagyságrenddel gyorsabbak, mint az egyetlen mikroprocesszorra épülő mikrogépek, az áruk viszont több nagyságrenddel nagyobb.

  A fentiek tipikus példája a „fürtözés”.

- **Nagyobb megbízhatóság**: redundanciával növelhető a rendszer biztonsága. Egy adott funkciójú eszköz hibája nem jelenti az adott funkció teljes megszűnését, ha a meghibásodott egység helyett egy másik átveheti a szerepét (pl. több nyomtató egy rendszerben). A háttértárak sérülésének hatását is kivédhetjük, ha a fontos adatokat, programokat több számítógép lemezegységén tároljuk.

- **Adatbázisok elérése**: a hálózat a gépek közötti adatcserével lehetővé teszi adatbázisok adatainak elérését, az adatbázis bővítését, több felhasználói végpontból.

- **A hálózat mint kommunikációs közeg** üzenetek, levelek vagy egyéb információk küldésére és fogadására is alkalmas (írott szöveg, álló- vagy mozgókép, hang).
  Ennek az alkalmazásnak egyre nagyobb a jelentősége, és számíthatunk rá, hogy a hálózatok sebességének növekedésével a multimédiás felhasználások kiterjedten megjelennek a hálózaton.

1. Hálózati struktúrák

A következőkben tekintsük át a hálózatok legfontosabb elemeit:

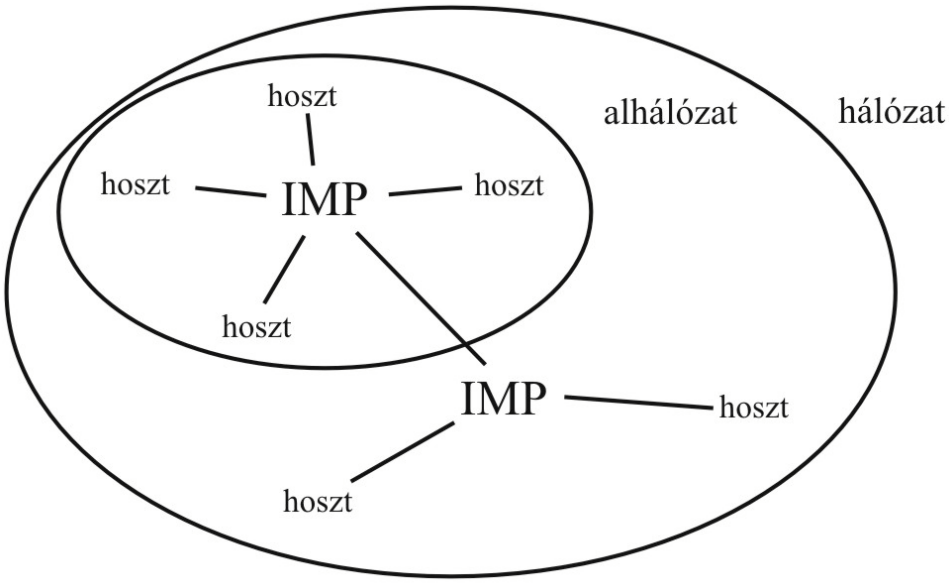
- **Hoszt**oknak (vagy gazdagépeknek) nevezzük azokat a számítógépeket, melyek egy számítógépes rendszerben össze vannak kötve. Ezeken futhatnak felhasználói programok, lehetnek rajtuk adatbázisok.
  (Ha egy számítógépben két hálózati kártya van, akkor az a gép két hosztnak tekintendő! Ezért a hálózatok témakörében nem számítógépeket szoktak alapegységnek tekinteni, hanem hosztokat.)

- A **kommunikációs alhálózatok** szerepe a hosztok közötti üzenetek továbbítása. Az alhálózatok funkcionálisan két fő részből állnak: a csatornákból és a kapcsolóelemekből.

  - **Csatornák**nak nevezzük az adatátvitelt biztosító vonalakat. Lényegében ezeken továbbítódnak a bitek, valamilyen fizikai jellemző változás formájában. (Pl. ha a levegőt tekintjük csatornának, adatátvitel a részecskék rezgésével történhet. Optikai adatátvitelnél a fény megléte vagy hiánya biztosítja az adatok átvitelét. Ezzel kapcsolatosan részletesebben a következő helyen olvashat: fizikai átvitel jellemzőinek változása.)

- A **kapcsolóelemek** a vonalakat kötik össze egymással. Ezek a kapcsolóelemek lehetnek egy hoszt részei (pl. hálózati kártyák), de leggyakrabban önálló célszámítógépek. A kapcsolóelemek további szokásos elnevezései: IMP (Interface Message Processor – interfész üzenetfeldolgozó), vagy hálózati kapcsoló pontok (internal network switching node)

Az alhálózat és a hosztok együttesen alkotják a hálózatot.



1. ábra
Hálózatfelépítés

Egy alhálózat logikai (egyes esetekben fizikai) egységbe tartozó részét szegmensnek szoktuk nevezni.

A hálózatba kapcsolt hosztok fizikai elrendezésének módját **topográfiá**nak, míg az összekapcsolás fajtáját **topológiá**nak nevezzük. Az összeköttetés-alapú hálózatok topológia szerint két nagy csoportra oszthatók:

- két pont közötti csatornával rendelkező, és
- üzenetszórásos csatornával rendelkező alhálózatokra.

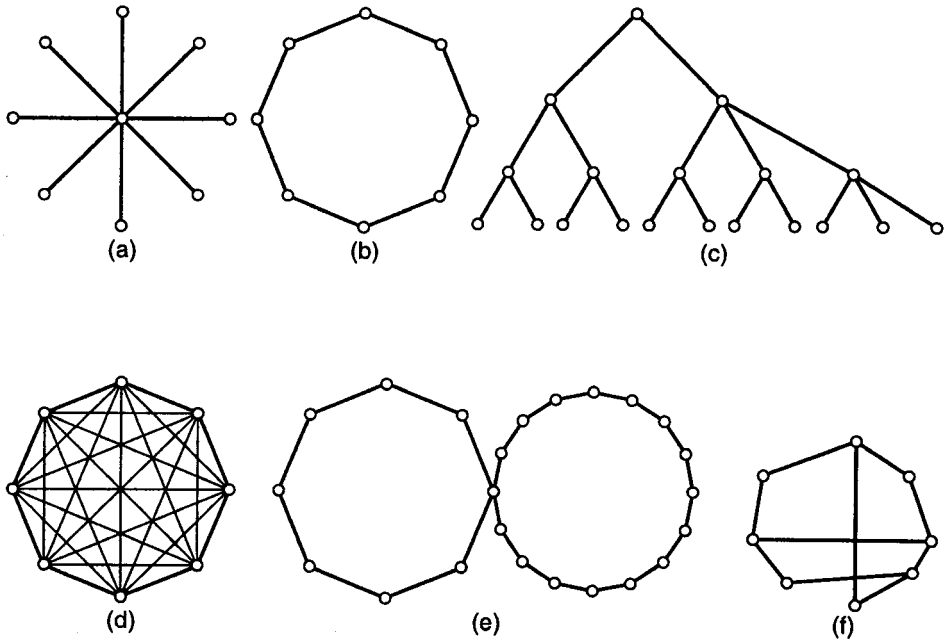
1.1. Két pont közötti csatornával rendelkező alhálózat (pont-pont összeköttetés)

Ennél a kialakításnál az egymással kommunikálni szándékozó hosztokat páronként összekötik, és az üzenetek az így létrejövő csatornán keresztül haladnak. Ha a rendszerben több hoszt is van a hálózatban, a csomagok egy vagy több közbülső állomáson áthaladva jutnak el a forrástól a célállomásig. Az adó által indított információt a vevő számítógép ellenőrzi (neki szólt-e az üzenet). Ha nem ő volt a címzett, továbbküldi a következő, vele kapcsolatban levő gépnek.

E kialakítás előnye, hogy a kommunikációs problémákat a két hoszt és a közöttük lévő csatorna hordozza, ezért adott esetben a hibakeresés jelentősen leegyszerűsödhet. Sok gépet tartalmazó hálózatban azonban nagyon sok összeköttetést kell létrehozni, ez viszont áttekinthetetlenné válhat.
Ennek a megoldásnak ez a legnagyobb hátránya.

Ha egy hálózatban – amelyben N darab gép van – azt szeretnénk elérni, hogy minden gép tudjon egymással kommunikálni, akkor N*(N-1)/2 darab összeköttetést kell létrehoznunk.

A hosztok összekapcsolása több különböző módon oldható meg, néhány lehetséges elrendezés az alábbi ábrán látható.



2. ábra
Két pont közötti alhálózati topológiák
(a) csillag, (b) gyűrű, (c) fa, (d) teljes, (e) egymást metsző gyűrű, (f) háló

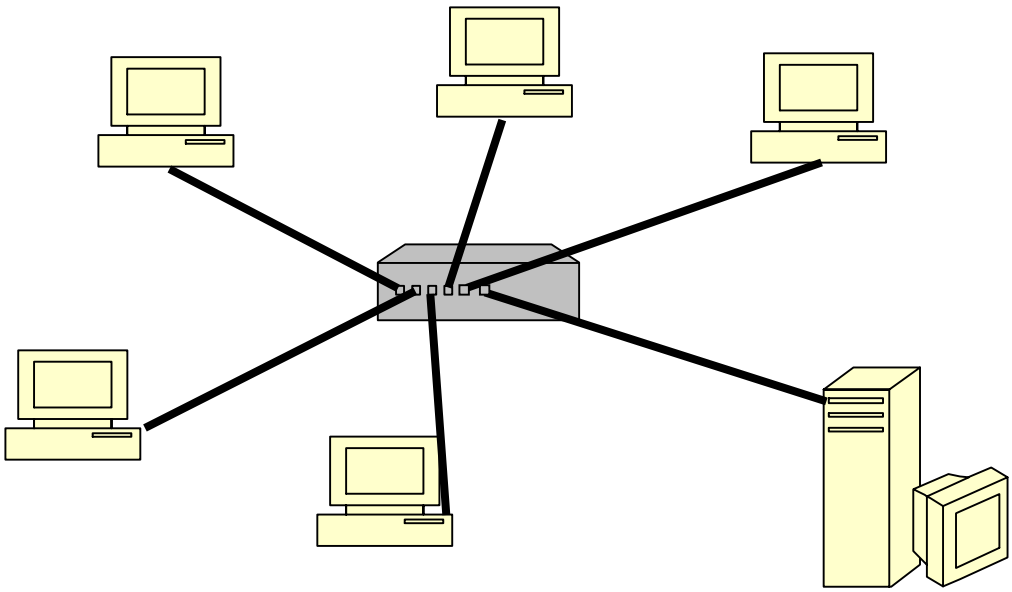
Csillag-topológia

A csillag-topológiában minden hoszt egy központi HUB-hoz (elosztóhoz) csatlakozik, amelyen keresztül valamennyi hoszt kommunikálhat bármelyik másik hoszttal. Egyszerre csak egy hoszt lehet aktív forgalmazó, a többinek meg kell várnia, míg az éppen adó befejezi, majd eltérő várakozási idő után újból megvizsgálják, hogy adhatnak-e, és az első szabad jelzést érzékelő elkezdi a csomagküldést.
Ez a topológia jól skálázható (azaz a megnövekedett igényeket viszonylag egyszerű bővítésekkel ki lehet elégíteni), bár ha a csillagágak száma meghaladja a HUB fogadóképességét, akkor a HUB bővítésére vagy cseréjére van szükség.

E topológia előnye, hogy egy hoszt kiesése nem zavarja a többi működését, és viszonylag könnyű a hibák felderítése. Hátránya, hogy a centrumban lévő HUB működési zavara az egész hálózatot lebénítja. További negatívum, hogy több kábel szükséges a kialakításához, mint a gyűrű- és a busz-topológia esetében.

Ez ma a leggyakrabban alkalmazott topológia, melyet CAT5-ös kábelezéssel valósítanak meg.
Fontos felhívni a figyelmet azonban, hogy a napjainkban használt technika fizikailag ugyan csillag-topológia, mivel minden hálózati végpont össze van kötve egy külön vezeték segítségével a HUB-bal vagy a switch-csel, ám logikailag busz-topológiát valósítanak meg, mivel üzenetszórásos technikával minden hálózati végpont megkapja az üzenetet.

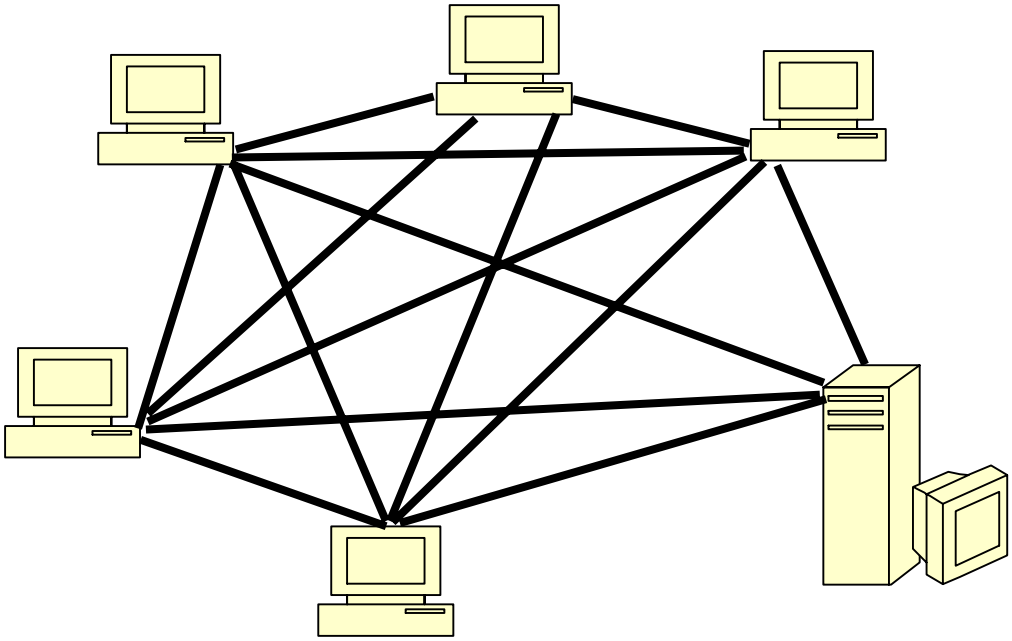
Az igazi csillag-topológia a ’70-es évek végétől használt ún. ArcNet-hálózat volt.



3. ábra
Csillag-topológia

Háló-topológia

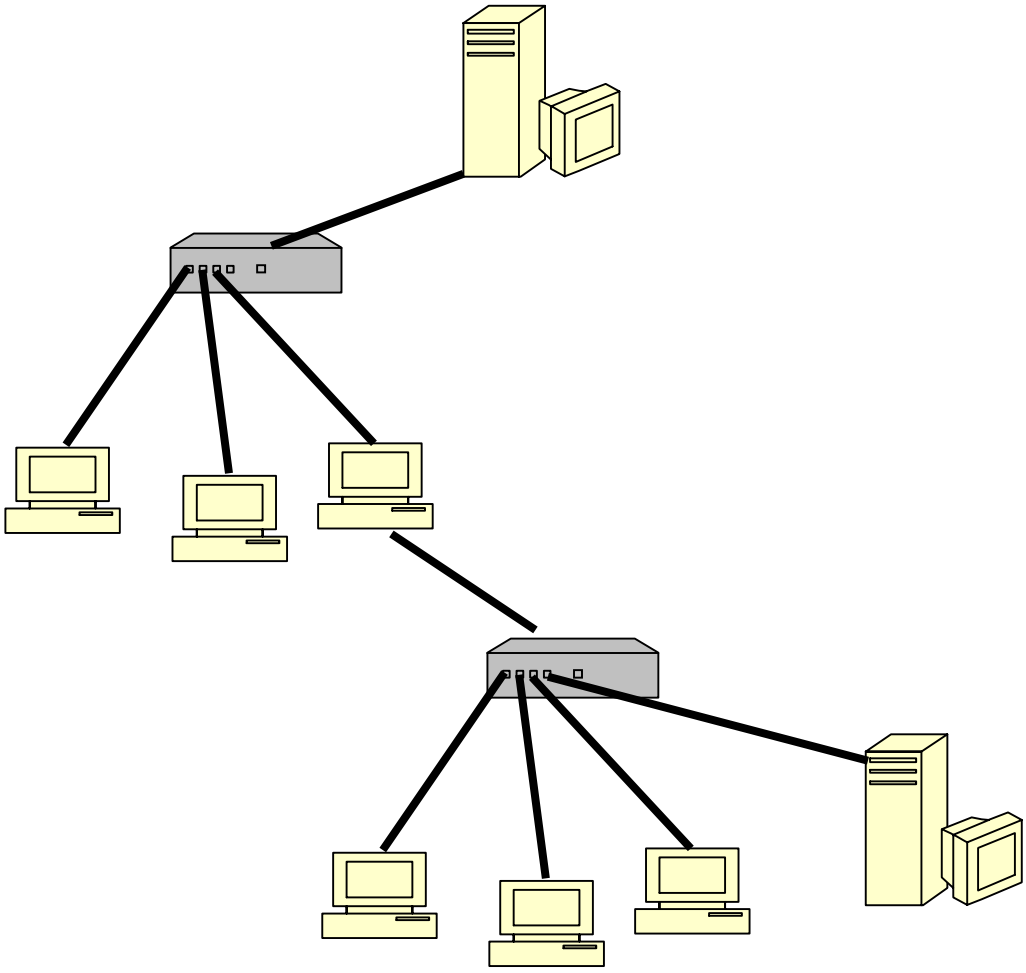
A háló-topológiában minden hoszt össze van kötve azzal a hoszttal, amelyikkel kommunikálni kíván, ezért rendkívül jó üzemelési biztonságot nyújt, nagy a hibatűrése. Jelentős hátránya viszont, hogy már néhány gép esetén is rendkívül nagy a hardverigény (hálózati csatoló és kábel). Ezért jobbára csak több szervert tartalmazó hálózat részeként, a szerverek összekötésére használják.
A háló-topológia egy speciális változata, mikor minden hoszt össze van kötve a többi hoszttal: ezt teljes topológiának nevezzük.



4. ábra
Teljes topológia

Fa-topológia

A pont-pont topológia ezen típusánál a hosztok a fa ágaihoz hasonlóan kapcsolódnak egymáshoz. A kapcsolódás egyfajta hierarchiát is jelent.
Szemléletes példa a topológia működésére az a módszer, ahogy az Internet DNS szerverek kapcsolódnak egymáshoz.
Ha egy számítógép az Internetről információt szeretne kérni, tudnia kell a célgép IP-címét. Ezért a gépünk (a fa legalján, mint egy levél) kérést intéz az egy szinttel felette található ún. névkiszolgálóhoz. Ez a névkiszolgáló nagy valószínűséggel még nem tud választ adni a kérdezőnek, ezért ő is a felette lévő kiszolgálót kérdezi, hasonlóan ahhoz, mint mikor egy vállalatnál a beosztott megkérdezi a főnököt valamiről, aki az ő főnökéhez fordul stb. mindaddig, míg a vállalat csúcsvezetőjéhez nem jut a kérdés.
Itt sem szokás minden aprósággal a „nagyfőnökhöz” rohanni. A fa elágazásai a központi gépek, vagy a vállalati hasonlatban a vezetők.



5. ábra
Fa-topológia

1.2. Üzenetszórásos csatornával rendelkező alhálózatok (multipont-összeköttetés)

Az üzenetszórásos összeköttetésnél **egyetlen közös üzenettovábbító csatorna** található, és az egyes hosztok erre kapcsolódnak, erre vannak felfűzve. Ha bármelyik gép küld egy üzenetet (egy csomagot) – a csatorna közös volta miatt – azt minden állomás veszi. Annak érdekében, hogy az adott hoszt fel tudja ismerni, hogy az információ neki szól-e, minden gépnek és eszköznek **egyedi címe** van. Ezt a címet természetesen el kell helyezni a küldendő információban (a csomagban). A hosztok ezt a címet ellenőrzik, és csak akkor foglalkoznak az üzenettel, ha nekik szól. Lehetőség van állomások csoportjainak közös címet adni, így a csoportba tartozó összes hoszt veszi az üzenetet, nem kell mindegyiknek külön-külön elküldeni. Ennek a módszernek a neve **csoportcímzés**, vagy idegen szóval multicasting.

Az üzenetszórásos rendszerek általában lehetővé teszik olyan üzenetek küldését is, amely a hálózat minden gépének szól. Ehhez egy **speciális címkód** szükséges. Ezt a módszert broadcastingnak nevezzük.

A közös csatorna használata miatt (nagyobb gépszám esetén) a kialakítás kevesebbe kerül a multipont-topológia esetében, mint a pont-pont összekapcsolásnál. A közös csatorna azonban nemcsak előnyökkel jár. Problémát jelent, hogy előfordulhat olyan eset, amikor egyszerre több állomás szeretne üzenetet küldeni a csatornán. Mivel a csatorna közös, az üzenetek összekeverednek, és az összekeveredett üzenetek értelmezhetetlenek. Ezt a jelenséget **ütközés**nek nevezzük. A probléma kivédéséhez különböző közeghozzáférési eljárásokat dolgoztak ki, amely feloldja a versenyhelyzetet a csatorna igénybevételéért, és ezzel együtt az ütközést is.

A közeghozzáférési eljárásokról a későbbiekben bővebben olvashatunk.

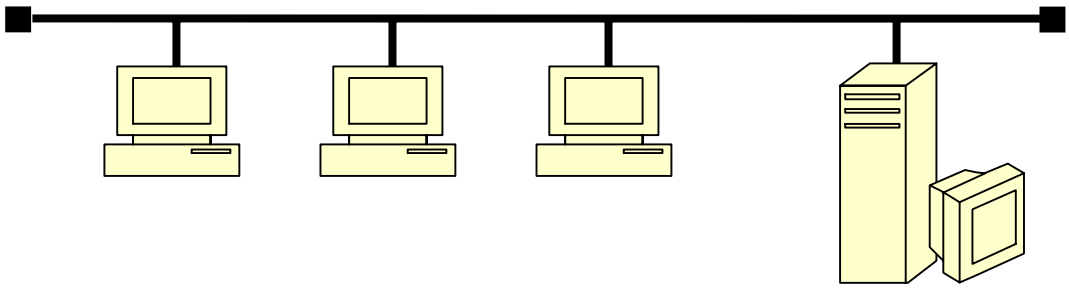
Az üzetnetszórásos összeköttetés is (hasonlóan a pont-pont összeköttetéshez) többféle topológia segítségével valósulhat meg. Ilyen topológiák: busz- vagy sín-topológia, gyűrű-topológia, hibrid (vegyes) topológia.

Busz-topológia

A hosztok egy gerinchálózatra kapcsolódnak, ez a **busz**. Valamennyi adó erre küldi adatait, és minden hoszt csak a neki szóló csomagokat veszi le. A buszon minden adat átáramlik.
Ha a busz megszakad, akkor hiába van két számítógép között fizikai kapcsolat, az adatkapcsolat mégsem üzemel.

A gerinc nyomvonalát úgy kell megtervezni, hogy a lehető legrövidebb legyen, és érintenie kell minden gépet. Az ilyen hálózat megvalósítása viszonylag olcsó, mert kevés hardverelemet igényel, továbbá megépítése is viszonylag egyszerű. Hátránya a fokozott hibalehetőség (takarításnál az irodában mindig leszedik a csatlakozókat, mert a gép alja annyira porosodik, és nem értik, hogy miért idegesek ilyenkor a többiek). Az esetleges hibahely is viszonylag nehezen határozható meg. Ilyen eset lehet egy hamis csomagokat gyártó hálózati kártya megtalálása, mely lényegében ellátja feladatát, csak éppen telíti a buszt senkinek nem szóló csomagokkal.
Ennek a topológiának a legismertebb megvalósítása a ’80-as–’90-es években elerjedt koax-kábeles Ethernet hálózat volt. Ez a hálózat 10 Mbit/s átviteli sebességre volt képes, a hossza maximum 150-180 m lehetett.

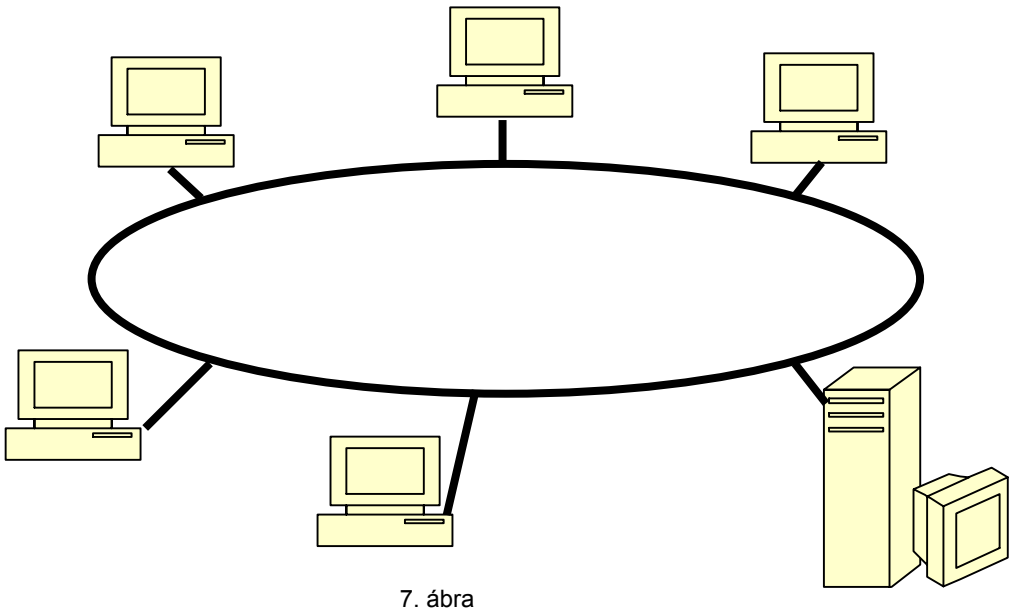
A 10 Mbps-os Ethernet sem túl gyors, de még ez az adatátviteli sebesség is erősen romlik a 10 fölötti gépszám növelésével.
Tehát alacsony gépszám (tíz alatt), szerény adatátviteli igények és egyszerűbb üzembiztonság esetén javasolt ez a topológia.



6. ábra
Busz-topológia

Gyűrű-topológia

A gyűrű-topológiának két típusát különböztethejük meg. Beszélhetünk gyűrű-topológiáról pont pont kapcsolat és üzenetszórásos kapcsolat esetében is. A mai hálózatok szempontjából egyértelműen az üzenetszórásos gyűrű-topológia a fontosabb.
A gyűrű-topológiánál a gerincvezeték egy **logikai gyűrűt** alkot, ehhez kapcsolódnak a hálózat hosztjai, és ezek képesek újból kisugározni a nem nekik szóló csomagokat, tehát a jel nem gyengül, azaz nem kell ismétlőket alkalmazni. Lezárókra nincs szükség.
A gyűrű-topológia könnyen megvalósítható és csekély a hardverigénye.
Hátránya a nehézkes karbantartás, mivel egy gép kiesése a teljes hálózati tevékenységet – ha rövid időre is – leállítja.

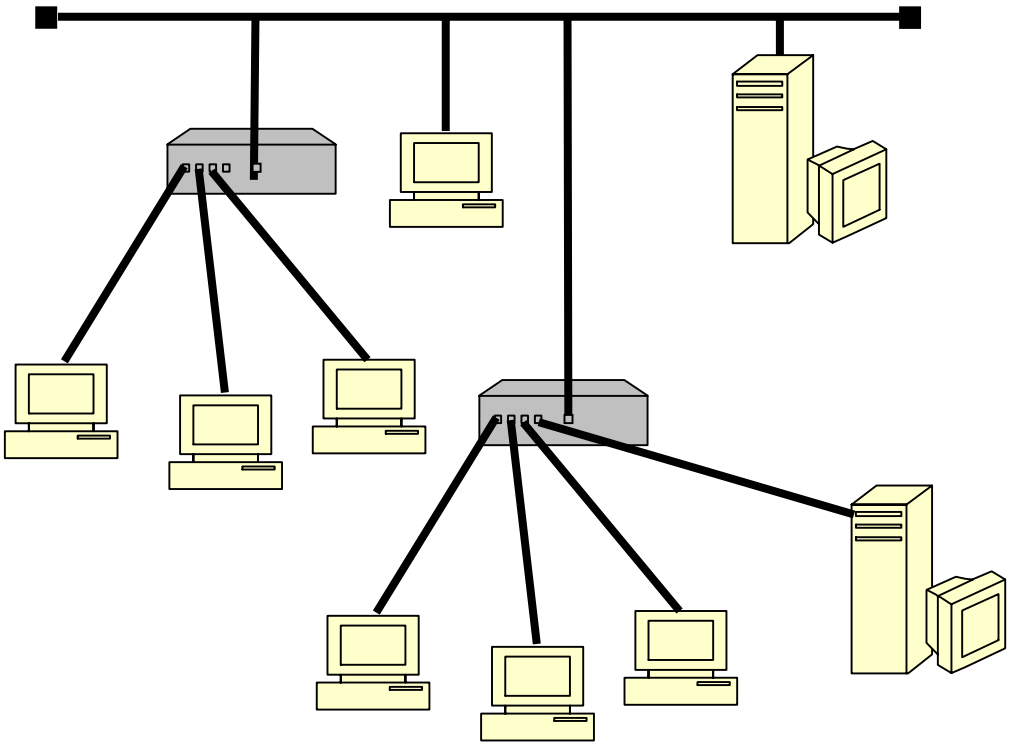


7. ábra
Gyűrű-topológia

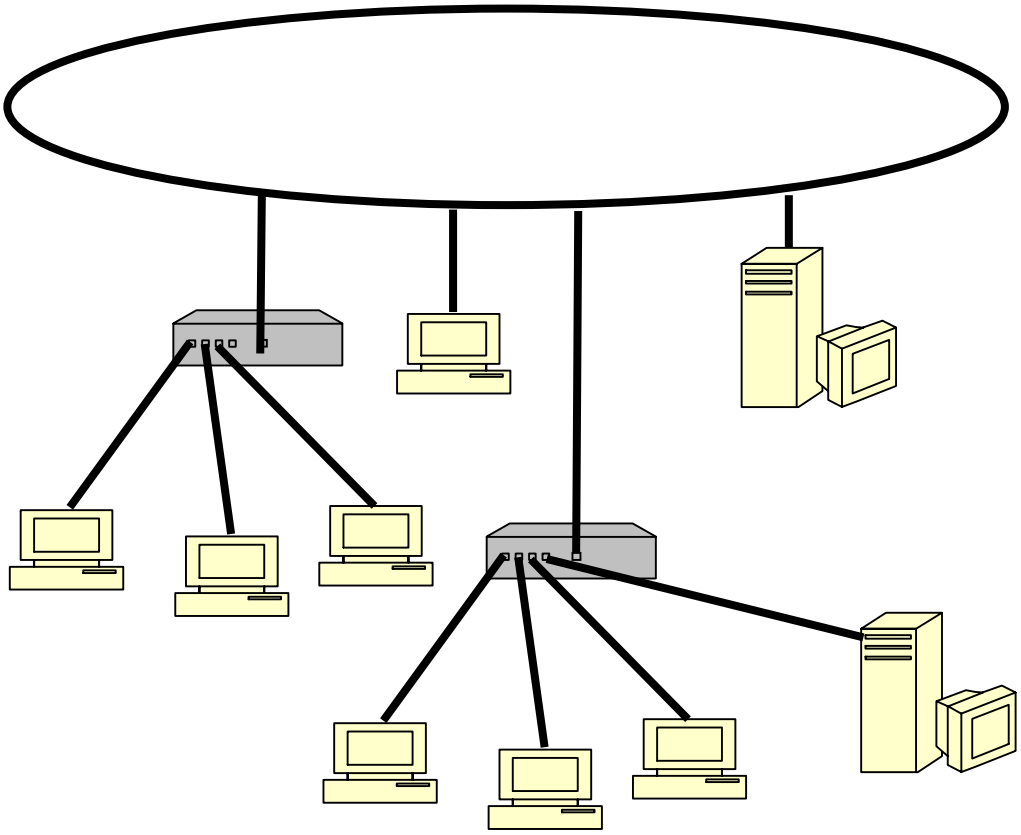
Hibrid topológiák

Lényegében bármely elemi topológia előfordulhat egy teljes hálózat részeként, leggyakoribb mégis a busz és a csillag együttes alkalmazása.
Általában elmondható, hogy a vegyes topológiák egyfajta optimumszámítás eredményei, amelyben az egyes elemi topológiák előnyei és hátrányai ötvöződnek a legjobb teljesítmény-ár viszony elérése érdekében.

Akkor szoktak a leggyakrabban vegyes topológiát használni, ha egy már meglévő, de régebbi, vagy eltérő topológiájú hálózathoz kell csatlakoztatni egy új hálózati egységet (pl. egy patinás egyetemen új épületszárnyat építenek, és ezt kell a teljes rendszerhez csatlakoztatni, vagy egy bankfiók meglévő hálózatát kellett az országos rendszerbe bekötni).



8. ábra
Busz-csillag topológia



9. ábra
Gyűrű - csillag topológia

2. Hálózati architektúrák

Kezdetben a számítógép-hálózatoknál a hardver jelentette a fő problémát, a szoftverekre csak a rendszerek egyre bonyolultabbá válása után terelődött a figyelem. Napjainkra a hálózati szoftverek nagymértékben strukturálódtak.

Annak érdekében, hogy csökkentsék a hálózatok bonyolultságát, a legtöbb hálózatot **rétegek**be (layers) vagy **szintek**be (levels) szervezik. Minden réteg vagy szint az alatta levőre épül. A rétegek száma, neve, tartalma és funkciója különböző hálózatban eltérő lehet. Az viszont minden hálózatban közös, hogy az egyes rétegek célja szolgálatok nyújtása a felettük levő réteg számára, oly módon, hogy közben a szolgálatok megvalósításainak részleteit azok elől elrejtse. Amikor azt mondjuk, hogy két gép kommunikál egymással tulajdonképpen a rétegek kommunikációját értjük alatta, mégpedig úgy, hogy az egyik gép k-adik rétege a másik gép k-adik rétegével tartja a kapcsolatot. A kapcsolatban minden réteg az alatta levő rétegnek vezérlőinformációkat és adatokat ad át.

A párbeszéd szabályait minden rétegre meghatározták és azt a réteg **protokoll**jának (protocol) nevezzük. A protokoll lényegében olyan megállapodás, amely az egymással kommunikáló felek közötti párbeszéd szabályait rögzíti.

Az egyes rétegek és közöttük folyó kapcsolati szabályok összességét **hálózati architektúrá**nak nevezzük.

Az architektúra megtervezése során a következő szempontokat kell figyelembe venni:

- Az **egyidejű adatáramlás** szempontja:

  - **Egyirányú** vagy **szimplex** adatátvitel: az adatok a két hoszt között csak egy irányba haladhatnak. A rendszer minden esetben rendelkezik adóval és vevővel, de ezek szerepet nem cserélhetnek (ilyen pl. a televízióadás).

  - **Félduplex** vagy váltakozva kétirányú adatátvitel: ebben az esetben az adatok a két állomás között már felváltva is áramolhatnak, azonban egyszerre csak az egyik irányba. A csatornát egyszerre csak az egyik irányú adatáramlás foglalhatja le (pl. CB rádió).

  - **Duplex** vagy kétirányú adatátvitel: az adatok egyidejűleg mindkét irányban áramolhatnak a két hoszt között. Ezzel a technikával egy hoszt adó és vevő funkciót is el tud látni egyszerre (pl. telefon).

- Minden rétegnek tartalmaznia kell olyan **funkcionális rész**t, mely alkalmas a kapcsolat felépítésére, majd az információcserét követő kapcsolatbontásra.

- **Hibavédelem** vagy **hibajelzés**: használja-e a rendszer, és ha igen, akkor az hogyan működjön?

- Szükség lehet az **adatátvitel szabályozásá**ra, ugyanis egy hálózatban az egyes hosztok különböző fizikai paraméterekkel rendelkezhetnek. A lassú és a gyors működésű számítógépeknek is tudniuk kell egymással kapcsolatot létesíteni. Ha a vevő gyorsabb működésű, mint az adó, az nem jelent problémát. Ha a helyzet fordított, akkor az már adatvesztéshez vezethet, ugyanis az adó gyorsabban adja a vonalra az információt, mint ahogy azt a vevő fogadni képes. Annak érdekében, hogy ez a helyzet ne állhasson elő, az adatátvitelt szabályozni kell, ezt nevezzük **folyamirányítás**nak, vagy angolul flow control-nak.

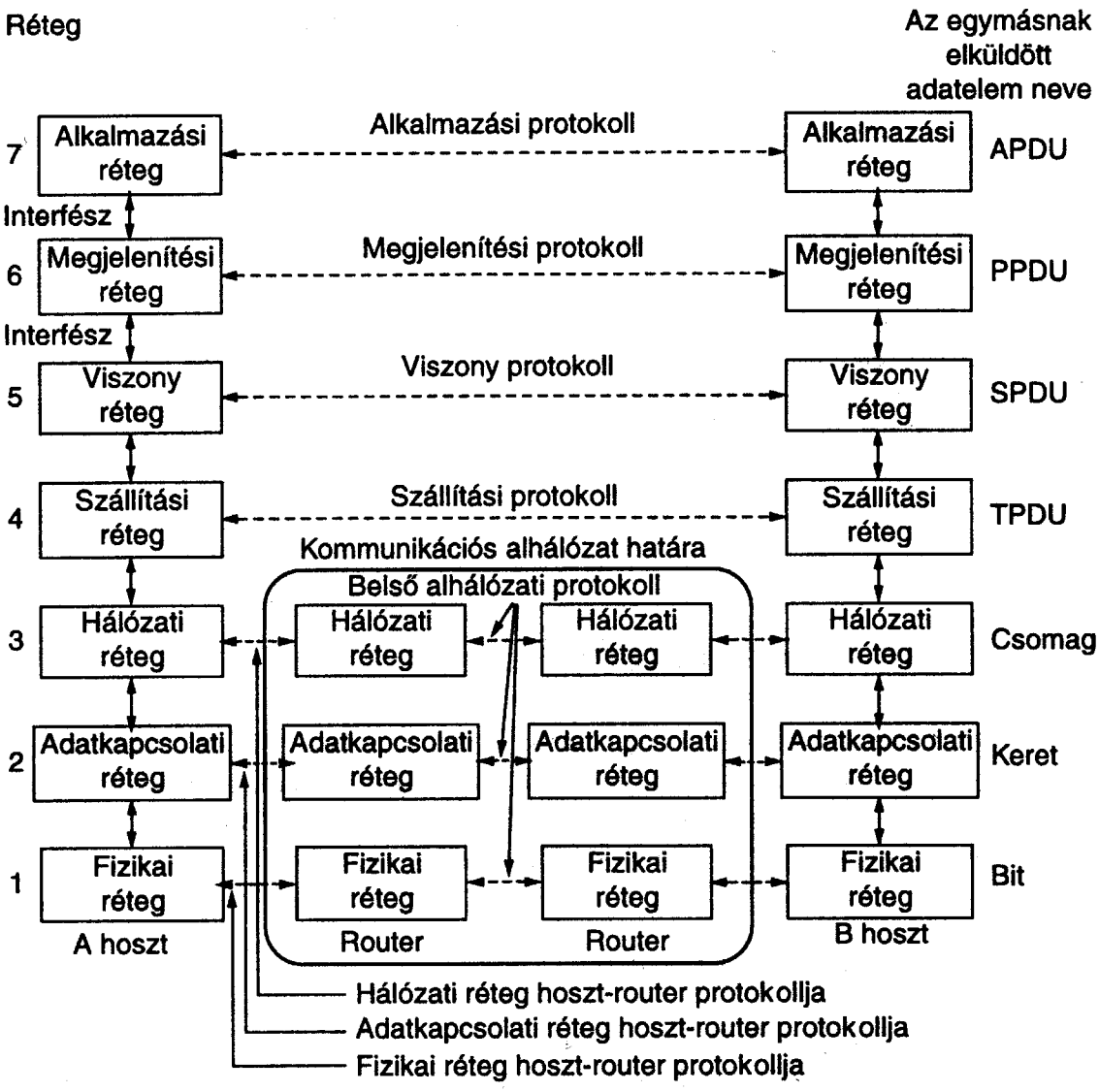
- Ha az üzenet nagy méretű, továbbításakor szükség lehet a **méret korlátozásá**ra. Ilyenkor a nagyobb méretű információcsomagokat fel kell darabolni kisebb részekre, és egyenként átküldeni a címzetteknek. Az így továbbított összes kisebb csomag célba juttatása után a címzett feladata ezek összeállítása, sorrendbe rakása. Ennek érdekében a csomagok fejrészében azonosítót kell elhelyezni, amely tartalmazza a csomag sorszámát is.

- A számítógépeink közötti kapcsolat legnagyobb költségét a kábelek kiépítése, azaz az adatátviteli vonal kialakítása jelenti. A vonal jobb kihasználására kell törekedni, és ennek növelése érdekében meg lehet és meg kell valósítani a **vonalmegosztás**t.

- A lehető leggyorsabb átvitel biztosítása érdekében törekedni kell a legrövidebb vagy a leggyorsabb **útvonal kiválasztásá**ra, mivel ha több gépünk van, általában két hoszt között nem csak egy útvonalon haladhatnak az adatok. Már a tervezés során meg kell határozni az útvonalak meghatározásának módját (útvonalvezérlés, útvonalválasztás). Ezt a feladatot a csomóponti számítógépeken futó megfelelő programok vagy célhardverek valósíthatják meg valamilyen algoritmus szerint.

3. Hálózatszabványosítás

A hálózati rétegek megvalósítására a **Nemzetközi Szabványügyi Szervezet** (International Standard Organisation, ISO) kidolgozott egy ajánlást, illetve definiált egy hivatkozási modellt, amely a csomópontok közötti kommunikáció folyamatát írja le. Az ajánlás a „Nyílt rendszerek összekapcsolása” (Open System Interconnect) nevet kapta. Az angol elnevezés szavainak kezdőbetűiből származik a modell elterjedt neve, az **OSI.** Az OSI-modell hét réteget határoz meg, melyből az alsó három réteg jellemzően a számítógép hardverével kapcsolatos, a felsőbb négy réteg megvalósítása viszont szoftverfeladat.



10. ábra
Az OSI-modell

A modell hét rétege különböző, jól meghatározott feladatot lát el. A hosztok közötti kommunikáció során csak a megfelelő rétegek kommunikálnak egymással a rétegek protokolljainak segítségével. Két réteg közti kommunikációt a réteginterfész segíti elő. Az egyes rétegek feladatai a következők:

3.1. Fizikai réteg (physical layer)

Ez a legalsó réteg kezeli közvetlenül a fizikai átviteli közeget, bitenként küldi és veszi az információt, felelős a tényleges fizikai kommunikáció megvalósításáért. A fizikai közeg sokféle lehet, ennek megfelelően a továbbítandó 0-khoz és 1-esekhez hozzárendeli a közegen továbbítható jeleket (feszültség, áram, fényimpulzusok stb.). Ebben a rétegben kell azt is meghatározni, hogy a kommunikáció csak egy vagy mindkét irányba történhet-e, és ha kétirányú, akkor váltakozva kétirányú vagy valódi kétirányú-e.

3.2. Adatkapcsolati réteg (data link layer)

Ez a réteg a fizikai réteg felett helyezkedik el, feladata a vonal két vége között az információ megbízható továbbítása. Biztosítania kell, hogy az adóoldali adatok a vevőoldalra is adatként jussanak el, és ne legyen belőlük értelmetlen jelek sorozata.
Ennek érdekében a réteg az átküldendő információt egyértelműen azonosítható adatkeretekre tördeli, ellátja a szükséges vezérlőbitekkel, majd sorrendben továbbítja azokat. A vevő a cím felismerése után az információt feldolgozza, majd ún. nyugtakerettel közli az adóval ezt a tényt.

3.3. Hálózati réteg (network layer)

A hálózati réteg az adatkapcsolati réteg felett helyezkedik el, és alapfeladata az adatkapcsolati réteg által elkészített keretek forrás- és célállomás közti útvonalának meghatározása, azaz a forgalomirányítás. Ez néha meglehetősen bonyolult feladat. A hálózat általában több alhálózatból áll, melyek felépítése is összetett lehet. Ilyen alhálózatokban két hoszt között több lehetséges útvonal is kialakítható. Természetesen ezek hossza, valamint a sebessége is jelentős mértékben eltérhet. A hálózati réteg az útvonalválasztás több lehetséges módját alkalmazhatja:

- statikusan: az útvonalak fixen meghatározottak;
- dinamikusan: ilyenkor az útvonalválasztáshoz a hálózat aktuális helyzetét térképezik fel. Ezzel a módszerrel figyelembe vehető a hálózat terhelése is. Természetesen igaz az, hogy két keret, amelynek ugyanaz a forrás- és célállomása is, nem biztos, hogy ugyanazon az útvonalon keresztül jut el a rendeltetési helyre, hiszen a hálózat pillanatról pillanatra változik. A vételi oldalon a keretek sorrendbe rakása a vevő feladata.

Ennek a rétegnek a feladata az is, hogy feloldja azt a helyzetet, amely az alhálózatban valamilyen ok miatt felszaporodott csomagok esetén jelentkezik.
Egy adatkeret útja során eltérő felépítésű hálózatokon is keresztülhaladhat. Az ilyen, ún. heterogén hálózatok összekapcsolásakor jelentkező problémát is a hálózati réteg szintjén kell megoldani.

3.4. Szállítási réteg (transport layer)

Ennek a rétegnek kell megoldania a hosztok közötti adatátvitelt. A hálózati réteg felett elhelyezkedő réteg biztosítja azt, hogy minden adat sértetlenül érkezzen meg a rendeltetési helyére. Ha szükséges, akkor ez a réteg vágja az üzenetet kisebb darabokra, ún. csomagokra, majd a csomagokat átadja a hálózati rétegnek. A hálózat két összekapcsolandó

gépe között szinte mindig vannak közbenső számítógépek (csomópontok) is. A szállítási réteg feladata annak a megvalósítása, hogy a két hoszt ezt a tényt „ne vegye észre”, tehát az összeköttetés pont-pont jellegű legyen. Ezen a rétegszinten a forrás- és a célállomás egymással kommunikál, míg az alsóbb rétegek szintjén a hosztok a szomszédjukkal folytatnak párbeszédet. Így a réteg ellenőrizni tudja, hogy hibátlan volt-e az adatok átvitele a teljes útvonalon.

3.5. Együttműködési vagy viszonyréteg (session layer)

A számítógépek a kommunikáció során kialakítanak egy viszonyt egymás között. Ilyen viszony lehet a bejelentkezés egy terminálról egy távoli számítógépre, vagy adattovábbítás két gép között. Az egyszerű adatátvitelt kiegészíti néhány praktikus szolgáltatással, például:

A kölcsönhatás-menedzselés, ami vezérli, hogy a két oldal egyszerre ne próbálkozzon ugyanazzal a művelettel. Ez úgy oldható meg, hogy vezérlőjelet tartanak fent, és csak az az oldal végezheti az adott műveletet, amelyiknél ez a vezérlőjel van.
A szinkronizáció egy másik fontos szolgáltatás. Egy nagyméretű, hosszú időt igénylő állomány továbbítása a hálózat valamilyen gyakran előforduló hibája miatt szinte lehetetlen lenne, ha a hálózati hiba miatt az átvitelt mindig az elejétől kellene kezdeni. Ha az adatfolyamba megfelelő számú ellenőrzési pontot iktatnak be, a hiba megszűnése után az átvitelt csak az utolsó ellenőrzési ponttól kell megismételni.

3.6. Megjelenítési réteg (presentation layer)

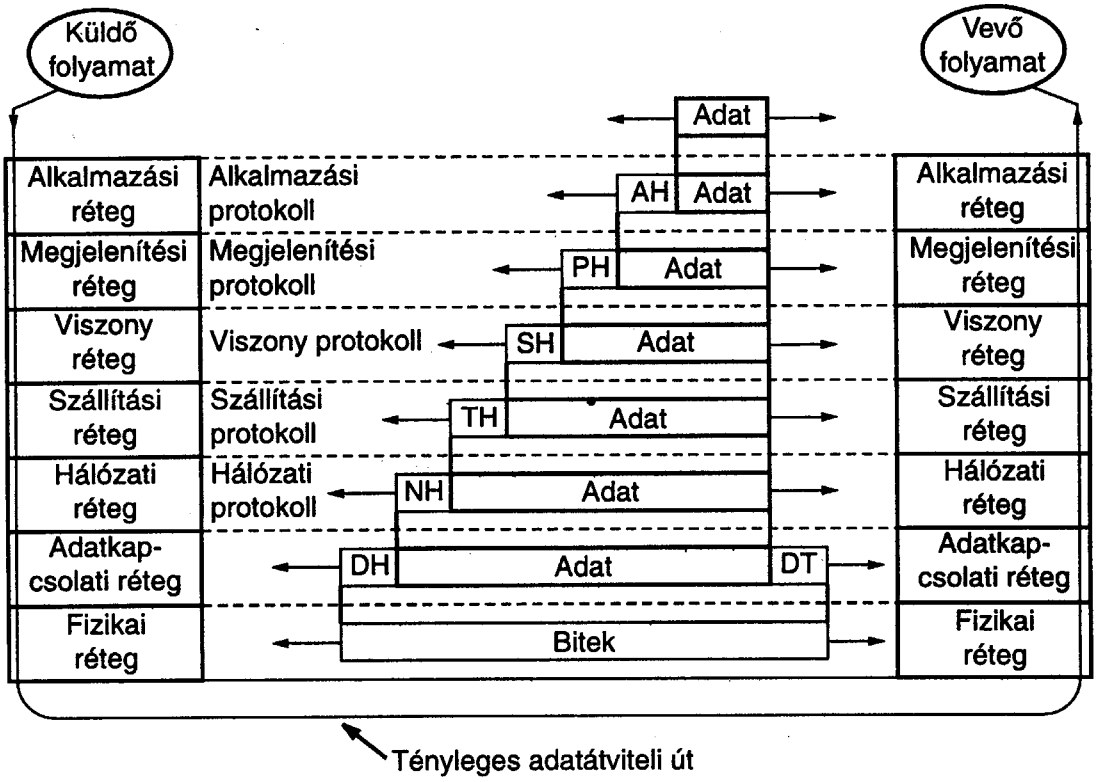
A viszonyréteg fölött helyezkedik el, és olyan szolgáltatásokat ad, amelyekre a legtöbb alkalmazói programnak szüksége van, amikor a hálózatot használja. Foglalkozik a hálózaton továbbítandó adatok ábrázolásával, hiszen munkánk során általában nem bináris számokkal dolgozunk, hanem annak valamilyen, az ember számára értelmezhetőbb megjelenési formájával. Ez azt eredményezi, hogy az egyes információk más és más formában jelennek meg. A megjelenítési réteg feladata az eltérő megjelenésű formájú adatok egységes kezelése. Fontos egységes adatstruktúrákat meghatározni és kialakítani, melyeknek a kezelését végzi ez a réteg. Itt lehet megvalósítani az adatok tömörítését és titkosítását.

3.7. Alkalmazási réteg (application layer)

Ez a legfelső réteg kapcsolódik a legszorosabban a felhasználóhoz. Ehhez tartoznak a felhasználói programok által igényelt protokollok. Az alkalmazási réteg léte a feltétele annak, hogy a különböző programok a hálózattal kommunikálhassanak. Többek között a réteg feladata pl. az elektronikus levelezést, az állománytovábbítást, a terminálemulációt irányító protokollok meghatározása.

Adatátvitel az OSI hivatkozási modellben

A rétegek között a tényleges adatátvitel valójában függőleges irányban történik, azonban az egyes rétegek úgy működnek, mintha vízszintes irányban továbbítanának az adatokat.
Ha egy küldő folyamat a vevő folyamatnak adatokat akar küldeni, az adatokat az alkalmazási rétegnek átadja, amely az adatok elé ún. fejrészt (headert) illeszt, majd az így kapott egységet továbbadja a megjelenítési rétegnek. (Az ábrában az AH: Application layer header, PH: Presentation layer header stb.)
A megjelenítési réteg a kapott egységet átalakíthatja, esetleg ő is kiegészíti egy fejrésszel, majd továbbadja a viszonyrétegnek. Az alsóbb rétegek nem vizsgálják, hogy amit a felettük levőtől kaptak, annak melyik része a fej-, illetve a valódi felhasználói adatrész. A további rétegek is hasonlóan viselkednek, és a folyamat egészen addig ismétlődik, amíg az adatok el nem jutnak a fizikai rétegig, ahol aztán valóban továbbítódnak a vevőoldali géphez.
A vevőoldali gépen – ahogy az üzenet az egyre magasabb rétegekhez kerül – az egyes rétegekben a különböző fejrészek leválasztódnak róla, végül megérkezik a vevő folyamathoz.



11. ábra
Adatátvitel az OSI hivatkozási modellben

A fejrészekkel kiegészített adatkeretek csak az azonos szinten lévő rétegek számára értelmezhetők. Az értelmezés szabályait a réteg **protokoll**ja írja le.

Ahhoz, hogy az adatkeretek eljussanak a legfelső rétegtől a legalsó rétegig - majd a vevő esetében a legalsó rétegtől a legfelsőig – igénybe kell venni a szomszédos rétegek ún. **szolgáltatásai**t.

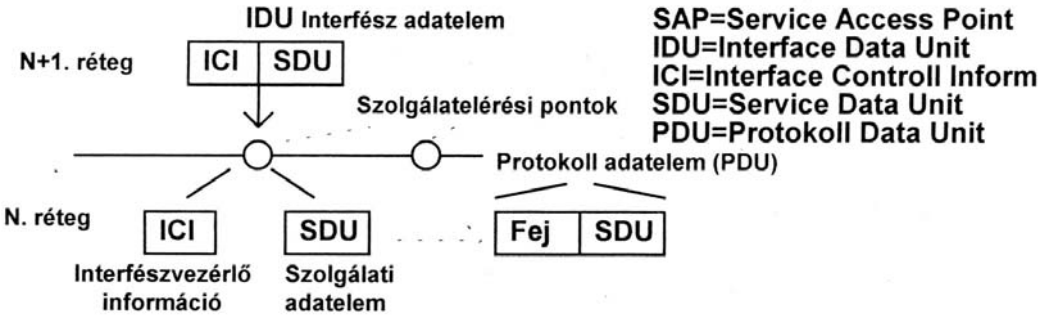
Az egymással szomszédos rétegek között **interfész** található. Az interfész azt definiálja, hogy az alacsonyabban levő réteg milyen elemi műveleteket és szolgálatokat nyújt a magasabban levő réteg számára. A **szolgálat** tehát olyan elemi műveletek (primitívek) halmaza, amelyet egy adott réteg a felette levő rétegek számára biztosít. A szolgálat két szomszédos réteg közötti interfésszel kapcsolatos, ahol az alsó réteg a szolgáltató, a felső réteg pedig a szolgálat felhasználója. A rétegtől elvárt funkciókat a rétegben aktív, működő elemek – ún. **funkcionális elem**ek (más, elterjedt néven: **entitás**ok) – valósítják meg. Ez lehet például egy hardverelem vagy egy program.

A rétegek és protokollok halmazát **hálózati architektúrá**nak (network architecture) nevezzük.

A rétegek feladatainak jó meghatározása és a közöttük átadandó információmennyiség minimalizálása a feltétele annak, hogy tiszta és egyszerű interfészek legyenek kialakíthatók a rétegek között. Ez pedig azért fontos, mert ez alapozza meg azt, hogy egy adott réteg implementációját egyszerűen cserélhessünk egy teljesen újra. Az új implementációval szemben csak annyi az elvárás, hogy pontosan ugyanazokat a szolgálatokat nyújtsa a felette levő rétegnek, mint az előző (pl. amikor a telefonhálózatot felváltjuk műholdas csatornákkal).

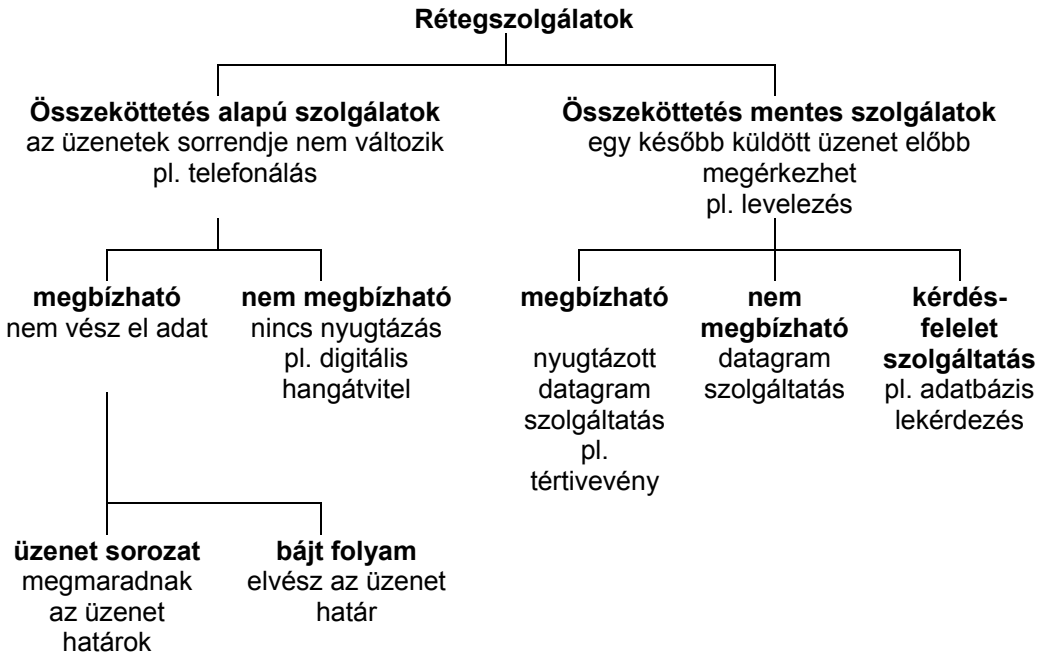
4. Szolgálatok a rétegek között

*A rétegek közötti kommunikáció a **szolgálatok** segítségével valósul meg. A szolgálatok a rétegek ki/bemeneti pontján: ún. **SAP**-ján (Sevice Access Point) keresztül érhetők el. Ezek mindig két szomszédos réteg között találhatók. Lényegében a két réteg közötti kommunikáció ténylegesen ezeken a pontokon keresztül valósul meg.*



12. ábra
Kapcsolat a rétegek között

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



13. ábra
Rétegszolgálatok osztályozása

4.1. Összeköttetés-alapú szolgálat

Az összeköttetés-alapú szolgálat lényegét a hagyományos távbeszélőrendszer működésével modellezhetjük. Ha valakivel beszélgetést akarunk kezdeményezni, akkor először felemeljük a kagylót, majd tárcsázunk, ezután van lehetőség a beszélgetésre, majd a beszélgetés végén letesszük a kagylót.
Egy összeköttetés-alapú hálózati szolgálat igénybevétele is hasonló fázisokból áll. A szolgálatot igénybe vevő felhasználó először létrehozza az összeköttetést, majd felhasználja, végül pedig lebontja azt. Az összeköttetés lényege az, hogy úgy működik, mint egy cső: az adó a "cső" egyik végén belerakja a biteket, a vevő pedig a másik végén ugyanabban a sorrendben kiveszi azokat. Ez azt jelenti, hogy amilyen sorrendben küldjük az információt, a vevő pontosan abban a sorrendben kapja meg. Az összeköttetés kialakítása jelentős időt vesz igénybe, így sok esetben csak akkor célszerű alkalmazni, ha nagyobb mennyiségű információt akarunk átvinni.

4.2 Összeköttetés-mentes szolgálat

Az összeköttetés nélküli szolgálat a levéltovábbító postai rendszerrel modellezhető. Minden egyes üzenet (levél) rendelkezik a célcímmel, és az egyes üzenetek akár a többitől független útvonalon is továbbíthatóak. Ilyenkor elképzelhető, hogy a részekre bontott információt a vevő nem az adó által küldött sorrendben kapja meg. Ahhoz, hogy a vevő oldalon az információ értelmezhető legyen, a csomagokat helyes sorrendbe kell rakni.

Az átvitel során természetesen az a cél, hogy az információ veszteség nélkül célba érjen. A szolgálatok jellemezhetők egy szolgálati minőséggel (Quality of Service).

Vannak **megbízható szolgálatok**, amelyek sosem vesztenek el adatot. Egy megbízható szolgálatot rendszerint úgy valósítanak meg, hogy a vevőnek minden megkapott üzenetet nyugtáznia kell, így a küldő biztos lehet abban, hogy az üzenet megérkezett. A nyugtázási folyamat időt igényel, késleltetést jelent; legtöbbször megéri, de van olyan eset, amikor nemkívánatos. A telefonon beszélgetők számára sokkal inkább elfogadható az, hogy egy kis zajt, torzulást halljanak, mint az, hogy a nyugtázások miatt késleltetés jelenjen meg a vonalon. Videofilm továbbításakor néhány hibás pixel vagy akár képkocka nem jelent nagy problémát, viszont elfogadhatatlan, ha a folyamatos megjelenítés a javítás miatt meg-megakad.

Sok esetben csak arra van szükség, hogy legyen olyan lehetőség az üzenetek elküldésére, ami nagy valószínűséggel célba juttatja azokat, de erre garanciát nem vállal. A nem megbízható (azaz nem nyugtázott) összeköttetés nélküli szolgálatot **datagram-szolgálat**nak (datagram service) nevezik, a távirat analógiájára, amelynél szintén nem lehet nyugtát küldeni a feladónak.

Ha a megbízhatóság alapvető fontosságú, de valamilyen ok miatt nem célszerű összeköttetést létesíteni (pl. egy rövidebb üzenet továbbításához), akkor **nyugtázott datagram-szolgálat**ot (acknowledged datagram service) kell használni, amely megbízható összeköttetés nélküli szolgálat, és a tértivevényes levélkézbesítéshez hasonlít. Ha a nyugta megérkezik a feladóhoz, biztos, hogy a küldemény útközben nem veszett el.

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

A szolgálatot leírhatjuk elemi műveletek (primitívek) halmazával, amelyek a szolgálatot elérhetővé teszik a felhasználó vagy más entitások számára. Ezek a primitívek utasítják a szolgáltatót arra, hogy hajtson végre egy feladatot, vagy számoljon be egy társentitás tevékenységéről. A szolgálat-primitívek az OSI-modellben négy osztályban jelennek meg:

Primitív	*Jelentés*
Kérés	*Valamilyen tevékenység végrehajtásának kérése*
Bejelentés	*A szolgáltató tájékoztat egy eseményről*
Válasz	*Válasz egy eseményre*
Megerősítés	*A kérést kérő informálása*

Egy összeköttetés-alapú szolgálat nyolc szolgálat-primitívből áll:

1. CONNECT.kérés	-	*a hívó összeköttetés létesítését kéri;*
2. CONNECT.bejelentés	-	*a hívó jelez a hívott félnek;*
3. CONNECT.válasz	-	*a hívott fél válasza a hívásra (elfogadja-elutasítja);*
4. CONNECT.megerősítés	-	*közli a hívóval, hogy a kérését elfogadta-e;*
5. DATA.kérés	-	*a hívott az adat küldését kéri;*
6. DATA.bejelentés	-	*a hívott az adat érkezését jelzi a hívónak;*
7. DISCONNECT.kérés	-	*a hívó az összeköttetés bontását kéri;*
8. DISCONNECT.bejelentés	-	*a hívott jelez a hívónak, hogy elfogadta.*

A használt terminológia szerint
CONNECT a kapcsolat létrehozását,
DATA az adatátvitelt,
DISCONNECT a kapcsolat lebontását jelenti.

A szolgálat lehet ***megerősített*** *(confirmed) vagy* ***megerősítetlen*** *(unconfirmed).*
Megerősített szolgálat *esetén van*

- *kérés (request),*
- *bejelentés (indication),*
- *válasz (response) és*
- *megerősítés (confirm) primitív.*

Megerősítetlen szolgálat *esetén csak*

- *kérés (request) és*
- *bejelentés (indication) primitív van.*

A CONNECT mindig egy megerősített szolgálat, mivel a távoli társfolyamattal meg kell egyezni az összeköttetés felépítéséről. Az adatátvitel viszont lehet megerősített és megerősítetlen is attól függően, hogy a küldőnek szüksége van-e nyugtára. A hálózatokban mind a két típusú szolgálatot használni szokták.
A primitívek legtöbbjéhez paraméter is rendelhető. Így például a CONNECT.request paraméterében kijelölhető az összeköttetésre kiválasztott gép, és megadható a kívánt szolgálat típusa vagy a megengedett maximális üzenethossz.

Összefoglalás

A fejezet képet ad a számítógép-hálózatok előnyeiről, így megérthető, hogy miért terjedt el manapság ekkora mértékben ezek használata.
A szakszerű fogalmazás, szóhasználat érdekében tisztáztuk a hálózatokkal kapcsolatos alapfogalmakat.
Fontos tudni, hogy a használt hálózat milyen topológiával rendelkezik, így korrektebb képet tudunk alkotni a hálózat működéséről, esetleges problémáiról.

Ezután következett a fejezet legfontosabb része, melyben az ISO OSI-ajánlást vizsgáltuk meg nagyvonalakban. Ez azért fontos, mert a tananyag további részeiben erre fogunk építeni, az egyes rétegeket pontosabban meg fogjuk ismerni. Természetesen nem egyformán fontosak az egyes rétegek, ezt a róluk szóló fejezetek nagysága is szemlélteti.
Ezek után a rétegek kapcsolatait elemeztük röviden, hogy jobban érthető legyen a rétegezési elv fontossága, működése.

Végül a rétegek közti szolgálatokat csoportosítottuk. Ez a téma is vissza fog köszönni a későbbiekben, tehát fontos a pontos elsajátítása.

Ellenőrző kérdések

1. Ismertesse a hálózatok létrehozásának előnyeit!
   Válasz

2. Ismertesse a hoszt fogalmát!
   Válasz

3. Mi a különbség a pont-pont és az üzenetszórásos összeköttetés között?
   Válasz

4. A gyűrű-topológia:
   - a pont-pont összeköttetés része
   - az üzenetszórásos összeköttetés része
   - a pont-pont és üzenetszórásos összeköttetésnél is van gyűrű-topológia, és ezek azonosak
   - a pont-pont és üzenetszórásos összeköttetésnél is van gyűrű-topológia, de ezek különbözőek

5. Ma melyik topológiát használják leginkább a gyakorlatban?
   - gyűrű
   - busz
   - csillag
   - vegyes

6. Ismertesse az ISO OSI-modell 7 rétegét, és pár szóban jellemezze azokat!
   Válasz

7. Csoportosítsa a rétegszolgáltatásokat!
   Válasz

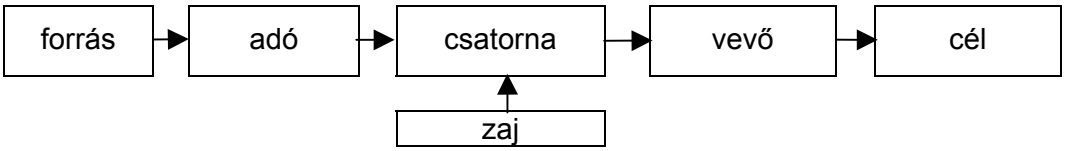
II. Fizikai átviteli jellemzők és módszerek

Bevezető

Ebben a fejezetben az ISO OSI-modell első szintjét részletezzük.
Tisztáznunk kell, hogy milyen adatátviteli közeget használunk, ezeknek milyen típusai, fajtái vannak. A legnépszerűbb vezetékes átviteli közegektől kezdve elemezzük a rádiófrekvenciás átvitelen át a műholdas átviteli közeget is.
Ezek után – a történeti fejlődést követve – megvizsgáljuk az adatátviteli vonalak logikai csatornákra osztásának lehetőségét, a multiplexelést.
A legegyszerűbb hálózatot a számítógépek telefonvonalon keresztül történő összekapcsolásával lehet megvalósítani. Így a fejezet utolsó felében a telefonos adatátvitelről lesz szó.

1. Általános elméleti alapok

Ahhoz, hogy a csatornán információt lehessen átvinni, az adónak meg kell változtatnia a csatorna fizikai közegének valamilyen tulajdonságát, ami a közegen továbbterjed, és a vevő ezt a fizikai közegváltozást érzékeli. Egy vezetékben változhat az átfolyó áram vagy a feszültség, ha antennával kisugárzott elektromágneses hullámot használunk, akkor a hullám amplitúdója, frekvenciája vagy kezdeti fázisszöge.



14. ábra
Az adatátvitel modellje

A jelátvitel fizikai korlátját a közeg fizikai jellemzőinek változása, a változás lehetséges sebessége, a továbbterjedés során fellépő veszteség jelenti, melyek jelgyengüléshez vezetnek.

Egy adott kommunikációs csatornára jellemző adat a **sávszélesség**: ez analóg rendszerek esetén használt fogalom, és egy adott analóg jel maximális és minimális frekvenciájának a különbségét értjük alatta.

A digitális hálózatokat **adatátviteli sebesség**ükkel (az időegység alatt átvitt bitek számával) jellemezhetjük, és bit/s-ban mérjük. Az átvitelt jellemezhetjük a felhasznált jel értékében 1 másodperc alatt bekövetkezett változások számával is, amit jelzési sebességnek vagy közismert néven **baud**-nak nevezünk.

1 baud = log2 P [bit/s], ahol P a kódolásban használt jelszintek száma.

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

1.1. Analóg átvitel

A híradástechnika, telefontechnika átviteli módszereit a múltban teljes egészében az analóg átvitel jellemezte (gondoljunk a telefonra, a rádióra és a televízióra). A jeleket a csatorna valamelyik fizikai jellemzőjének (pl. feszültség, frekvencia, fázis) időben történő folytonos változtatásával vitték át. A kialakított kommunikációs infrastruktúra egy jelentős része ma még analóg. Bár a digitális technika fokozatosan leváltja, még hosszú ideig jelen lesznek ezek a megoldások és rendszerek a műszaki gyakorlatban. A számítógépek összekapcsolására szolgáló távbeszélő hálózatot nem hagyhatjuk figyelmen kívül, így előnyeit és hátrányait érdemes megismerni.

1.2. Analóg jelek átalakítása digitálissá

Eredeti formájukban a hangok analóg, időben és értékben folytonos jelek. Továbbításkor ezek az analóg jelek módosítják (modulálják) a csatorna valamelyik fizikai jellemzőjét.
A hangok digitális továbbításakor az eredeti analóg hangjelekből mintákat vesznek, emiatt a mintavételezett hangjel időben és értékben nem folytonos, hanem egymástól elkülönülő impulzusok sokaságából áll. A mintavételezett impulzusok amplitúdóértékét bináris formában megadva, megkapjuk az analóg hanganyag digitális megfelelőjét, ami elkülönülő (diszkrét) minták sorozatából áll. Ezt hívják impulzus-kódmodulációnak (PCM - Pulse Code Modulation). Visszaalakításkor a PCM impulzussorozatból digitális áramkörök segítségével folytonos, analóg jeleket alakítanak ki, ezeket az analóg jeleket lehet meghallgatni.
Az analóg jelek digitalizálásának folyamata két fázisból áll, ezek

- a mintavételezés és
- a kvantálás, diszkrét minták sorozatának létrehozása.

Mintavételezés

Mintavételezéskor az időben és értékben folytonos analóg jelekből impulzussorozatot állítunk elő, azaz a mintavételezési gyakoriságnak megfelelő időközönként megmérjük a jel amplitúdóját. Ebben az impulzussorozatban tehát minden egyes impulzus amplitúdója azonos az analóg jelnek az adott ponton felvett értékével. A mintavételezett impulzussorozat, más néven mintavételezett jelsorozat információtartalma – meghatározott feltételek teljesülése esetén – megegyezik az eredeti, időben folytonos analóg jel információtartalmával.

Hogy milyen gyakran vegyünk mintát, **Shannon tételé**ből tudhatjuk meg. Ennek értelmében egy mintavételezett jelből akkor lehet az eredeti

jelet információveszteség nélkül visszaállítani, ha a mintavételezési frekvencia értéke legalább kétszerese az eredeti analóg jelben előforduló legnagyobb frekvenciának.

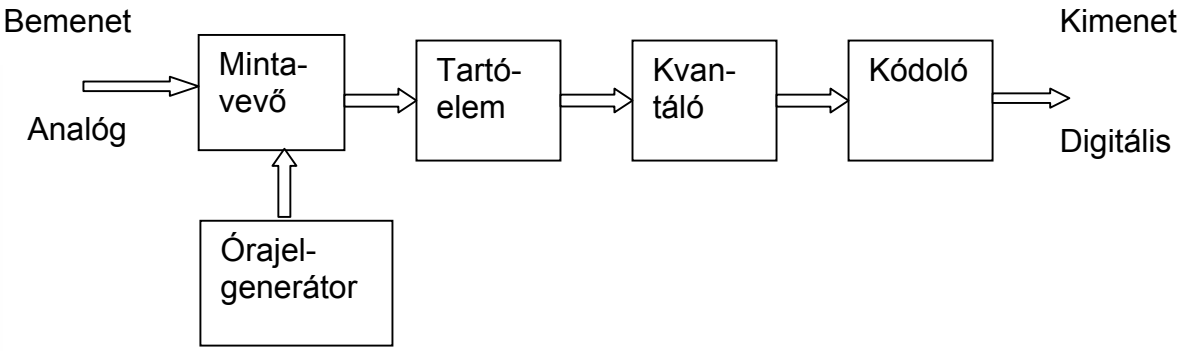
Mivel a továbbítandó jel (pl. hang) minőségével kapcsolatban különböző elvárások lehetnek, különböző célra eltérő, de szabványos mintavételi frekvenciákat használunk:

Mintavételezési frekvencia	Felhasználás	Mintavételezési frekvencia	Felhasználás
8 kHz	A telefontechnika használja	32 kHz	MPEG Audio
11,025 kHz	I/4 CD-DA mintavételezési frekvencia	44,1 kHz	CD-DA
22,1 kHz	1 /2 CD-DA mintavételezési frekvencia	48 kHz	Digital Audio Tape, MPEG Audio, Dolby Digital

15. ábra
A legfontosabb mintavételezési frekvenciák

Kvantálás

A mintavételi frekvencia mellett a digitalizálásának másik legfontosabb paramétere a kvantálás hossza. A mintavételezés során nyert sorozat impulzusainak amplitúdója végtelen sok értéket vehet fel, ezért ez a jelsorozat még analóg impulzussorozatnak tekinthető. Az amplitúdók digitális jellé történő átalakítását az analóg-digitális átalakító (A/D konverter) végzi. Az A/D konverter bemenetére vezetett impulzusok bináris adatokká átalakítva jelennek meg a konverter kimenetén. Feladata lényegében egy ***kerekítés****, hiszen az analóg jel amplitúdója bármilyen értéket felvehet, azonban az A/D konverter csak meghatározott számú bitet használ a jel digitális ábrázolásához.*
Minél finomabb felbontású a kvantálás, minél több lépcsője van, minél több bitből állnak a kódszavak, annál pontosabban lehet rekonstruálni az eredeti analóg jel amplitúdóját.
A gyakorlatban a hangfrekvenciás jelek digitalizálásánál a kvantálás általában 8 vagy 16 bites szóhosszúsággal történik. A 8 bit (1 byte) hosszúságú adatszóval dolgozó A/D konverter 256 különböző értékű adatszót, míg a 16 bit (2 bájt) hosszúságú adatszóval dolgozó A/D konverter 65 536 különböző értékű adatszót képes előállítani.
A kvantálással átfogható hangerő tartomány a kvantálás hosszától függ. Minél több bitet használnak az amplitúdó megadására, annál nagyobb az átfogható hangerő tartomány.



16. ábra
Analóg-digitális átalakító blokkvázlata

A bemenetre érkező analóg jelből a mintavevő a megadott gyakorisággal mintát vesz. A tartóáramkör a következő mintavételi időpontig ezt a mintavett értéket tartja. A két mintavétel közötti időben a kvantáló ezt az értéket megfelelteti egy kódszónak.

2. Vonalak megosztása

Ahhoz, hogy információcserét valósíthassunk meg két végpont között, szükségünk van a végpontok között az összeköttetést biztosító vonalakra. Sok esetben azonban a kommunikáció jellegéből fakadóan nincs folyamatos információcsere rajtuk, azaz a legtöbb kapcsolatban a vonalhasználat csak időszakosan jelentkezik. Nem ésszerű tehát egy kommunikációs csatorna számára kisajátítunk egy teljes vonalat. Ezek a vonalak igen jelentős költséggel épültek meg, célszerű minél jobban kihasználni azokat.

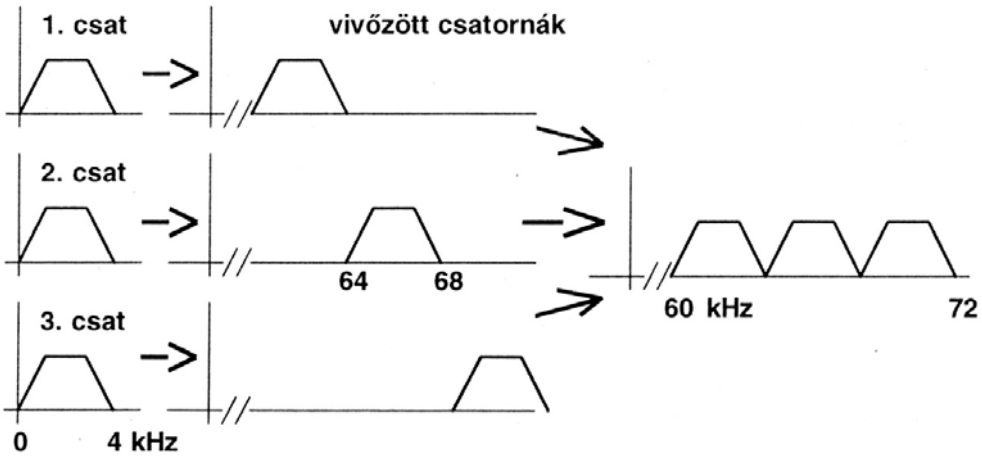
A funkciókat különválaszthatjuk **csatorná**ra, amelyeken az információcsere történik, és a felhasznált, tényleges, fizikailag létező összeköttetéseket biztosító **vonalak**ra, akkor lehetőség nyílik a gazdaságosabb kihasználásra. Mivel az adó- és vevőoldal számára csak a végeredmény, az információ a fontos, egy vonalon több csatorna is kialakítható, a megvalósítás pedig többféleképpen is elképzelhető:

- **Multiplexelés**: a fizikai közeget több csatorna között osztjuk meg, annak érdekében, hogy a vonalat több adó és több vevő vehesse igénybe. A multiplexelés olyan eljárás, amelynek során egy adatvonalat előre meghatározott, rögzített módszer szerint elemi adatcsatornákra osztunk fel. Minden bemenő elemi csatornához egy kimenő csatorna is tartozik. A mutiplexelést el lehet az időtartományban és a frekvenciatartományban is végezni, így beszélhetünk frekvenciaosztásos és időosztásos multiplexelési módszerekről, valamint ezek kombinációjáról.
- Az **üzenet-** és **csomagkapcsolási módszerek** alkalmazásával hatékony vonalkihasználás érhető el. Az átvitelre szánt információt kisebb egységekre kell bontani, majd a vonalon egymás után átvinni, végül az egységekből újra összerakni. A csomagok folyamatos áramlása az adó és a vevő számára úgy tűnik, mintha folyamatos összeköttetés lenne.

- A **vonalkapcsolás**nak hívott módszer a harmadik lehetőség. Az adatvezetéket a kommunikálni szándékozó felek csak a kommunikáció időtartamára kapják meg. Az adatvezeték tehát nem egy adóhoz és egy vevőhöz tartozik, hanem csak attól függ, hogy szükségük van-e rá, valamint más attól, hogy nem használja-e a vonalat. A kapcsolat a kommunikáció befejezésekor megszűnik.

2.1. Multiplexelés frekvenciaosztással

A módszer (FDM – Frequency-Division Multiplexing) alapelve azon a felismerésen alapul, hogy ha szinuszos hullámok összegéből előállítunk egy jelet, abból bármelyik összetevőt a csatorna másik oldalán eredeti formájában kinyerhetjük egy alkalmas szűrő segítségével. Az adóoldalon a csatornák jeleit egy-egy vivőfrekvenciára ültetik (a vivőfrekvenciát a jelekkel modulálják), majd ezeket összegzik, az összegzett jelet átviszik a vevőoldalra, és ott ezeket szűrőkkel szétválogatják.
A frekvenciaosztásos multiplex rendszerek legfőbb felhasználási területét a távbeszélő-hálózatok vivőfrekvenciás rendszerei jelentik. A széles frekvenciasávban időben is egyszerre haladnak a különböző vivőfrekvenciákra ültetett jelek. Másik, mindenki által ismert felhasználása a módszernek a rádió-, televízióadások.



17. ábra
Frekvenciaosztásos multiplexelés

Mivel a rendszert elsősorban távbeszélő hálózatok vivőfrekvenciás berendezéseinél használják, a sávszélesség mértékegységét az ekvivalens beszédcsatornák számával adják meg (12, 24 csatornás rendszerek).
Egy-egy beszédcsatorna 4 kHz sávszélességű. Tizenkét beszédcsatornát távbeszélőcsoportba fognak össze, amely a 60–108 kHz-es frekvenciatartományban helyezkedik el.
Öt távbeszélőcsoport egy távbeszélő főcsoportot alkot, ez 240 kHz sávszélességű. Ezeket további magasabb rendű csoportokba lehet összefogni, azonban ez a módszer nem igazán alkalmas számítógépek közötti információátvitelre, a csatornák emberi beszédre alapozott sávszélessége miatt.

2.2. Multiplexelés szinkron időosztással

(STDM – Synchronous Time-Division Multiplexing)
Digitális átvitelnél az időosztásos multiplexelésnél a nagyobb sávszélességű adatvonalat időben osztják fel több, elemi adatcsatornára.
A rendszerben minden elemi adatcsatorna egy-egy időszeletet kap. A fővonal két végén elhelyezkedő vonali multiplexerek előre meghatározott időben, periodikusan, egymással szinkronban működve kapcsolják össze egy-egy rövid időre az összetartozó be-, illetve kifutó vonalakat.
Ilyen elven működik a TV teletext adása is. Egy-egy időszeletet kap a műsor és a szöveges információ. Az adó- és vevőberendezések oldják meg, hogy felváltva érkezzen meg a kép, majd a teletext oldal. Mindezt olyan gyorsan teszik, hogy a nézőnek mind a kép, mind a teletext szövege folyamatosnak tűnik.

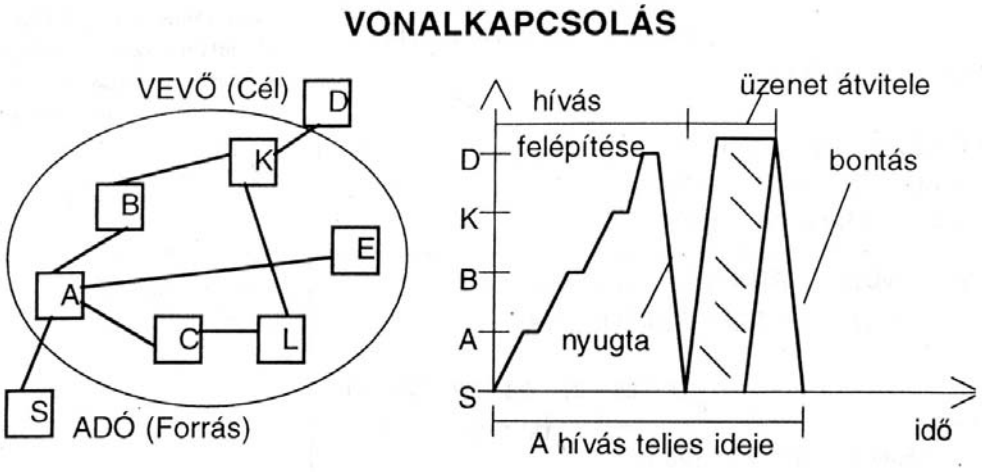
A demultiplexer feladatát néha a nagy sebességű fővonalhoz közvetlenül csatlakozó feldolgozó számítógép (illetve annak adatátviteli vezérlőegysége) látja el. A rendszer működéséhez szükség van arra, hogy a vonal két végén elhelyezett multiplexerek szinkronizmusát biztosító periodikus jeleket is elhelyezzük az információegységek között. Ezek a szinkronjelek csökkentik a fővonal kihasználhatóságát. A frekvenciaosztással és időosztással működő multiplexerek egyaránt akkor felelnek meg jól rendeltetésüknek, ha jelenlétük nem befolyásolja az adatkapcsolat-szintű vezérlést.

2.3. Vonalkapcsolás

Ahhoz, hogy az adó és a vevő kommunikálni tudjon egymással, összeköttetést kell létesítenünk közöttük: kell alakítani azt az útvonalat, melynek részeit kapcsolóközpontok kötik össze.

- Az első lépésben fizikai kapcsolat létesül az adó és vevő között, ami az összeköttetés idejére áll fenn.
- Az összeköttetésen keresztül megvalósul az adatátvitel.
- Az adatátvitel befejeztével a kapcsolat lebomlik.

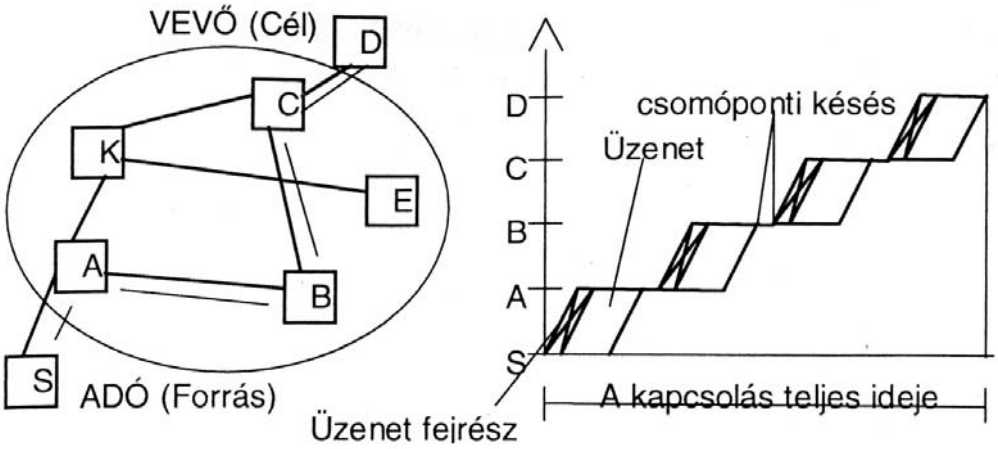
A folyamatot a távbeszélő technikában **hívás**nak nevezik. A módszer szempontjából fontos, hogy az információátvitelt meg kell előznie a híváskérés hatására létrejövő összeköttetésnek. Előnye a tényleges fizikai összeköttetés létrehozása. Ezek után a két állomás úgy képes kommunikálni, mintha pont-pont összeköttetés valósult volna meg közöttük. Hátrányaként említhető, hogy a kapcsolat felépítése néha hosszú ideig tart, valamint ha éppen adatátviteli szünet van, addig is foglalja a vonalat.



18. ábra
A vonalkapcsolás elve

2.4. Üzenetkapcsolás

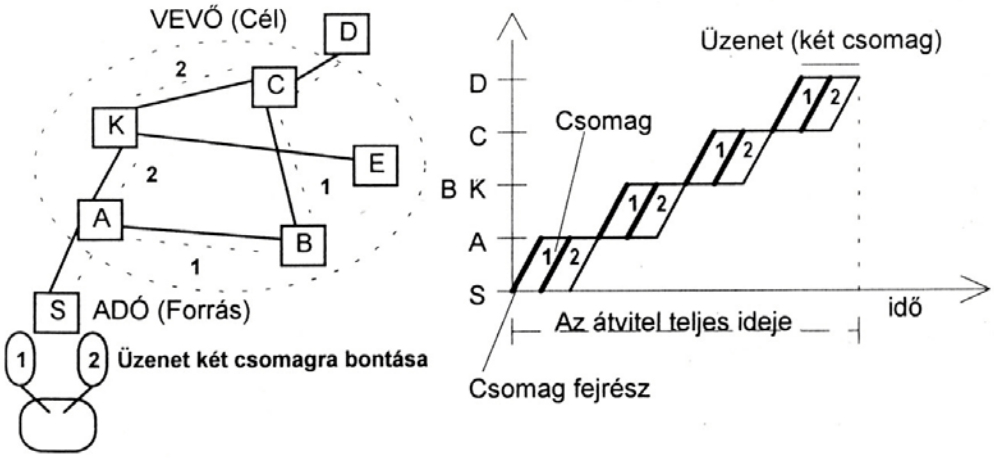
A vonalkapcsolás hátrányát felismerve – miszerint az adat elindításától kezdve a megérkezéséig lefoglalja a teljes vonalat – a fejlesztők kidolgoztak egy jobb eljárást, melynek a neve üzenetkapcsolás.
Ennél az eljárásnál az üzenet küldéséhez nem foglalják le a teljes vonalat, csupán a szomszédos hosztig építik fel a kapcsolatot. Ha az adat átment a szomszédos hosztig, bontható az adó és a közbülső állomás közti összeköttetés. Így csak egy-egy vonalszakaszt terhel az adás.
Az ilyen hálózatok a **tárol és továbbít** (store and forward)-hálózatok. Az üzenetkapcsolás esetén nincs az adatblokk méretére korlátozás, ami nagy tárolókapacitású fogadó és továbbító IMP-ket igényel.
Ez a korlátlan csomagméret egyben a hátránya is az eljárásnak. Előfordulhat ugyanis, hogy hiba történik az adatok átvitele során. Ebben az esetben a teljes üzenetet meg kell ismételni, ami tetemes időt vehet igénybe.
Ehhez az elvhez hasonló a levélküldés mechanizmusa: szintén több állomáson megy keresztül a levél, és ha pl. megsérül (elázik, elszakad), itt is újra kell küldeni a teljes levelet.
A következő módszer ezt a hátrányt is kiküszöböli.



19. ábra
Az üzenetkapcsolás elve

2.5. Csomagkapcsolás

Csomagkapcsolásról akkor beszélünk, amikor a továbbítandó üzenetet a vonal kihasználtságának növelése érdekében kisebb adagokra bontjuk, és ezeket egyenként küldjük el a címzettnek. A csomagkapcsolás nagyon hatékonyan képes a vonalak kihasználására, mivel az adott két pont közötti összeköttetést több, más irányból érkező és továbbhaladó csomag is használja. Minden csomag küldése előtt végre kell hajtani egy hálózatvizsgálatot, melynek során meg kell állapítani a küldő és a címzett közötti legrövidebb, és ezzel együtt a leggyorsabb útvonalat is. Felléphet olyan eset is, amikor a később küldött csomag előbb érkezik meg. A vevő feladata a csomagok helyes sorrendbe állítása, majd azok egyesítése. A csomagkapcsoló hálózatok hatékonyan alkalmazhatók interaktív forgalom (ember-gép kapcsolat) kezelésére is, mivel biztosítják, hogy bármelyik felhasználó csupán néhány ezredmásodpercre sajátíthat ki egy vonalat.
Ezt a technikát használják az Internetnél is, sőt manapság már a telefonbeszélgetéseknél is előfordul a csomagkapcsolásos módszert alkalmazó eljárás.



20. ábra
A csomagkapcsolás elve

3. Vezetékes átviteli közegek

Az előzőekben megismertük az adatok átvitelének lehetséges technikáit. Szó esett a vonalak megosztásáról is. Azt azonban még nem vizsgáltuk meg, hogy ezek a vonalak a valóságban milyen adatátviteli közegek lehetnek.
Napjainkban többfajta átviteli közeg is elterjedt. Ezeket két nagy csoportra oszthatjuk:

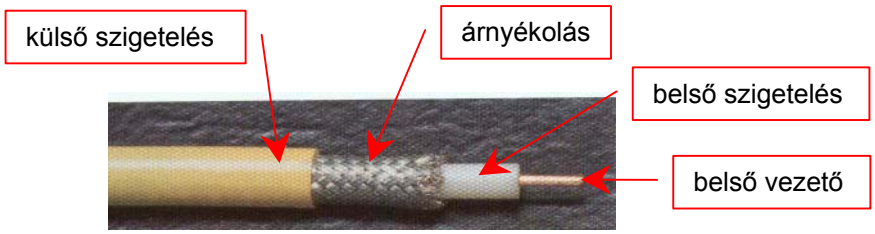
- vezetékes átviteli közeg
- vezeték nélküli átviteli közeg

Ezeken belül is létezik többfajta adatátviteli lehetőség, melyeket a következőkben részletesen elemzünk.

A vezetékes rendszerek előnye, hogy lehallgatás ellen védettebbek, a rövidebb távolságok kisebb költséggel építhetők ki, mint az összeköttetés-mentes hálózatok esetében. Hátránya a helyhez kötöttség.
Ha jelek átvitelére egymás mellett futó, párhuzamos vezetékeket használunk, ezek antennaként funkcionálnak, tehát összeszedik a környezet jeleit, illetve sugároznak is zavarjeleket. Ezt a kellemetlen tulajdonságot kiküszöbölendő a gyakorlatban a csavart érpár (twisted pair) és az árnyékolt, vagy más néven koaxiális kábel használata terjedt el.

3.1. Koaxiális kábelek

Ennek a típusnak a felépítése alapvetően eltér a csavart érpártól, felépítése a következő ábrán látható.



21. ábra
A koaxiális kábel felépítése

A jeleket a belső, tömör vagy vékony vezetékekből álló sodrony továbbítja. Ezt néhány mm vastag szigetelőanyag veszi körbe. A szigetelőt árnyékoló réz „harisnya” vagy vékony fémfólia fonja vagy tekeri körbe. Erre ismét egy szigetelő réteg kerül, amely a kábelt a fizikai behatások során elszenvedett károsodásoktól védi.

A kábel három jellemzője:

- A **hullámellenállás**, amely általában 50 és 75 ohm értékű.
- A **késleltetési idő** a kábel szigetelésének dielektromos állandójától függ.

- A **csillapítás** a kábel ohmos ellenállásából, a dielektrikumon belül keletkező és a sugárzás okozta veszteségekből tevődik össze.

A tömör belső erű kábel késleltetése és csillapítása kisebb, mint a több fémszálból összefonotté, viszont jóval merevebb is. A környezet zavarainak kiküszöbölését lehet fokozni úgy, hogy az árnyékolást két rétegben készítjük el. Ezt a technikát elsősorban olyan helyeken alkalmazzák, ahol a jelvezetékek fokozattan ki vannak téve a környezet zavarainak.

A koaxiális kábelek két fajtáját használják az iparban információ továbbítására.

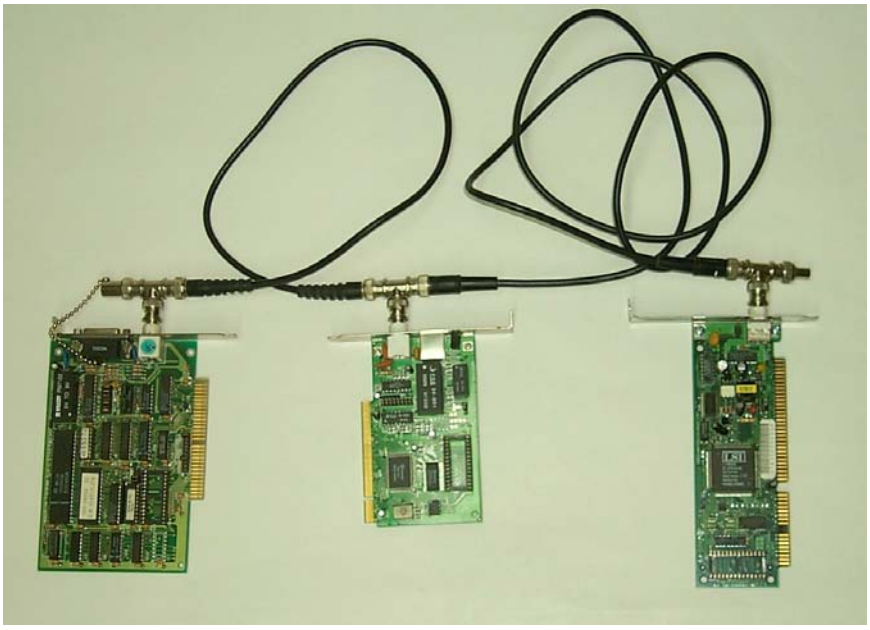
3.1.1. Alapsávú koaxiális kábelek

Az **alapsávú koaxiális kábel**eket elsősorban digitális átvitelre, leggyakrabban a számítógép-hálózatok kialakításánál alkalmazzák. Az elérhető adatátviteli sebesség 10-100 Mbit/s. Amennyiben a távolság kisebb, a sebesség növelhető, és ez fordítva is igaz. (Leggyakrabban a 10 Mbit/s használatos.)
Az alapsávú kábeleknek két fajtája van, a vékony és a vastag koax.
A **vékony koax**ot elsősorban ethernet-hálózatok kialakítására használták 10Mbit/s sebességig.

22. ábra
Vékony koaxiális kábel

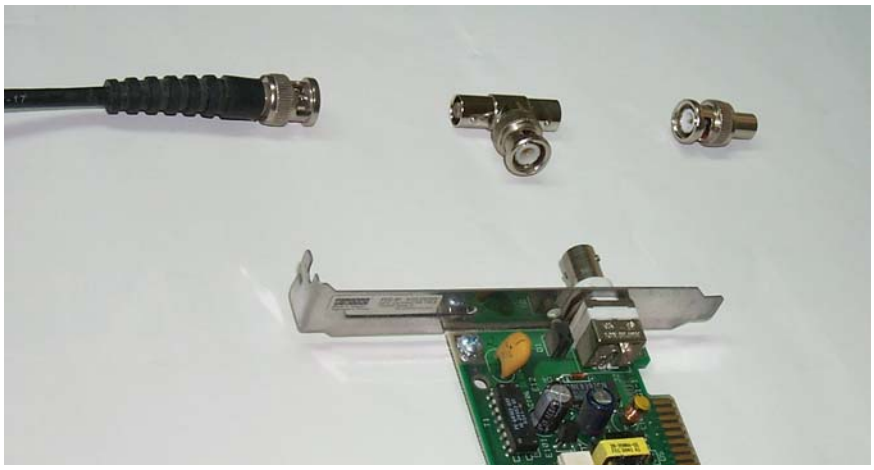
A **vastag koax** előnye, hogy a csillapítása kisebb, mint a vékony változaté, és emiatt az áthidalható távolságok nagyobbak lehetnek ugyanakkora sebesség mellett.

23. ábra
Vastag koaxiális kábel



24. ábra
Koaxális kábellel összekötött hálózat, különböző hálózati kártyákkal

A koaxiális kábelhálózat szegmense lényegében egy folytonos kábelnek tekinthető, melynek végeit lezáró ellenállásokkal határoljuk. A szegmens mentén speciális csatlakozókkal valósítható meg a leágazás, vastag Ethernet kábel (RG-8 és RG-11) esetén a kábelt megfúró úgynevezett „vámpír” elemmel (AUI: Attachment Unit Interface), vékony Ethernet kábelnél (RG-58) pedig egy „BNC-T” elemmel, mely újból megteremti a folytonosságot a megszakított kábelrészek között. A leágazás a hálózati kártyára kapcsolódik. Vastag kábel esetén az AUI a vezérlőkártyán lévő DB15 típusú csatlakozóhoz, vékony kábelnél a „T” elem pedig BNC (British Naval Connector) típusúhoz kapcsolódik.



25. ábra
A vékony Ethernet-kábel csatlakoztatása

26. ábra
„Vámpír”-csatlakozó

Az Ethernet-hálózatoknál a következő korlátozások mérvadóak:

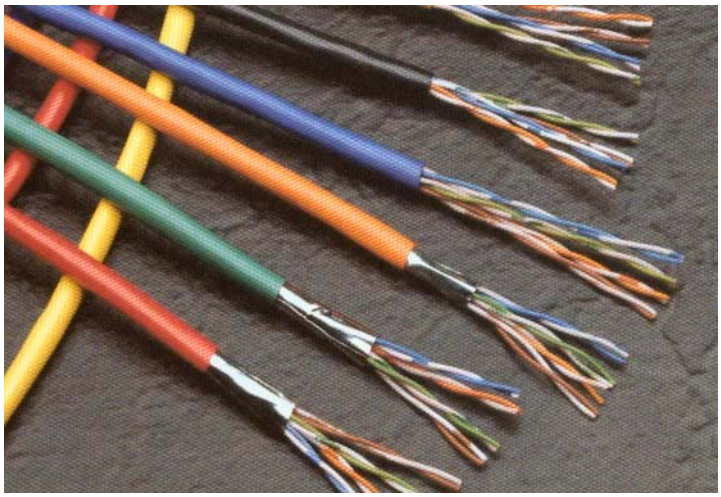
- Maximum 5 főszegmens kapcsolható össze 4 jelismétlő vagy koncentrátor felhasználásával, és maximum három lehet teljesen betelepítve munkaállomásokkal.
- Szegmensenként legfeljebb 30 állomás lehet.
- A teljes hálózatban maximum 1024 gép lehet.
- A hálózati szegmensek teljes hossza nem lehet több, mint 925 méter.

3.1.2. Széles sávú koaxiális kábel

A **széles sávú** koaxiális kábeleket **analóg átvitel**re használjuk, ilyen pl. a televízióadások jeleinek továbbítására kialakított kábelrendszer. Ahhoz, hogy a kábelt adatátvitelre használhassuk, a számítógépből kikerülő digitális jeleket át kell alakítani analóg jelekké, majd a fogadóoldalon el kell végezni az ellenkező irányú konverziót. A széles sávú koaxiális kábelek akár GHz-es jelek átvitelét is lehetővé teszik. Ez a sávszélesség nagyon nagy, ezért ezekben a rendszerekben a vonalat több, kisebb sávszélességű csatornára osztják, amelyeken egymástól független információátvitel valósulhat meg. E csatornák sávszélességét úgy határozzák meg, hogy közöttük ne jöhessen létre átfedés, mivel ez jelkeveredést okozna.

3.2. Csavart érpár

A két ér csavarása a kábel átviteli jellemzőit javítja az egyszerű párhuzamosan futó vezetékpárhoz képest. A jelkisugárzás így minimálisra csökkenthető. Minél több az egységnyi hosszra jutó csavarások száma, annál nagyobb sebességig használható a vezeték. A kábelben több érpárt fognak össze egy közös szigetelőben. Ezek egymásra és a külvilágra való hatását tovább lehet csökkenteni, ha a párokat is egymásra csavarják. Minden érpár egységnyi hosszon eltérő számú csavarást tartalmaz, a köztük lévő áthallás csökkentése miatt. Ha az érpár körül árnyékolás is található, akkor árnyékolt sodrott érpárnak (Shielded Twisted Pair, STP), míg az árnyékolás nélkülit UTP (Unshielded Twisted Pair) kábelnek nevezzük.



27. ábra
Csavart érpáras kábelek

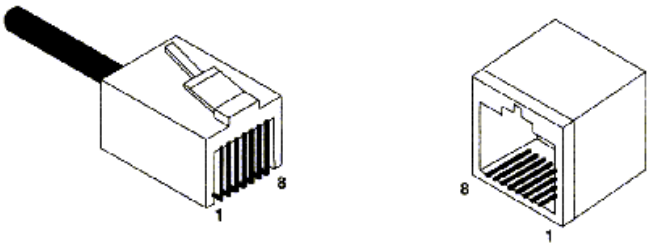
A kábelek minősége a telefonvonalakra használtaktól a nagysebességű adatátviteli kábelekig változik.
A kábeleket jelátviteli tulajdonságaik alapján több kategóriára osztják, ezek természetesen árban is eltérnek egymástól.

Szabványos osztályozásuk:

1. kategória	hangátvitel (telefon)
2. kategória	4 Mbit/s-os adatvonal
3. kategória	10 Mbit/s-os adatvonal (Ethernet)
4. kategória	20 Mbit/s-os adatvonal
5. kategória	100 Mbit/s-os adatvonal (Fast Ethernet)

Az Ethernet-hálózatokban ma túlnyomó részben az 5. kategóriájú kábeleket használják. A kábeleken alkalmazott csatlakozó típusa RJ-45. Ennek nyolc érintkezője a kábel négy szabványos színkódolású érpárjához csatlakozik. Az adatátvitel két sodrott érpáron működik, az egyik érpár adásra, míg a másik vételre szolgál. A maradék érpárak kiegészítő funkciókra használhatók fel (távtáplálás, eszközjelenlét-

érzékelés stb.) Az UTP-kábel esetében a megengedett legnagyobb, jelregenerálás nélkül áthidalható távolság (szegmenshossz) 100 méter. A vezetékek megkülönböztetésére színkódolást alkalmaznak. Négy különböző színű vezeték van, a maradék négy pedig ezek és a fehér szín keveréke.



Színjelzés	Csatlakozó	Jel neve	Jel neve	Csatlakozó	Színjelzés
Fehér-narancs	1			8	Fehér-narancs
Narancs	2	DTR	DSR	7	Narancs
Fehér-zöld	3	TxD	RxD	6	Fehér-zöld
Zöld	4	GND	GND	4	Zöld
Fehér-kék	5	GND	GND	5	Fehér-kék
Kék	6	RxD	TxD	3	Kék
Fehér-barna	7	DSR	DTR	2	Fehér-barna
Barna	8			1	Barna

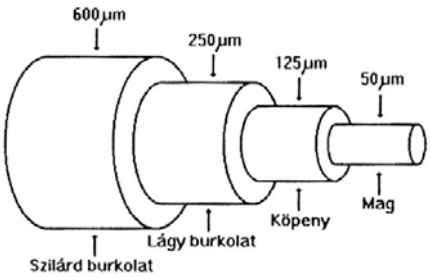
28. ábra
Az RJ-45 típusú csatlakozó bekötése

Fontos jellemzője a csavart érpárnak, hogy nem csak számítógép-hálózatok építésére lehet használni, hanem telefonhálózatoknál is. Ezt az előnyt kihasználva manapság ún. strukturált hálózatokat építenek az irodákban. Ez azt jelenti, hogy az irodában a fali csatlakozóktól egy központi helyig (az ún. patch panelig) kiépítik a kábeleket, de ekkor még nem döntik el, hogy a kábelt számítógép vagy telefon csatlakoztatására fogják használni. A csatlakozás mindkét feladat ellátására alkalmas, és bármikor megváltoztatható a csatlakozó funkciója. Így rendkívül rugalmas hálózatot lehet kiépíteni.

3.3. Üvegszálas kábel

Manapság már egyre kiterjedtebben használják, ami kiváló paramétereinek és egyre csökkenő árának köszönhető. Az információ fényimpulzusok formájában terjed egy olyan közegben, ami ezt lehetővé teszi. E közeg lehet a levegő is, azonban ebben az esetben szükséges, hogy az adó és a vevő egymás számára látható legyen. Ez nagyobb távolságok esetében nem megoldható, több tényező (a Föld görbülete, tereptárgyak, időjárás stb.) miatt. A megfelelő választás ezért az optikai szál.

Az optikai szálban a fényvezető egy speciális, nagyon vékony mag, és mint egy csőben, ebben halad a fénysugár. A mag körül helyezkedik el a köpeny, amelynek az a célja, hogy megakadályozza a fény kilépését a magból. A köpenyen egy lágy burkolat található, aminek a szerepe a nagyobb ellenállóság biztosítása a fizikai terhelésekkel szemben. Az egészet egy kemény, műanyag burkolat védi a környezet behatásaival szemben.



29. ábra
Az optikai szál felépítése

Attól függően, hogy a fény milyen módon halad a csőben, beszélhetünk egy- és többmódusú optikai kábelről.
A **többmódusú kábel** esetében a teljes fényvisszaverődés fizikai jelenségét használják fel. Ha két közeg törésmutatójának különbsége megfelelő, akkor az erre a felületre eső fény nem lép át a másik közegbe, hanem teljes egészében visszaverődik. A cső anyagának kiválasztásánál is ezt a szempontot kell figyelembe venni. Ha a paraméterek megfelelőek, akkor létrejön a teljes visszaverődés és a fénysugár gyakorlatilag csillapodás nélkül tud a vezetőben haladni.

Az **egymódusú kábel** esetében a cső átmérője a fény hullámhosszával megegyező. Ez azért különleges eset, mert ekkor a fény nem fog ide-oda verődni. Ezzel a módszerrel nagyobb távolság hidalható át erősítés nélkül.
Az optikai kábeleknél nagyon fontos szempont, hogy a vezeték egységnyi hosszon mekkora jelcsillapítással rendelkezik. A csillapítást dB-ben adják meg, egységnyi hosszúságra vonatkoztatva (pl. dB/km).

A fényforrás egy LED, vagy lézerdióda. Ezek az eszközök félvezetők, melyek nagyon jól fókuszálható fényt állítanak elő a rajtuk átfolyó áram erősségétől függő intenzitással (erősséggel). Fényérzékelőként fototranzisztort alkalmaznak. Ez szintén félvezető, ami a kristályra eső fény erősségétől függő kimeneti jelet állít elő.
Az optikai adatátvitel esetében az áthidalható távolságot a fényveszteség határozza meg, ami három jellemzőnek a függvénye:

- A két közeg összeillesztésénél a fény egy része visszaverődik. Ezen segíteni lehet a lehető legpontosabb illesztéssel. E célra ma már rendelkezésre állnak a megfelelő eszközök. Vagy nagyon pontosan összecsiszolják a két üvegszálat, vagy speciális ragasztót használnak. (Manapság inkább ez utóbbi az elterjedtebb a gyorsasága miatt.)

- Ugyanezt a hatást okozzák az átviteli közegben lévő szennyeződések is, melyeket a megfelelő anyagválasztással lehet csökkenteni.

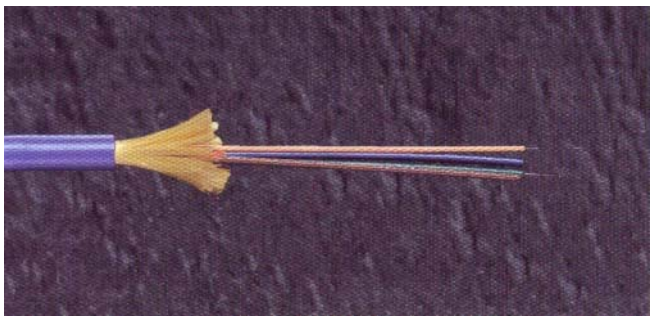
- A harmadik veszteség abból adódik, hogy ha a fény nem megfelelő szögben érkezik a közeg határfelületére, akkor a fény egy része nem verődik vissza. Ezen is az anyagválasztással és a fény hullámhosszának helyes meghatározásával tudunk segíteni.

Az optikai szál nagyon kényes a fizikai terhelésre. Mivel a kábel nagyon vékony és viszonylag merev, a fizikai megterhelést nehezen viseli. Minden nagyobb vagy hosszan tartó terhelést más szerkezeti elemnek kell átvennie. Annak érdekében, hogy azért a vezeték kezelhető legyen, a lágy burkolatban a köpenyt és a magot hullámosítva helyezik el. Ez biztosítja a bizonyos szintű nyújthatóságot és a hajlíthatóságot.

Ennél az átviteli közegnél a legproblémásabb és legfontosabb kérdés a jelek be- és kicsatolása, amire alapvetően kétféle csatolótípust alkalmaznak:

- a passzív illesztő két, az optikai szálra kapcsolódó csatlakozóból áll. Az adót egy LED, míg a vevőt egy fényérzékelő félvezető valósítja meg;
- az aktív illesztő annyiban több, mint a passzív, hogy a vett jeleket átalakítja villamos mennyiséggé, felerősíti és visszaalakítja fényimpulzusokká és újra a közegre kapcsolja.

Az optikai adatátvitel során az információt különböző hullámhosszúságú fényjel hordozza. Könnyű belátni, hogy a kétirányú adatátvitelhez két optikai szál szükséges. Ez nem okoz gyakorlati problémát, mivel a szilárd szigetelőben rendszerint több kábelt fognak össze. Megoldható üzenetszórásos topológia is, mivel léteznek ehhez szükséges interfészek.



30. ábra
Üvegszálas optikai kábel

31. ábra
Optikaikábel-szerkezet

Az ábrán található angol kifejezések magyarázata:

Fiber bunded detail	szálakból álló nyalábrészlet
Outer jacket	külső borítás
Aramid strength elements	erősítő elemek
Color coded jacket	színkódolású borítás
Specified fiber (color coded)	speciális szál (színkódolt)
Upjacketed central strength element	központi megerősítő elem
Bundles of up to 12 fibers each	max. 12 szálból álló köteg
Overall polyester type barrier	átfogó poliészterborítás
fibers	szálak

4. Adatátvitel rádióhullámok segítségével

A II. világháborúban az Egyesült Államok hadserege használt először rádiójeleket adatátviteli célokra. Ők fejlesztették ki a rádión keresztüli adatátvitelt, amit kódoltak is. Ez adta az alapot, hogy 1971-ben néhány kutató a Hawaii Egyetemen létrehozza az első csomagalapú (packet) rádiós adatátviteli technológiát. Ez lett az első vezeték nélküli hálózat (WLAN = wireless local area network), és az ALOHNET nevet kapta. Ez a hálózat 7 számítógépből állt és kétirányú csillag-topológia kialakítású volt.

Bár eddig a vezetékes hálózatok egyértelműen uralták a piacot, az utóbbi években terjed a vezeték nélküli változatok használata is. Legsűrűbben egyetemi környezetben, az egészségügyben, az iparban és a raktározásban láthatjuk ennek példáit. Mindeközben a technológia fejlődik, egyre egyszerűbb és olcsóbb megoldások látnak napvilágot.

4.1. Mikrohullámú átvitel

Nagyobb távolságok áthidalására pont-pont közötti összeköttetésre gyakran használják a mikrohullámú átvitelt. A frekvenciatartomány 2–40 GHz között lehet. A rádióhullámokat parabolaantennák segítségével nyalábolják, és az antennákat – a mikrohullám egyenes vonalú terjedése miatt – magas antennatornyokon helyezik el. Az áthidalható távolság akár száz kilométer is lehet. (A hullámok egyenesvonalú terjedése miatt a láthatóság itt is feltétel.) Ha nagyobb távolságot kell áthidalni, a jelismétlést ún. relézőállomásokkal oldják meg, azaz a vett jelet erősítés után egy másik frekvencián a következő relézőállomásnak továbbítják. A viharok, a villámlás és egyéb légköri jelenségek zavarhatják az adatátvitelt.

4.2. Műholdas átvitel

A fényhez hasonló terjedési jellemzői miatt a mikrohullámú rendszerek csak korlátozott távolságig használhatók a Föld felszínének görbülete, valamint domborzati adottságai miatt.
A műholdas rendszerek kiküszöbölik ezt a problémát, ugyanis a hálózat végpontjai egy VSAT (Very Small Aperture Terminal – igen kis nyílásszögű antennavégpont) végberendezéssel és egy kisméretű parabolaantennával, műholdon keresztül tartják fenn a kapcsolatot a központi számítógéppel. A VSAT átviteli rendszerek általában csomagkapcsolt protokollal működnek.

A VSAT-szolgáltatók a műholdak kapacitását nemzetközi műholdas szervezetektől (például EUTELSAT) bérlik.
A műholdakon lévő ún. transzponderek a felküldött mikrohullámú jeleket egy másik frekvencián felerősítve visszasugározzák. A műholdak geostacionárius pályán, az Egyenlítő fölött kb. 36.000 km magasságban keringenek, sebességük megegyezik a Föld forgási sebességével, így a Földről állónak látszanak. Ez lehetővé teszi, hogy a földi antennákat ne kelljen mozgatni. A mai technológia mellett 90 geostacionárius műhold helyezhető el ezen a pályán (4 fokonként).

A távközlési műholdak frekvenciatartományai:

- *felfelé irányuló nyaláb számára: 5,925...6,425 GHz*
- *lefelé irányuló nyaláb számára: 3,7...4,4 GHz*

A műhold tipikus sávszélessége 500 MHz (amely mintegy 800 db 64 kbit/s-os hangcsatorna).
A műholdas átvitel késleltetése a földi mikrohullámú, illetve a vezetékes rendszerekhez képest jelentős a nagy távolság miatt: kb. 0,2-0,3 másodperc.

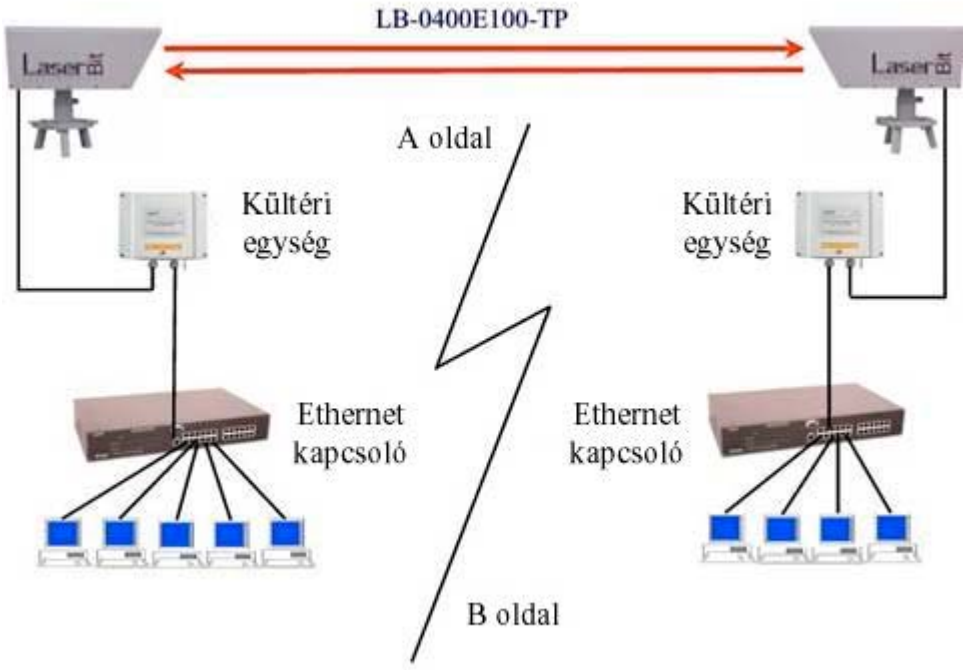
5. Vezeték nélküli adatátvitel

A hordozható számítógépek, mint például a notebookok és **PDA-k** (personal digital assistants – kézi számítógép), a számítástechnikai iparág leggyorsabban fejlődő területéhez tartoznak. Általános az igény, hogy ezek a hordozható eszközök a mobiltelefónia eszközeivel, vagy más módon, de kábeles csatlakoztatás nélkül is alkalmasak legyenek hálózati szolgáltatások igénybevételére.

Vezetékes összeköttetést nem lehet minden esetben megvalósítani. Gyakran adódik olyan helyzet, amikor a költségek (kábelek fektetése, közterületen utcák felásása) nem állnak arányban a létesítendő hálózati kapcsolat előnyeivel. Hordozható, kisméretű eszközeink egymáshoz kábelekkel történő csatlakoztatása is problémát jelenthet. Ilyen esetekben segítenek a vezeték nélküli átviteli megoldások, amikor átviteli közegként fényt (infravörös, lézer) vagy rádióhullámot használunk.

5.1. A lézeres adatátvitel

A lézeres átvitelt alkalmazó adó-vevő párokat pont-pont közötti adatátvitelre használhatjuk. A kommunikáció teljesen digitális, a lézerfény irányított energiakoncentrációja nagyobb távolság (néhány km) áthidalását teszi lehetővé. Az illetéktelen lehallgatás, illetve külső zavarás ellen viszonylag védett. Az időjárási viszonyok azonban befolyásolják a fény terjedését, így az eső, a köd, a légköri szennyeződések zavarként jelentkeznek. Felhasználható lokális hálózatok, telefonközpontok összekötésére. A megvalósított adatátviteli sebesség jelenleg 2 és 100 Mbps között van.



32. ábra
Lézeres adatátvitel

5.2. Adatátvitel infravörös fény segítségével

Az Infravörös Adat Egyesülés – angol rövidítéssel IrDA – egy vállalatok feletti tömörülés, amely kidolgozta az infravörös fényen alapuló adatátviteli ajánlást. Az IrDA-Data infravörös kapcsolat két eszköz között. Az IrDA-Control perifériális egységek, pl. PC egér, billentyűzet rendszerbeli illesztését teszi lehetővé.
Az IrDA-Data eszközöket elsősorban a vezetékes kapcsolat alternatívájaként kezelhetjük. Az infravörös eszközökkel történő adatátvitel feltétele, hogy a két berendezés lássa egymást (azaz semmilyen optikai akadály ne legyen a két egység között), a távolságul kb. 1 méternél ne legyen több. Egy infravörös eszköz kb. 30 fokos szögű kúp alakú területet tud besugározni, ezen belül kell lennie a vevő berendezésnek.

Az IrDA általános jellemzői

- Elterjedt megoldás a vezetékes kapcsolat helyettesítésére, több tízmillió eszköz világszerte, „célozd meg és mehet az átvitel" stílusú, egyszerű használat.
- Rendkívül sokféle hardver- és szoftvermegoldása létezik, szinte minden elterjedt számítástechnikai rendszerhez.
- Nem kell számolni más eszköztől származó zavarással, így nem szükséges speciális biztonsági eljárás használata.
- Nagy adatátviteli sebesség (fejlesztés alatt a 16Mbit/sec-os kapcsolat).

Hol használható az IrDa-kapcsolat?

- notebookok, asztali PC-k, kézi számítógépek (különös tekintettel a mobilkommunikációs megoldásokra),
- nyomtatók,
- mobiltelefonok, személyhívók, modemek,
- digitális fényképezőgépek és kamerák, LAN eszközök,
- orvosi és ipari berendezések,
- szórakoztatóelektronikai termékek, órák stb. esetében.

A világon már több mint 50 millió ilyen eszközt használnak és évente 40%-os növekedés várható.

5.3. Szórt spektrumú sugárzás WLAN (wireless LAN – kábel nélküli rádiófrekvenciás LAN)

Míg a műholdas és a mikrohullámú jeltovábbítás általában a szolgáltatók nagytávolságú adatátviteli problémáit oldja meg, a lézeres átvitel pont-pont közötti összeköttetésre használható, az infravörös fény pedig csak maximum néhány méterig használható, addig az ún. szórt spektrumú, kábelnélküli (wireless) rendszerek a felhasználó szintjén adnak megoldást számítógépek, hálózatok összekapcsolására, közepes távolságig. Az infavörös fényt és a szórt spektrumú rádiófrekvenciás átvitelt használó megoldásokra – mivel lényegében csak a fizikai réteg

szintjén különböznek egymástól – közös szabványt dolgoztak ki, ezzel segítve elő a tömeggyártást és a mielőbbi széleskörű elterjedést.
A szabvány biztosítja a különböző gyártók eszközei közötti kompatibilitást, és ezt a IEEE (Institute of Electrical and Electronics Engineers – Műszaki és Elektronikai Mérnökök Intézete) készítette el. A rendszer az ipari, tudományos és orvosi, az ún. ISM (industrial, scientific, medical) célra fenntartott sávban dolgozik.

Az eredeti szabvány, az IEE 802.11 az infravörös fény és a szórtspektrumú 2,4 GHz-es rádiófrekvencia használatát engedi meg az 1, illetve 2 Mbps-os (millió bit per másodperc) adatátviteli sebesség mellett.
A szabvány bővítései – az IEEE 802.11a és az IEEE 802.11b – az 5 GHz-es, valamint a 8 GHz-es hullámsáv használatát is megengedik. Az adatátviteli sebességek is változtak, 5 és 11 Mbps-re, valamint az IEEE 802.11a esetében az 54 Mbps-os adatátviteli sebesség is elérhetővé vált.

A kapcsolatot megvalósító szórt spektrumú adó nagyon kis teljesítmény mellett a moduláció ütemében változtatja adási frekvenciáját, ellentétben a keskenysávú rádiókkal, amelyek a teljes energiát egyetlen frekvenciára koncentrálják. Széles frekvenciasávot használ, amit egy hagyományos vevő fehér zajnak érzékel (azonos amplitúdó minden frekvencián). A szórt spektrumú vevő azonban felismeri és „megérti" az adást.

A szabvány két különböző modulációs módszert tartalmaz:
- *a közvetlen sorrendes szórt spektrumú rádiós összeköttetést (DSSS = Direct Sequence Spread Spectrum) és*
- *a frekvenciaugrásos szórt spektrumú rádiós összeköttetést (FHSS = Frequency Hopping Spread Spectrum).*

Az adatátvitel frekvenciája folyamatosan, adott algoritmus szerint változik, amit mind az adó-, mind a vevőállomás ismer. Ez meglehetősen biztonságos módszer: illetéktelenek nem tudhatják, hogy mikor melyik frekvenciára váltsanak ahhoz, hogy a teljes adatfolyamot megkaphassák.

WLAN topológiák

Vezeték nélküli környezetben alkalmazott hálózati topológiák: a csillag és a háló.
A csillag-topológia – amely ma a világon a legelterjedtebb – olyan hálózatot jelent, amely egy központi bázisállomás, vagy más néven hozzáférési pont (AP = Access Point) köré épül. A kiinduló csomópont elküldi az információcsomagot a központi állomásnak, amit az a célcsomópontra irányít.

Az Access Point jelentheti a hidat egy vezetékes hálózat (ezentúl LAN = Local Area Network helyi hálózat) számára, amin keresztül elérhet további vezetékes klienseket, az Internetet vagy más hálózati eszközöket.
A háló-topológia (mesh) csak annyiban különbözik a csillagtól, hogy ott nincs központi bázisállomás. Minden csomópont szabadon kommunikálhat a hatósugarán belül levő bármely másikkal.

A 802.11 szabványban leírt LAN a mobiltelefon-rendszerekhez hasonlóan cellarendszerben is kiépíthető, és ezáltal nagy területek is lefedhetők. Az egyes cellákat egy-egy bázisállomás hatótávolságán belüli terület jelenti.

A Bluetooth-rendszer

A Bluetooth a fenti elveken alapuló adatátviteli eljárás, rövid hatótávval, pont-több pont kapcsolattal. Hatótávolsága 10 cm-től néhányszor 10 m-ig terjed, épületeken belül és kívül is. Használható rövid távolságú adatkapcsolathoz mobil- és telepített eszközök esetében. Kis távolságok esetén a falak sem jelentenek akadályt.

A Bluetooth általános jellemzői:

- Működési tartománya a szabadon felhasználható 2.4 GHz-es, ún. ISM-rádiósáv.
- Megvalósítása az ún. frekvenciaugrásos, szórt spektrumú technológiával történik. A frekvenciasávok ugrócsatornákra vannak osztva, amelyeket a kapcsolat idején ál-véletlen módszerrel (valamilyen algoritmus alapján, előre nem meghatározható módon) választanak ki az eszközök.
- A rendszer számára definiált ún. piconet-hálózatot (nagyon kis kiterjedésű hálózat) egyszerre 8 eszköz használhatja a csatornákon megosztozva.

A Bluetooth gyakorlatilag minden, az IrDA-nál már említett eszköznél használható, kiegészülve a nagyfokú kényelemmel, mobilitással, ami abból fakad, hogy az eszközöknek látniuk sem kell egymást. Ilyen eszközök lehetnek:

- telefonok, személyhívók,
- modemek,
- LAN-eszközök,
- fejhallgató/mikrofon egységek,
- asztali, notebook és kéziszámítógépek.

6. Telefónia

A hagyományos nyilvános, kapcsolt távbeszélő hálózatok (Public Switched Telephone Network, PSTN) kialakításakor a cél az emberi hang felismerhető módon való átvitele volt. Ehhez a hangot analóg villamos jelekké kellett alakítani, továbbítani, valamint a vonal túlsó végén visszaalakítani hallható hanggá.

Ha ezt az analóg átvitelre tervezett rendszert digitális adatátvitelre akarjuk használni, az elérhető, néhányszor 10kbit/s adatátviteli sebesség számítógépek kommunikációjára csak kompromisszumok árán alkalmas. A számítógépek között, az erre a célra kiépített összekötő adatátviteli kábeleken az adatátviteli sebesség Mbit/s-os nagyságrendű, igen kis hibaarány mellett. Ez azonban feltételezi, hogy a gépek viszonylag közel vannak egymáshoz, lényegében így működnek a lokális hálózatok.
Amennyiben a távolságok viszont már nagyok, több gépről van szó, vagy a magánvezetékek lefektetése megfizethetetlenül drága, szükségképpen igénybe kell venni a már meglevő távközlési eszközöket. Ha a világ összes előfizetői készüléke és a helyi központok közötti ún. előfizetői hurkot egymáshoz fűznénk, akkor a Föld és a Hold közötti távolság sokszorosát kapnánk.
Fontos tehát, hogy ezeket a digitális jelátvitel szempontjából sok esetben erősen korlátozott paraméterekkel rendelkező rendszereket minél nagyobb hatékonysággal tudjuk felhasználni.

6.1 A telefon működési elve

Az emberi hang analóg villamos jellé alakítására a mikrofon szolgál. A telefontechnikában használatos mikrofonok a hanghullámok amplitúdóváltozásának megfelelően változtatják ellenállásukat, és ennek megfelelően változik az áramkörben az áram.
Ez a változó áram a vevőoldali hallgató elektromágnes tekercsén átfolyva, a kialakuló mágneses tér változásaival mozgásra kényszeríti a membránt, az pedig hallható hangot fog kibocsátani.
Ha csak egy mikrofon és egy hallgató (mint adó és vevő pár) lennének a rendszerben, az átvitel csak szimplex lehetne, ezért az áramkört fordított irányban duplázni kell. Ahhoz, hogy egy beszélgetést kezdeményezni lehessen, egy újabb funkcióra is szükség van: a jelzésre. Erre a célra külön csengető áramkör hozzáadása szükséges. Ilyen módon két huzallal összekötve két távbeszélő állomás már képes egymással teljes duplex módon kapcsolatba lépni.

6.2 A távbeszélőrendszer felépítése

Amikor Alexander Graham Bell (http://www.szechenyi-szfvar.sulinet.hu/erdekes/telefon/a_g_bell/ag_bell.htm) 1876-ban feltalálta a telefont, csak a párosával árusították a telefonkészülékeket. A készülékek közötti vezeték kihúzása a felhasználó feladata volt. Ha egy telefontulajdonos nem csak egy, hanem több másik telefontulajdonossal akart beszélni, akkor minden házhoz külön vezetéket kellett kihúznia. Hamarosan nyilvánvalóvá vált, hogy más megoldást kell keresni. Bell felismerte ezt a problémát, és megalapította a Bell Telefontársaságot, amely 1878-ban létrehozta az első telefonközpontot. A telefonközpont és az előfizető között már csak egy vezetékpár kihúzására volt szükség, mert a telefonközpont kézi kapcsolással létre tudta hozni a kapcsolatot a két előfizető között, így egy városon belül megoldódott a probléma. A városok közötti beszélgetések igénye azonban újabb, második szintű központok beállítását tették szükségessé.

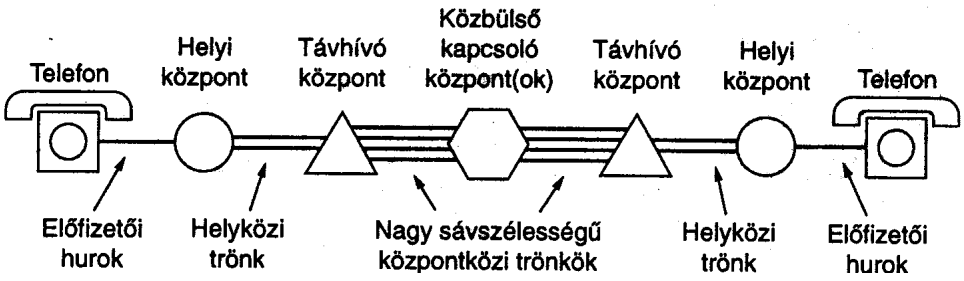
Az 1800-as évek végére kialakult a távbeszélőrendszer három fő része:

- *a kapcsolóközpontok,*
- *az ügyfelek és a kapcsolóközpontok közti vezetékek (előfizetői hurkok), valamint a*
- *telefonközpontok közötti nagytávolságú vezetékek (trönkök).*

A távközlésben igen sokféle átviteli közeget használnak. Az előfizetői hurkok manapság csavart érpárból állnak. A kapcsolóközpontok között koaxiális kábeleket, mikrohullámú összeköttetést, leggyakrabban pedig fényvezető kábeleket használnak.

Fentiek alapján több állomás esetén az egymással való beszélgetés telefonközpont közbeiktatásával lehetséges. A központon keresztüli kommunikációhoz azonban az összekapcsolódáshoz egy vezérlő információt (jelzést) is el kell juttatni: a hívott állomás számát, és ez a jelzés szintén a beszélgetés céljára szolgáló vezetéken jut el a központba. A telefonközpont a szám vétele után létrehozza az összeköttetést a hívott állomással.

Elvi lehetősége megvolna annak, hogy a világ összes telefonját egy hatalmas központon keresztül kapcsoljuk össze, ez azonban teljesen értelmetlen lenne. A valóságban a központok többszintű hierarchikus rendszerként épülnek fel, hiszen hívásaink legnagyobb része is lakóhelyünk közelében élő családtagjaink, barátaink, üzletfeleink elérésére irányul.



33. ábra
Telefonhálózat felépítése

A **helyi központ**hoz kapcsolódó előfizetők egymást a központon keresztül érik el, az összeköttetés a beszélgetés idejére létrejön.
A **távhívó központ**ok a nem ugyanazon helyi központok közötti kapcsolat kialakításában játszanak fontos szerepet. A helyi központok több vezetékpárral (nevük: **helyközi trönk**) kapcsolódnak a távhívóközponthoz. Ezeken keresztül a helyi központok közötti információcsere valósul meg. Két előfizető egy adott távhívóközponton keresztül történő összekapcsolása csak akkor lehetséges, ha mindkét előfizető helyi központja ugyanazon távhívóközponthoz kapcsolódik.
Ha a távhívóközpont nem közös, akkor az összeköttetés kialakítása a kapcsolóközpont-hierarchia következő szintjén történik. Ezek a magasabb szintű kapcsolóközpontok segítségével valósulnak meg.

7. Digitális átvitel

Egészen a közelmúltig a távbeszélőrendszerekben az átvitel analóg módon történt. Az aktuális hangjeleket változó villamos feszültségjel formájában juttatták el a forrásállomástól a célállomásig. A digitális elektronika és a számítógépek kifejlődése lehetővé tette a digitális átvitelt. A digitális rendszerekben csak két feszültségszint megengedett, például a digitális 0 értékhez 0V, a digitális 1 értékhez 5V tartozik.
A digitális átvitelnek számos előnye van az analóg átvitellel szemben. Ha a jeleket nagyobb távolságra akarjuk továbbítani, a csatorna veszteségei miatt erősíteni kell azokat. Az analóg jelek erősítéskor mindig veszítenek valamennyi információt, ráadásul ez a veszteség halmozódik (jelalaktorzulás, zajnövekedés). A digitális jelek is torzulnak a csatornában, mivel azonban csak két jelalak van, a torzulás ellenére könnyebben felismerhetőek. Meghatározott távolságonként ún. jelismétlőket (digital regenerator) tehetünk a vonalra, amelyek helyreállítják az eredeti jelet. A digitális jel tetszőleges számú jelismétlőn mehet keresztül anélkül, hogy veszteséget szenvedne, így nagy távolságok esetén sem vész el az információ.

A digitális adatátvitel leglényegesebb előnyei:

- a hibaarány alacsony szinten tartható;
- beszéd, adat, zene és kép (pl. televízió, fax, videó) együttes (integrált) továbbítását is lehetővé teszi;

- a jelenleg meglévő vonalakon is jóval nagyobb adatsebességet lehet elérni ezzel a módszerrel;
- a digitális átvitel sokkal olcsóbb, mint az analóg;
- egy digitális rendszer üzemeltetése egyszerűbb, mint egy analóg rendszeré.

Ennek köszönhetően az összes nagytávolságú trönköt folyamatosan lecserélik digitális vonalakra, a régi analóg átvitel rézvezetékeit digitális átvitelre alkalmas fényvezető kábelekkel váltják ki.

7.1. Adatátviteli eljárások

A digitális átvitel során mindig biteket viszünk át. Eleinte csak szövegeket és számokat kellett továbbítani, ezért az átvitt információ egysége a bitcsoport volt. A bitcsoport a szöveg egy karakterét kódolta, lényegében az egy billentyű lenyomásakor keletkező kódot jelentette. Az ilyen, bitcsoportokat átvivő módszert szokták karakterorientált átviteli eljárásnak nevezni. Az átvitt információ egysége a karakter. A hosszabb üzenetek átvitelének vezérlését speciális ún. vezérlő karakterek biztosítják.

A bitorientált adatátviteli eljárásokra azért volt szükség, mert a hálózatok elterjedésével a szöveges jellegű információk mellett más jellegű információk átvitele is szükségessé vált a sokszor eltérő szóhosszúságú és adatábrázolású számítógépek között. A bitorientált eljárások tetszőleges bitszámú üzenet átvitelére alkalmasak.

Az adó által útjára bocsátott biteket a vevőnek helyesen kell tudnia értelmeznie. Mivel az átvitel sorban, bitenként történik, valahogy biztosítani kell az adó és a vevő szinkronizmusát, azaz azt, hogy az n-ediknek elküldött bitet a vevő szintén az n-ediknek érkező bitként érzékelje.

Az adatok helyes felismerésére két módszert dolgoztak ki.

- a **szinkron** és
- az **aszinkron** átviteli módszert.

A **szinkron átviteli módszernél** az egyes bitek jellemző időpontjai (kezdetük, közepük és a végük) egy meghatározott alapidőtartam egész számú többszörösére helyezkednek el egymástól. Ez azt jelenti, hogy egy üzenet bitjei szigorú rendben követik egymást. A szinkronizmust egy speciális bitcsoport (szinkron bitek) érzékelése biztosítja. A vevő ezt érzékelve – mintegy átvéve az adó adási ütemét – már helyesen tudja az ezt követő biteket vagy bitcsoportokat (karaktereket) értelmezni.

A karakterorientált eljárások az **aszinkron, ún. START-STOP átvitel**t használják. Ez a legrégibb adatátviteli módszer, melyben a szinkronizmus az adó és a vevő között csak egy-egy karakter átvitelének idejére korlátozódik, és az adatvonalon lévő adatjelek maguk végzik el a szinkronizálást. Az egyszerre átvitt adatmennyiség 5, 6, 7

vagy 8 bit lehet. Az adatfolyamot egy START bit előzi meg, amely egy 1 bitnyi ideig tartó alacsony szintű jel. Ezt követik az adatok, először a legalsó helyiértékű bit, majd a többi, végül a legfelső helyiértékű bit kerül átvitelre. Az adatokat követi a paritásbit, melyet nem kötelező használni. A paritásbit után egy vagy két STOP bit következik. Ezek tartoznak egy adatkeretbe (frame). A szinkronizálás az átvitel elején, a START bittel kezdődik, az adó és a vevő órájának annyira együtt kell futnia, hogy a szinkronból ne essenek ki egy keret átvitele alatt. A paritásbit az adatátvitel biztonságát növeli. A STOP bit vagy bitek feladata a keret lezárása. Ezzel az egymás utáni kereteket is szétválasztja.

Adó- és vevőoldalon az átvitel kezdetekor az adatátviteli paramétereket egyeztetni kell, pontosan ugyanazokat beállítva mind az adónál, mind a vevőnél. A START bit fix hosszúságú, mindig egy bit. A STOP bitek száma 1, 1,5 és 2 lehet. Az adatbitek száma 5, 6, 7 vagy 8 lehet. A PC-k esetében az adatok szinte kivétel nélkül 1 bájt hosszúságúak, azaz 8 bitesek.

A paritásbit használatának öt lehetséges kombinációja van:

a. Nincs paritásbit
b. A paritásbit mindig alacsony szintű (0)
c. A paritásbit mindig magas szintű (1)
d. Páros paritás: akkor beszélhetünk erről, ha a paritásbit az adatblokkban lévő 1-eseket páros számúra egészíti ki.
e. Páratlan paritás: ebben az esetben a paritásbit olyan értéket vesz fel, hogy az adatblokkok 1-eseivel együtt páratlan számú 1-es legyen.

A vevő újra kiszámítja a paritásbit értékét, és ha a vett érték nem egyezik a számítottal, azt jelenti, hogy hiba történt az adatátvitel során. Egy paritásbittel páratlan számú bithibát lehet felismerni. Ha viszont két bit változik meg, akkor a paritásbit értéke még mindig helyes lesz, tehát a hiba rejtve marad.

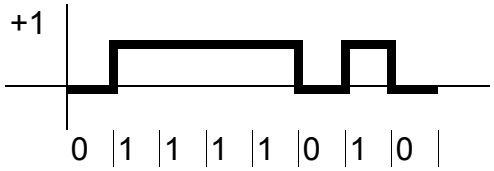
7.2.Digitális jelek kódolása

Ahogy az előbb már tisztáztuk, a digitális átvitelben két állapotot különböztetnek meg: az 1-et és a 0-t.
Tekintsük át azokat az eljárásokat, melyek segítségével a digitális jelek a fizikai adatcsatornán el tudnak jutni a vevőtől a célig. Az eljárásoknál a következő tényezőket kell figyelembe venni:

- a kódolás minél kisebb sávszélességű legyen (kevés váltást tartalmazzon), hogy minél több információt lehessen átvinni,
- minél kisebb legyen a jelek egyenfeszültség-összetevője (azaz legyen benne lehetőleg +1 és -1 is, illetve ezek nullázzák ki egymást), így kevésbé gyengül a jel,
- az adó és a vevő közti szinkronizáció segítségével biztosítható legyen a két kommunikáló fél azonos ütemű adatfeldolgozása.

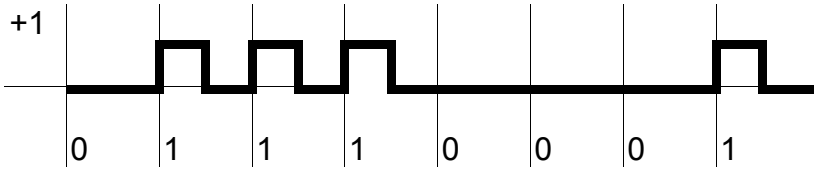
NRZ (Non Return to Zero) – Nullára vissza nem térő kódolás:
A digitális értéket veszi fel a jel is. Csak akkor használható, ha biztosítjuk az adó és a vevő közti szinkronizációt egy külön csatornán. Erre azért van szükség, mert sok egyes, vagy sok nulla esetén a vevő „eltévesztheti” a kapott jelek számát.

Elvileg az adó és a vevő órajele azonos. Az adás úgy történik, hogy minden órajelkor ad egy jelet az adó. A vevő pedig minden órajelkor megnézi, hogy milyen jel érkezett. Ha az adó és a vevő órajele egy picit eltér, hosszú idő alatt akkora lehet az eltérés, hogy rossz időben nézi meg a vevő, hogy milyen jel érkezett. Ennek legegyszerűbb kiküszöbölési technikája, ha a szinkron jel határozza meg, hogy mikor kell jelet küldeni, illetve mikor kell jelet fogadni.

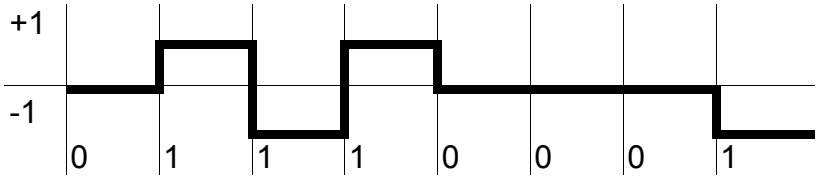


A sok egyes esetében a szinkronizációt a „bitbeszúrás” módszerével oldják meg. A módszer lényege, hogy 5 egymást követő egyes után az adó automatikusan beszúr egy 0-t, amit a vevő is érzékel, de nem dolgoz fel.

RZ (Return to Zero) – Nullára visszatérő:
Minden 1-es után visszatér a 0 szintre. Így az 1-esek esetében nem kell külön szinkronjel. Sok nulla esetében azonban még ez az eljárás sem hatékony.

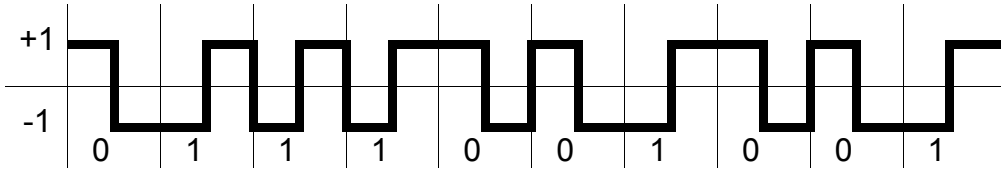


AMI (Alternate Mark Inversion) – Váltakozó 1 invertáló:
A szinkronizációt úgy oldja meg, hogy az egymás utáni 1-esek +1, illetve -1 között váltakoznak. (Sok 0 esetén még mindig megoldott a szinkronizálás.)



Manchester-kódolás:
0 és 1 esetén is van jelváltozás, így teljes a szinkronizáció. A jelváltozás iránya határozza meg, hogy 1-es vagy 0 következik. 1-es esetében negatívból változik a jel pozitívba, 0 esetében pozitívból negatívba.

Ezt a módszert használják a napjainkban legelterjedtebb Ethernet-hálózatokban.



7.3. Karakterek ábrázolása

Az előző fejezetben megismerhettük a fontosabb kódolási eljárásokat, melyeknek a segítségével a számítástechnikában 0 és 1-es biteket tudnak továbbítani. Ez azonban az adatok értelmezése szempontjából még nem elegendő. Nagyon bonyolult lenne az életünk, ha csak kettes számrendszerbéli számokkal kellene mindent leírnunk. (A számítástechnika őskorában ez így is történt. Ekkor elegendő volt egy 1-est egy 0-val felcserélni, és teljesen fals számítási eredményt kaptak.) Az emberek mindennapi életének egyszerűsítése érdekében bizonyos számú 1-est és 0-t tartalmazó bitsorozatot összevontak, és egy-egységnek kezeltek. Ezeknek az egységeknek a segítségével kódolták a betűket és az írásjeleket. A technika fejlődésével ezek a karakterábrázolási technikák is egyre fejlődtek. Tekintsük át röviden a fejlődés fontosabb állomásait!

ASCII kódrendszer

Az ASCII-karakterkészlet a legrégebbi karakterábrázolás a személyi számítógépekben. 7 bites jelábrázolást valósított meg (azaz 1 karaktert egy 7 bitből álló kód jelentett). 128 jelet tudtak vele tárolni. Így az angol abc betűin, a számokon, vezérlőjeleken kívül alig fért bele egy-két speciális jel. (Például szerepelt benne az á betű kódja, de a Á betűé már kimaradt; nem tartalmazott ő betűt, csak ô jelet.) Ezért igényes magyar nyelvű szövegek írására, nyomtatására nem volt alkalmas. Így volt ez más nemzetek karaktereivel is, ezért kénytelenek voltak a karakterábrázolással foglalkozó szakemberek egy fejlettebb jelkészlet kialakítására.

Latin 1 karakterkészlet

Ez már 8 bites karakteráblázolás volt, ebből következően több jelet tudott előállítani. Magában foglalta számos európai ország nemzeti jelkészletét (francia, görög stb.). A teljes magyar jelkészlet ábrázolására azonban még ez a jelkészlet sem adott megoldást.

852-es kódlap

Az ASCII-kódkészlet nem gyakran használt jeleinek a nemzetek jeleivel való lecserélése révén jöttek létre az ún. kódlapok. Ezek már teljes megoldást nyújtottak a nemzeti írásjelek alkalmazására. A magyar a 852-es kódlap volt. A megoldás hátránya, hogy ha több kódlapot egyszerre kellett használni, akkor a memóriába való be-ki lapozásával lehetett csak megoldani. (Ugyanis a memóriában egyszerre csak egy kódlap lehetett. Ha pl. magyart és franciát kellett felváltva használni, akkor kódlapot kellett cserélni a különböző nyelveknél.)

Unicode

A végleges megoldást 1987-ben a XEROX cég fejlesztése jelentette. Készítettek egy olyan jelkészletet, ami 16 bites, tehát 216=65536 karaktert lehet vele ábrázolni. Ebbe már belefér a világ összes abc-jének kódolása, még a holt nyelveké is. Az ASCII kóddal ellentétben nincsenek benne vezérlőjelek, ebből következően sokkal egyszerűbb, bár a kompatibilitás érdekében szerepelnek benne az ASCII-vezérlőjelek. Hátránya viszont, hogy sokkal több helyet foglal, mint az ASCII.

8. Az RS 232 szabvány

Most már tudunk jeleket kódolni, a kódolás során karaktereket kialakítani. Kérdés, hogy hogyan juttathatjuk el karaktersorozatunkat az egyik számítógéptől a másikig? A legegyszerűbb megoldás, ha a meglévő telefonos hálózatot használjuk az adatátvitelre. A telefonhálózat viszont analóg jeleket használ, a számítógép pedig digitális jeleket. A két rendszer összehangolására szolgál a modem, amelyet valamilyen módon csatlakoztatni kell a számítógéphez. Ennek egyik módja a soros vonali kapcsolat.

A soros vonal nemcsak a modem miatt fontos, hanem két számítógépet is össze tud kapcsolni, illetve tetszőleges berendezés és a számítógép közötti kommunikáció is megvalósítható a segítségével.

Soros kommunikációt megvalósító interfész ma már minden számítógépben található. (Az interfészt Amerikában az RS 232C, míg Európában a CCITT V24/V28 szabvány jelöli). Nem csak számítógépek összekapcsolására, hanem perifériák géphez csatlakoztatására is használják (pl. egér).

A soros működés azt jelenti, hogy az adatbitek egy vezetéken egymás után magadott ütemben kerülnek átvitelre. Egy vezetékén egyirányú átvitel valósítható meg. Az RS-232-ben a 0 logikai szintnek +3 és +15V, a logikai 1 szintnek -3 és -15 V (volt) közötti értékek felelnek meg. A nagy tűrési tartomány a zavarérzéketlenséget növeli. Az interfész segítségével az áthidalható távolság két gép között néhányszor tíz méter lehet.

Adatátvitelnél fontos jellemző a sebesség. Az aszinkron kommunikáció tipikus értékei:

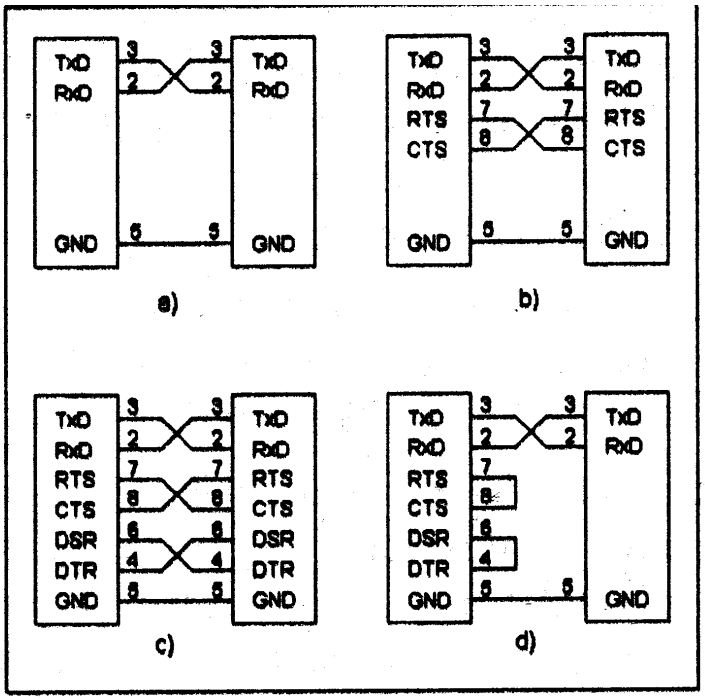
56 75 110 300 600 1200 2400 4800 9600 19200 38400 bit/s

Az interfész jeleit a szabványban rögzítették. Számítógépeinken a soros interfész 9 vagy 25 pólusú csatlakozón van kivezetve.

25 pólusú csatlakozó	9 pólusú csatlakozó	A jel rövidítése és neve
2	3	TxD (Transmit Data, átvitt adat)
3	2	RxD (Receive Data, vett adat)
4	7	RTS (Request to Send, adáskérés)
5	8	CTS (Clear to Send, adásra kész)
6	6	DSR (Data Set ready, modem üzemkész)
7	5	GND (Ground, föld)
8	1	DCD (Data Carrier Detect, vivőérzékelés)
20	4	DTR (Data Terminal Ready, számítógép üzemkész)
22	9	RI (Ring Indicator, telefoncsengés)

34. ábra
Az RS-232C interfész csatlakozóinak jelkiosztása

A soros kommunikációra képes eszközöket többféleképpen is összeköthetjük. Ezekre az alábbi ábra. mutat példát
Minden lábszámozás (azaz processzor lábainak sorszáma) a 9 csatlakozóra vonatkozik. A bal oldali csatlakozók a PC-hez, a jobb oldaliak a csatlakoztatott eszközhöz tartoznak.



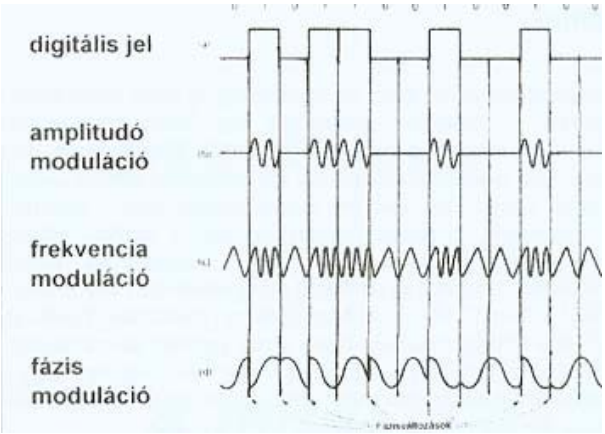
35. ábra
RS 232-es összeköttetések

Az a) esetben a legegyszerűbb kétirányú átvitelre alkalmas összekapcsolást láthatjuk. Ebben az esetben a két irányhoz összesen két adatvezeték kell.
Még egyszerűbb ennél, ha csak egyirányú átvitelt szeretnénk. Ebben az esetben csak egy adatvezetékre van szükség. Az egyenrangú kapcsolat megvalósításához az összetartozó jelpárokat keresztbe kell kötni. A PC az adatokat a TxD vonalon adja ki, a soros eszköz pedig az RxD bemeneten érzékeli azt. Az a) ábrán látható összekapcsolási módot nevezzük nullmodem kapcsolatnak. Ebben az esetben nincs hardveres visszajelzés az adat hibamentes átvitelére vonatkozóan, hibátlan vételt feltételeznek. A hibaellenőrzést természetesen szoftveres megoldással lehet pótolni. Ebben az esetben az RS 232-es porton keresztül két számítógépet kötnek össze, a modemek kihagyásával. Az összeköttetés lényege, hogy az egyik gép adókapuját közvetlenül összekötik a másik gépével.

9. Modemek

A soros kommunikáció digitális jeleket használ. Az interfész jelei nagy távolságra nem juttathatók el közvetlenül. Természetes igény, hogy az egész világot behálózó telefonrendszert használjuk fel gépeink kapcsolatához. A hagyományos telefonrendszereket azonban beszéd céljára tervezték. Az emberi beszédhang nem tartalmaz 300 Hz-nél alacsonyabb és 3,3 kHz-nél magasabb hangokat. Így a telefonrendszerek az ezen kívül eső tartományt levágják, nem továbbítják. Ha a telefonrendszeren adatokat akarunk átvinni, az információt az ebbe a tartományba eső szinuszos jel valamely jellemzőjéhez kell rendelni. Ezt a szinusz hullám modulálásával tudjuk megvalósítani. Azt az eszközt, mely a bemenetére adott digitális jel vezérlésével a modulációt elvégzi, illetve a modulált, analóg jelet visszaalakítja (demoduláció) digitális bitsorozattá, modemnek (**mo**dulátor-**dem**odulátor) nevezzük.

Minden szinuszos jelnek három olyan jellemzője van, melyet a moduláció során felhasználhatunk:



36. ábra
Szinuszos jelek modulációja

Amplitúdómoduláció

A legrégibb modulációfajta. Ezt használták a morzejelek küldésekor. Egy adott amplitúdójú jel (hang) megléte vagy a hiánya hordozza a az információt. Az 1-et jelentő adatbit küldésekor egy vagy több szinuszhullám továbbítódik, a 0-át jelentő adatbit esetén nincs továbbított szinuszjel. A megvalósítás jól működik, de nagyon zavarérzékeny, és csak lassú átvitelre használható.
A jelek visszaállítására komparátorokat használnak. Ezek olyan eszközök, melyek összehasonlítják a vett jel amplitúdóját egy viszonyítási szinttel, és ennek alapján értelmezik a jelet 0-nak vagy 1-nek.

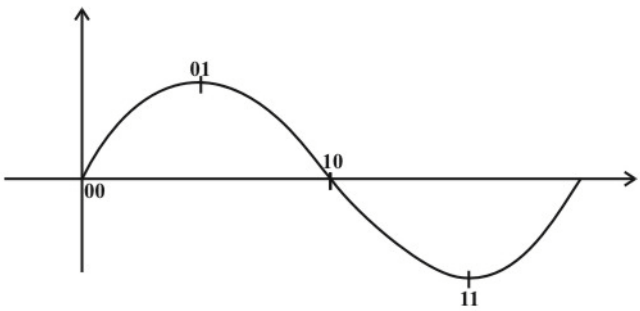
Frekvenciamoduláció

Ezt a modulációt használták először a modemeknél. Nagyon jó a zajtűrése és a biteket hordozó frekvenciákat egyszerű szűrőkkel nagyon könnyű szétválasztani.
A digitális információban lévő 1 és 0 állapotokat azzal valósítják meg, hogy a szinuszos hordozójel frekvenciáját változtatják meg. Az információt hordozó két állapothoz külön frekvencia tartozik. A frekvenciakülönbségnek elég nagynak kell lennie ahhoz, hogy biztonságosan szét lehessen választani.

Fázismoduláció

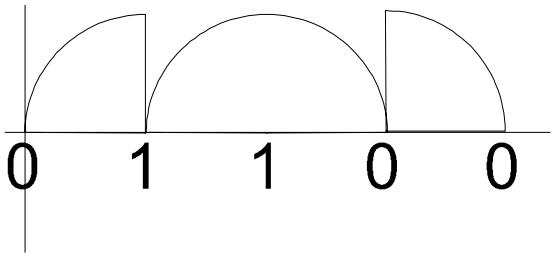
Az elnevezés utal a módszer lényegére. A modulációs módok közül ez a legbonyolultabb. Az adatbitek a jel kezdőfázisát változtatják meg, pl. 0 adatbit esetén a kezdőfázis 0 fok, 1-es adatbit esetén 180 fok. Ha a 360°-os fázistartományt felosztjuk négy részre, akkor a sík négy irányába mutató, egymással 90°-os szöget bezáró fázisvektorral lesz leírható. Mivel itt négy különböző állapot van, ezért négy fázisvektorral összesen két bitet lehet kódolni. A négy állapot a következő:

00 adatbitek esetén az átmenetnek a 0°-os fázistolás felel meg, tehát nem történik fáziseltolódás.
01 adatbitek esetén az átmenetet 90°-os fázistolás jelenti.
10 adatbitek esetén az ilyen átmenethez a szinuszos jelet 180°-kal kell eltolni a kezdőfázishoz képest.
11 adatbitek esetén az átmenetet 270°-os fáziseltolás fogja jelölni.



37. ábra
Fázismoduláció 90 º-is fázistolással

Ha tovább szeretnénk az adatátviteli sebességet növelni, ez két módon lehetséges: vagy a fázishelyzetek, vagy az amplitúdóértékek számát növeljük meg. 8 fázisszög és 2 amplitúdó 4 bites információ átvitelét teszi lehetővé azáltal, hogy a 16 különböző fázisszög-amplitúdó kombináció lehetséges.
Kepzeljünk el egy egyszerű szinuszhullámot: ha a vizsgálat pillanatában szinuszhullámunk 0 értékű, akkor nincs jel, ha 1 értékű, akkor van jel. Tehát 0 értéket 0, 180, 360 fokoknál tudna felvenni, míg +-1 értéket 90, 270 fokoknál. Így 2 bitet tudunk megkülönböztetni.
0 1 1 0 0 kódolása:



Modulálás esetében a kezdő fázist a 0 értéktől eltoljuk. Pl 45° fázistolás esetében:

0000 értéket kapunk 0°-nál

0001 értéket kapunk 45°-nál

0010 értéket kapunk 90°-nál

0100 értéket kapunk 135°-nál

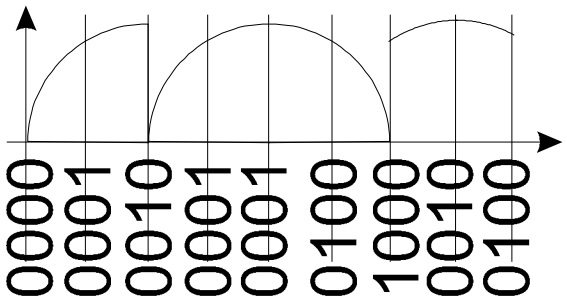
1000 értéket kapunk 180°-nál

1001 értéket kapunk 225°-nál

1011 értéket kapunk 270°-nál

1111 értéket kapunk 315°-nál

Tehát ha a szinuszhullámot időegységenként megvizsgáljuk, hogy éppen hány fokhoz tartozó értéket mutat, akkor meg tudjuk határozni, hogy milyen bináris értéket hordoz a jel. Pl:



9.1. A modemek felépítése

A modemek önállóan működni képes számítógép-perifériák. Egy modem csak akkor használható, ha a számítógép először felprogramozta, majd pedig szabványos, csak az adott modemre érvényes parancsokkal vezéreli, és az állapotát folyamatosan ellenőrzi. A kapcsolatban két egységet különböztetünk meg:

- ***DTE** (Data Terminal Equipment): adat-végberendezés, a számítógép vagy a terminál neve.*
- ***DCE** (Data Circuit-Terminating Equipment): adatáramköri végberendezés, a modem hivatalos elnevezése.*

Két modem a telefonhálózaton keresztül kerül összeköttetésbe.
A modem a számítógép soros portjára (RS232) csatlakozik. Ma már olyan modemeket is használunk, amelyek a számítógép alaplapjára csatlakoztathatóak tehát a szabványos ISA vagy PCI sínbe illeszkednek. Az előbbieket nevezzük külső, míg az utóbbiakat belső modemeknek. A belső modemek felépítése néhány különbségtől eltekintve azonos, az eltérés csupán a csatolófelület kialakításában van. A működés lényege azonban ugyanaz.

10. Cellás mobiltelefonok

A mobiltelefont eleinte a beszédalapú kapcsolattartásra használták, napjainkban azonban egyre inkább mindentudó adatátviteli eszközzé – mobil „társsá” – válik. Az élet minden területén tanúi vagyunk a mobil kommunikáció és a mobil internet-hozzáférés iránti igény radikális erősödésének.

- Mobil telefonálásra kezdetben az analóg rendszerek szolgáltak a 450MHz-es sávban. Ezek adatátvitelre maximum 1,2 kbps, vagy jó esetben 2,4 kbps sebességgel voltak használhatók (első generáció).

- A 900 MHz-es és az 1800 Mhz-es sávot használó GSM, valamint az ezt követő rendszerek már digitális átvitelre épültek. A GSM adatátviteli sebessége 9,6 kbps lehet. Bár beszédátvitelre maradéktalanul megfelel a rendszer, ez a sebesség adatátviteli igényeinket már ma sem elégíti ki. További hátrány, hogy ezek a rendszerek adatátvitel esetén is idő alapú elszámolást alkalmaznak, tehát nem az átvitt adatmennyiséget, hanem a vonal lefoglalását fizetjük ki (második generáció).

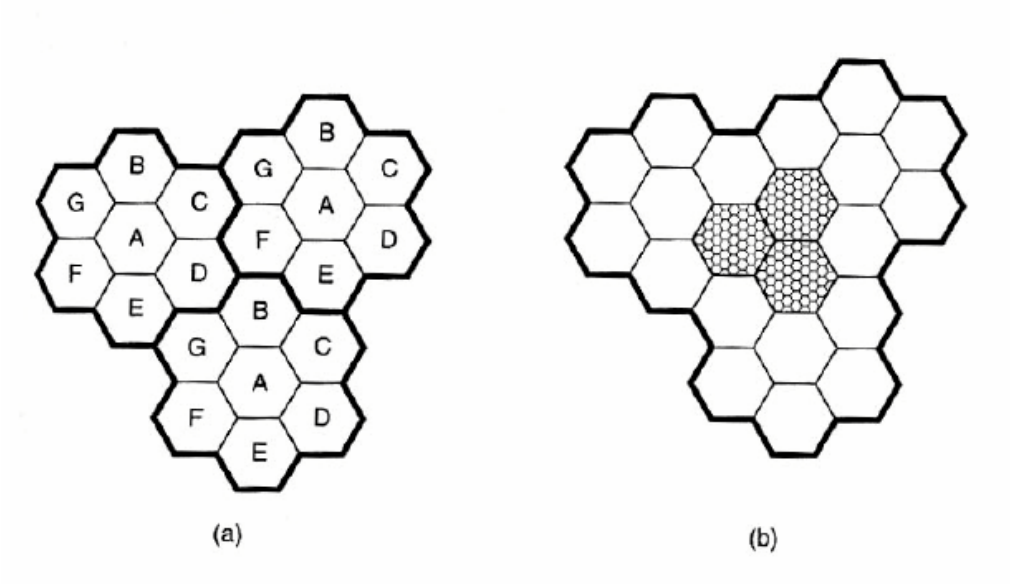
- A GPRS (General Packet Radio System – csomagkapcsolt mobiltelefon-szolgáltatás) a hagyományos beszédátvitel mellett kb. 60-70 kbps sebességet biztosít adatátviteli célra, és már nem vonalkapcsolásos elven működik, azaz fizetni csak az átvitt információs csomagokért kell („2,5” generáció – felfogható úgy is, mint a második generáció továbbfejlesztése).

- Az UMTS rendszer (Universal Mobile Telecommunication System –Univerzális Mobiltelefon-rendszer) az elkövetkező években várhatóan nevének megfelelően univerzális megoldást fog nyújtani szinte az egész világon az egyéni, az otthoni és az üzleti felhasználók számára. A 2 GHz-es sávban működő rendszer 2 Mbps adatátvitelre is képes, azonban teljesen új hálózat kiépítését igényli, és a rendkívül magas beruházási költségek egyelőre késleltetik térhódítását (harmadik generáció).

A rádiótelefonok által használt frekvencián nem lehet nagy távolságokat áthidalni, és a frekvenciasáv is elég szűk, nem kaphat minden előfizető külön frekvenciát a beszélgetéshez.
A megoldást a cellás szerkezetű rádiótelefon-rendszerek jelentik, melyek az igényeket a rendelkezésre álló frekvenciatartomány kihasználtságának növelésével elégítik ki.

A cellás technika alapja:
- a cellaosztás és
- a frekvenciák ismételt felhasználása.

A területet kisebb részekre osztják. A cellákon belül egy központi rádióállomás tartja a mozgó előfizetőkkel a kapcsolatot. Mivel egy cella mérete nem túl nagy, és központi rádióállomás jele a cellán kívül már nem fogható, a hullámterjedés sajátosságai lehetővé teszik, hogy egy bizonyos távolság felett újra fel lehessen használni a frekvenciasávot. Így ugyanaz a frekvencia egyidejűleg több, egymástól megfelelő távolságban lévő cellában is kioszthatő. A cellák alakját szabályos hatszög alakra érdemes tervezni, ez azonban csak elméleti megközelítésként használható, ugyanis a valós cella alakját és méretét a helyi körülmények figyelembevételével kell meghatározni (domborzati viszonyok, épületek stb.). A celláknak azt a legkisebb csoportját, ahol a használható frekvenciákat tartalmazó csatornakészlet kiosztásra kerül, cellacsoportnak (clusternek) nevezik.
Az azonos frekvenciákat használó cellák közötti távolságot úgy kell megválasztani, hogy az azonos csatornák zavaró kölcsönhatása elfogadhatóan kicsi legyen.



38. ábra
A cellás rádiótelefon-rendszer felépítése
a) A szomszédos cellék különböző frekvenciát használnak
b) Több felhasználó esetén csökkenteni lehet a cellák méretét

A mobil felhasználók egy cellán belül a helyi bázisállomáson keresztül tartják a rádiós kapcsolatot. A bázisállomás-hálózat a mobilközponthoz csatlakozik rádiós vagy vezetékes összeköttetéssel.

A mobilközpont feladata:

- a cellás rendszer működésének vezérlése
- kapcsolattartás a mobil felhasználókkal
- a cellarendszer illesztése a nyilvános postai távbeszélő hálózathoz (azaz a nyilvános vezetékes telefonhálózat és a mobilhálózat összeköttetésének megteremtése)

Előfordulhat, hogy a mobiltelefonáló éppen a folyamatban lévő beszélgetés közben lép át egy cellahatárt. A rendszereknek gondoskodniuk kell arról, hogy ilyenkor az összeköttetés ne szakadjon félbe. A hívást átkapcsolják a következő cella egy csatornájára. Ennek feltétele, hogy a fogadó cella rendelkezzen kioszthatóbeszédcsatornával. Ezt a váltást **handover**nek vagy **handoff**nak nevezzük.

A cellák méretük szerint lehetnek:

- hiper: R >10 km, vidéken (R = sugár)
- makro: 0,5km < R < 10 km, városi területeken
- mikro: 0,1km < R < 0,5 km, nagyvárosok központjában
- nano: 50m < R < 100 m, épületen belül
- piko: 20m < R < 50 m, épületen belül

A cellaméret csökkentésére a nagy forgalmi igények miatt került sor, ugyanis ekkor a nagy cellás rendszerekben már elfogadhatatlanul nagy frekvenciakészletre lenne szükség.
A rendszer kapacitása szerint lehet kis-, közép- és nagykapacitású. Kiskapacitású hálózatok nagycellás felépítéssel a 450 MHz alatti frekvenciasávokban, a közép- és nagykapacitásúak kiscellás felépítéssel a 450 és 900 MHz-es, illetve a 900 Mhz fölötti sávban (1800 MHz) üzemelnek.

10.1. Barangolás

A mobilitás feltétele a barangolás (roaming). Ha egy mobilállomás a térerősség csökkenését, a jel/zaj viszony romlását érzékeli, automatikusan keresni kezd egy másik bázisállomást, és ha talál, arra kapcsolódik át.
A roamingot kapcsolatvesztés nélküli cellaváltásként lehet definiálni. Ez a művelet két alapvető pontban tér el a cellás telefonok handoverétől:

Egy csomagkapcsolt LAN-rendszerben a cellaváltás létrejöhet két csomag átvitele között, ellentétben a telefóniával, ahol akár beszélgetés közben kell megvalósítani a cellák közti átmenetet. Ez könnyebbé teszi a LAN-roamingot, viszont egy beszédátviteli rendszerben egy pillanatnyi kapcsolatszakadás nem befolyásolja jelentősen a beszélgetést, míg egy csomagalapú környezetben a felsőbb rétegprotokollok újraadása komolyan lecsökkenti a teljesítményt.

10.2. Csatornakiosztás

Egy-egy cellában csak meghatározott számú frekvenciát lehet kiosztani, így a cellán belül a telefonálók száma korlátozott. (Mindenkinek ismerős lehet a szilveszteri „csúcsforgalom” mikor percekig nem lehet vonalat kapni, vagy „a hálózat túlterhelt” üzenetet kapjuk.) Ezt a korlátot igyekeznek a mobiltelefon társaságok enyhíteni a következő elvek segítségével:

- **Fix csatornakiosztás**: ebben az esetben a központ előre meghatározott számú csatornákkal dolgozhat, nincs módja a forgalom növekedése során új csatornákat nyitni. Ha elfogy a rendelkezésre álló csatorna, az új hívást kezdeményezőnek várnia kell az egyik csatorna felszabadulására.
- **Dinamikus csatornakiosztás**: ebben az esetben, ha elfogy az előre definiált csatorna, a központ a szomszédos cellából kérhet szabad csatornát. Így dinamikusan változhat a csatornák száma a leterheléstől függően. Gond csak akkor van, ha elvon a szomszédos cellától csatornát és a szomszédnak is szüksége lenne szabad csatornára. Ekkor ő is elvon a szomszédos cellától. Így tovább gyűrűzhet a cellák elvonása.
- **Hibrid csatornakiosztás**: a rendelkezésre álló csatornák egy részét fix módon, másik részét dinamikus módon használja a cella.
- **Adaptív csatornakiosztás**: a csatornákat először fix módon kiosztja, majd hagyja, hogy bizonyos ideig dinamikusan kezeljék a cellák. Az előre meghatározott idő letelte után azonban újra kiosztja a csatornákat, így próbálja megakadályozni a túlzott csatornaelvonásokat.

11. ISDN - integrált szolgáltatású digitális hálózat

Az **ISDN** (Integrated Services Digital Network – integrált szolgáltatású digitális hálózat) létrehozásának célja az volt, hogy egyesítsék a hang és nem hang átviteli szolgáltatásokat egy gyors, digitális rendszerbe.

Az ISDN legfontosabb szolgáltatása továbbra is a hangátvitel, de kiegészült más szolgáltatásokkal is, például:

- hívószám-azonosítás
- hívószámkijelzés
- zárt felhasználói csoportok kialakításának lehetősége
- videotex (távoli adatbázis lekérdezési lehetősége)
- elektronikus levelezés lehetősége
- távérzékelők összekapcsolásának lehetősége (riasztók, füstjelzők…)

Az ISDN működési elvének az alapja a digitális bitcső. Ez egy olyan képzeletbeli cső, amely az adót és a vevőt köti össze. A bitcsőben bitek áramolnak. A bitcső időosztásos multiplexeléssel több csatornára osztható, így egyszerre több kapcsolat is létrejöhet. Attól függően, hogy milyen terheléssel szeretnénk használni a bitcsövet, két szabványt dolgoztak ki. Egy kisebb sávszélességűt az otthoni felhasználásra, és egy nagyobb sávszélességűt üzleti felhasználásra.
A szolgáltató és a vevő között digitális vonalat építenek ki. A vevőnél a szolgáltatás igénybevételére egy ún. **NT** (Network Terminator) felszerelése szükséges.

Magáncélú (esetleg kisebb vállalati célú) ISDN felhasználásra **NT1**, míg üzleti célú felhasználásra **NT2** kerül felszerelésre.

- NT1 esetében legfeljebb 8 ISDN telefon, terminál (számítógép), riasztó fűzhető fel.
- NT2 esetében az NT-re egy alközpont kapcsolódik. Erre az alközpontra kapcsolhatók ugyanazok a berendezések, mint a magáncélú hálózatnál, azzal a különbséggel, hogy a rákapcsolt berendezések számát a központ korlátozza, nem az NT.

Az ISDN bitcső időosztásos multiplexelés segítségével egyszerre több csatornát is tud kezelni. A csatornák a következők lehetnek:

A 4 kHz-es analóg telefoncsatorna
B 64 kbit/s-os PCM csatorna hang- és adatátvitel céljából
C 8 vagy 16 kbit/s-os digitális csatorna
D 16 vagy 64 kbit/s-os digitális csatorna vezérlésre
E 64 kbit/s-os digitális csatorna belső ISDN-jelzésre
F 384, 1536, 1920 kbit/s-os digitális csatorna

Hazánkban a csatornák kiosztásának két változata terjedt el: magáncélra az ISDN2-t ajánlják, ami 2 B csatornát és 1 D csatornát tartalmaz. Ehez általában 3 telefonszámot lehet rendelni, és a két B csatorna függetlenül használható. Az internethasználat meggyorsítása érdekében a 2 B csatorna összevonható.
Üzleti célra az ISDN 30 az ajánlott. Itt 30 B csatornát és 1 D csatornát bocsátanak az ügyfél rendelkezésére. Ez 30 egymástól független telefonszámot és beszélgetést jelent.

További információk az ISDN-ről:
Kérdések és válaszok az ISDN-ről
Várnai Tamás: ISDN
Az ISDN-ről

12. ADSL

Az Asymmetric Digital Subscriber Line – aszimmetrikus digitális előfizetői vonal (ADSL) – technológiáját tíz évvel ezelőtt találták fel. A lényege, hogy a rézkábelek normál összeköttetésben ki nem használt áteresztőképességének segítségével az átviteli sebesség megszázszorozható.

Az ADSL-1 elméletileg 1,53 Mb/s-os felhasználó felé irányuló és 16 Kb/s-os központ felé irányuló sebességet kínál. Az ADSL-2 és ADSL-3 ennek javításával kedvezőbb arányt tud elérni, de a központ felé történő adatátvitel mindenképpen lényegesen kevesebb, mint a felhasználó felé történő adatátvitel. Ez azonban általában nem zavaró, mert internetezéskor inkább lekérünk adatokat, mint küldünk.

13. Terminálkezelés

Sokszor előfordul az alkalmazott számítógép-hálózatok esetében, hogy a hálózati vonalak végén lévő berendezéseknek nem kell komoly adatfeldolgozási vagy számítási feladatot ellátniuk, csupán adatbeviteli pontként kell működniük (pl. bankkártyaterminál, jegykezelő-terminál stb.) Ebben az esetben valamilyen kommunikációs vonal segítségével összekötik a terminált a központi számítógéppel.

Terminálnak nevezzük a monitorból, billentyűből, kommunikációs egységből álló megjelenítési és beviteli eszközt. Feladata, hogy kapcsolatot tartson a központi egységgel.

Fajtái

- **Egyszerű, intelligencia nélküli terminál**: csak arra képes, hogy a központ felé küldje a bevitt adatokat, illetve, hogy a központból érkező adatokat megjelenítse. Minden adatfeldolgozási művelet a központi számítógépen történik (ilyen pl. egy kártyás ajtónyitó).
- **Korlátozott intelligenciával rendelkező terminál**: bizonyos műveletek elvégzésére önállóan is képes, saját memóriája, processzora is van (ilyen pl. egy bankkártya terminál, melynek saját menüvezérelt programmal rendelkezik, de minden fontos műveletnél a központi géppel kommunikál.)
- **Terminálként működő kisszámítógép**: egy hagyományos PC, amely önállóan is képes munkát végezni, de bizonyos esetekben terminálként üzemel. Ilyenkor nem a saját processzorát, memóriáját használja, hanem a központi gépét (ilyen pl. a Telnet program).

Előfordulhat, hogy egy kommunikációs vonalon több terminál osztozik; ezek irányítását az ún. **terminálvezérlő** valósítja meg.
A kapcsolódás **többpontos** (multidrop) illetve **pont-pont** típusú lehet. Többpontos esetben a terminálok egy kommunikációs vonalon osztoznak, míg pont-pont kapcsolatnál minden terminál saját vonalon kapcsolódik a terminálvezérlőhöz.

A terminálvezérlő feladata a terminálok adási jogosultságának kiosztása, megakadályozva azt, hogy több terminál egyszerre próbáljon adást kezdeményezni. Az adási szándékról úgy szereznek tudomást, hogy lekérdezik a terminálokat. Ezt a lekérdezést **pooling**nak nevezik.
Ennek megvalósítására több eljárás létezik:

- **Körbekérdezés** (roll-call pooling): lényege, hogy a terminálvezérlő elküld egy üzenetet az összes terminálnak, melyben egy adott terminál adási szándékáról érdeklődik. A címzett terminál válaszként vagy adatot küld vissza, vagy egy speciális jelet, amely azt jelent, hogy nincs adási szándéka. Természetesen az adás nem lehet korlátlan hosszúságú. Ha több adatot szeretne elküldeni, mint ami egy adásba belefér, megvárja a következő adási lehetőséget. Elméletileg a központnak lehetősége van prioritás kiosztására úgy, hogy egymás után többször kérdezi le ugyanazt a terminált. Hátránya az időpazarlás: a terminálnak mindig válaszolnia kell.

- **Központ felé haladó lekérdezés**: az előbb említett hátrányt a terminálvezérlő úgy küszöböli ki, hogy a legtávolabbi terminálnak küld lekérdező üzenetet. Ha a terminál adni kíván, akkor az adatot visszaküldi a terminálvezérlőnek. Viszont ha nincs adási szándéka, a következő terminálnak elküldi az adási jogot, így megtakarítja a központba, majd a következő terminálhoz küldendő kéréseket.

Összefoglalás

E fejezetben megvizsgáltuk, hogy milyen lehetőségek vannak a vonalak megosztására. A vonalmegosztásnak azért van nagy jelentősége, mert a gyakorlatban lehetetlen annyi adatátviteli vonalat kiépíteni, amennyire szükség lenne, így meg kell oldani, hogy egy vonalon több kommunikáció is megvalósulhasson. A vonalmegosztási módszerek közül a mai technika szempontjából a legfontosabb a csomagkapcsolás. Ezt a módszert ma már nem csak a számítógép-hálózatoknál alkalmazzák, hanem a távközlésben, a műsorszórásban is.
Ezek után kitértünk a régi és a napjaink adatátviteli közegeire, a kábelek típusaitól kezdve a vezeték nélküli adatátviteli lehetőségekig.

Az átviteli közegtől függetlenül a digitális jeleket valamilyen módszer segítségével kódolni kell a jelátvitel megvalósításához. Ezeket a kódolási módszereket is áttekintettük, melyek közül gyakorlati szempontból a Manchester-kódolás a legfontosabb.
A fejezet utolsó egységeként az adatátvitel legrégebbi formájáról, a telefonrendszerekről esett szó. Itt áttekintettük a hagyományos analóg technikától kezdve a napjainkban használatos mobiltechnikán át az ISDN- és ADSL-adatátviteli lehetőségeket is.

Ellenőrző kérdések

1. Igaz-e a következő állítás? 1 baud = 1 bit/sec
   - Nem
   - Igen
   - Csak akkor, ha 2 különböző jelet kódolunk

   Válasz

2. Mit jelent a vonalak megosztása?
   - Azt, hogy egy adatvonalat több logikai csatornára osztunk.
   - Azt, hogy egy adatcsatornát több logikai vonalra osztunk.
   - Azt, hogy egy fizikai vonalat több fizikai csatornára osztunk.

   A megoldás bővebben itt

3. Melyik multiplexelési eljárást használja az Internet?
   - Vonalkapcsolás
   - Üzenetkapcsolás
   - Csomagkapcsolás

   A megoldás bővebben itt

4. Mit jelent a szinkron adatátvitel?
   - Az adatokat küldéskor egy speciális jel vezeti be.
   - Az adatokat csak meghatározott időközönként lehet küldeni.

   A megoldás bővebben itt

5. Melyik kódolási eljárást használják az Ethernet-hálózatokban?
   - RZ
   - AMI
   - Manchester

   A megoldás bővebben itt

III. Közeghozzáférési módszerek

Bevezető

Az előző fejezetekben már részleteztük, hogy a számítógép-hálózatok két nagy csoportba sorolhatók: a pont–pont kapcsolatot megvalósító és az üzenetszórást alkalmazó típusba.
Mivel napjainkban inkább az utóbbi eljárást alkalmazzák, mi is az üzenetszórásos rendszerrel foglakozunk részletesebben.
Ennél a módszernél a kommunikáló hosztok egy adatcsatornát használnak. Minden hoszt a közös csatorna segítségével próbál kommunikálni a másik hoszttal. Ez egy konferenciabeszélgetéshez hasonlítható, ahol ugyanazon a vonalon egyszerre több ember próbál hozzászólni. Elképzelhető, hogy mekkora zűrzavar keletkezik, ha mindenki egyszerre próbál beszélni. Az is világosan látszik, hogy minél többen vesznek részt a konferenciában, annál nehezebb az átlátható kommunikációt biztosítani.
Ugyanez érvényes a számítógép hálózatoknál is. Amíg a pont–pont kapcsolatnál jól elkülöníthető a két fél adása, az üzenetszórásnál mindenki ugyanabba a csatornába „dobja” be az adását, és ha a csatornán már volt adat, akkor a két adat értelmetlen jeleket eredményez. Ezért meg kell oldani, hogy több hoszt is tudja használni a csatornát anélkül, hogy az adataik összekeveredjenek.
Tehát a közeghez (mint adatátviteli közeghez) való hozzáférést biztosítani kell.

A probléma megoldására sok módszert dolgoztak ki, melyeket a következő csoportosítás alapján lehet kategorizálni:

Véletlen vezérlésű közeghozzáférés	**Osztott vezérlésű közeghozzáférés**	**Központosított vezérlésű közeghozzáférés**
ALOHA	Vezérjeles gyűrű	Lekérdezéses eljárás
Ütközést figyelő, ütközést jelző (CSMA/CD)	Vezérjeles sín	Vonalkapcsolásos eljárás
Réselt gyűrű	Ütközést figyelő, ütközést elkerülő (CSMA/CA)	
Regiszterbeszúrásos gyűrű		

1. Véletlen vezérlésű közeghozzáférés

Az eljáráscsoport lényege, hogy nincs szabály arra vonatkozóan, melyik hosztnak mikor kell adást kezdeményeznie, a csatorna használata véletlenszerű. Ha egyszerre több állomás próbálja használni a csatornát, akkor **ütközés** keletkezik, azaz mind a csatornán lévő jel, mind pedig az új adás jelei feldolgozhatatlanok lesznek. Erről mindkét adó értesül, és valamennyi idő elteltével megismétli az adást.

1.1. ALOHA

Az 1970-es években a Hawaii Egyetemen dolgozták ki ezt az igen egyszerű módszert. Az eljárás lényege, hogy az adók korlátozás nélkül adhatják üzeneteiket. Bármelyik hoszt bármikor, bármekkora adást kezdeményezhet. Ha az adás pillanatában már van üzenet a csatornán, akkor természetesen ütközés keletkezik, és mindkét üzenet elvész. Viszont az adók visszacsatolással adják le üzeneteiket, így értesülnek az üzenet elvesztéséről. Ekkor a küldő véletlen hosszúságú ideig vár, majd megismétli az adást. A rengeteg ütközés miatt a csatorna kihasználtsága nagyon rossz, a legjobb esetben is csak 18%-os.

1.2. Réselt ALOHA

Az egyszerű ALOHA túlságosan sok ütközést enged meg amiatt, hogy bármikor adást kezdeményezhetnek az állomások. Ezt úgy lehet csökkenteni, hogy nem engedélyezzük az állomásoknak a tetszőleges időpontban történő adást. A réselt ALOHA eljárásnál a felhasználóknak meg kell egyezniük abban, hogy mekkora időközönként lehet adást kezdeményezni. A meghatározott időközöket szokták időrésnek nevezni. (Pl. fél másodperces időrés azt jelenti, hogy fél másodpercenként kezdeményezhetnek adást a hosztok, ebből következően egy adás is fél másodperc hosszú lehet.) A szinkronizálás egy lehetséges módja, hogy egy kijelölt állomás speciális jelet bocsát ki az időintervallumok kezdetekor.
Réselt ALOHA esetében a munkaállomásoknak az adással meg kell várniuk a következő időrést.
Ezzel az egyszerű változtatással a csatorna kihasználtsága megduplázható. Bár még így se mondható hatékonynak az eljárás, de az egyszerű ALOHA eljárásnál jóval hatékonyabb.

1.3. Ütközést figyelő, ütközést jelző (CSMA/CD)

CSMA/CD: Carrier Sense Multiple Access with Collision Detection, azaz vivőjel érzékeléses többszörös hozzáférés, ütközésérzékeléssel.
Ellentétben az előző eljárásokkal, itt már van feltétele az adás megkezdésének. Csak akkor kezdeményezhet adást a hoszt, ha a csatorna üres, így nem ütközhet bele a már csatornán lévő adatba. Ezért az adó az adás megkezdése előtt belehallgat a csatornába, ha a csatorna üres, adást kezdeményez. Az elővigyázatosság ellenére az viszont előfordulhat, hogy két (vagy több) állomás egyszerre veszi észre, hogy üres a csatorna, és azonos pillanatban kezdeményez adást.

Ebben az esetben ismét ütközés keletkezett. Ennek a problémának a feloldása érdekében az adók az adás ideje alatt is figyelik a csatornát, és összehasonlítják a csatornán található adatot az általuk kibocsátott adattal. Ha azt tapasztalják, hogy a két adat nem egyezik meg, akkor minden bizonnyal ütközés keletkezett. Ebben az esetben fölösleges tovább folytatni az adást, tehát leállítják azt. Valamennyi ideig várakoznak, majd (ha szabad a csatorna) újra megpróbálkoznak az adással.
Fontos kérdés még, hogy mennyi ideig várakozzon az adó? Ezt a perzisztencia (kitartás) határozza meg.

- **1 perzisztens:**
  ütközés után 1 valószínűséggel ad, ha a csatorna üres (azaz ütközés után leállítja az adást, majd azon nyomban újrakezdi);
- **0 perzisztens** (non perzisztens):
  ütközés után véletlen hosszúságú ideig várakozik, majd újra megpróbálkozik az adással, ha a csatorna üres;
- **p perzisztens:**
  ha a csatorna szabad, akkor p valószínűséggel ad, vagy q valószínűséggel visszalép adási szándékától a következő időrésig(q = 1 – p), ahol ismét megvizsgálja, hogy a csatorna szabad-e.

A CSMA/CD sokkal jobb protokoll, mint az előzőek, mert egyrészt figyel a csatorna foglaltságára, másrészt ütközés esetén rögtön befejezi az adást, így nem foglalja a sávszélességet és az időt.

A protokoll szerint működő állomások három állapotot vehetnek fel:

- **versengéses állapot**: vár az adás megkezdésére, versenyez a többi állomással, hogy ki adhatja le hamarabb az üzenetét,
- **átviteli állapot**: éppen adatátvitelt bonyolít le,
- **tétlen állapot**: nincs továbbítandó adata.

A CSMA/CD az egyik legfontosabb protokoll, mivel a gyakorlatban az Ethernet-kártyák ennek egy változatát, az IEEE 802.3-at alkalmazzák.

1.4. Réselt gyűrű

Gyűrű-topológia esetén alkalmazható protokoll. A réselt ALOHA-hoz hasonlóan itt is egységnyi részekre (keretekre) kell feldarabolni az átküldendő adatokat. Csak akkor adhat az állomás, ha a szinkronjel a rés kezdetét jelzi. Egy–egy állomás mindig csak egy „résnyi” adatot adhat le.
A módszer lényege, hogy az állomások megvizsgálják a csatornát, azaz az éppen hozzájuk érkezett keretet. Ha a keret foglaltságát jelző marker szabadnak mutatja a keretet, az adó a csatornára teheti az adás egy keretnyi részét, és bebillenti a markert foglalt állásba. A többi részével az adásnak meg kell várnia a következő szabad keretet.
A vevő szintén megvizsgálja a keretet: ha neki címezték az adatot, kiolvassa azt, és a keret foglaltságát jelző markert szabadra állítja.
Ha a markert valamiért nem sikerül átállítani, előbb utóbb a keretek megteléséhez vezet. Ezért be kell iktatni egy kitüntetett munkaállomást, amely figyeli a kereteket, és ha egy keret már egyszer körbeért, és még mindig nem távolították el a csatornából, akkor kitörli.

1.5. Regiszterbeszúrásos gyűrű

A gyűrű-topológiák léptető- és tárolóregisztereit használja ki a protokoll. A léptetőregisztert használják arra, hogy a csatornáról a biteket egyesével bemásolja a hoszt a regiszterbe. A regiszter elejére kerül a célhoszt címe. Ha ez nem egyezik meg a másolást végző hoszt címével, egyszerűen eldobja a regiszter tartalmát. Ha a cím a hoszt címe, a regiszterből a biteket átadja a munkaállomásnak feldolgozásra.
Adás esetén a tárolóregisztert tölti fel a munkaállomás a csatornára küldendő bitekkel. A csatornára való másolást azonban nem teheti meg tetszőleges időben, mert ütközést okozhatna. Csak akkor lehet elkezdeni a bitek kiküldését, ha a léptetőregiszter üres, mert akkor a csatorna is üres.

2. Osztott vezérlésű közeghozzáférés

Az ütközéseket úgy küszöbölik ki, hogy egyszerre csak egy állomás adhat. Az adás joga azé az állomásé, amelyik a „tokent” (adásjogot) birtokolja. Az állomás a tokent adás után valamilyen elv alapján továbbadja egy másik hosztnak.

2.1. Vezérjeles gyűrű

Gyűrű-topológia esetén alkalmazható protokoll. Lényege, hogy a legelső adást kezdeményező hoszt előállít egy tokent, melyet adás után továbbad a szomszéd hosztnak. Az vagy adást kezdeményez, vagy adási szándék hiányában egyszerűen továbbadja a tokent. Az elv megengedi a prioritás létrehozását, ebben az esetben a prioritással rendelkező állomásnak nem kell rögtön tovább adnia a tokent, hanem csak meghatározott számú adás után.
Gond lehet, ha a token elvész, vagy leblokkol egy állomásnál. Ennek az elkerülésére felügyelő állomást neveznek ki, amelynek joga van a token újrakiosztására.

2.2. Vezérjeles sín

Ugyanaz az elv, mint előbb, azzal a különbséggel, hogy a fizikai sín topológiát logikai gyűrűvé alakítják.

2.3. Ütközést figyelő, ütközést elkerülő (CSMA/CA)

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance, azaz vivőjel-érzékeléses többszörös hozzáférés, ütközéselkerüléssel.
A CSMA/DC protokoll nagy számú hoszt esetében már nem működik kielégítően. Túlságosan sok hoszt kerül versengési állapotba, és így kénytelen lesz várakozni. A probléma egyik megoldása, hogy szétválasztjuk a teljes hálózatot alhálózatokra. Ekkor viszont az alhálózatok összeköttetése lesz a szűk keresztmetszet.
Sokkal jobb megoldás az ütközés lehetőségének elkerülése. Az ütközés elkerülését úgy oldják meg a CSMA/CA típusú protokollok, hogy az adás befejezése után a hosztok egy logikai listában elfoglalt helyüktől függő ideig várakoznak, és csak utána kezdik meg az adást. (A logikai lista felépítésének több módja is van, melyek részletezésétől most eltekintünk.)

3. Központosított vezérlésű közeghozzáférés

Ebbe a csoportba tartoznak az olyan protokollok, melyek úgy kerülik el az ütközést, hogy egy központi egység szabályozza az állomások csatornahasználatát.

3.1. Lekérdezéses eljárás

Az állomások között létezik egy kitüntetett, master (fő) állomás, ami szabályozza a mellékállomások működését. A főállomás egymás után szólítja fel a mellékállomásokat adásra. Csak az az állomás adhat, amelyiket a főállomás felkért adásra. Ha nincs szüksége a csatornára az állomásnak, akkor ezt tudatja a főállomással, amely a következőt kérheti fel adásra.
Előnye az eljárásnak, hogy rugalmasan változtatható a mellékállomások prioritása. Csak a főállomásnál kell rögzíteni, hogy egy adott mellékállomást hányszor lehet lekérdezni.
Hátránya viszont, hogy a főállomás sérülése a teljes rendszer üzemképtelenségét okozza.

3.2. Vonalkapcsolásos eljárás

Itt is létezik főállomás és mellékállomások. Ám ebben az esetben a főállomás egy kapcsolóközpontként köti össze a kommunikálni kívánó mellékállomásokat.

Összefoglalás

Ebben a fejezetben olyan eljárásokat ismertünk meg, melyek az üzenetszórásos rendszereknél az adatátviteli közeghez való hozzáférést szabályozzák.
Ezeket az eljárásokat három nagy csoportba lehet sorolni:

- véletlen vezérlésű közeghozzáférés,
- osztott vezérlésű közeghozzáférés,
- központosított közeghozzáférés.

Mindegyik eljáráscsoportban több módszer is található a közeg elérésének szabályozására. Ezek közül a legfontosabbak:
A CSMA/CD ütközést figyelő, ütközést jelző eljárás, az egyik legelterjedtebb hálózati szabvány, az Ethernet-hálózatok ezt a módszert alkalmazzák. Ennél az eljárásnál az adók csak akkor adhatnak, ha a csatorna üres. Ha ekkor mégis ütközés keletkezik a perzisztenciától függően várnak valameddig, és újra megpróbálják az adást.
Ezen kívül fontos eljárás még a vezérjeles eljárás, mivel ezt is gyakran alkalmazzák a gyakorlatban. Itt nem a munkaállomás dönti el, hogy mikor ad, hanem egy token (adásjog) jár körbe, és akinél az adásjog van, az kezdeményezhet adást. Így biztos nem keletkezik ütközés.
A többi eljárás ismerete is fontos az átfogó képalkotáshoz, de a fenti két eljárás kiemelkedő jelentősséggel bír a gyakorlati élet szempontjából.

Ellenőrző kérdések

1. Mi az Aloha?
   - Véletlen vezérlésű közeghozzáférés.
   - Osztott közeghozzáférés.
   - Központosított közeghozzáférés.
   - A Hawaii Egyetem neve.

2. Mit jelent a CSMA/CD?
   - Ütközést figyelő, ütközést elkerülő eljárás.
   - Egy Compact Disc-szabvány.
   - Ütközést figyelő, ütközést jelző eljárás.
   - Egyfajta központosított eljárás.

3. Az 1 perzisztens-módszer esetében:
   - Ütközés után az adó véletlen hosszú ideig vár, majd adni kezd.
   - Ütközés után az adó azonnal adni kezd.
   - Ütközés után lehet, hogy visszalép az adó az adási szándékától.

4. A CSMA/CD protokoll szerinti hosztok három állapotot vehetnek fel:
   - versengéses, átviteli, tétlen
   - versengéses, feldolgozási, tétlen
   - adási, átviteli, tétlen

5. A regiszterbeszúrásos módszer a következő regisztereket használja:
   - léptetőregiszter, címregiszter
   - léptetőregiszter, tárolóregiszter
   - adatregiszter, tárolóregiszter
   - adatregiszter, címregiszter

6. A lekérdezéses eljárásnál:
   - Nem lehet prioritást rendelni az állmásokhoz.
   - Az állomás megkérdezi a központot, hogy adhat-e.
   - Több központ is lehetséges.
   - Csak az az állomás adhat, amelyiket a vezérlő megszólított.

IV. Adatkapcsolati réteg és protokolljai

Bevezető

Az eddigiekben az ISO OSI modell fizikai rétegét elemeztük. A modell szerint a fizikai rétegre épül az adatkapcsolati réteg. A következőkben ezt a réteget fogjuk egy kicsit részletesebben megvizsgálni.
Az adatkapcsolati réteg feladata, hogy szolgáltatást nyújtson a hálózati rétegnek. A legfontosabb szolgáltatás az adatok átvitele az adó hoszt hálózati rétegétől a vevő hoszt hálózati rétegéig. Az átvitel során adatkapcsolati protokollokkal kommunikálnak.
Ahhoz, hogy az adatkapcsolati réteg szolgáltatást nyújthasson a hálózati rétegnek, igénybe kell vennie a fizikai réteg szolgáltatásait. A fizikai réteg csupán annyit tesz, hogy a kapott bitsorozatot továbbítja.
Ez a továbbítás egyáltalán nem feltétlenül tökéletes. Lehet, hogy bitek vesznek el, duplázódnak vagy tévesen érkeznek meg. A hibamentességet a fizikai réteg nem garantálja. Az adatkapcsolati réteg feladata, hogy jelezze, illetve lehetőség szerint ki is javítsa a hibát.
A hiba felismerésének, esetlegesen kiküszöbölésének érdekében az adatkapcsolati réteg **keretekre** tördeli a bitfolyamot, és minden kerethez készít egy ellenőrzőösszeget. (Az ellenőrzőösszeg előállításának több módszere is létezik, ezekre később térünk vissza.) Amikor a keret megérkezik a célhoz, a vevő újra kiszámítja az **ellenőrző** összeget. Ha azt tapasztalja, hogy a két érték eltér egymástól, akkor hiba történt az átvitel során, ilyenkor meg kell ismételni az átvitelt, vagy ha tudja, kijavítja a keretet.

A keretek képzése több módszer szerint történhet. E módszerek két nagy csoportra bonthatók. Az egyik módszer a **karakterorientált** átvitel, ez esetben mindig ugyanannyi számú bit szerepel a keretben. Ez a módszer azonban főleg szöveges jellegű adatok átvitele esetén használható hatékonyan, manapság pedig egyre több nem szöveges információt kell továbbítani.
Ezért a mai viszonyok között inkább a **bitorientált** átvitelt szokták megvalósítani.

1. Keretek képzése

Nézzünk most egy pár eljárást a bitfolyam keretekre való tördelésére.

1.1. Karakterszámlálásos módszer

A módszer lényege, hogy a keret fejlécében megadják, hány karakter van a keretben. A keret megérkezésekor a célállomás tudja, hogy hány karakter fog következni, tehát pontosan tudja, hol a keret vége, és mekkora a keret hossza. Nincs megszabva, hogy a kereteknek mennyi karaktert kell tartalmazniuk, ez mindig a legelső karakterből derül ki a vevő számára.

5	1	2	3	4	**8**	1	2	3	4	5	6	7	**5**	1	2	3	4
1. keret 5 karakter					2. keret 8 karakter								3. keret 5 karakter				

Ezzel az eljárással az a probléma, hogy átviteli hiba esetén módosulhat a keret hosszát jelző karakter értéke is. Pl. a 2. keret 1. karaktere 8 helyett 7-re módosul. Ekkor azt hinné a vevő, hogy csak 7 karakter következik, és innét kezdve az összes keret hibás lenne.

1.2. Kezdő- és végkarakterek alkalmazása karakter-beszúrással

Ez az eljárás hatékonyabb, mint az előző, mert itt már van mód az újraszinkronizálásra.
E módszernél egy speciális karaktersorozat vezeti be a keretet, és egy másik jelsorozat zárja le. Ez a jelsorozat a DLE STX és a DLE ETX. Tehát minden keret a DLE STX (DLE: Data Link Escape – adatkapcsolati ESC karakter, STX: Start of TeXt – szöveg kezdete) jellel kezdődik, majd tetszőleges számú adatkarakter, és az adat végén DLE ETX (ETX End of TeXt – szöveg vége) karakterekkel záródik.
A módszer problémája lehet bináris típusú adatok átvitelénél, hogy a kezdő- és vég karakterek bitmintája előfordulhat a keret adat-részében. A probléma kiküszöbölhető, ha az adó adatkapcsolati rétege minden véletlen előforduló DLE ASCII karaktersorozat elé egy plusz DLE karaktersort szúr be, melyet a vevő automatikusan eltávolít. Ezt a módszert karakterbeszúrásnak nevezik.

A fő hátránya ennek a keretezési módszernek, hogy nagyon kötődik az ASCII karakterkódoláshoz.

1.3. Kezdő- és végjelek alkalmazása bitbeszúrással

Az előző metódus hátránya miatt új módszert kellett kidolgozni, amely megengedi tetszőleges méretű karakterek használatát.
Ez az eljárás lehetővé teszi, hogy tetszőleges számú bit legyen egy keretben. A módszer működési elve:
Minden keret egy speciális bitsorozattal kezdődik és fejeződik be. Ez a speciális minta két 0 között hat 1-es. Hogy a minta ne ismétlődhessen meg az adás folyamán, az adó minden öt egymást követő 1-es után automatikusan beszúr egy 0-t, melyet a vevő szintén automatikusan eltávolít. Ez a bitbeszúrás módszere.
Például:

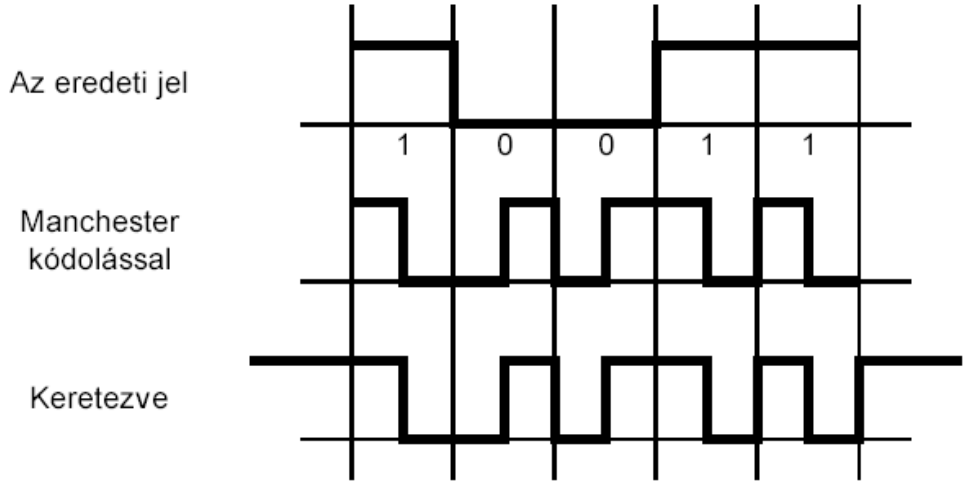
A hálózati réteg által küldött adat: 11111↓11111↓011111↓0100110

Az adatkapcsolati réteg keretezése: 01111110111110111110111110100110**0**01111110

A fenti példában látszik, hogy sok egymást követő 1-es volt eredetileg az adatban. A ↓ jelöli az ötödik 1-es utáni bitbeszúrás helyét. Az is látható, hogy amennyiben az ötödik 1-es után eredetileg is 0 állt, akkor is elvégzi az adó a bitbeszúrást. Nagyobb betűk jelzik a keretet és a beszúrt biteket, melyeket a vevő majd el fog távolítani, és így visszakapja az eredeti adatot.

1.4. Keretezés a fizikai rétegben nem használt állapottal

Ezt a módszert olyan hálózatok esetén alkalmazzák, ahol a fizikai réteg kódolása redundanciát tartalmaz, azaz van még fel nem használt jel. Például Manchester-kódolást használva minden jelnek van jelváltozása (tehát minden jel vagy pozitív, vagy negatív irányba változik). Ez történhet magas szintről alacsony szintre, vagy alacsony szintről magas szintre: ebben az esetben kezdő jelnek alkalmazható a két cikluson át tartó magas szintű jel. Biztos, hogy más jel nem fogja felhasználni ugyanezt a kombinációt.



39. ábra
Keretezés nem használt állapottal

A biztonság növelése érdekében sok adatkapcsolati protokoll karakterszámlálós és másik eljárást együttesen alkalmaz.

2. Hibakezelés

Az előzőekben tárgyalt keretezés csupán arról biztosítja az adót és a vevőt, hogy az adat eljutott a címzettig. A belső tartalom helyessége még egyáltalán nem biztos, ezért a keretezés után meg kell vizsgálni a hibázás lehetőségeit is.

Attól függően, hogy hány bit változott meg vagy veszett el a hiba során, beszélhetünk **egyedi,** illetve **csoportos bithibáról**.
A bithiba azt jelenti, hogy csupán egy-egy bit tér el az eredetihez képeset. A csoportos bithiba esetén több, egymást követő bit is meghibásodik.

A hibák kezelésére két, alapjaiban eltérő módszert dolgoztak ki. Az egyik módszer szerint a kóddal együtt még annyi információt küldünk a vevőnek, hogy az egyértelműen ki tudja következtetni, hol volt a hiba, és mi volt az eredeti jel, tehát a hiba ki is javítható. Erre a célra **hibajavító kódokat** használnak. A másik módszer szerint csak annyi plusz információt küldenek a kóddal, hogy eldönthető legyen a hiba ténye. Ebben az esetben a vevő képtelen kijavítani a kódot, viszont sokkal rövidebb redundáns részt kell mellékelni. Ez a módszer a **hibajelző kód** alkalmazása. Ilyen kódolás használatakor a hibás kódot újra kell adni.

Tehát egy keretnyi adat, amit az adónak továbbítania kell, két részből tevődik össze. Először az adatbiteket kell kódolni, majd az adatbitek ellenőrzésére szolgáló biteket kell kódolni. Ezt az egységet gyakran szokták **kódszónak** is nevezni.
Könnyen kiszámolható, hogy az eredeti kódszó hány bitben különbözik a kapott kódszótól, azaz hány bit hibás. Egyszerűen a két kódszó közt kizáró vagy műveletet kell végezni, és meg kell számolni, hogy hány db 1-est kaptunk. Pl.:

az eredeti kód:	10100101
a kapott kód:	11110101
XOR után:	01010000

tehát 2 bit hibás!

Ez alapján **Hamming-távolság**nak nevezzük azt, hogy egy kódszó hány egybites hibával alakítható át egy másik kódszóvá.

Nézzünk most egy-egy példát a hibajelző és a hibajavító kódokra!
Klasszikus hibajelző kódolási módszer a **paritásbit** használata. A módszer lényege, hogy az adó megszámolja, hogy a küldendő kódban hány darab 1-es van. A kapott szám párosságának megfelelően egy 1-sel vagy egy 0-val egészíti ki a kódot. Megállapodás szerint 1-est ír, ha páros darabszámú 1-es volt a kódban, és 0-t, ha páratlan darabszámú 1-es volt a kódban. Pl. az eredeti kód: 10110101, ebben 5 db 1-es van, ezért 0-val kell kiegészíteni: 101101010. A módszer előnye, hogy nagyon egyszerű, és a hibajelző bit kevés helyet foglal az átvitel során. Hátránya viszont, hogy csak páratlan számú hiba esetén jelez, és nem lehet tudni a hiba pontos helyét.

A másik nagyon gyakran használt módszer a **CRC** (Cyclic Redundancy Check – ciklikus redundancia-ellenőrzés) nevű kódolási eljárás. Ez a módszer már a hibajavító kódolások közzé tartozik. Itt sokkal hosszabb az ellenőrző rész, de ennek fejében csoportos hibát is tud jelezni, és vissza is állítható az eredeti kód. A módszer lényege, hogy az eredeti kódot egy előre meghatározott (speciális feltételeket kielégítő) bitsorozattal (polinommal) elosztják, és a maradékot a kóddal együtt továbbítják. A vevőnek is ismernie kell az osztó bitmintát, így ő is el tudja végezni az osztást. Ha a két eredményt összehasonlítja, meg tudja határozni, hol volt hiba az átvitel során. Mivel az osztó polinom nem lehet tetszőleges, három polinom vált nemzetközi szabvánnyá: CRC-12: 6 bites karakterek átvitelénél alkalmazzák, CRC-16 és CRC-CCITT: 8 bites karakterek átvitelénél alkalmazott polinom.

3. Elemi adatkapcsolati protokollok

Az adatkapcsolati réteg a hálózati réteg által a réteginterfészen keresztül eljuttatott csomagokat továbbítja a vevő hoszt hálózati rétegének. Az átvitel során az adatkapcsolati rétegnek a kapott adatokat egyformán kell továbbítania, függetlenül attól, hogy az adat egy részét a hálózati réteg fejrész-információnak kezeli.
Az adatkapcsolati réteg a hálózati rétegtől kapott csomagokat keretekre tördeli, ellátja fejrésszel és ellenőrző résszel, majd a fizikai réteg segítségével továbbítja a másik hoszt adatkapcsolati rétegének. A vevő oldalán az adatkapcsolati réteg ismét előállítja a keretet.
Csakhogy egyáltalán nem biztos, hogy a keret megérkezik, vagy hibamentes. A legtöbb protokollban megbízhatatlan csatornát tételezünk fel, amelynél megvan annak veszélye, hogy egész keretek vesznek el.

A fentiek figyelembevételével vizsgáljunk meg most néhány adatkapcsolati protokollt, a közel ideális helyzetet feltételező legegyszerűbből kezdve.

3.1. Korlátozás nélküli, szimplex protokoll

Ennél a protokollnál feltételezzük, hogy csak egy irányba történik adás, illetve hogy az adó és a vevő hálózati rétegei mindig készen állnak keretek küldésére és fogadására. A keretek feldolgozásának idejétől eltekintünk, feltételezzük, hogy végtelen méretű puffer áll rendelkezésre. Feltételezzük továbbá, hogy az adás során nem vesznek el keretek, és minden keret hibamentesen megérkezik.
A protokoll algoritmusa is nagyon egyszerű:

- Az adó adatkapcsolati rétege elkészíti az adatkereteket, és a fizikai rétegnek továbbítja.
- A vevő adatkapcsolati rétege fogadja a kereteket, feldolgozza, és a hálózati rétegének továbbítja.

3.2. Egyirányú „megáll és vár” protokoll

Ennél a protokollnál már nem tételezzük fel azt, hogy a vevő ugyanolyan sebességgel (vagy gyorsabban) tud dolgozni, mint az adó, ugyanakkor még mindig feltételezzük, hogy az adás hibamentes.
A fő probléma ennél a protokollnál, hogy meg kell akadályozni az adót abban, hogy gyorsabban adja a kereteket, mint ahogy a vevő fel tudná dolgozni azokat. Több lehetséges megoldás közül az a legcélravezetőbb, ha a vevő visszajelzést ad az adónak a keret feldolgozásáról, így csak a visszajelzés megérkezésekor küldheti a következő keretet az adó.

A protokoll algoritmusa:

Adó esetében:

Eljárás eleje.
Keret előállítása.
Keret elküldése.
Ciklus kezdete.
Érkezett nyugta az előző keretre?
„Igen” esetben kilépés a ciklusból.
„Nem” esetben visszalépés a ciklus elejére.
Ciklus vége.
Ha van még küldendő adat visszalépés az eljárás elejére.
Eljárás vége.

Vevő esetében:

Eljárás eleje.
Érkezett keret?
„Nem” esetben várakozás, eljárás elejére lépés.
„Igen” esetben keret feldolgozása, nyugta küldése, eljárás elejére lépés.
Eljárás vége.

Tehát az adó minden keret elküldése után megáll és vár, míg visszajelzést nem kap a keret feldolgozásáról.
Bár ebben az esetben is egyszerre csak egy irányba folyik adatáramlás, a vevőnek választ kell küldenie, tehát az adó felé is történik adatmozgás. Következésképpen a csatornának alkalmasnak kell lennie kétirányú adatforgalom lebonyolítására. Mivel egyszerre csak az egyik irányba történik adatáramlás, a protokoll megvalósításához félduplex csatorna az ideális.

3.3. Egyirányú összetett protokoll

Mi történik akkor, ha a hibátlan adatátvitel feltételétől is eltekintünk (ahogy az a valóságban is előfordul)?
Ekkor a keretek megsérülhetnek vagy elveszhetnek. Ha a keret adásakor történt hiba, a nyugta visszaküldhető, és a hibás keret megismétlését kell kérni. Ha a keret elvész, az adó nem kap nyugtát, és rövid várakozás után újra küldi a keretet. Tehát a keret adása már biztosan megtörténik. A vevőtől az adó felé történő visszaigazolásnál viszont problémák adódnak. Ha a nyugta vész el, akkor nem tudhatja az adó, hogy megérkezett-e a keret. Feltételezheti, hogy nem jutott el a vevőig a keret, ezért nem kapott nyugtát. Ekkor egy kis várakozás után megismétli az adó a keret elküldését. Ez pedig hiba, mert a keret így két példányban kerül feldolgozásra a vevőnél. Ezt a jelenséget **keretduplázódásnak** nevezzük.
Tehát meg kell oldani, hogy a vevő meg tudja különböztetni az első alkalommal küldött keretet, az ismételten küldött kerettől. Ennek egyszerű megoldása, ha az adó egy számot helyez a keret fejrészébe, amiből kiderül, hogy ezt a keretet most küldi először, vagy már ismételten küldi.

A fentieknek megfelelően az algoritmus is bonyolultabb lesz:

Adó esetében:

Eljárás eleje.
Keret előállítása.
Nyugtabit 0-ra állítása.
Keret elküldése.
Ciklus kezdete.
Érkezett nyugta az előző keretre?
„Igen” esetben kilépés a ciklusból.
„Nem” esetben:
Ha nem járt le a várakozási idő, akkor visszalépés a ciklus elejére.
Ha lejárt a várakozási idő, akkor a nyugtabit 1-re állítása, keret újraküldése, ciklus elejére lépés.
Ciklus vége.
Ha van még küldendő adat, visszalépés az eljárás elejére.
Eljárás vége.

Vevő esetében:

Eljárás eleje.
Érkezett keret?
„Nem” esetben várakozás, eljárás elejére lépés.
„Igen” esetben:
Nyugtabit 0?
„Igen” esetben keret feldolgozása, nyugta küldése, eljárás elejére lépés.
„Nem” esetben keret eldobása, nyugta küldése, eljárás elejére lépés.
Eljárás vége.

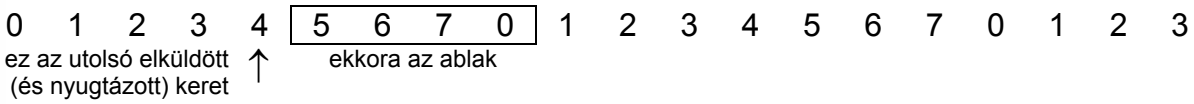
4. Kétirányú protokollok

Az eddigi protokolloknál csak egy irányba haladtak az adatkeretek, visszafele csupán nyugtákat küldtünk. A valóságban azonban a keretek küldése mindkét irányba szükséges. Az eddigi technikákkal ez úgy valósítható meg, hogy két kommunikációs csatornát építünk ki, és az egyiken az adás megy, a másikon a nyugták. Ez is elképzelhető, de ehhez nem érdemes kiépíteni két csatornát. A most következő protokollok egy csatorna segítségével tudnak megvalósítani kétirányú kommunikációt.
Az eljárások lényege, hogy összekapcsolják az ellenirányú nyugtát és az adást. Pontosabban, az adást követő nyugtát ráültetik a másik irányba tartó adásra. Ha belátható időn belül nincs visszafelé irányuló adás, akkor a vevőnek külön kell küldenie a nyugtát.
A ráültetés alkalmazásának fő előnye a különálló nyugtakeretekhez képest a csatorna rendelkezésre álló sávszélességének jobb kihasználása.

4.1. Csúszóablakos protokoll

A csatorna még jobb kihasználását teszi lehetővé, ha megengedjük, hogy egyszerre több nyugtázatlan keret is a csatornán legyen. A protokollban minden elküldhető keret kap egy sorszámot (a sorszám 0 és maximális érték közti). Az adó egyszerre több keretet is elküld. Ezeknek a kereteknek a sorszámait megjegyzi, úgy szoktuk mondani, hogy ezeket a kereteket az adási ablakba teszi. Az ablak mérete a protokolltól függ, van olyan csúszóablakos protokoll, ahol állandó az ablak mérete (azaz mindig azonos a kiküldött keretek száma), de létezik olyan protokoll is, ahol az ablak mérete változhat. Ha egy keretre megérkezik a nyugta, az ablak feljebb csúszhat, így újabb keret kerül a csatornára. Ráadásul nem szükséges minden keretet egyesével nyugtázni. Ha egy keretre megérkezik a nyugta, az azt jelenti, hogy az összes előtte lévő keret is megérkezett a vevőhöz. Tehát egyszerre több keretnyit is léphet feljebb az ablak.
A vevő egy vételi ablakot tart fenn, amely az elfogadható keretek sorszámait tartalmazza. Ha véletlenül olyan keret érkezik, melynek a sorszáma az ablakon kívül esik, azt eldobja. Egy beérkezett keretet a vevő akkor nyugtázhat, ha a keret még nem került nyugtázásra, és az össze előtte lévő keret is beérkezett.

Magyarázatképpen nézzünk erre egy példát:
Az adó a kereteket 0-tól 7-ig sorszámozza.



Tehát 4 ablakos keretünk van, amiben most a 5. 6. 7. 0. sorszámú keretek vannak a csatornán. Tegyük fel, hogy a vevőhöz beérkezett az 5. és a 6. sorszámú, szinte egy időben, és még egyikre se adott nyugtát. Ekkor a vevő nyugtát küld a 6. keretről. Ezt az adó úgy értelmezi, hogy az 5-re és a 6-ra is érvényes, ezért feljebb csúsztatja a keretet, és kiküldi az 1. és a 2. kereteket is.

4.2. Az n visszalépést alkalmazó protokoll

Az eddigiekben azzal a hallgatólagos feltételezéssel éltünk, hogy a keret megérkezése a vevőhöz és a nyugta visszaérkezése elhanyagolható időt vesz igénybe. Egy lokális hálózatnál ez így is van, azonban pl. egy műholdas hálózatnál ez az idő már jelentős lehet. Az előző módszer már egy kicsit segít a dolgon, mert egyszerre „ablaknyi” keret lehet a csatornán, de nem mellékes az ablak mérete. Úgy kell megválasztani az ablak méretét, hogy folyamatosan legyenek várakozó, vevő felé haladó, és adó felé haladó keretek, nyugták is a csatornán, tehát folyamatos legyen a „körbeforgás”. Ezt a technikát **csővezetékezésnek** (pipelining) nevezik, utalva arra, hogy a keretek mintegy csőben haladnának egymás után.

Ez a technika azonban komoly veszélyt rejt magában megbízhatatlan csatorna esetében. Mi történjen akkor, ha egy hosszú adatfolyam közepén egy keret menet közben elvész, vagy meghibásodik? Mikorra az adó tudomást szerez a hibáról, már rengeteg keretet kiküldött. A vevő természetesen el fogja dobni a hibás keretet, de mi történjen a hibás keret után érkezett jó keretekkel? Fontos szempont, hogy az adatkapcsolati rétegtől a hálózati réteg mindenképpen helyes sorrendben várja a kereteket.

Két lehetséges megoldás közül az egyik a **„visszalépés n-nel”** (go back n) eljárás.
Ekkor a vevő eldobja az összes keretet, amely a hibás keret után érkezett. Ennek az lesz az eredménye, hogy a csővezeték kiürül, az adónak lejár az időzítése, és megismétli az adást. Remélhetőleg ekkor már hiba nélkül megérkeznek a keretek, és újra beindulhat a folyamat, feltöltődhet a csővezeték. Látható, hogy nagy veszteséget okoz az átviteli sebességre nézve.

4.3. Szelektív ismétlő protokoll

A másik megoldás a hibás keret kezelésére a **szelektív ismétlés** (selective repeat).
Ennek az a lényege, hogy a vevő hiba esetén nem dobja el az összes keretet, hanem egy pufferbe menti, és a hibás keret ismétlését kéri az adótól. Ha a hibás keret újraküldése megérkezett, a pufferből áttölti a többi jó keretet, így megőrizte a kerek sorrendhelyességét. Ekkor a legnagyobb sorszámú keretet, ami a pufferben volt, nyugtázza, így az adó egyszerre több keretet léphet feljebb, és nem ürül ki a csővezeték. Hátránya a módszernek, hogy nagy pufferrel kell rendelkeznie az adatkapcsolati rétegnek.

Összefoglalás

E rétegben történik a csomagok keretekké történő tördelése. A hibátlan átvitel érdekében a rétegben a keretekhez ellenőrző összeget csatolnak. A fejezetben először megismerhettük a keretek képzésének módszereit, majd a keretek átvitelekor keletkezett hibák kezelésének módozatai foglaltuk össze. Végül a keretek célba juttatásának módszereit vizsgáltuk meg a legegyszerűbb, ideális esetet feltételező módszertől a valóságban használt kétirányú protokollokig.

Ellenőrző kérdések

1) Hogyan működik a karakterszámlálásos módszer?
   - Átvitel előtt közli az összes karakter számát.
   - Átvitel végén tudatja, hány karaktert küldött.
   - Keretenként megadja az átvitt karakterek számát.

2) Mi a bitbeszúrás?
   - 5 egymást követő 1-es után automatikusan beszúr egy 0-t.
   - Minden karakter után beszúr egy speciális bitmintát.
   - A keret elejére és végére egy speciális bitmintát szúr be.

3) Mi a Hamming-távolság?
   - A csoportos hiba mérete.
   - Két kódszó hány egybites hibával különbözik.
   - Egyfajta hibajelző eljárás.

4) A csúszóablakos protokoll egyszerre hány keretet küld a csatornára?
   - Amennyit az adási és vételi ablaka megenged. (Tetszőleges számú, de előre definiált.)
   - Ahányat az adási és vételi ablaka megenged. (Tetszőleges számú, előre nem definiált.)
   - 5-öt.
   - 7-et.

5) Mi történhet hibás keret érkezésekor?
   - A vevő csak a hibásat ismételteti.
   - A vevő az összes hibás utánit eldobja, és újra kéri.
   - A vevő vagy minden hibás utánit megismételtet, vagy csak a hibásat ismételteti (protokollfüggő).

V. Hálózati réteg

Bevezető

A hálózati réteg feladata, hogy a csomagokat eljuttassa a forrástól a célig. A cél eléréséig a csomagoknak általában több csomóponton is át kell jutniuk. E tevékenység már jóval szélesebb körű, mint az adatkapcsolati réteg feladata, mivel az csupán kereteket továbbít az egyik hoszttól a másikig (pont–pont kapcsolat). Ebből következően a hálózati réteg a legalacsonyabb réteg, amely két végpont közötti átvitellel foglalkozik (több csomópont két végén lévő hosztokat kapcsol össze).
Mivel több csomópontot is érint a kapcsolat létrehozása, a hálózati réteg tanulmányozásához figyelembe kell venni a hálózat topológiáját, és megfelelő utat kell találni. Ügyelni kell arra, hogy a hálózatban ne okozzunk túlterhelést egy–egy kommunikációs vonalon, vagy ne legyen túlságosan kihasználatlan a hálózatnak egy bizonyos része. Végül a hálózati réteg feladata az is, hogy két különböző hálózatban lévő hoszt összeköttetését megoldja.
Ebben a fejezetben ezeket a kérdéseket fogjuk részletezni.

1. Forgalomirányítás

Az alhálózatok megszervezésének két fajtája lehetséges. Az egyik összeköttetéseket használ a hálózat létrehozására, a másik összeköttetések nélkül működik. Az összeköttetés alapú hálózatoknál az egyes hosztok **virtuális áramkörök** segítségével kapcsolódnak egymáshoz. Az összeköttetés-mentes hálózatban pedig az adatátvitel ún. csomagok, **datagramok** segítségével történik.

- Virtuális áramkörök használatakor az adó és a vevő összeköttetése úgy jön létre, hogy a kapcsolathoz szükséges hosztok összekapcsolódnak az adat átvitelének idejére. Ebből következően nem kell minden csomagra forgalomirányítási döntést hozni, azaz minden csomag ugyanazon az útvonalon halad. A megvalósításához a hálózat csomópontjainak rendelkezniük kell egy–egy táblázattal, amely tartalmazza a lehetséges útvonalakat. Tehát minden virtuális áramkörre fel van jegyezve a kiinduló és a célhoszt, illetve az áramkört azonosító szám. Így egy csomag célba juttatása úgy történik, hogy csomópontonként kiolvassa a táblázatból, hogy az azonosítójának ismerete mellett merre kell tovább haladnia.
  Erre a módszerre talán a telefonhálózat a legszemléletesebb példa.
- Ezzel szemben a datagram típusú alhálózatokban nincs előre meghatározott útvonal. Minden egyes csomag a hálózat aktuális állapotától függően halad csomópontról csomópontra. Így gyakran előfordul az is, hogy az egymást követő csomagok más útvonalon érkeznek meg a célba. Ebből következően az sem biztos, hogy a csomagok sorrendhelyesen fognak megérkezni. Mindezeket csak úgy lehet megvalósítani, ha a csomagok tartalmazzák a forrás- és a célhoszt címét, és a sorban elfoglalt helyüket (sorszám).

Mindkét megoldásnak vannak előnyei és hátrányai is. Mindig a feladat alapján kell mérlegelni, hogy melyik módszer a kedvezőbb.

1.1. Forgalomirányítási algoritmusok

A hálózati réteg fő feladata, hogy a csomagokat a forráshoszttól a célhosztig irányítsa. A forgalomirányítási algoritmus (routing algorythm) a hálózati réteg szoftverének az a része, amelyik meghatározza, hogy egy bejövő csomag melyik kimeneten menjen tovább a célhoszt felé. Ezt a döntést igyekszik úgy meghozni, hogy a legjobb úton juttassa el a csomagot, azaz azon az úton, amelynek a megtétele a legkevesebb időt veszi igénybe. Az utak viszont változó leterheltségűek. Ezért minden csomópontnak, minden datagramra újra és újra meg kell határozni a megfelelő utat. Az útvonal meghatározásában fontos szerepet játszik az ún. routing tábla, ami a csomóponttal kapcsolatban álló másik csomópontok adatait tartalmazza (pl. távolság).

A forgalomirányító algoritmusok két nagy csoportba sorolhatók: determinisztikus és adaptív forgalomirányítás:

Determinisztikus (előre meghatározott) forgalomirányítás: a csomópontok úgy hoznak útválasztási döntéseket, hogy nem veszik figyelembe a hálózat pillanatnyi leterheltségét. A hálózat építésekor meghatározzák a routolási táblákat, és azokat nem változtatják. Előnye, hogy a csomópontoknak nem kell útvonalszámító algoritmusokat használni, de hátránya, hogy a csomópontok nem képesek alkalmazkodni az útvonalak változó leterheléséhez.

Adaptív (alkalmazkodó) forgalomirányítás: a csomópontok úgy hoznak forgalomirányítási döntést, hogy figyelembe veszik a hálózati forgalom változását. A csomópontok valamilyen módszer alapján figyelik a hálózat terhelését, és ezt figyelembe veszik a csomagok továbbítási irányának meghatározásakor.
Az adaptív forgalomirányítási eljárásnak három típusát lehet megkülönböztetni:

- **Elszigetelt adaptív forgalomirányítás:** lényege, hogy a csomópontok ugyan döntéseket hoznak, de ezek a döntések csak a csomópont információira támaszkodnak, azaz a csomópontok csak azt vizsgálják, hogy a tőlük induló vonalak mennyire vannak leterhelve. Ennek a módszernek az lehet a hátránya, hogy egy olyan útvonalra küldi el a csomagot, amelynek a csomópont és a következő csomópont közti szakasza nagyon kicsi leterheltségű, de a következő csomópont után nagyon nagy leterheltség következik. Így sokkal rosszabb eredményt ér el, mintha folyamatosan közepes terhelésű útvonalra küldené a csomagot.

- **Elosztott adaptív forgalomirányítás:** lényege, hogy a csomópontok a belőlük kiinduló útvonalszakaszok leterheltségének adatait bizonyos időközönként kicserélik egymással. Előnye, hogy a csomópontok így tisztában lesznek a teljes útvonalak terheléseivel, és optimalizálni tudják a csomagok útját. Hátránya viszont, hogy jól kell meghatározni az információcserék időzítését, mert túl gyakori információcsere esetén a hálózatot a routolási táblák küldözgetése foglalja le, túl ritka küldés esetén pedig a csomópontok nem kapnak értesítést a változásokról.
- **Központosított adaptív forgalomirányítás:** lényege, hogy a csomópontok bizonyos időközönként megküldik a központnak a helyi terhelésüket. A központ összegzi ezeket, és kiszámolja az ideális útválasztást, melyet visszaküld a csomópontoknak. A módszer előnye, hogy a központ határozza meg az optimális útválasztást. Hátránya viszont az időzítés problémája (mint előbb), illetve, hogy az információ központhoz, majd vissza történő utaztatása nagyon sok időt vesz el, így nem aktuálisak a csomóponti útválasztások.

Az algoritmusok elvi hátterének megvizsgálása után nézzünk néhány példát az algoritmusok konkrét megvalósítására!

Determinisztikus forgalomirányítási algoritmusok:

a) **Véletlen forgalomirányító eljárás – „bolyongás”:** az eljárás során a csomópont egy véletlen szám segítségével határozza meg, hogy melyik kimenő vonalára küldje a csomagot. Így a csomagok véletlenszerűen haladnak csomópontról csomópontra, hátha egyszer „belebotlanak” a címzett csomópontba. A túl régóta bolyongó csomópontok eltávolítása céljából a csomag mellé rendelnek egy számot, amely az átlépett csomópontok számát tartalmazza. Ha ez a szám egy előre meghatározott értéken túlhalad, akkor a csomagot eltávolítják. Ez az eljárás nem mondható túl hatékonynak.

b) **Elárasztásos forgalomirányító eljárás:** lényege, hogy a csomópontok a biztonság kedvéért minden lehetséges kimeneti vonalukra elküldik a csomagot. Így biztos, hogy a címzetthez is elér a csomag, csakhogy ezzel hatalmas felesleges forgalmat generálnak.

Adaptív forgalomirányítási algoritmusok:

a) **Elszigetelt adaptív** forgalomirányítási módszerek:

- **„forró krumpli”:** ennél a módszernél a csomópont a csomagot arra a kimeneti vonalra küldi tovább, ahol a legkevesebb csomag vár a továbbításra. Így minél gyorsabban meg akar szabadulni a csomagtól, nem érdekli, hogy a vonal mennyire jó. A folyamat olyan, mintha egy embernek forró krumplit adnánk a kezébe, és az minél gyorsabban igyekezne megszabadulni tőle.

- **fordított tanulás módszere:** ennél a módszernél minden csomópont elindít egy csomagot, amelyben nyilvántartják, hogy honnan indult és hány csomóponton lépett már át. Ezeknek a csomagoknak a segítségével a csomópontok megtudhatják, hogy ha nekik kellene küldeni egy csomagot a másik csomópont számára, hány átlépéssel jutnának el a célig. Ha egy másik irányból érkezett csomagból az derül ki, hogy kedvezőbb átlépési számmal lehet elérni a küldőt, akkor az előző útvonalat eldobja, és ezt jegyzi meg.

b) **Elosztott adaptív** forgalomirányítás:

A gyakorlatban ezt a módszert szokták alkalmazni a hálózatokban.

- **Távolságvektor-alapú** forgalomirányítás: minden csomópont rendelkezik egy routolási táblával, amelyből megtudhatja, hogy a célállomásig mekkora a legrövidebb távolság, és megismerheti a célhoz vezető út azonosítóját. Ezeket a táblázatokat a csomópontok rendszeresen újraszámolják, és megküldik egymásnak. A táblák szétküldésének két módszere lehetséges: vagy szinkron módon bizonyos időközönként szétküldik a táblákat, vagy aszinkron módon csak akkor kezdeményeznek táblacserét, ha változás állt be a vonalak terhelésében.
- **Kapcsolatállapot-alapú** forgalomirányítás: az előző algoritmust az ARPANET-ben használták 1979-ig. Ekkor azonban felváltotta a kapcsolatállapot-alapú irányítás, mivel a távolságvektor-alapúnál nem vették figyelembe a vonalak eltérő sávszélességét, illetve az előző módszer túl lassúnak bizonyult. Manapság ennek a módszernek a különféle változatait alkalmazzák.

A módszer alapja igen egyszerű, 5 lépésből áll. Minden routernek a következőket kell tenni:

1. Felkutatni a szomszédait, és megtudni a hálózati címüket.
2. Megmérni a késleltetést vagy a költséget minden szomszédjáig.
3. Összeállítani egy csomagot, amely az előző adatokat tartalmazza.
4. Elküldeni ezt a csomagot az összes többi routernek.
5. A kapott adatok alapján kiszámítani az összes routerhez vezető legrövidebb utat.

c) Központosított adaptív forgalomirányítás:

Hierarchikus forgalomirányítás: A hálózatok növekedésével a routerek táblázatai is növekednek. Előbb-utóbb elérik azt a méretet, amikor már nem lehet őket kezelni. Ezért a nagy hálózatokat tartományokra, régiókra kell bontani. Régiókon belül minden forgalom hagyományosan történik, de ha a régióból kifele irányuló adatforgalom van, akkor a régió központi routeréhez kell irányítani a csomagot, ahonnan egy másik régió központi routeréhez kerül a csomag. Ez a módszer nem teljesen központosított forgalomirányítás, de a központi routerek szerepe itt sokkal nagyobb, mint az előző módszerekben.

2. Torlódásvédelem

A hálózatokban csak ideális esetben egyezik meg az a sebesség, ahogy az adó, vagy a csomópontok elküldik a csomagokat azzal a sebességgel, ahogyan a vevő vagy a következő csomópont fogadni tudja a csomagokat. A valóságban nagyon gyakori az olyan eset, mikor a csomagok valamiért nem tudnak lekerülni a vonalról, és ezt a tényt az adó nem ismeri fel, hanem folytatja a további csomagok kiküldését. Ekkor egyre több feldolgozatlan csomag halmozódik fel. Ezt az eseményt szoktuk **torlódás**nak nevezni.

A torlódást sok tényező okozhatja:

- hirtelen sok csomag érkezik több bemeneten, melyek ugyanazt a kimenetet szeretnék használni,
- lassú a csomagok feldolgozása, nem képes követni az adás sebességét,
- kevés a routerek puffere.

A torlódásnak egy speciális esete, mikor valamilyen hiba miatt (elsősorban tervezési hiba) véglegesen leállnak a hálózatok. Ez a szélsőséges esetet **befulladás**nak (lock-up) nevezik.

A torlódás másik, még súlyosabb esete a **holtpont**. Ebben az esetben két csomópont egymásra vár. Ilyenkor az egyik nem tud kilépni a holtpontról, mert vár a másikra, de a másik is vár az elsőre. Ebből a helyzetből nincs kiút, csak a hálózat (routerek) újraindításával.

A torlódás lehet helyi jellegű, mikor a hálózatnak csak egy alhálózatát érinti, de lehet egész hálózatot érintő is. Mindenképpen védekezni kell a torlódások ellen. A védekezés egyik módja, hogy megpróbáljuk elkerülni a torlódások kialakulását. A másik módszer, ha már nem sikerült elkerülni a torlódást, legalább próbáljuk meg elhárítani a problémát, és megelőzni a teljes hálózat leállását.
A torlódásvezérléssel azért kell a forgalomirányítás témakörénél foglalkozni, mert gyakran a torlódások kiküszöbölése forgalomirányítási beavatkozást igényel.

2.1. Torlódásmegelőző módszerek

A torlódások kialakulását próbálják elkerülni úgy, hogy az adatkapcsolati, hálózati és szállítói rétegeknél olyan elveket vezetnek be, amelyek lehetetlenné teszik a torlódás kialakulását. Befolyásolja a torlódás kialakulását ezekben a rétegekben az, hogy:

- az adónak milyen gyorsan jár le az időzítése, mikor kezd újra adni,
- a visszalépés n-nel vagy a szelektív ismétlés technikáját alkalmazza,
- azonnal nyugtáz-e, vagy megvár több keretet, és egyben nyugtázza,
- mekkora a csúszóablak mérete,
- mennyi ideig „élhetnek” a csomagok,
- megfelelő-e forgalomirányítási algoritmusuk.

2.2. Torlódásvédelmi algoritmusok

- **Belépés-ellenőrzéssel**: abban az esetben, ha már fellépett a torlódás, megoldást jelenthet a helyzet további romlásának megakadályozására, hogy letiltjuk a kapcsolatteremtést addig, míg a torlódás meg nem szűnik.
- **Pufferek foglalása**: a csomópontok azokat a csomagokat, amelyeket nem tudnak feldolgozni, átmenetileg egy pufferben tárolják. A csomópontok csak akkor nyugtázzák a csomag továbbítását, ha van szabad puffer, tehát nem fog kialakulni torlódás. A nyugta egyben jelzi azt is, hogy a csomópont kész a következő csomag feldolgozására.
- **Csomageldobásos módszer**: a csomópont rendelkezik egy bizonyos méretű pufferrel. Ha nem tudja olyan gyorsan feldolgozni a csomagokat, mint ahogy azok érkeznek, akkor a pufferben gyűjti addig, amíg meg nem telik. Ha a puffer megtelt, a további csomagokat egyszerűen eldobja, azaz nem vesz róluk tudomást.
- **Izometrikus torlódásvédelem**: mivel a torlódást az okozza, hogy a hálózatban túl sok csomag van jelen, korlátozzuk a csomagok számát. Ezt úgy lehet kivitelezni, hogy ún. engedélycsomagokat vezetünk be. Csak az adhat, akinél az engedélycsomag van. Természetesen több engedélycsomag is lehet. Ez a módszer azonban nem garantálja, hogy egy csomópontba nem érkezhet be esetleg az összes engedéllyel rendelkező csomag. Ekkor pedig ismét torlódás keletkezik.
- **Lefojtó csomagok használata**: a routerek könnyen meg tudják figyelni kimenő vonalaik terhelését. Ez alapján el tudják dönteni, hogy torlódásveszély kezd-e kialakulni. Az algoritmus csak ekkor, torlódásveszélyes helyzetben lép működésbe. Működésének az a lényege, hogy megengedi a csomagok beérkezését, igyekszik gondoskodni a csomag kijutásáról, de hogy a csomagok fölszaporodását, és így a torlódást elkerülje, a csomag küldőjének egy ún. lefojtó csomagot küld vissza. Ennek a csomagnak csupán annyi a szerepe, hogy az eredeti csomag küldőjét lefoglalja, és így egy darabig nem ér rá újabb csomagot küldeni.

Összefoglalás

A hálózati rétegnek egyik fontos feladatához – a csomagok célba juttatásához – forgalomirányítást kell végeznie. Ebben a fejezetben ezért először a forgalomirányítási elveket tekintettük át. A módszerek közül az elosztott adaptív forgalomirányítás a legfontosabb, ugyanis a gyakorlatban a routerek is ezt az elvet használják.
A routolási algoritmusok után fel kellett ismernünk, hogy a hálózatokban nagyon gyakran előfordulhatnak torlódások, melyek a rendszer sebességét nagyban lecsökkentik. Ezért megvizsgáltunk néhány módszert, mellyel a torlódás megszüntethető vagy elkerülhető.

Ellenőrző kérdések

1) Mit jelent az adaptív forgalomirányítás?
   - A csomópontok úgy hoznak forgalomirányítási döntést, hogy nem veszik figyelembe a terhelésváltozást.
   - A csomópontok a terhelés változását figyelve képesek megváltoztatni a forgalomirányítást.
   - A csomópontok adaptálják a mellettük lévő csomópont döntését.

2) Melyik eljárásra jellemző: a csomópontok a belőlük kiinduló útvonalszakaszok leterheltségi adatait időközönként kicserélik egymással.
   - Elszigetelt adaptív forgalomirányítás.
   - Elosztott adaptív forgalomirányítás.
   - Központosított adaptív forgalomirányítás.
   - Determinisztikus forgalomirányítás.

3) A „forró krumpli” eljárás melyik forgalomirányítási módszerhez tartozik?
   - Elszigetelt adaptív forgalomirányítás.
   - Elosztott adaptív forgalomirányítás.
   - Központosított adaptív forgalomirányítás.
   - Determinisztikus forgalomirányítás.

4) Mi a befulladás?
   - Beragad egy csomag.
   - A torlódás másik elnevezése.
   - A torlódás egy szélsőséges esete.

5) Mi a közös az alábbi módszerekben: csomageldobás; lefojtó csomag; pufferelés?
   - Torlódásvédelmi eljárások.
   - Torlódásmegelőző eljárások.
   - Routolási eljárások.

VI. A felsőbb rétegek

Bevezető

Ebben a fejezetben az OSI-modell felső négy rétegét foglaljuk össze röviden. Ellentétben az alsó rétegekkel, ezek a rétegek már csak a forráshoszt és a célhoszt megfelelő rétegei között jönnek létre.

1. A szállítási réteg

A réteg feladata nagyon fontos, megbízható, gazdaságos adatszállítás biztosítása a forrástól a célig, úgy, hogy független maradjon a fizikai hálózattól. Tulajdonképpen e réteg feladata biztosítani azt, hogy a két kommunikáló hoszt pont–pont kapcsolatban lássa egymást, azaz úgy tudjanak kommunikálni, hogy közben ne kelljen tudomást venniük arról, hogy a kapcsolat létrehozásában akár több száz csomópont is részt vett.

1.1. A szállítási szolgálat

A szállítói rétegnek szolgálatokat kell nyújtani a felette lévő rétegeknek. Ezekhez a szolgálatokhoz fel kell használnia a hálózati réteg által nyújtott szolgálatokat. A szállítási rétegen belül azt a hardver- vagy szoftverelemet, amely a szolgálatot végzi, szállítási funkcionális elemnek vagy **szállítási entitásnak** nevezzük. Ez lehet az operációs rendszer magjának egy része, a hálózati kártya vagy önálló felhasználói folyamat, esetleg egy hálózati alkalmazáshoz tartozó könyvtár.
A hálózati szolgálatokhoz hasonlóan a szállítási szolgálatoknak is két típusa van: **összeköttetés-alapú** és **összeköttetés-mentes** szolgálat. Mindkét esetben az összeköttetésnek három fázisa van:

- összeköttetés létesítése
- adatátvitel
- összeköttetés bontása

A címzés és a forgalomszabályozás szintén hasonló a két rétegben.

A két réteg fenti egyezősége alapján felmerül a kérdés, hogy miért van szükség külön hálózati és szállítói rétegre, miért nem elegendő egy réteg?
A válasz a hálózatok működtetésére vezethető vissza. Amíg a hálózati réteg működtetése a szolgáltató feladata, a szállítói réteg már a felhasználóhoz tartozik.

Képzeljük el, hogy mi történhet akkor, ha a hálózat nem elég megbízható. A felhasználó nem nyúlhat bele az alhálózat működtetésébe, így nem tudja módosítani, javítani a gyenge minőségű szolgálat problémáját. Lehetősége van viszont a hálózati réteg fölé egy olyan réteget létrehozni, amely javítja a szolgálat minőségét.
Lényegében a szállítási réteg teszi lehetővé, hogy a szállítási szolgálat megbízhatóbb legyen az alatta lévő hálózati szolgálatnál. A szállítási réteg észleli a hibákat, az elveszett csomagokat vagy az adatok sérülését, és képes azokat kijavítani vagy újra kérni. Ezen kívül a szállítási szolgálat primitívjeit úgy tervezték, hogy függetlenek legyenek a hálózati réteg primitívjeitől, melyek hálózatonként jelentős eltérést mutatnak.

Ha minden valódi hálózat hibamentes lenne, és azonos szolgálati primitívekkel rendelkeznének, nem lenne szükség szállítási rétegre.

1.2. A szállítási protokollok elemei

A szállítási szolgáltatást egy szállítási protokollnak kell megvalósítani. Néhány feladatát tekintve a szállítási protokoll az adatkapcsolati protokollra emlékeztet: a feladataik közé tartozik a hibakezelés, sorszámozás és forgalomirányítás. Ugyanakkor jelentős eltérés is mutatkozik a két réteg protokolljai között. Az adatkapcsolati rétegben két csomópont közvetlenül a fizikai hálózaton kapcsolódik, ezzel szemben a szállítási protokollnál egy alhálózat van a két kapcsolódó hoszt között. Ebből következik, hogy az adatkapcsolati rétegben nincs szükség a címzett kijelölésére, a szállítási rétegben viszont több címzett is lehetne, melyek közül meg kell adni a kapcsolódó hosztot. Tehát a hálózati rétegben az összeköttetés létesítése sokkal bonyolultabb, **címzési** eljárást kell alkalmazni.

Egy másik különbség, hogy az adatkapcsolati szinten elküldött adat vagy megérkezik a címzetthez, vagy elvész, nincs olyan lehetőség, mint a szállítási szinten, hogy néhány másodpercig a router tárolja az adatot, és csak utána fog megérkezni. Tehát a **forgalomirányítási** eljárások is megváltoznak.

Végül az is egy lényeges különbség, hogy a szállítói réteg által használt **pufferelési** elvek is más jellegűek, mint az adatkapcsolati rétegben. Ennek az az oka, hogy a szállítási rétegben sokkal nagyobb mennyiségű összeköttetést kell létesíteni, mint az adatkapcsolati rétegben.

1.3. Szállítási protokollok

A gyakorlatban több szállításiprotokoll-kialakítás is létezik. Ezek közül talán az Internet szállítási protokollja a legismertebb, az Internet elterjedtsége miatt.
Az Internet szállítási rétegében két protokoll található: a TCP, ami egy összeköttetés-alapú szállítási protokoll, és az UDP, ami összeköttetés-mentes szállítási protokoll. Mindkét protokollal később fogunk részletesen foglalkozni.

2. Viszonyréteg

A viszonyréteg feladata, hogy a felhasználók között viszony létesítését tegye lehetővé. Az adatok átvitelén túl, a felhasználó számára nyújt szolgáltatásokat: pl. segítségével beléphet egy távoli gépre, fájlokat másolhat különböző gépeken. Egy viszony használhat egy vagy több szállítási összeköttetést is.

A viszonyréteg egyik szolgálata a párbeszédek irányítása is. A viszony egyirányú, illetve kétirányú adatforgalmat tesz lehetővé. A párbeszédes kapcsolat esetén fontos lehet, hogy nehogy ugyanazt a műveletet kezdeményezze mindkét fél. Ennek az elkerülés érdekében ún. **vezérjelezést** (token management) is végez a réteg. A kritikus műveleteket mindig az az oldal végezheti, amelyiknél a vezérjel (token) van.

Másik fontos feladata a viszonyrétegnek a **szinkronizálás**. Ez azt jelenti, hogy a viszonyréteg ellenőrzési pontokat iktat az adatátvitelbe, hogy az esetleges adásmegszakadások esetén ne kelljen a teljes adatsort újra átküldeni, csak az ellenőrző ponttól ismételjen az adó. Egy adást csak a két fél egyetértésével és együttes engedélyével lehet bontani.

3. Megjelenítési réteg

Ez a réteg felelős az adatok szabványos módon való kódolásáért. Ennek eredményeként tetszőleges lehet a kommunikáló gép típusa, operációs rendszer. Minden adó és vevő olyan adatátalakítást végez ezen a szinten, hogy a másik fél képes legyen feldolgozni az adott vagy kapott adatokat.
Ezenkívül fontos feladata még a rétegnek a hatékony adatovábbítás érdekében végzett tömörítés, illetve a biztonságos adat átvitel érdekében végzett titkosítás.

3.1. Adatábrázolás

Különböző számítógépek kommunikálnak egymással a hálózat segítségével. Fokozottan igaz ez napjainkban az Internet térhódításának következtében. Egy e-mailt el kell tudni olvasni Magyarországon és Kínában is. Mindegy, hogy milyen számítógépen írták, mindegy, hogy milyen operációs rendszert futtattak a gépen. Erre klasszikus példa az ASCII kód és a nagy számítógépek által használt EBCDIC kód. A két kódrendszer nem igazán volt kompatibilis. Mégis, ha régen egy nagygépen megírtak egy levelet, és elküldték egy személyi számítógépeket használó rendszernek, annak úgy kellett konvertálnia a levél jeleit, hogy mindkét rendszerben értelmes szöveget kapjanak. Ugyanez elképzelhető a számábrázolás terén is. Ennek a problémának a kiküszöbölése is a megjelenítési réteg feladata.

3.2. Adattömörítés

Mivel a hálózatok adatátviteléért általában fizetni kell, egyáltalán nem mindegy, hogy mekkora az átadni kívánt adat, és ebből következően mennyit kell majd fizetni. A adatátvitel gyorsaságát próbálják növelni a különböző tömörítési eljárásokkal.
A tömörítést az teszi lehetővé, hogy az adatok tárolása általában redundáns. Ennek a redundanciának a megszüntetése vagy csökkentése a tömörítés.

A tömörítési eljárásokat több csoportba sorolhatjuk.
Megkülönböztethetünk például szimmetrikus és aszimmetrikus tömörítési eljárásokat aszerint, hogy a betömörítési és kitömörítési módszerek azonosak vagy különbözőek-e.
Másik csoportosítási elv a veszteséges vagy veszteségmentes tömörítés. A veszteséges tömörítés esetében az eredeti adatból leválasztunk olyan részeket, amelyeket nem feltétlen szükséges átvinni, nélkülük még mindig használható az adat. Ezek a részek a tömörítés során elvesznek, kitömörítésnél nem lehet visszaállítani őket. Ilyen pl. a

legtöbb képtömörítési módszer. A képnek bizonyos információit eldobjuk a betömörítés során, így még mindig élvezhető, bár rosszabb minőségű képet kapunk. Ennél a módszernél az a legfontosabb kérdés, hogy hol a határ az eldobható és a fontos adatok között.
A veszteségmentes tömörítésnél a teljes adathalmazt betömörítjük, és kitömörítésnél minden tökéletesen helyreállítható. Ez sokkal rosszabb tömörítési arányt jelent, viszont minden adatelem megmarad.
Vizsgáljunk meg most néhány adattömörítési eljárást!

Darabszám-kódolás: ha az adathalmazban gyakran ismétlődő jelek vannak, akkor használható ez a módszer. Lényege, hogy egy speciális jel után megadjuk, hogy milyen adatelem ismétlését váltjuk ki, és hányszor ismétlődött az elem eredetileg. Pl. 16 „b" betű ismétlődik eredetileg. Ez kódolt formában mondjuk &b16. Így a 16 jel helyett 4 jelet kell csak átvinni.

Szimbólumsor-helyettesítés: gyakori, azonos jeleket tartalmazó szimbólumsort helyettesítünk egy jellel. (Pl. a TAB karakter helyettesíthet 8 szóközt.)

Minta-helyettesítés: gyakran ismétlődő karaktersorozatot kiváltunk egy speciális szimbólumsorral. (Pl. a programozási nyelvek fordítói az utasításokat – FOR, INPUT … – egy–egy speciális ASCII karakterre cserélik. FOR = 80H, INPUT = 81H.)

Sorozathossz-kódolás: sok nullát tartalmazó bitsorozatban jó módszer lehet, ha a nullák számát binárisan kódoljuk.
Pl.: 000100100000100001 összesen 18 bit
a nullák száma az egyesek között: 3 2 5 4
a nullák számát 3 biten kódolva: 011 010 101 100 összesen 12 bit
tehát csak 12 bitet kell átvinni a 18 helyett.

Statisztikai kódolás: ennél a módszernél a leggyakrabban használt jel kapja a legrövidebb kódot, és a legkevésbé használt jel a leghosszabb kódot. Ilyen elvet használ pl. a Morse-abc.

Huffmann-kódolás: lényege, hogy egyes jelek vagy jelsorozatok előfordulásának a gyakoriságát figyeli, és ettől teszi függővé a kód hosszát.

Transzformációs kódolás: a folytonos, összetett függvények leírhatók különböző vektorok összegeként, ilyen pl. a Fourier-transzformáció is.

Predikciós vagy relatív kódolás: ha ismert a kiinduló jel, akkor a többi jelnél elegendő csak az előzőhöz képest történt változást kódolni.

3.3. Titkosítás

A számítógép-hálózatok kialakulásakor nem fektettek túl nagy hangsúlyt a biztonság kérdésére. Akkor még nem nagyon keringtek bizalmas információk a hálózaton, vagy ha erre volt szükség, akkor saját hálózatot használtak. Manapság azonban a világháló elterjedésével bizalmasan kezelendő információk továbbítására is szükség van: ilyen lehet egy banki átutalásnál a kódunk, de a hitelkártyánk száma is. Ma már üzletek köttetnek a Neten, ezért a feleknek biztosnak kell lenniük abban, hogy valóban azzal kötöttek üzletet, akivel akartak.

A hálózattal kapcsolatos biztonsági problémák nagyjából négy, egymást átfedő területre oszthatók: titkosság, hitelesség, letagadhatatlanság és sértetlenség.

- A titkosság feladata az adat védelme az illetéktelen felhasználók ellen.
- A hitelesség azt garantálja, hogy valóban attól származik az üzenet, aki fel van tüntetve küldőként.
- A letagadhatatlanság az aláírással foglalkozik, azaz letagadhatatlan legyen egy elektronikus tranzakciónak minden részlete.
- A sértetlenség biztosítja azt, hogy az üzenet ugyanaz a megérkezés pillanatában is, mint az elküldés pillanatában.

Ugyanezek a problémák előfordulnak a hagyományos (nem elektronikus) adatátvitelnél is, de megoldási módjuk itt alapvetően más.
Az előbb említett biztonsági problémák mindegyike a titkosításra vezethető vissza, ezért nézzünk néhány titkosítási módszert.
Kezdetben még nem lehettek túl bonyolultak ezek a módszerek, mert embereknek kellett visszafejteni a kódokat.

Alapjában két módszert használhatunk:

Helyettesítéses rejtjelezést: ennek az a lényege, hogy az eredeti jelet valami mással helyettesítjük. Ennek a módszernek is több változata ismert.

- **Caesar-féle rejtjelezés**: az eredeti abc-t egy három (általános esetben k) jellel eltolt abc-vel helyettesítik. (Pl: a kutya ezek alapján nzwcd titkosítva.)
- **Többábécés rejtjelezés**: 26 Caesar-abc sort tartalmazó mátrix az alap. A kódolandó szöveg (más néven nyílt szöveg) fölé egy kulcssort írnak. A mátrixból kell behelyettesíteni egy betűt. A behelyettesítendő betű sorát a kulcs sor határozza meg, az oszlopot pedig a nyílt szöveg betűje.

  Pl.: Az első betű z, mert az e-sorának és t-oszlopának a metszéspontjában z van.

A	B	C	D	E	…
B	C	D	E	F	
C	D	E	F	G	
D	E	F	G	H	
E	F	G	H	I	
…					

kulcs: ezakulcssorezakulcssor
nyílt szöveg: titkosszoveg
titkosítva: zhtuidupgjvk

(A teljes mátrixot, és a titkosító függvényt, a mellékelt Excel tábla tartalmazza.)

Felcseréléses rejtjelezés: a jelsorozat elemei megmaradnak, csak a sorrendjük változik meg a rejtjelezés során.

- **Keverőkód:** olyan kulcsot használunk, amiben minden betű csak egyszer fordul elő. A kulcs alá, balról jobbra sorba beírjuk a titkosítandó szöveg betűit. Így egy mátrixot kapunk. A mátrix oszlopaiban lévő betűk fogják alkotni a titkosított szöveget. Úgy kell az oszlopokat összeolvasni, hogy a kulcs betűit az ábécé alapján megszámozzuk, és 1-től növekvő sorrendbe olvassuk össze az oszlopokat.

pl.:

kulcsszó:	**k**	**u**	**l**	**c**	**s**
az abc-ben hányadik betű:	11	21	12	3	19

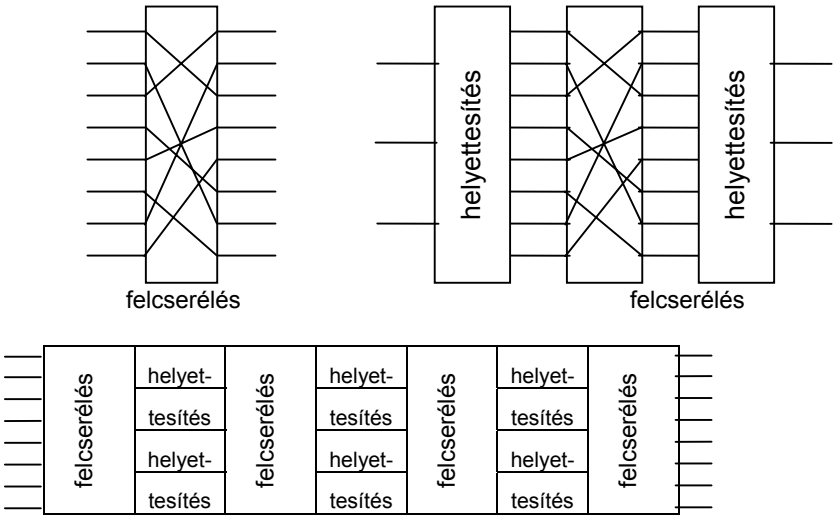
s	o	r	b	a
l	e	k	e	l
l	í	r	n	i
a	t	i	t	k
o	s	í	t	á
s	r	a	v	á
r	ó	s	z	ö
v	e	g	e	t

Titkosítandó: „sorba le kell írni a titkosítandó szöveget”
(természetesen a szóközöket ki kell hagyni)

Titkosított szöveg: benttvzesllaosrvrkriíasgalikáäötoeítsróe

- **Egyszer használatos bitminta:** ezzel az eljárással valóban feltörhetetlen kódot lehet készíteni, tulajdonképpen számítógép használata nélkül.
  A módszer egész egyszerű: a kulcs egy tetszőleges hosszúságú véletlen bitsorozat. A kódolandó üzenet betűit szintén bitsorozattá kell alakítani, például ASCII kódjuk alapján. A két sorozatra kizáró vagy műveletet kell végezni, és így megkaptuk a titkos szöveget.
  Azért feltörhetetlen, mert a titkos szöveg semmiféle információt nem tartalmaz a kiindulásról, és kellően hosszú szöveg esetén minden betű és betűpár azonos gyakorisággal fordul elő benne.
  Feltörhetetlensége ellenére meglehetősen nagy hátrányai vannak:
  - nagyon nehéz megjegyezni a kulcsot, ezért a küldőnek is és a fogadónak is rendelkeznie kell egy leírt kulccsal, amit meg lehet szerezni,
  - mivel csak annyi karaktert tudnak kódolni, amilyen hosszú a kulcs, nem biztos, hogy elég a kódoláshoz, vagy nagyon hosszú kulcsra van szükség,
  - érzékeny a bithibákra, azaz egy bit eltévesztése már gondot okoz a visszafejtésben.

  A számítógép megjelenésével már hosszabb szövegek is kódolhatók ezzel az eljárással, de elég körülményes az alkalmazása.
- **DES (Data Encryption Standard) – Adattitkosítási szabály:**
  A számítógép megjelenésével lehetőség nyílt arra, hogy a klasszikus kódolási eljárásokat gépesítsék, és összevonják, így olyan bonyolult titkosítást lehet elérni, amit még számítógéppel is csak nagyon sok idő alatt lehet visszafejteni.
  A hagyományos helyettesítéses és felcseréléses kódolásokat egymás után tudják fűzni számítógéppel, így többszörös kódolás után keletkezik a végleges kód. A helyettesítést és a felcserélést egyszerű áramkörökkel meg lehet valósítani.



40. ábra
Helyettesítés és felcserélés

Ezt az elvet dolgozza fel a DES is, ami az IBM 1977-ben szabványosított kódolási eljárása. Az eljárás a nyílt szöveget 64 bites blokkonként kódolja. A kódolás után szintén 64 bites titkos üzenetet kap. Az algoritmus egy 56 bites kulcsszót használ, és 19 lépés során felcserérést és helyettesítést végez.

- **Nyilvános kulcsú titkosítás:**
  A legtöbb kriptográfiai rendszer problémája hosszú ideig az volt, hogy a titkosítási kulcsot mindkét félnek ismernie kellett, így nem lehetett biztos az üzenet küldője, hogy nem került-e illetéktelen kezekbe a kulcs.
  1976-ban két stanfordi kutató merőben új eljárást javasolt, amiben a kódoló és a dekódoló kulcs nemhogy nem volt egyforma, de még elő sem lehetett állítani az egyikből a másikat.
  A módszer működési elve: az adatsort az A kódolási eljárással titkosítják, így egy olyan adatsor áll elő, melyet csak a B kódolási eljárással lehet visszafejteni. A és B eljárásoknak az a különlegessége, hogy függetlenek egymástól, és az egyikkel kódolt adatsort a másikkal vissza lehet fejteni. Azaz, ha A-val titkosítunk egy adatsor, azt csak B-vel lehet visszafejteni, és megfordítva, ha B-vel titkosítunk egy adatsort, azt csak A-val lehet visszafejteni. Másképp fogalmazva: két kulccsal rendelkezünk, melyek egymás párjainak tekinthetők. Ezen eljárás kétkulcsos titkosítás néven is közismert.
  Mivel a két kulcs csak párban használható, adott a lehetőség, hogy az egyiket a felhasználó tartsa meg saját magának (ezt nevezik „titkos kulcsnak”, ezt szigorúan őriznie kell a tulajdonosának), a másikat viszont nyugodtan odaadhatja ismerőseinek (ez a „nyilvános kulcs”, melyet akár az Interneten keresztül is lehet terjeszteni).

A két kulcs több lehetőséget biztosít a felhasználók számára:

- Az adat (pl. e-mail) hitelességét lehet vele bizonyítani: úgy, hogy a levél írója a saját titkos kulcsával titkosítja a levelet. Így csak a kulcs párjával lehet azt elolvasni, ez pedig az ún. nyilvános kulcs, amihez bárki hozzáférhet. Ekkor bárki el tudja olvasni az adatot, és biztos lehet abban, hogy a feladója a kulcspár tulajdonosa. Így tulajdonképpen digitálisan aláírta a levelet az írója.
- Úgy lehet titkosítani a kulcspár segítségével az adatot (pl. e-mail), hogy csak a címzett olvashatja el. Ehhez a levelet titkosítani kell a nyilvános kulccsal. Ezt bárki megteheti, de csak a titkos kulcs birtokosa tudja kicsomagolni az adatot.
- Ha az adat (pl. e-mail) küldője két kulcspárt is használ, akkor teljesen biztos lehet abban, hogy a levelét csak a címzett tudja elolvasni, a címzett pedig biztos lehet benne, hogy a levél a feladótól érkezett. Ennek a folyamatnak a következők a lépései:
  - F a feladó, C a címzett, mindkettejüknek van nyilvános (Fn, Cn) és titkos (Ft, Ct) kulcsa is.
  - A feladó megírja a levelet, és hitelesíti azt a saját titkos kulcsával (adat+Ft). Ekkor még bárki elolvashatná, ezért a címzett nyilvános kulcsa segítségével titkosítja a levelet (adat+Ft+Cn). Így csak a címzett tudja megfejteni az üzenetet.
  - A címzett – miután megkapta a titkosított levelet – először a saját titkos kulcsa segítségével kicsomagolja azt (adat+Ft+Cn-Ct), majd a feladó nyilvános kulcsa segítségével meggyőződik arról, hogy valóban a feladóként megjelölt személytől érkezett a levél (adat+Ft+Cn-Ct-Fn). Így visszakapja a levél eredeti tartalmát.

PGP (Pretty Good Privacy – „elég jó” titkosítás)

A PGP az előző elvnek egy a gyakorlatban is jól bevált megvalósítása: egy komplett e-mail biztonsági csomag, amely felkínálja az adatvédelem, a hitelességvizsgálat, a digitális aláírás és a tömörítés lehetőségét egyszerűen kezelhető formában. A titkosításhoz 128 bites kulcsot használ.

4. Alkalmazói réteg

Az alkalmazói réteg feladata hálózati szolgáltatások biztosítása a felhasználó és a felhasználói programok részére.
A legfontosabb hálózati szolgáltatások:

- állomány-hozzáférés biztosítása, állományok továbbítása (pl. FTP),
- elektonikus levelezés (e-mail),
- virtuális terminálok kezelése (pl. Telnet),
- egyéb szolgáltatások, pl. névszolgáltatások (pl. DNS).

Összefoglalás

Ebben a fejezetben az OSI-modell további rétegeit elemeztük röviden. Először a szállítási réteg feladatáról és a szállítási protokollokról volt szó, majd szóltunk a viszonyrétegről, melyet a megjelenési réteg kifejtése követett.
A megjelenési réteg feladatai közül kiemeltük

- az adatábrázolást,
- az adattömörítést,
- és a titkosítást.

Ebben a fejezetben leginkább a tömörítésre és a titkosításra tettük a hangsúlyt. Ennek megfelelően a következő adattömörítési eljárásokat részleteztük:

- darabszám-kódolás,
- szimbólumsor-helyettesítés,
- mintahelyettesítés,
- sorozathossz-kódolás,
- statisztikai kódolás,
- Huffmann-kódolás,
- transzformációs kódolás,
- predikciós vagy relatív kódolás.

Mindezek mellett a következő titkosítási algoritmusokat tárgyaltuk:

Helyettesítéses rejtjelezés:

- Caesar-féle rejtjelezés,
- többábécés rejtjelezés.

Felcseréléses rejtjelezés:

- keverőkód,
- egyszer használatos bitminta,
- DES – Adattitkosítási szabály,
- nyilvános kulcsú titkosítás.

Ellenőrző kérdések

1) Az Internet szállítási rétegének két protokollja van. Melyek ezek?
   - IPX, SPX
   - TCP, IP
   - TCP, IPX
   - TCP, UDP

2) Melyik tömörítési eljárásra jellemző: ugyanazzal a módszerrel kell az adatokat betömöríteni és kitömöríteni.
   - Szimmetrikus tömörítés.
   - Aszimmetrikus tömörítés.
   - Redundáns tömörítés.
   - Veszteséges tömörítés.
   - Veszteségmentes tömörítés.

3) Melyik tömörítési eljárásra jellemző, hogy a gyakori, azonos jeleket helyettesíti egy jellel?
   - Darabszám-kódolás.
   - Szimbólumsor-helyettesítés.
   - Mintahelyettesítés.
   - Sorozathossz-kódolás.
   - Statisztikai kódolás.

4) Melyik tömörítési eljárásra jellemző, hogy a nullák számát binárisan kódolja?
   - Darabszám-kódolás.
   - Szimbólumsor-helyettesítés.
   - Mintahelyettesítés.
   - Sorozathossz-kódolás.
   - Statisztikai kódolás.

5) Melyik tömörítési eljárásra jellemző, hogy sok egyforma karakter helyett a számát tartja nyilván?
   - Darabszám-kódolás.
   - Szimbólumsor-helyettesítés.
   - Mintahelyettesítés.
   - Sorozathossz-kódolás.
   - Statisztikai kódolás.

6) Mi a felcseréléses rejtjelezés?
   - A karaktert egy másikkal helyettesíti.
   - A karaktereket összekeveri.
   - A karakterek olvasási irányát fordítja meg.

7) Mi a PGP?
   - Egyfajta DES-eljárás.
   - Egyfajta nyilvános kulcsú eljárás.
   - A nyilvános kulcsú eljárás elterjedt neve.

VII. Lokális hálózatok

Bevezető

A számítógép hálózatok fejlődésével mindennapossá vált a helyi (lokális) hálózat használata. Ez a fejezet a helyi hálózatok típusaival, lehetséges szolgáltatásaival foglalkozik. Bemutatjuk a hálózatok eszközeit, szabványosításaikat, majd említést teszünk a legelterjedtebb lokális hálózati operációs rendszerekről.

1. A lokális hálózat definíciója

Ha két vagy több számítógépet összekapcsolunk úgy, hogy azok kommunikálni tudnak egymással, már számítógépes hálózatról beszélünk. Az ember hajlamos mindent, így e hálózatokat is bizonyos szempontok szerint osztályozni. Meg is született három bűvös szó: **LAN, MAN, WAN**. Születésükkor még jól meg lehetett különböztetni, hogy az éppen vizsgált hálózat melyikbe is tartozik, ma azonban már átfedések lehetnek az egyes kategóriák között. Ennek ellenére még napjaikban is ez a legelterjedtebb csoportosítása a hálózatoknak.

LAN: Local Area Network	**lokális (helyi) hálózat**
MAN: Metropolitan Area Network	**városi hálózat**
WAN: Wide Area Network	**nagy kiterjedésű hálózat**

A csoportosítás alapvetően kiterjedés (az összekapcsolt számítógépek távolsága) szerinti megkülönböztetést sugall, amely valóban a legfontosabb jellemző tényező.

De vajon lokális hálózatban van-e az a két gép, melyek egy irodaház két szobájában helyezkednek el egymástól néhány tíz méter távolságban, és a kapcsolatot egymással egy távközlési szolgáltató kapcsolt vonalán tartják? Nem, mert a kapcsolóközpont távol van. És ha ugyanabban az épületben van a kapcsolóközpont? Nem, mert a kapcsolóközpont más kapcsolatot is biztosíthat, ezért már egy nagyobb hálózat része lehet a két gép. És ha bérelt vonalat vesznek igénybe a két gép összekötésénél? Nincsenek lokális hálózatban, ha a kapcsolatot létrehozó kábelek hossza kilométeres nagyságrendű? És vannak, ha csak néhány száz méter? Nincsenek, ha egy távközlési szolgáltatást vesznek igénybe, még ha néhány száz méteres távolságról is van szó? Távközlési szolgáltatás használatának minősül-e, ha egy toronyház optikai kábeles kommunikációs hálózatát bérlem? Ha igen, akkor egy tizenegy emeletesé? Tíz, kilenc…?
Vagy a technikai megoldás mérvadó, amit az elnevezés nem is tükröz? Elfogadott helyi hálózatépítési technológiával összeköthetők néhány száz méterre lévő épületek számítógépei. Sőt, megfelelő eszközökkel a távolság növelhető. Mikor válik egy LAN MAN-ná?

Nos, azt kell mondanunk, hogy nincs egzakt definíció, ám ez mégsem okoz gondot. Egyrészt azért, mert mégiscsak az a legfontosabb, hogy az összekapcsolt számítógépek hálózata működjön, mindegy, melyik mozaikszó büszke birtokosaként. És ha elfogadjuk azt az ismérvet, hogy a lokális hálózat elemei néhány száz méter kiterjedésű körzetben helyezkednek el, akkor az esetek túlnyomó többségében meg tudjuk mondani, hogy hálózatunk belül van-e a LAN keretein, vagy túllépte azt. Az is igaz, hogy vannak jellemzően LAN építési technológiák, melyek nem alkalmazhatók például WAN esetében.

Talán jól példázza a kategorizálást a következő, a valós életből vett eset.
Egy tipikus helyi hálózat az irodai hálózat. Ez lehet egy 32 m2-es kicsi iroda a maga 2 számítógépével és egy nyomtatójával, de lehet egy 10 emelet irodaház is több száz számítógéppel. Közös jellemzőjük, hogy azonos alhálózatban vannak.
Ebből következően LAN-oknak tekinthetők az egyes OTP fiókok helyi hálózatai. Azonban a Budapesten lévő összes OTP fiókot összekötő hálózat már a MAN kategóriába tartozik. Ha ugyanennek a vállalatnak az országos hálózatát nézzük, akkor az már a WAN kategóriába tarozik.
Ugyancsak a WAN kategóriába tartozik a Siemens irodákat összekötő nemzetközi méretű hálózat is, de tulajdonképpen ide tartozhat a teljes Internet is. Ha azonban ezt az utóbbi két hálózatot összehasonlítjuk, jól látható, hogy nagy különbség van köztük méretüket tekintve is. Ezért ezt a nagy méretű WAN hálózatot néha a GAN (Global Area Network - globális hálózat) elnevezéssel is illetik.

2. Funkcionális szervezés

A számítógép-hálózat kialakításának célja, hogy a kapcsolatban álló gépek használhassák egymás erőforrásait (perifériáit); továbbá hogy egyszerű adatközlés küldő és/vagy fogadó oldalán szerepelhessenek. E feladatok, szerepek és a munkamegosztás tekintetében alapvetően két modellt különböztetünk meg.

2.1. Egyenrangú (peer to peer) hálózat

Az ilyen hálózatba kötött számítógépek mindegyike ugyanolyan jogokkal, lehetőségekkel bír, továbbá egyiknek sincs kitüntetett szerepe a hálózati munka szempontjából. Nevezetesen: mindegyik gép (azaz a rajta dolgozó) a saját erőforrásait mások rendelkezésére bocsáthatja és meghatározhatja az ezekhez hozzáférők körét. Két gép közötti üzenetküldésnél nincs szükség további szereplőre.

E pozitív tulajdonságok áttekintésekor felmerül a kérdés: miért nem ilyen minden hálózat?
Válaszként azt mondhatjuk, vannak hátrányok is, melyek főleg nagyobb számú (kb. 10 vagy több) gép esetén jelentkeznek. Ha egy gép erőforrásához (pl. winchesteréhez) fordulnak sokan egyszerre (pl. a főnök gépéhez, aki épp reggel tette ezen közzé a csapatmunka alapját

képező, kellően nagy adatbázisát), és mondjuk csak egy átlagos teljesítményű berendezésről van szó, akkor ha több tízen fordulnak a gépéhez egyszerre, főnökünk bizony jobban teszi, ha a titkárnőnek ad addig instrukciókat (pl. kávé), míg lanyhul az adatbázisa utáni érdeklődés. Ugyanígy a titkárnő is kérheti főnökét, hogy lassabban diktáljon, ha másvalaki egy komolyabb anyagot küld a gépére csatlakoztatott és megosztott nyomtatójára.
További problémát jelenthet még a sokszereplős egyenrangú hálózatnál a szervezetlenség (mit hol találok?).

Kényelmetlenséget okozhat, hogy ahány gép, annyi azonosító (és esetlegesen jelszó) szükségeltetik az idegen erőforrások igénybevételéhez. Ha ugyanazt a jelszót használom minden gépen, akkor az időnkénti módosítást minden gépen el kell végeznem. Ez meglehetősen körülményes, továbbá minden gépen egyénileg kell elvégezni az adatmentést, amely így elosztva és összegezve többletmunkát, több archiváló perifériát (pl. CD-írót), háttértárat (írható CD-lemezt) igényelhet. Szintén gépenként kell elvégezni az egyén (mondjuk az új alkalmazott) hozzáférési jogainak beállítását.

Mennyi hátrány – azonban mégsem kell mindenkit lebeszélni az egyenrangú hálózatról. Jó beruházás lehet kis számú számítógép egyenrangú hálózatba kötésénél, mert itt egyáltalán nem jelentkeznek a fenti negatívumok, vagy legalábbis nem zavaróan. Néhány gép hálózati munkájának ellátására pedig fölösleges, pazarló beruházás lenne külön hálózati kiszolgáló gép beállítása, ami nem csak a hardver, hanem a hálózati operációs rendszer és az ügyfél-licencek megvásárlását is jelenti (persze típustól függően). Tudni kell azt is, hogy egy kiszolgáló gép (szerver) lényegesen drágább, mint az átlag szóló vagy hálózati munkaállomásgép, továbbá nem kell magasabb informatikai képzettségű alkalmazottat is foglalkoztatni a hálózat felügyeletére. Annak is lehet előnye, hogy mindenki saját maga adminisztrálhatja gépét, nem kell bürokratikus erőfeszítést tenni a hálózati rendszergazda megnyerésére.

2.2. Kliens-szerver (ügyfél-kiszolgáló) hálózat

Mindazok a problémák, amelyek egyenrangú hálózatnál gondot okoznak, egy jól megtervezett és megvalósított kliens-szerver hálózatnál nem jelentkeznek. Lényege, hogy a hálózati munkát egy (vagy több) kiszolgáló gép koordinálja, és teljesíti a hálózati kéréseket.
Milyen feladatokról lehet szó? Az elsődleges az, hogy a hálózat munkaállomás gépeit, felhasználóit azonosítani kell, és nyilván kell tartani a felhasználók jogait. Ez önmagában nem sok hasznot jelentene, de szükséges ahhoz, hogy igénybe lehessen venni a kiszolgáló szolgáltatásait: egyrészt a gép saját erőforrásainak megosztását, vagyis a szerver-periféria szolgáltatást; másrészt olyan speciális szoftver-szolgáltatásokat, amelyeket kimondottan kiszolgáló gépekre készítettek.

Minthogy a szervernek a kliensek egyidejű kéréseit is hatékonyan kell tudnia teljesíteni, továbbá a rajta futó programok is „nagyobbak”, összetettebbek, ezért „erősebb” gépet, jobb paraméterekkel rendelkező hardvert igényel, mint általában egy munkaállomás, így drágább is.

A szerver paramétereit nyilván a feladatai figyelembevételével kell megállapítani. Elsődleges szempont, hogy hány gépet kell kiszolgálnia egy időben, további szempont pedig a periféria-szolgáltatások típusa, volumene, a szoftver-szolgáltatások mennyisége, azok hardverigénye. Egy szerver munkába állítását követően biztosítani kell annak felügyeletét, a hálózati rendszergazda feladatainak ellátását. Ez a tevékenység több informatikai tudást igényel, mint egy helyi operációsrendszer-ismeret, tehát plusz (kvalifikált) emberi erőforrás igénybevételére van szükség, amely persze az üzemeltetési költséget növeli. A centralizált hálózati szolgáltatás a teljes rendszer leállásának valószínűségét növeli, amit a kiszolgáló üzembiztos működését valószínűsítő minőségének biztosításával lehet ellensúlyozni. Természetesen ez is közvetlenül (ha nem is mindig hűen) megjelenik az árban. További extraköltség lehet a megtervezett, de nem az alapmunkaidőben és gyorsan végrehajtott karbantartás; a nem tervezett, váratlanul bekövetkezet hiba elhárítása; vagy az ilyen esetekre megkötött szerződés, az átalánydíjas hibaelhárító szolgáltatás. Az üzemelési biztonságot persze növelni lehet például a szerverszám emelésével, a háttértárak megfelelő szervezésével, működés közbeni cserélhetőségével is.

A kliens-kérések hatékony kiszolgálása mellett további előny a felhasználók központi adminisztrációja. A hálózati rendszergazda egy helyen állítja be a felhasználók jogait, a felhasználónak egy azonosítóval és egy jelszóval kell csak rendelkeznie ahhoz, hogy hozzáférhessen a számára biztosított központi erőforrásokhoz. A szerver-kliens programok használata esetén többnyire csak a szerveroldali programot kell módosítani ahhoz, hogy a javított vagy a módosított szolgáltatás lépjen életbe, ami gyors és egyidejű átállást jelent, nem beszélve a munkamegtakarításról. Képzeljünk el például egy 100 gépből álló lokális hálózatot! Minden munkaállomáson ugyanazzal a kliens programmal kezelik a kiszolgálón található adatbázist, így a rendszergazdának a módosításhoz végig kell látogatni mind a 100 helyet, kicsit várni, míg megkapja a gépet, keveset beszélgetni (már csak az illendőség miatt), válaszolni néhány „de jó hogy itt van” típusú kérdésre stb. – érdemes beszorozni.
Viszont ha a frissítést csak a szerveren kell elvégezni, akkor a művelethez elég lehet 5-10 perc is. Az igaz, hogy az első megoldás „kalandosabb”, de egy idő után, megfelelő hálózati szerverprogram birtokában előnyben részesítjük az utóbbi megoldást.
Nem kis előny továbbá, hogy könnyen és jól áttekinthető, menedzselhető rendszert valósíthatunk meg a kliens-szerver technológiával. Nincsenek bővítési korlátok, a rendszer skálázható, azaz a megnövekedett igényeket viszonylag egyszerű bővítésekkel ki lehet elégíteni.

Úgy tűnhet, mintha fent élesen szembeállítottuk volna a két szervezési típust, azonban nem kell kizárólag csak az egyikben gondolkodni. Ugyanazon a fizikai hálózaton ugyanazok a gépek működhetnek az egyik, illetve a másik hálózati típus szerint is. A felhasználó (persze megfelelő jogok alapján) dönti el, hogy egy kiszolgálóra vagy egy egyenrangú ügyfelek alkotta munkacsoportba jelentkezik be.

3. Szerverszolgáltatások

Az adott szerverszolgáltatást vagy maga a hálózati operációs rendszer végzi (melyet a kiszolgálóra telepítenek), vagy kimondottan az adott feladatra készített és a hálózati operációs rendszer fölé telepített program.
Ezek a programok – normális üzemben – párhuzamosan futnak a kiszolgálón, hogy bármikor tudják a kliens-kéréseket teljesíteni.

A párhuzamos programfutás egy processzoros szervernél persze látszólagos. Az elindított programok váltakozva apró időszeleteket kapnak, amely időintervallumban övék a processzor. Ennek letelte után a program várakozik, míg újból neki jut egy kis processzoridő. A felhasználó mindezt úgy látja, hogy az elindított programok párhuzamosan, egy időben futnak. Természetesen a szerver egy adott tevékenysége gyorsabb, ha nem kell osztozni más tevékenységekkel a processzor kapacitásán. Az előbb vázolt időosztásos (multitasking) technika kulcsszereplője a processzor, ezért a szervergép processzorának paramétereit a kiszolgáló programok, azok processzorigénye figyelembevételével kell megválasztani. A hatékonyság növelhető többprocesszoros gépek alkalmazásával is, ahol már valódi párhuzamos feladat-végrehajtásról beszélhetünk. Ekkor is az időszeletek kapnak szerepet, csak egy időben annyi folyamat fut, ahány processzor áll rendelkezésre.

Jegyezzük meg, hogy az esetek túlnyomó többségénél mégsem a processzor teljesítménye, hanem a perifériák (winchesterek) kezelési sebessége határozza meg a kiszolgáló hatékonyságát.

A következőkben néhány jellemző lokális hálózatban alkalmazott szerverszolgáltatást tárgyalunk.

3.1. Hálózati felhasználók adminisztrációja

A hálózat felhasználóinak, azok jogköreinek nyilvántartását a szerver operációs rendszere végzi. Adott szerver hálózati szolgáltatását az veheti igénybe, aki rendelkezik az illető szerverre vonatkozó azonosítóval és a hozzá tartozó jelszóval. Fontos megjegyeznünk, hogy egy fizikai hálózaton több szerver is működhet egy időben, és e szervereken lehetnek különböző típusú hálózati operációs rendszerek.

Előfordulhat az is, hogy e különböző kiszolgálókhoz ugyanazt az azonosítót és jelszót használjuk, amely persze külön-külön nyilván van tartva az egyik, illetve a másik szerveren is. A hálózati szolgáltatás igénybevétele előtt „be kell jelentkezni” az adott hálózati szolgáltatást koordináló szerverre, ami az – adott operációs rendszertől függő módon történő – azonosító és jelszó megadását jelenti. A rendszergazda által felvett és a szerveren tárolt felhasználói jogkör alapján veheti igénybe a bejelentkezett felhasználó a számára hozzáférhető szolgáltatásokat.

A különböző hálózati operációs rendszerek némiképp eltérő hálózati filozófiát valósítanak meg. Lehetnek olyan szerverek, melyek nem foglalkoznak a felhasználó adminisztrálásával, ezt egy olyan másik szerverre bízzák, melynek a logikai hatókörében (tartományában) működnek.
Egyszerűsítheti a felhasználói jogok kezelését (sok hálózati felhasználó esetén különösen), ha felhasználói csoportokat hozunk létre. Így nem kell minden egyes egyénnél külön-külön elvégezni a – sokszor aprólékos munkát igénylő – jogkörbeállítást, elég csak a csoportra megtenni. A felhasználónál ezután elég csak a csoporttagságot megjelölni.

3.2. Fájlkiszolgáló

Ez a szerverek „legnépszerűbb” szolgáltatása, melyben a kiszolgáló – a beállított hozzáférési jogosultságoknak megfelelően – a hálózati felhasználók rendelkezésére bocsátja merevlemezes erőforrásának egy részét. Ehhez persze az kell, hogy a kiszolgáló nagyobb kapacitású, gyorsabb elérésű háttértárral rendelkezzék, mint egy átlagos kliens.

Erre azért van szükség, mert

- így lehetőség nyílik adatok központi szolgáltatására, és a módosítást csak a kiszolgálón kell végrehajtani;
- csoportmunkánál egy helyen van az aktuális anyag, melyet mindenki elérhet;
- nem kell a másolatok szinkronizálására figyelni. Nem kell több helyen (a klienseken) tárolni ugyanazt;
- kényelmesen és egyszerűen „kibővíthető” a kliensgép winchesterének kapacitása;
- bizonyos munkafolyamatok egyszerűsödhetnek, például a központi adatmentés;
- a hálózati felhasználó nincs ugyanahhoz a kliensgéphez kötve;
- az adatok tárolása biztonságosabb egy műszakilag megbízhatóbb szerveren, és maga a tárolóegység is fizikailag biztonságosabb helyen lehet;
- olcsóbb a szerverben biztosított adott (nagyobb) kapacitású tárterület, mint megosztva a kliensgépekben.

3.3. Nyomtatószerver

A nyomtatószerver is – általában – egy hálózati operációs rendszer beépített szolgáltatása. Többnyire pazarlás lenne, és nincs is arra lehetőség, hogy minden kliensgéphez külön nyomtatót vásároljunk. A szervergépre csatlakoztatott nyomtatót a szerver mint saját erőforrását – a jogosultsági előírásoknak megfelelve – a felhasználók rendelkezésére bocsátja. A nyomtatószerver-program gondoskodik a kliensoldali nyomtatási kérések fogadásáról, sorba állításáról, ütemezett végrehajtásáról.

A nyomtatószervernek több változata létezik, a fenti leírás a klasszikus, szerver által nyújtott szolgáltatást írja le.
Azonban meg kell említenünk a szervertől független „nyomtatószervert" is. Ez egy olyan berendezés, mely közvetlenül kapcsolódik a hálózatra, és ehhez csatlakoztatják a nyomtatót. Pont ez a hatalmas előnye, hogy nem kell sem számítógéphez csatlakoztatni, sem pedig szerverhez vagy munkaállomáshoz: önmaga dolgozza fel a nyomtatásokat, maga tárolja a nyomtatási sorokat.
Napjainkban egyre nagyobb teret hódít ez a megoldás, olyannyira, hogy nagyobb teljesítményű, kimondottan hálózati nyomtatásra szánt nyomtatókba eleve beépítik.

Több gyártó is készít ilyen hálózati nyomtatószervert. Néhány példa a nyomtatószerver technikai megvalósítására (a linkek az egyes gyártók oldalaira, leírásaira mutatnak):

HP külső nyomtatószerverek

Miért jobb a hp JetDirect, mint egy pc-nyomtatószerver?

Compaq:

Digicom:

Axis:

3.4. Alkalmazásszerver

Vannak olyan, hálózatot feltételező programok – vagy programrészek –, melyek a kiszolgálón futnak. A kliensek ezeket egyidejűleg használhatják. Előnye, hogy a programfrissítés gyorsabb, egyszerűbb, mert többnyire csak a szerveren kell végrehajtani az aktualizálást. Szerényebb paraméterekkel rendelkező kliensgépek is elegendők, és a hálózati felhasználó nincs konkrét munkaállomáshoz kötve, viszont a hatékony működéshez gyors hálózat szükséges.

3.5. Adatbázisszerver

Az adatbázis az információhalmaz adott logikájú, strukturált informatikai megjelenítési formája. A tárolt információ egy meghatározott részének gyors kinyerése, illetve az egész adathalmaz adott szempontok szerinti kiértékelése fontos feladat. A kiszolgálón futó adatbázis-szerverprogram épp ezt a funkciót látja el. Egyszerre tudja fogadni a kliensoldali kéréseket, és ütemezve hajtja azokat végre. Minden felhasználó úgy érzi, hogy folyamatos kiszolgálásban részesül.

3.6. Levelezőszerver

A kiszolgálón futó levelezőszerver-program nyilvántartja a levelező felhasználók adatait, és tárhelyet biztosít mindenkinek a ki- és bejövő levelek számára. Alapértelmezésben minden felhasználó csak saját leveleihez férhet hozzá. Amikor a felhasználó a saját gépén egy levelező kliensprogram segítségével elkészíti és elküldi a levelét, azt a lokális hálózaton a levelezőszerver fogadja, majd a többi küldendő levéllel együtt továbbítja a lokális hálózat egy kilépési pontján (megfelelő hardverelemen) keresztül – általában egy távközlési szolgáltató segítségével – egy elektronikus levelező szolgáltatást biztosító számítógépre.

Abban az esetben, ha a levél a lokális hálózat egy belső felhasználójának szól, akkor a szerverprogram a levelet a címzett „beérkező levelek” postafiókjába helyezi, ami a kiszolgáló háttértárolóján található. A levelezőszerver – a külső kapcsolat típusától függően – időszakosan vagy folyamatosan lekéri, illetve fogadja a külső leveleket, és behelyezi azokat a címzett postafiókjába. A felhasználó tetszőleges időben bejelentkezhet a szerverre, és kliens-levelezőprogramja segítségével lekérheti leveleit.

3.7. Proxy szerver

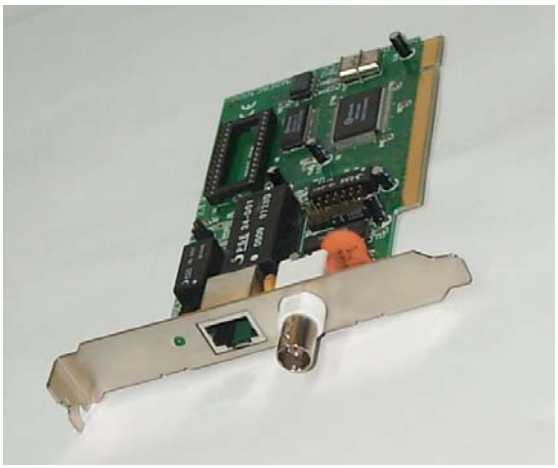
Az olyan programokat nevezzük proxy szervernek, amelyek a kliensprogram és az Internet között végzik az adatok átvitelét. A kliens a proxy szerverhez, az pedig a lokális hálózaton kívüli tényleges internetszerverhez kapcsolódik.
A proxy szerver felhasználónként és csoportonként szabályozza az Internet kívánság szerint beállított címeihez való hozzáférést. Intelligens, aktív gyorsítótára (cache) segítségével pedig jelentősen felgyorsítja a gyakran kért oldalak elérését. A biztonság alapvetően két dolgot jelent internetelérés esetén: kívülről ne törhessenek be, és belülről ne tudjon bárki bárhova kijutni.

A lokális hálózat gépein általában intranetes címtartományt használnunk (ha nincs hivatalos címtartományunk), a szervernek pedig az Interneten érvényes IP-címmel kell rendelkeznie. Ilyenkor lokális hálózatunk gépei rejtve maradnak az Internet felől, és az általuk küldött kéréseket a szerver úgy továbbítja, mintha eredetileg is tőle származtak volna. A válaszokat is a szerver kapja meg, majd továbbítja a kérdező gép felé. Lehetőség van az átmenő forgalom korlátozására is, a címek (kiinduló és célállomás), protokollok (tcp, udp, icmp) és portok (pl. ftp, http ...) szabályozásával.

4. A lokális hálózat eszközei

4.1. Hálózati csatoló (Network Interface Card)

A hálózati csatoló olyan hardvereszköz, mely a számítógépet a hálózati közegre (kábelre) csatlakoztatja. A csatoló rendszerint egy kártya, mely az alaplap ISA (a régebbi típusoknál) vagy PCI foglalatába kerül. Vannak olyan alaplapok, melyek integráltan tartalmazzák ezt az interfészt.
A mai korszerű csatolók már automatikusan konfigurálódnak, a „Plug and Play” eszközöket felismerő operációs rendszerek a kártyával történő bővítéskor vagy az operációs rendszer telepítésekor megteszik a szükséges beállításokat. Ha nem történik meg ez az automatizmus valamilyen okból, akkor direkt módon kell beállítani megszakításvonalukat (IRQ) és a be/kimenetei címüket (I/O address). Figyelnünk kell arra, hogy más eszközökkel ne legyen ütközés, azaz ezek az értékek ne legyenek ugyanazok.
A hálózati kártyák fontos jellemzője, hogy milyen hálózati közegre (kábelre) csatlakozhatnak. Vannak úgynevezett „combo” kártyák, melyek kétfajta jelátvivő csatlakozóval is rendelkeznek.



41. ábra
Combo hálózati csatoló

42. ábra
Koax(BNC) hálózati csatoló

43. ábra
UTP csatlakozós 100 Mb/s hálózati kártyák:

44. ábra
PCMCIA csatlakozós, notebookokba csatlakoztatható hálózati kártya

Napjaink egyik divatos, fejlődő technológiája a vezeték nélküli számítógép-hálózat. Ennek megvalósításához szükséges egy bázisállomás és a PC-kbe csatlakoztatható vezeték nélküli hálózati kártya.

45. ábra
Bázisállomás és a PC-kbe csatlakoztatható vezeték nélküli hálózati kártya

További információk: SMC, 3com, Intel

4.2. Kábelek

Az IEEE 802.3 specifikáció rögzíti a fizikai jellemzőket. A kábelek típusát megadó név három részből áll:

- az első a hálózat sebessége megabitben másodpercenként (Mbps),
- a középső az alapsávot (Base) vagy a széles sávot (Broad) jelöli,
- a harmadik a szegmens hosszát adja 100 méterben.

Néhány gyakrabban használt kábel adatai:

	Ethernet	IEEE 802.3			
		10Base5	10Base2	10BaseT	10BaseT4
Átviteli sebesség (Mbps)	10	10	10	10	10/100
Szegmens -hossz (m)	500	500	185	100 (UTP)	100 (UTP)
Közeg	50 ohm koaxiális (vastag)	50 ohm koaxiális (vastag)	50 ohm koaxiális (vékony)	Árnyékolatlan csavart érpár (UTP)	UTP CAT5 4 érpár
Topológia	busz	busz	busz	csillag	csillag

4.3. Jelátvivő (Transceiver)

A jelátvivő az a berendezés, mely közvetlenül az adott hálózati közegben továbbítható fizikai formára alakítja a neki átadott információs jeleket. Jelátvivőnek nevezzük azt az eszközt is, mely két típusú hálózati közeg közötti fizikai jelkonverziót végez; és a hálózati kártya azon részét is, mely a kívánt fajtájú kábelre illeszti a jeleket.

46. ábra
UTP/koax jelátvivő

A képen látható berendezés például koax-kábeles közeget és UTP csatlakozós csavart érpáras közeget tud összekötni.

4.4. Jelismétlő (Repeater)

A hálózati közegben a továbbított jelek gyengülnek, ezért csak az adott típusú közegre jellemző maximális távolságra vihetők át biztonságosan. Jelismétlő berendezés alkalmazásával az áthidalható távolságok megnövelhetők. A jelismétlő felismeri a fogadott jelet, és letisztítva (illetve újra előállítva) továbbküldi. Fontos kihangsúlyoznunk, hogy a felismert fizikai jelet nem értelmezi, csak egyszerűen „után” továbbít: amit az egyik oldalon hallott, azt elmondja a másik oldalon.
Azt gondolhatnánk, hogy ezzel a technikával akkor a hálózatépítésnek nincsenek kiterjedésbeli korlátjai, hiszen csak megfelelő számú jelismétlőt kell alkalmazni. A problémát az okozza, hogy a jelismétlő késleltetéssel dolgozik, és ha sok ilyen berendezésen kell az adatcsomagoknak átküzdeniük magukat, akkor a hálózati forgalom rendkívül lelassul.
A jelismétlő a digitális adattovábbításnál használatos, az analóg megfelelője az erősítő.
A jelismétlők az OSI-modell fizikai rétegéhez tartoznak.

47. ábra
Repeater (Jelismétlő)

4.5. Elosztó (HUB)

A HUB (angol szó, jelentése: középpont) a csillag-topológia középpontjának feladatát ellátó eszköz, melynek három típusa lézezik: passzív, aktív és intelligens.
A passzív HUB-ok egyszerű fizikai kapcsolatot valósítanak meg, az aktívak jelerősítést is végeznek, ezért a passzívtól eltérően külső áramforrást is igényelnek. Az intelligensek csomagkapcsolást vagy forgalomirányítást is végeznek az aktív HUB-ok szolgáltatásán túl.
Az elosztók az OSI-modell fizikai rétegéhez tartoznak.

48. ábra
HUB

4.6. Híd (Bridge)

A híd a hálózati szegmensek között biztosít adatforgalmat intelligens módon. Az érkező adatcsomagból kiolvassa a feladó és a címzett hardvercímét, és tárolt áthidalási táblázata segítségével eldönti, hogy továbbítsa-e a csomagot, vagy ne. Ha a küldő és a címzett ugyanabban a szegmensben van, akkor azt nem jeleníti meg a többi kimenetén (a többi szegmensben), ha különbözőben, akkor csak a címzett szegmensébe továbbítja. A híd ezzel a céltudatos tevékenységével csökkenti a fölösleges hálózati forgalmat, azonban csak a MAC-címekre (Media Access Control: közeghozzáférés vezérlési alréteg) vonatkozóan, mivel az **OSI adatkapcsolati rétegé**ben tevékenykedik. A magasabb szintű címek (ilyen az IP) átirányítására nem képes. A mindenkinek szóló üzenetszórásos (broadcast) forgalmat természetesen minden kimenetükön megjelenítik.

A híd – típusától függően – további szolgáltatásokat is végezhet: ilyen lehet a különböző típusú hálózati közeg (pl.: 10BaseT – 10Base2) összekapcsolása, a különböző hálózati metódus (pl.: Ethernet – Token Ring) közötti kapcsolat megteremtése. Ezek az úgynevezett tolmácshidak. Vannak tanulóhidak, amelyek – figyelve az adatforgalmat – automatikusan frissítik adattovábbítási táblázatukat.
A hidak viszonylag drága aktív elemei a hálózatoknak, további negatívumuk, hogy nagyobb késleltetési idővel dolgoznak, mint az egyszerű jelismétlők.

49. ábra
Bridge (híd)

4.7. Útválasztó (Router)

Az útválasztók a hidakhoz hasonló filozófiával működnek, csak éppen az OSI-modell hálózati rétegében. A csomagok hálózati címét vizsgálják és elvetik a csomagot, ha ugyanabba a szegmensbe van címezve, egyébként pedig csak arra a szegmensre küldik, amelyikben a címzett található. A döntést egy útválasztási táblázat alapján hozzák. Statikus útválasztókról beszélünk, ha e táblázatot a rendszergazdának kell elkészítenie, a dinamikus útválasztó ezt maga intézi.
Az útválasztók csak úgynevezett routolható protokollokat képesek kezelni (TCP/IP, IPX/SPX), melyek tartalmazzák a hálózati réteghez tartozó címzési információt. Nem routolható protokoll például a NetBEUI.

Az útválasztó lehet egy célberendezés, egy hardvereszköz, de útválasztó feladatot elláthat egy számítógép is, a rajta futó program segítségével. Ekkor annyi hálózati kártya szükséges a gépbe, ahány szegmens között kell közvetítenie. Az útválasztók intelligens módon figyelik a forgalmat, és torlódás vagy meghibásodás esetén képesek más útvonalat kijelölni a csomagoknak.
Az útválasztók különböző architektúrák, illetve eltérő hálózati közeghozzáférési metódust használó szegmensek között is meg tudják teremteni a kapcsolatot.
A fenti előnyökkel szemben azonban negatívumként felhozható, hogy az útválasztók lassabbak, mint a hidak, viszonylag drágák, és a dinamikus típusúak adatbázisuk frissítése miatt növelik a hálózati forgalmat.

50. ábra
Router (útválasztó)

4.8. Kapcsoló (Switch)

A kapcsolók az OSI-modell adatkapcsolati rétegben tevékenykednek, ezért speciális hidak halmazának tekinthetjük őket. Az egyik oldaluk egy közös belső szegmenshez, buszhoz; másik oldaluk pedig valamilyen hálózati csomóponthoz vagy egy további szegmenshez kapcsolódik. A kapcsolók működésének egyik módja az, amikor már ismertté vált a csomag címe: ilyenkor a csomag egyből továbbításra kerül. A másik esetben a teljes csomag begyűjtése után történik csak a cím dekódolása, amely alapján a továbbítás megtörténhet. Ez utóbbi esetben speciális szűrőfeltételek is megfogalmazhatók, továbbá virtuális hálózatok kialakítására is lehetőség nyílik. A kapcsolókat elláthatják útválasztó képességekkel is, így alkalmassá válnak – egy berendezés alkalmazásával – hálózati szintű címek alapján megvalósuló kapcsolásra is.

51. ábra
Switch (kapcsoló)

4.9. Átjáró (Gateway)

Az átjárók elsődlegesen az **alkalmazási réteg**ben működnek, de kaphatnak feladatot például a viszonyrétegben is.
Protokollok, illetve adatformátumok közötti konverziót hivatottak ellátni. Protokollok közötti fordítás történik akkor, ha a küldő például IPX/SPX-et használ, a fogadó pedig TCP/IP-t: ekkor az átjáró a csomag adatait átalakítja a célprotokoll keretformátumára. A híd is hasonlót csinál, de az nem alakítja át a keretformátumot, hanem becsomagolja az egyiket a másikba. Az átjárók különböző technológiát használó szegmensek közötti kapcsolat megvalósítását is szolgálják, például erőforrás-megosztást közvetíthethetnek egy IBM kompatibilis gépekből, illetve Apple Macintosh gépekből álló hálózat között.

Az átjáró további jellemző alkalmazási területe még a speciális levélformátumok általános formátumra való átalakítása.
Hátrányként viszont elmondható, hogy a protokollfordítás lassabb folyamat, mint a többi aktív hálózati elem adattovábbító tevékenysége, és komolyabb feladatot jelent az átjárók telepítése és konfigurálása is.

52. ábra
Gateway (átjáró)

4.10. Szünetmentes tápegység (UPS)

Ez a berendezés nem a hálózati közeghez illeszkedik közvetlenül (ellentétben a fenti berendezésekkel), és nem is kimondottan csak hálózat esetén lehet szerepe, hanem kliens-szerver hálózat esetén fokozott jelentőséggel bír. A kiszolgáló működési zavara megbéníthatja a teljes hálózati munkát, hirtelen leállása adatvesztést eredményezhet. Ilyen zavart okozhat az elektromos hálózat feszültségkimaradása, -ingadozása. A szervergépeket – központi szerepük miatt – indokolt ellátni szünetmentes áramforrásokkal. Az UPS-ek jellemző adata, hogy mennyi ideig képesek tovább működtetni a rájuk kapcsolt számítógépet (gépeket). Ez nyilván tárolókapacitásuk és a szerver fogyasztásának függvénye. Nem csak a teljes feszültségmegszűnés esetén lépnek működésbe, hanem feszültségszintesés esetén ki is pótolják azt. Az intelligens tápegységek a kiszolgálók kommunikációs portjára csatlakozva tájékoztatják a szervert állapotukról, ami alapján az értesítést küldhet a klienseknek. Ez általában olyan üzenet, hogy mennyi ideig kell a felhasználóknak beszüntetni a hálózati tevékenységet. Ha a szerver üzemét biztosító energia már vészesen fogy a tápegység akkumulátoraiból, akkor az UPS szerverre telepített programja leállítja a hálózati operációs programot és kikapcsolja a gépet.

53. ábra
UPS

5. Hálózati szabványok

A korábbi részekben megismerkedhettünk az ISO OSI hétrétegű referenciamodell ajánlásával. Ennek néhány rétegét szabványosították az IEEE (Institute of Electrical and Electronic Engineers – Villamos- és Elektronikai Mérnökök Szervezete) 802-es specifikációival 1980-as évek elején. Ez a kiterjesztés a fizikai és adatkapcsolati rétegben jelentkezik, ahol azt definiálja, hogyan férhet hozzá a hálózati közeghez egy időben több számítógép anélkül, hogy egymás zavarnák a kommunikációban. E specifikációk 12 kategóriát határoznak meg:

802.1	Internetworking: bevezetést nyújt a szabványhalmazba és meghatározza az alapegységeit	
802.2	LLC: Logical Link Control – logikai kapcsolatvezérlés	
802.3	CSMA/CD LAN	MAC: Media Access Control – közeghozzáférést vezérlő alréteg
802.4	Token Bus LAN	
802.5	Token Ring LAN	
802.6	MAN hálózatok	
802.7	Broadband Technical Advisory Group – szélessávú átvitel	
802.8	Fiber Optic Advisory Group – optikai átvitel	
802.9	Integrált hang- és adathálózatok	
802.10	Network Security Technical Advisory Group – hálózatok biztonsági kérdései	
802.11	Drótnélküli hálózatok	
802.12	Demand Priority Access LAN, 100VG-AnyLAN – alacsony szintű hozzáférés a LAN-okhoz	

Az ISO OSI-modell és az IEEE 802 szabvány kapcsolatát mutatja a következő ábra:

2. Adatkapcsolati réteg	LLC	IEEE 802.2		
	MAC	802.3	802.4	802.5
1. Fizikai réteg				

Mivel a szabvány elfogadásának idejében már több hálózati eszközöket gyártó cégnek is kiforrott megoldása volt a fizikai réteg megvalósítására, nem lehetett egyet kiválasztani közülük.
Ezt a problémát úgy oldották meg, hogy a gyártók hálózati kártyáihoz tartozó fizikai rétegeket elfogadták, és megoldották az adatkapcsolati réteg kimenetének szabványosítását. Így tetszőleges fizikai réteg fölött a további rétegek már azonosak lehetnek.

Az adatkapcsolati réteg kimenetének szabványosítását úgy tudták elérni, hogy két részre osztották magát a réteget: az MAC (Media Access Control - közeghozzáférést vezérlő) és az LLC (Logical Link Control – logikai kapcsolatvezérlés) részre.
Az MAC a fizikai réteggel, azaz a hálózati kártyák hardverével teremti meg a kapcsolatot.
Az LLC pedig a logikai kapcsolatot hozza létre a hálózati kártyák kimenete és a közös 8002.2-es szabvány között.

A Xerox hetvenes években kifejlesztett Ethernetje volt az alapja a 802.3-as specifikációnak. Az eltérés lényegében csak az, hogy a 802.3 tartalmazza a fizikai réteg leírását, de az adatkapcsolati réteg nem határozza meg az LLC protokollt, mert azt a 802.2 szabvány rögzíti.

6. Hálózati azonosítás

6.1. A számítógépek azonosítása

A különböző OSI-rétegekben különböző lehet ugyanannak a számítógépnek a neve, hasonlatosan ahhoz, hogy minket is máshogy szólít a villamoson az ellenőr, a szomszéd és a barátunk/barátnőnk. Ezen címek konverziójára különböző névfeloldási szabványok alakultak ki, hogy más rendszerekkel, különböző protokollokon képes legyen az adott gép kommunikálni.

6.1.1. Hardvercím (hardware address)

A fizikai rétegben az eszközök és programok az úgynevezett hardvercímet használják. Minden hálózati csatoló rendelkezik egy ilyen „beégetett” címmel (a kártya ROM-jában tárolt hexadecimális számmal), amely minden általa küldött, illetve neki címzett csomag fejlécében is szerepel.

6.1.2. NetBIOS név

A Windows operációs rendszerek az alkalmazási rétegben egy – a felhasználó által könnyen megjegyezhető – maximum 15 karakter hosszú nevet használnak. A TCP/IP protokollal kommunikáló gépeknek le kell fordítaniuk e NetBIOS nevet IP-címre, hogy fel tudják venni a kapcsolatot. Ezt a konverziót a WINS (Windows Internet Name Service – a Windows névfeloldó rendszere) vagy a DNS (Domain Name Service – Internet névfeloldó rendszere) programok végzik a Windows operációs rendszer alatt.

A WINS egy szerver/kliens szolgáltatás, amely lényege az, hogy a kliens gép indulásakor a kiszolgálótól egy – a NetBIOS nevéhez rendelt – ideiglenes IP-címet kap, amellyel már azonosítható az IP alapú hálózaton. A gépek tehát IP-címüket NetBIOS-nevük WINS-függvényértékként kapják.

A DNS szolgáltatás is gépnevet konvertál IP-címre, csak itt a DNS-ben (Domain Name System – domainnév-rendszer) szereplő teljes alakú tartománynevek (azaz FQDN-ek: Fully Qualified Domain Names) az átváltásra kerülő nevek.

6.1.3. FQDN név

Ezek a nevek „gépnév.tartománynév” alakúak, ahol a gépnév (hostID) egyértelműen azonosít egy gépet a tartományán belül (Windows NT esetén NetBIOS név). Egy tartománynévhez (netID) tartozó gépeket pedig egy tartománykiszolgáló adminisztrál. A tartománynév hierarchikus szerkezetű, az egymásba skatulyázott tartományok egyre szűkülő körét jobbról balra haladva, egymástól ponttal elválasztva adja meg.
(Pl. Anci néni gépének neve: „ancigepe.osztaly.vallalat.hu”)

6.1.4. TCP/IP-cím

Az ilyen címek ccc.ccc.ccc.ccc alakúak, ahol az egyes „c” betűk egy-egy decimális számjegyet jelölnek. Ez lényegében egy 32 bites számot jelent.
A TCP/IP protokollpáros napjaink leggyakrabban használt protokollja, ezért jellemzőit külön fejezetben ismertetjük részletesen.

6.1.5. DHCP

A kliens gépek „eldönthetik” (ez beállításuktól függ), hogy állandó IP-címmel kerülnek hálózati azonosításra, vagy igénybe veszik egy kiszolgáló DHCP (Dinamic Host Configuration Protocol – dinamikus IP-cím-kiosztás) szolgáltatását. A DHCP szerver egy meglévő (általában belső) címtartományból oszt ki IP-címeket a bejelentkező klienseknek. Így egy kliensnek a különböző bejelentkezések alatt eltérő lehet az IP-címe.

6.2. Felhasználók azonosítása

A hálózati folyamatok felhasználótól függő része egyfajta megszemélyesítéssel valósul meg, nevezetesen hogy ki generálta azt az OSI alkalmazási rétegében. Fordítva, csak az folytathat ilyen tevékenységet, aki egyfajta „logikai lét" állapotával rendelkezik a hálózaton, tehát „valaki". Persze e folyamatok nem föltétlen rögzítik elindítójuk azonosságát.
Ilyen hálózati „valakivé" az válik, aki rendelkezik érvényes hálózati névvel és általában hozzá tartozó jelszóval.

Szerver/kliens hálózaton ezeket az adatokat a szerver tárolja, és rajta keresztül lehet adminisztrálni. Többszerveres hálózaton a szerverek közti feladat- és tartományfelosztástól függ, hogy melyik feladata a felhasználókezelés.
Az egyenrangú hálózatok esetében nincs kitüntetett hálózati feladatot ellátó számítógép, így a hálózati azonosítást is az a gép intézi, melyről a bejelentkezés történik.
A felhasználó hálózati azonosításának egyik fő célja, hogy a folyamatok kiindulási és célobjektumait azonosítsák. Példa lehet erre a hálózati levelezés, melyben egy adott felhasználói azonosítóhoz rendelt tárhelyről a levél a cím szerinti azonosítóhoz rendelt tárhelyre kerül.
Másik nagyon fontos cél az, hogy felhasználói azonosítóhoz rendelten lehet szabályozni az elérhető hálózati tevékenységek körét, a hálózati jogokat.

A hálózati azonosítót szerver/kliens hálózatnál a rendszergazda biztosítja, melyhez ad egy átmeneti jelszót. Ezt a jelszót a felhasználó módosíthatja, vagy – beállítástól függően – kötelező megváltoztatnia az első bejelentkezésnél. A rendszergazda (az adott hálózati operációs rendszertől függően) beállíthat jelszóra vonatkozó megszorításokat:

- minimális hossz;
- a kis- és nagybetűk, számok egyidejű használata;
- időtartam, melyen belül a felhasználónak ismételten meg kell változtatni jelszavát;
- egyedi jelszóhasználat, melyben már használt jelszó ismételt használatára nincs lehetőség;
- grátisz bejelentkezés, melyben mennyi jelszómódosításra való figyelmeztetés után tiltódik le a felhasználói azonosító;
- kapcsolatok száma, mely megadja, hogy egy időben maximum hány helyről jelentkezhet be a felhasználó;
- lejárati dátum, mely megadja, meddig használható az azonosító;
- tiltás, azonosító (átmeneti) szüneteltetésére;
- hány egymást követő rossz jelszómegadás után tiltódjon le az azonosító;
- rossz jelszómegadás miatti azonosító-szüneteltetés azt az időt jelenti, amely után ismét rendelkezésre áll a rossz próbálkozás miatt letiltott azonosító.

Ki ismerheti a felhasználó jelszavát? Bárki, akinek megmondja. Alapvetően két ok lehet irányadó ebben a kérdésben: az egyik a – más által nem szabályozott – saját, egyéni érdeke, mely persze lehet a munkájával kapcsolatos, de lehet magánügy is.

Nézzünk erre egy példát: Munkaerő Pál programozó. A programfejlesztés során régebben megírt kódrészleteket is fel szokott használni. Külön-külön egyik sem nagy érték, de ez a kis gyűjtemény könnyíti, gyorsítja munkáját. A szerveren tárolja, hogy ne foglaljon helyet a munkaállomás-gépén. Mivel az ő jelszavával lehet csak hozzáférni, biztos lehet abban, hogy ellenőrzés nélkül használhatja bármelyik modult, mert annak idején alaposan tesztelte már őket. Vajon ilyen rendíthetetlen nyugalommal bízna tárolt kincsei érintetlenségében, ha a szemben ülő kolléga kisfia is tudná a hozzáférési jelszót, aki pedig suli után mindig beugrik egy kicsit nyüstölni a papa gépét?
Valljuk be, elő szokott fordulni, hogy a vállalati levelező rendszeren keresztül magántermészetű élménybeszámolót is kapunk. Sőt, nem is mindig ugyanaz a feladó. Ez a levélmennyiség azért nem sodorja a gazdasági csődeljárás felé a céget, de a takarító Nusi néninek túl sok pletykatémát adna, és ez ugye visszavetné munkájában.

A tevékenységünkkel kapcsolatos előírások, szabályzatok szolgáltatják a másik alapvető okot, ami miatt diszkréten kezeljük a felhasználói jelszavunkat. Itt már nem dönthetünk szabadon, hogy megbízunk-e valakiben, vagy nem. Munkajogilag felelősségre vonhatnak a jelszó kiadásáért.

Hol tároljuk a jelszavunkat? Ha csak nincs külön előírás, akkor legjobb, ha a fejünkben tartjuk. Ezért kell megjegyezhető jelszavat választani. Ne legyen azonban olyan kézenfekvő, amit mások könnyen kitalálhatnak. Ne legyen például a gyerekünk beceneve, születési évszámunk, feleségünk édesanyjának gyakran emlegetett megszólítása stb. Vannak rendszerek, melyek például a „titok” szót már meg sem engedik jelszóként választani. Persze, feledékenyek vagyunk, és memóriánkat cetlikkel bővítjük. A jelszót tartalmazó cetli elhelyezése csak a feladat áttranszformálása. Vannak olyan esetek, amikor például egy felhasználó jelszavával fontos adatok fölött diszponál. Kiesése, átmeneti elérhetetlensége adatvesztést, károkat okozna vállalatának. Ezért előírják számára jelszavának borítékolását, hogy szükség esetén az illetékesek hozzáférhessenek. Rendkívüli esetben történő jelszóhasználatról később – ha lehetséges – a tulajdonost tájékoztatják. Normál üzemben a felhasználó természetesen bármikor ellenőrizheti a jelszavát tartalmazó boríték sértetlenségét.
Mi a teendő, ha „elvész” a jelszó? A rendszergazda nem tud ennek utánanézni, de új jelszó megadásával tud segíteni. Az viszont már elég nagy probléma, ha a rendszergazda jelszava veszik el. Ilyenkor a kiosztott hozzáférési jogok szerinti teljes mentést kell végezni, majd egy alkalmas időben végre kell hajtani az újratelepítést.
A jelszó begépelésénél általában csak csillagok jelennek meg, ami csak abban segít, hogy hányadik karaktert vittük be. Ez nyilván azt szolgálja, hogy a monitorunkat figyelő ne tudja leolvasni jelszavunkat. A billentyűk leütését látva is megtudható jelszavunk, ezért körültekintően kell eljárnunk.

7. Felhasználói jogok

7.1. Egyedi felhasználó

A felhasználó jogai a hálózati tevékenységének lehetőségeit definiálja, bár sokszor ennek megfogalmazása éppen korlátozással történik. A teljesség igénye nélkül tekintsünk néhány jellemző hálózati jogot, melyek alkalmazása különböző hálózati operációs rendszereknél eltérhet:

- hetes periódussal meghatározható, hogy a nap mely szakában jelentkezhet be a felhasználó,
- milyen fizikai címmel rendelkező kliensekről jelentkezhet be a felhasználó (általában alapértelmezés, hogy mindről),
- milyen protokollokkal történhet a felhasználó bejelentkezése,
- a hálózathasználathoz számlázás-szolgáltatást rendelve, adott keret eléréséig használhatja a felhasználó a hálózatot, ezt túllépve letiltódik az azonosítója,
- a felhasználó rendelkezésére bocsátott (a fájlkiszolgálón lévő) lemezterület méretének korlátozása,
- könyvtár- és fájljogok.

Az egyes hálózati operációs rendszerekben eltérő jogrendszer működik, mégis valamilyen formában mindegyikben megtalálhatók a következő táblázatban felsoroltak:

A jogosultsági szint megnevezése	A jogosultsági szint magyarul	A jogosultsági szint jelentése
Supervisor Administrator	Minden jog	A hálózatban az összes létező jogot megkapja.
Read	Olvasás	Állomány megnyitása és tartalmának olvasása, programok futtatása.
Write	Írás	Állomány megnyitása és tartalmának változtatása.
Create	Készítés	Új fájl vagy részkönyvtár készítése.
Erase	Törlés	Könyvtárak, fájljaik, részkönyvtáraik törlése.
Modify	Módosítás	Név- vagy attribútummódosítás

7.2. Felhasználói csoportok

Több felhasználó egy időben történő kezelésére csoportokat hozhatunk létre, vagy használhatjuk az eleve létező, beépített csoportokat. A jogok beállítását így csak a csoportokra kell elvégezni, az egyes felhasználóknál (vagy akár egy csoportnál is) csak a csoporttagságot kell megadni.

8. Lokális hálózati operációs rendszerek

Eddig igyekeztünk általánosan jellemezni a lokális hálózatokat, kerülve az egyes hálózati operációs rendszerekre jellemző specialitásokat. Az alábbiakban megnevezzük a legjelentősebbeket, és áttekintjük néhány sajátos vonásukat.
(Az operációs rendszerekről részletesebben a „Hálózati felhasználói ismeretek” és a „Linux” tantárgyak keretében lesz szó.)

8.1. Microsoft hálózati operációs rendszerek

Munkaállomás-operációsrendszerek, melyek egyenrangú hálózat kialakítására alkalmasak:

- Windows’95
- Windows’98
- Windows ME
- Windows NT 4.0 Workstation
- Windows 2000 Professional

A **Windows NT 4.0 Server**t tartományi, mentés- és fájlkiszolgáló telepítési lehetőségekkel vállalati feladatokra használják. Ma már felváltotta a Windows 2000 szervercsalád.

A **Windows 2000 Server** a kis- és közepes vállalatok, alkalmazásfejlesztők, webszolgáltatók, munkacsoportok és vállalati részlegek ideális hálózati kiszolgálója. Integrált címtárszolgáltatása, az Active Directory egyszerűsíti a rendszerfelügyeletet, fokozza a biztonságot és segíti a felügyeleti műveletek elosztását, központosítja a felhasználók, csoportok, biztonsági beállítások és hálózati erőforrások felügyeletét.
A Windows 2000 Server széleskörű internetes szolgáltatásokat nyújt a legújabb webtechnológiákat alkalmazó vállalatok részére. A Windows 2000 Server a Windows NT Server 4.0-t váltotta fel a hálózati kiszolgálók sorában.

A **Windows 2000 Advanced Server**, a Windows NT Server 4.0, Enterprise Edition utódja, erős hálózati kiszolgáló operációs rendszer. A széleskörű fürtözési (clustering) lehetőség, a még jobb megbízhatóság és méretezhetőség, a nyolcszoros SMP, valamint az elérhető 8 GB fizikai memória a nagyvállalati alkalmazások igényeinek felel meg.

A **Windows 2000 Datacenter Server** a Windows 2000 szervercsalád csúcsmodellje. Optimális eszköz nagy adattárolók, tudományos és mérnöki feladatok, szimulációk, online tranzakciók, kiszolgálóegyesítési projektek, nagy internet- és webszolgáltatók számára. A Windows 2000 Datacenter Servert több mint 10 000 felhasználó egyidejű kiszolgálására tervezték, mindamellett a legjobb ár-teljesítmény viszonyt éri el a tranzakciós folyamatok feldolgozásában. Tudása túlmutat a lokális hálózatok igényein.

További információk:
A Windows család legújabb tagjai

8.2. NetWare hálózatok

A **Novell Kisvállalati Csomag** az 50 vagy kevesebb felhasználós hálózatok legmegbízhatóbb és legkifizetődőbb megoldása. Az egyszerűen kezelhető, telepíthető és felügyelhető rendszer a hagyományos NetWare funkciókon kívül tartalmaz csoportmunka-támogatást (GroupWise), teljes körű internetkapcsolatot (MultiProtocol Router), távoli munkavégzés-lehetőséget és még sok hasznos programot, pl. faxszerver, vírusellenőrző, Windows menedzsment-funkciók (Z.E.N.works Starter Pack).

A **NetWare 6** a NetWare első olyan verziója, amely tökéletesen megvalósítja a Novell one Net (egységes Novell-hálózat) jövőképét és kiváló példája annak, milyen irányba fejlődnek a Novell hálózati szolgáltatásai. Ez az első alkalom, hogy a NetWare maga is egy sor hálózati szolgáltatás együttese, amelyet mindenki szabadon használhat, operációs rendszertől vagy helyszíntől függetlenül. Ma a cégeknek és intézményeknek arra van szükségük, hogy többféle, különálló hálózataikat egy egységes hálózattá (one Net) legyenek képesek integrálni, és a NetWare 6 segít ennek kialakításában. A NetWare 6 használatával az információ bármikor, bárhonnan elérhető a különféle hálózati platformokról, asztali operációs rendszerekről és vezeték nélküli eszközökről.

A **NetWare Cluster Services** (NCS) lehetővé teszi, hogy különálló hálózati szerverek együttműködhessenek a célból, hogy hozzáférést biztosítsanak a felhasználók számára a kritikus fontosságú hálózati erőforrásokhoz, az adatokhoz, az alkalmazásokhoz és más szolgáltatásokhoz. Ha a cluster egy hálózati szervere (csomópontja) meg is hibásodik, a cluster egy másik csomópontja automatikusan átveszi a meghibásodott csomópont által biztosított erőforrásokkal és szolgáltatásokkal kapcsolatos teendőket, így a clusterbe kötött erőforrások rendkívül magas szintű rendelkezésre állása biztosítható.
Az NCS valódi, többcsomópontos clustermegoldás, amelyik clusterenként 32 csomópont kezelésére képes. Ennek eredményeképpen az NCS nemcsak hogy magasabb szintű rendelkezésre állást képes biztosítani, mint más Intel-alapú clusterrendszerek, hanem mivel képes a meghibásodott csomópont erőforrásainak szétosztására több csomópont között, a hálózat működése lényeges teljesítménycsökkenés nélkül folytatódik.

A **Novell StandbyServer** és a **StandbyServer Many-to-One** (MTO) két, a lehető legnagyobb rendelkezésre állást biztosító szoftvermegoldás. Az első termék esetében egy készenléti (másodlagos) szerver egy, a második esetében több elsődleges szerver állapotát figyeli. Az elsődleges szerver(ek) bármilyen hardver- vagy szoftver-meghibásodása esetén a készenléti szerver automatikusan átveszi annak feladatait. Kifejezetten olyan cégek számára tervezte a Novell a StandbyServert, ahol a létfontosságú adatokat védeni kell, ugyanakkor elengedhetetlen a folyamatos rendelkezésre állás.
A StandbyServer teljességgel redundáns rendszer, az MTO készenléti szervere pedig maximum 20 elsődleges szerver őrzésére képes; ezek adatait egy külön lemezmeghajtóra menti.
Az inaktívvá vált elsődleges szerver(ek) a hálózat további működése közben vizsgálhatók és javíthatók; sőt, a rendszergazda a tervszerű karbantartás céljából szándékosan is lekapcsolhatja az elsődleges szerver(eke)t – a hálózat mindeközben továbbra is működik. Az MTO nem követeli meg, hogy a készenléti szerverek tökéletesen megegyezzenek a tükrözött elsődleges szerverekkel – sőt, az sem feltétlenül szükséges, hogy közel legyen hozzájuk. A készenléti szerver ezenfelül használható teljes funkcionalitású NetWare-szerverként – például faxszerverként vagy CD-szerverként –, miközben ellátja a készenléti funkciókat is.

További információk:
Novell termékek

8.3. Unix-hálózatok

A **Sun-rendszerek** alapköve, a **Solaris 8** operációs rendszer képessé teszi használója IT szervezetét arra, hogy nagyteljesítményű, folyamatos, valós idejű számítógépes hátteret nyújtson. A Sun felismerte, hogy a vállalati hálózatok kezelése a felhasználók egyik legfontosabb megoldandó feladatát jelenti. Ennek eredményeképpen született meg a Solstice rendszermenedzsment-eszközcsalád, melynek segítségével a vállalati hálózatok maximális hatékonysággal működtethetők. A Solstice révén a Sun és más rendszerek egyszerűen kezelhetők, konfigurálhatók a hálózaton, megkönnyítve ezáltal az eszközkezelést és a biztonsági ellenőrzést.
További információk:
www.sun.hu

A **LINUX** az Internetnek köszönheti létét, nagyhálózatos feladatokra készítették. A fejlesztése a világhálón keresztül kapcsolatot tartó „önkéntes” programozók munkájával történt, és folyik tovább. Az egyes fejlesztési változatok különböző disztribúciós gondozásban jelennek meg (FreeBSD, OpenBSD, NetBSD, Slackware, OpenLinux, SuSe, Redhat stb.).

A Linux több program egy időben történő futtatására képes, többfelhasználós, TCP/IP alapú operációs rendszer: lokális hálózatban megfogalmazódó feladatok bármelyikét képes ellátni, jól skálázható (azaz teljesítménye, az erőforrásmegosztása jól beállítható). Lehet munkaállomás vagy kiszolgáló operációs rendszere. Kliensként a „legkisebb" gépen is képes futni, szerverként az adott feladatra hangolva telepítik. Előszeretettel használják lokális hálózatok külső kapcsolattartását biztosító szerverén (proxy szerver, tűzfal, levelező szerver ...), de adatbázisszerver-szolgáltatóként is népszerű.
Különös jellemzője, hogy ingyenes szoftver, különböző web-helyekről szabadon letölthető.

Linux-letöltési címek:
sunsite.unc.edu
tsx-11.mit.edu
nic.funet.fi

9. A lokális hálózat védelme

Manapság szinte minden munkahely rendelkezik helyi hálózattal, sőt ezek a helyi hálózatok általában kapcsolódnak az Internethez is. Ez rendkívül sok előnyt jelent, azonban biztonsági problémákat is felvet. Ki ne hallott volna arról, hogy betörtek erre vagy arra a szerverre, és jobb esetben csak a weblapot írták át, rosszabb esetben letöröltek fontos adatokat, vagy felhasználói listákat szereztek meg. E külső vagy belső támadások ellen fel kell készíteni a hálózatunkat.

Több megoldás létezik. Beszélhetünk hardveres és szoftveres védelemről egyaránt. Hardveres védelem esetében egy célberendezés vigyáz a biztonságunkra, szoftveres védelem esetében az egyik szerveren futó program igyekszik megtenni ugyanezt.
Nézzünk most ezek közül néhányat, a teljesség igénye nélkül.

9.1. Tűzfal

Korszerű, nagy biztonságú, felügyelt rendszer a kívülről történő behatolások megakadályozására és a belülről való kijutás szabályozására.
Minden, ami az Internetről a helyi hálózatba bejut, illetve onnan az Internetre kikerül, át kell haladjon a tűzfalon,a folyamatok tehát jól kézben tarthatók.
Így a vállalat összes bejövő és kimenő levele áthalad a tűzfalon, ezek naplózásra kerülnek, valamint esetleges veszélyes csatolmányaik ellenőrizhetők, melyek küldése és fogadása bizonyos esetekben akár meg is tagadható.

A routerek nagy része számos tűzfal-funkciót képes ellátni. A tűzfal egyik típusa az ún. "külső tűzfal" a teljes helyi hálózatot részben izolálja az Internettől, míg az ún. "belső tűzfal" a helyi hálózat egy különösen védendő részét zárja el annak többi részétől (és így az Internettől is).

A tűzfal használata titkos, érzékeny adatok védelme vagy nagy üzembiztonságot kívánó hálózatok esetén elengedhetetlen.
A tűzfal szerepét játszhatja egy intelligens router, megfelelő konfigurációjú Unix gép, vagy ezek közül akár több is. Internet-tűzfalnak egy PC-n futó szoftver is kiválóan megfelelhet.

9.2. Vírusfigyelő rendszerek

A külső hálózatról vagy közvetlen a helyi gépekről is kerülhet ártó szándékú szoftver a lokális hálózatba.
Megfelelő hálózati szoftverrel védhetjük a szerver- és kliensgépeket egyaránt.
A több maggal működő víruskereső sokkal hatékonyabban dolgozik, mint ha különböző magú víruskeresőket futtatnánk külön-külön. Mivel egyetlen keretrendszerbe vannak integrálva, nincs inkompatibilitási probléma, és a keresési idő is rövidebb, mint pl. három különböző vírusirtó szoftver párhuzamos használatakor.
A korszerű vírusfigyelők talán legszembetűnőbb előnye a hálózati funkciók gazdagsága. Modern, háromrétegű hálózati architektúrát képez: az adminisztrátor saját gépéről, akár notebookjáról felügyeli a teljes hálózat víruseseményeit a menedzser-modulon keresztül. Az adminisztrátor utasításait, beállításait a második réteg, a Policy Manager Server továbbítja a felhasználók felé. Emellett pedig minden gépen fut a Management Agent komponens, amely a víruskereső beállításaiért és a hálózati kommunikációért felelős. Ezen a rendszeren keresztül kiterjedt, több ezer számítógépből álló hálózatok is hatékonyan menedzselhetők, akár WAN-kapcsolatokon keresztül is.

Miután az adminisztrátor teljes kontrollt kap a hálózat összes gépén futó víruskereső felett, a felhasználók elől szinte száz százalékosan el lehet rejteni a kereső jelenlétét. Mindössze a tálcán található ikon jelzi a felhasználóknak, hogy gépük folyamatosan „biztonságban” van.
A vírusfigyelő valós időben végzi a vírusok felkutatását. Minden elindított programot, betöltött dokumentumot ellenőriz, de figyel az Internetről letöltött állományokra vagy a gépbe helyezett floppylemezekre is. A tömörített állományokba (ZIP, ARJ, RAR, LZH, …) is belelát, automatikus eltávolító képessége révén pedig a felfedezéskor azonnal képes a fertőzött állományok megtisztítására.

Összefoglalás

Az előző fejezetekben megtanulhattuk a számítógép-hálózatok legfontosabb alapfogalmait, valamint megismerhettük a legtöbb hálózat elméleti alapjának tekinthető ISO OSI-modellt. Ezek után ebben a fejezetben megvizsgáltuk a hálózatok közül a legelterjedtebbek – a lokális hálózatok – elemeit a gyakorlatban is.
A hálózatok két nagy típusát különböztettük meg: a peer to peer és kliens-szerver típusú hálózatokat. Végigvezettük a két típus előnyeit és hátrányait, azt a következtetést leszűrve, hogy a nagy hálózatokban (több mint 10 számítógép) a kliens-szerver típust érdemes alkalmazni.

Mivel a hálózatok többsége a kliens-szerver típusba tartozik, áttekintettük a legelterjedtebb szerverszolgáltatásokat.
Típustól függetlenül minden hálózatban vannak passzív és aktív elemek. A passzív elemek közül az adatátviteli közegekről az előző fejezetben szóltunk. Most az aktív elemeket vettük sorra, a legegyszerűbb hálózati illesztőkártyától indulva a switchen keresztül a gatewayig.
Fontos kérdés a hálózati szabványok kérdése, mellyel egy külön fejezetrész foglalkozik.
Végül, de nem utolsó sorban rövid áttekintést kaptunk a fejezet végén a legelterjedtebb hálózati operációs rendszerekről.

Ellenőrző kérdések

1. Kiterjedés alapján hogyan hívjuk a lokális hálózatokat?
   a. LAN
   b. MAN
   c. WAN
   d. GAN

   A megoldás bővebben itt

2. Hol előnyös a peer to peer hálózat alkalmazása?
   a. egy 200 helyiséges irodaházban
   b. egy 5 gépet működtető irodában
   c. egy otthoni „hálózatban” a papa és a kisöcsi gépének összekötésére
   d. egy atomreaktorban

   A megoldás bővebben itt

3. Hol előnyös a kliens-szerver típusú hálózat alkalmazása?
   a. egy 200 irodás irodaházban
   b. egy 5 gépet működtető irodában
   c. egy otthoni „hálózatba” a papa és a kisöcsi gépének összekötésére
   d. egy atomreaktorban

   A megoldás bővebben itt

4. Mi a fájlkiszolgáló?
   a. kérésre elküldi a fájlokat a felhasználónak
   b. a háttértárolóján lévő fájlokhoz jogosultság alapján hozzáférést biztosít
   c. kiszolgálja a fájlok kéréseit

   A megoldás bővebben itt

5. Melyek szerverszolgáltatásokaz alábbiak közül? (Több megoldás is lehetséges.)
   a. nyomtatószerver
   b. UPS
   c. proxy szerver
   d. hálózati hozzáférés

   A megoldás bővebben itt

6. Mi a különbség a HUB és a switch között?
   a. semmi
   b. a HUB nem erősít, a switch igen
   c. a HUB minden portjára elküldi a beérkező jelet, a switch külön-külön tudja címezni a portjait

   A megoldás bővebben itt

7. Az IEEE 802.2 szabvány fizikai adatkapcsolati rétege a/az
   a. LLC
   b. MAC
   c. MPC

   A megoldás bővebben itt

8. Mi ellen kell védeni a hálózatokat?
   a. vírusok
   b. behatolók
   c. tűzkár
   d. földrengés ellen

   A megoldás bővebben itt

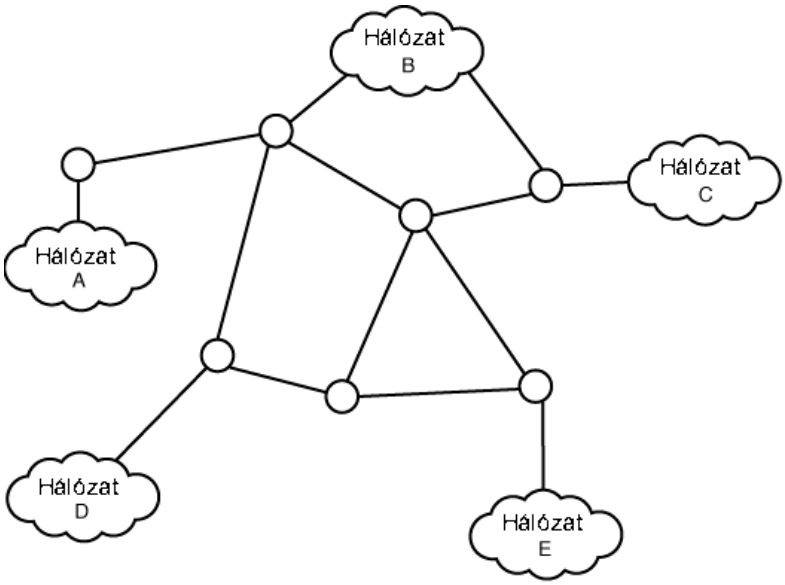
VIII. A TCP/IP protokoll és az Internet

Bevezető

A fejezet célja, hogy megismertesse az olvasót az Internet kialakulásával, fontosabb jellemzőivel és szolgáltatásaival. Ez a fejezet szervesen kapcsolódik az előzőekhez, mivel a korábbi ismeretekre épít, ugyanakkor egy kicsit más az előző fejezetekhez képest, mivel csak egy hálózattal, az Internettel foglalkozik. Erre azért van szükség, mert napjainkban szinte mindenki kapcsolatba kerül az Internettel, így fontos, hogy tisztában legyen a lehetőségekkel. A hálózat adta lehetőségek megismeréséhez szervesen hozzá tartozik az elméleti háttér ismerete is.
Így ebben a fejezetben az Internet alapját képező TCP/IP protokoll páros keresztül megismerkedünk a fontosabb jellemzőkkel, szolgáltatásokkal.

1. A TCP/IP protokoll

A TCP/IP protokollkészlet a különböző operációs rendszerekkel működő számítógépek, illetve számítógép-hálózatok közötti kapcsolat létrehozására szolgál. E protokollkészletet arra dolgozták ki, hogy hálózatba kapcsolt számítógépek megoszthassák egymás között az erőforrásaikat. A TCP és az IP a legismertebb, ezért az egész családra a TCP/IP kifejezést használják. Segítségével különálló számítógép-hálózatok hierarchiája alakítható ki, ahol az egyes gépeket, illetve helyi hálózatokat nagy távolságú vonalak kötik össze.



54. ábra
Helyi hálózatok összekötése nagytávolságú vonalakkal

A protokoll-készletet először az USA Védelmi Minisztériumának (Department of Defense, DoD) DARPA bizottsága definiálta az ARPANET projekt keretében, 1969-ben. A protokollkészlet (a

későbbiekben egyszerűen csak protokollnak is nevezzük) a 70-es években került az USA felsőoktatási intézményeihez, ahol a 80-as években az intézmények közötti kapcsolattartás fő eszközévé vált. Ettől kezdve kezdték az e protokollal működő hálózatot Internetnek nevezni. Ma már az Internet kisebb kiterjedésű lokális számítógépes hálózatok (LAN-ok) összekapcsolásából álló globális számítógépes rendszer.

A TCP/IP protokollkészlet egymásra épülő rétegekből áll; nem követi az OSI hétrétegű felépítését. Alapvetően a referenciamodell két rétegének funkcióját valósítja meg: ezek a hálózati és a szállítási réteg. A TCP/IP protokollkészletre épülő hálózati modell négy rétegből áll, melyet az alábbi ábra mutat.

55. ábra
A TCP/IP protokollok az OSI réteg-modell viszonya

Lássunk egy példát a fentiekre: tipikus hálózati feladat a levelezés megvalósítása, amit protokoll (SMTP-Simple Mail Transfer Protocol – egyszerű, levéltovábbító protokoll) szabályoz. A protokoll az egyik gép által a másiknak küldendő parancsokat definiálja, például annak meghatározására, hogy ki a levél küldője, ki a címzett, majd ezután következik a levél szövege. A protokoll feltételezi továbbá, hogy a kérdéses két számítógép között megbízható kommunikációs csatorna létezik.

A levelezés, mint bármely más alkalmazási rétegbeli protokoll, a küldendő parancsokat és üzeneteket definiálja. A tervezésekor a TCP/IP-t vették alapul, azaz azzal együtt használható. A TCP a felelős azért, hogy a parancsok biztosan elkerüljenek a címzetthez, figyel arra, hogy mi került át, és ami nem jutott el a címzetthez, azt újraadja. Amennyiben egy rész – pl. az üzenet szövege – túl nagy lenne (meghaladja egy datagram, vagyis az egy üzenetben küldhető csomag méretét), akkor azt a TCP széttördeli több datagramra, és biztosítja,

hogy azok helyesen érkezzenek célba. Mivel a fenti szolgáltatásokat jó néhány alkalmazás igényli, ezeket nem a levelezés, hanem egy külön protokoll tartalmazza. Az egész TCP tulajdonképpen nem más, mint rutinok olyan gyűjteménye, amelyet a különböző alkalmazások vesznek igénybe, hogy megbízható hálózati kapcsolatot építsenek ki más számítógépekkel.

A TCP szolgáltatásait sok alkalmazás igényli, azonban vannak olyanok, amelyeknek nincs rájuk szükségük. Persze léteznek olyan szolgáltatások, amelyeket minden alkalmazás megkíván; ezeket szedték egybe az IP-be. Ugyanúgy, ahogy a TCP, az IP is egy rutingyűjtemény, de ezt a TCP-t nem használó alkalmazások is elérhetik.

A különböző protokolloknak ezt a szintekbe rendezését rétegezésnek nevezik. Ennek megfelelően az alkalmazási programok (mint például a levelezés), a TCP, illetve az IP külön réteget alkotnak, amelyek mindegyike az alatta lévő réteg szolgáltatásait használja. A TCP/IP alkalmazások általában a következő négy réteget veszik igénybe:

- alkalmazási protokollok (pl. levelezés);
- a TCP-hez hasonló protokollok, amelyek rengeteg alkalmazás számára biztosítanak szolgáltatásokat;
- IP, amely a datagramok célba juttatását biztosítja;
- a felhasznált fizikai eszközök kezeléséhez szükséges protokollok (pl. Ethernet).

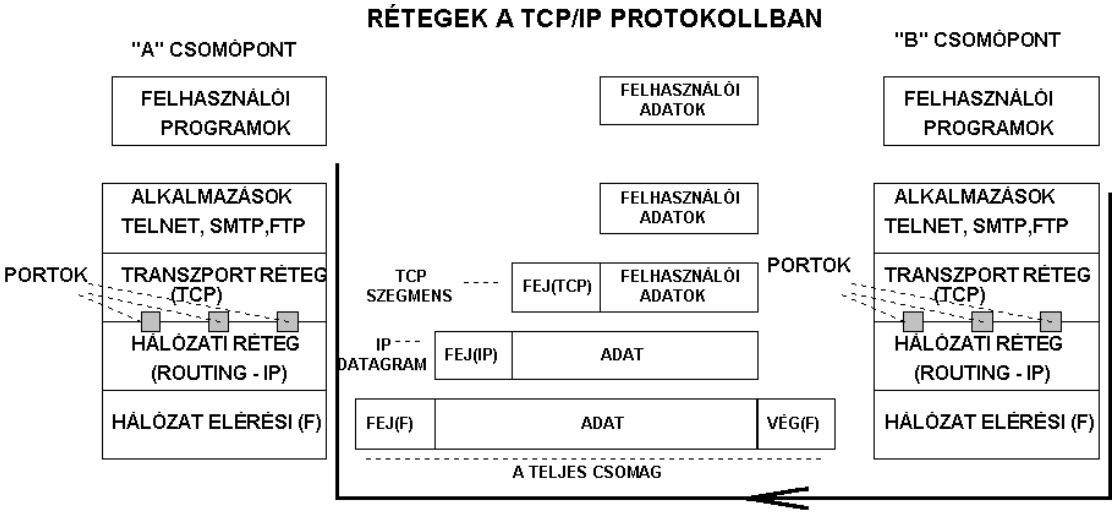
Az alapfeltevés az, hogy nagyszámú különböző hálózat áll egymással összeköttetésben átjárók (gateway) segítségével. Ezeken a hálózatokon lévő bármely számítógépet vagy erőforrást a felhasználónak el kell tudnia érni. Az adatcsomagok esetleg több tucat hálózaton is keresztülmehetnek mielőtt a célállomásra érkeznének. Az ezt megvalósító útvonalválasztásnak természetesen láthatatlannak kell maradnia a felhasználó számára, abból ő mindössze egy internetcímet kell, hogy ismerjen. Ez egy olyan számnégyes, mint például a 128.6.4.194, amely tulajdonképpen egy 32 bites számot reprezentál. A felírás 4 darab 8 bites decimális szám formájában történik.

A TCP/IP összeköttetés-mentes hálózati protokollokat tartalmaz, ami azt jelenti, hogy az információ a datagramok sorozataként terjed tovább. A datagram adatok együttese, amely egy egyszerű üzenetként kerül továbbításra. A datagramok egymástól függetlenül, egyesével indulnak útjukra. (Az adott adatkapcsolat időtartamára vonatkozóan persze vannak előrejelzések.) A küldendő információt a protokollok a fenti adatokra tördelik, amelyeket aztán a hálózat egymástól teljesen különállóként kezel.

Tegyük fel például, hogy egy 15.000 bájt méretű állomány továbbításáról van szó. Mivel a legtöbb hálózat nem tud ekkora datagrammal mit kezdeni, a protokollok ezt 30 darab 500 bájtos darabra szedik szét, amelyek mindegyikét elküldik a célállomásra, ahol összerakják belőlük az eredeti, 15 000 bájtos állományt. A datagramok

adása közben a hálózaton semmi nem utal arra, hogy közöttük bármiféle kapcsolat is létezne; előfordulhat, hogy egy a sorrendben eredetileg hátrább álló megelőz egy előtte állót. Az is lehetséges, hogy a hálózaton valahol hiba keletkezik és néhányuk nem érkezik meg a rendeltetési helyére. Ilyenkor újra kell adni a hiányzó datagramot.

A datagram és a csomag kifejezés gyakran egymással felcserélhetőnek tűnik, azonban ez nem minden esetben van így. A TCP/IP leírásakor a datagram a helyes kifejezés: azt az adategységet jelöli, amellyel a protokollok operálnak, míg a csomag egy fizikailag létező dolog, amely a kábeleken jelenik meg. A legtöbb esetben egy csomag egyetlen datagramot tartalmaz.



56. ábra
Az információáramlás

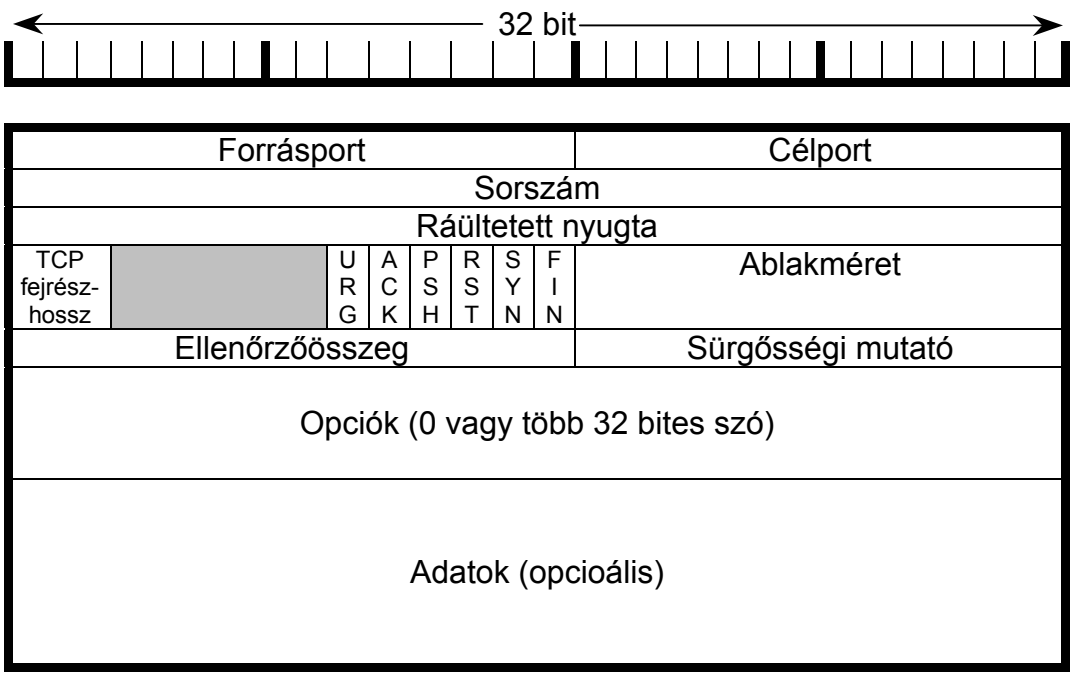
1.1. Az Internet szállítási rétege: a TCP

A TCP/IP datagramok kezelésében két különböző protokoll játszik szerepet. Az üzenetek széttördelését, összeállítását, az elveszett részek újraadását, a datagramok helyes sorrendjének visszaállítását mind a TCP (transmission control protocol – átvitelvezérlési protokoll) végzi. Az egyes datagramok útvonalának a meghatározását (routing) az IP (internet protocol) hajtja végre. Mindez azt a látszatot kelti, hogy a munka tetemes része a TCP-re hárul. Kis kiterjedésű hálózatokban ez így is van, azonban az Interneten egy datagramnak a rendeltetési helyre való juttatása igen összetett feladatot jelenthet. Egy datagram több hálózaton mehet keresztül, míg végül eljut a célállomásra. A különböző átviteli közegekből adódó inkompatibilitások kezelése és a célállomásokhoz vezető útvonalak végigkövetése komplex feladat. (Előfordulhat, hogy egy Ethernet-hálózatból kell eljuttatni egy datagramot egy másik Ethernet-hálózatba, csakhogy ez a másik hálózat a tengeren túl van. Ekkor biztos, hogy más közegeket is igénybe kell venni, például ISDN, műholdas stb. átviteli közegeket.)
Meg kell jegyezni azonban, hogy a TCP és az IP közti interfész rendkívül egyszerű: a TCP egy datagramot ad át az IP-nek egy

rendeltetési címmel együtt. Az IP semmit sem tud arról, hogy ez az információ hogyan viszonyul más datagramokhoz.

Nyilván nem elegendő csupán a datagramnak a helyes címre való továbbítása. A TCP-nek még azt is tudnia kell, hogy az adott datagram melyik kapcsolathoz tartozik. A probléma megoldását a demultiplexálás v. nyalábbontás néven ismert eljárás adja, amely a TCP/IP-ben valójában több különböző szinten folyik. A demultiplexáláshoz szükséges információt a fejlécek hordozzák. A fejléc azokat a kiegészítő információkat jelenti, amelyeket a különböző protokollok ragasztanak a datagramok elejére, hogy azokat nyomon tudják követni. A dolog hasonlít ahhoz, amikor a levelet a borítékba tesszük, majd azt megcímezzük. A különbség annyi, hogy a modern hálózatokban ez jóval többször történik: úgy, mintha a levelet egy kis borítékba tennénk, majd azt a titkárnőnk egy nagyobb borítékba helyezné, amit a központ egy még nagyobb borítékban továbbítana stb.

Az alábbiakban a tipikus TCP/IP hálózaton keresztülhaladó üzenetre rárakódó fejléceket tekintjük át. A TCP csomagformátumot a következő ábra mutatja.



57. ábra
A TCP csomagformátum

Minden datagram elé egy TCP fejléc kerül, amely legalább 20 bájtból áll. Ezek közül a legfontosabbak: egy **forrás-** és egy **célport**, valamint egy sorszám. A portok az összeköttetések végpontjait azonosítják. Tegyük fel például, hogy egyszerre 3 felhasználó továbbít állományokat. A TCP ezekhez az átvitelekhez az 1000, 1001 és 1002 portokat rendelheti. Datagram küldésekor a kiválasztott port válik a forrásporttá, mivel innen indul ki a datagram. A kapcsolat másik végénél lévő TCP szintén hozzárendeli a saját portját az átvitelhez.

A küldőoldali TCP-nek a célport számát is tudnia kell (ezt az információt a kapcsolat felépülésekor szerzi meg), amelyet az a fejléc célport mezőjébe helyez. Ha a másik oldalról érkezik egy datagram, akkor annak TCP fejlécében a forrás- és a célportok tartalma ellentétes, hiszen ekkor az a forrás, ez pedig a rendeltetési hely.

Minden datagramnak van egy **sorszáma**, amely a vevőoldalt arról biztosítja, hogy minden adatot helyes sorrendben kapjon meg, és ne veszítsen el egyet se a datagramok közül. A TCP valójában nem a datagramokat, hanem a bájtokat sorszámozza. Ha például minden datagram 500 bájt adatot tartalmaz, akkor az első datagram sorszáma 0, a másodiké 500, a következőé 1000, az az utánié 1500 stb... lesz.

A datagramnak a rendeltetési helyre való megérkezését a vevő egy nyugtával hozza a küldő oldal tudomására. Ez a szám a datagram TCP fejlécében a „**Ráültetett nyugta**” mezőben jelenik meg. Például egy olyan csomag elküldése, amelynek nyugtamezőjében 1500 szerepel, azt jelenti, hogy az 1500-as bájtig bezárólag minden datagram eljutott a rendeltetési helyre. Amennyiben a küldőoldal egy adott időn belül nem kap nyugtát, akkor újból elküldi az adatot.
A **TCP fejrészhossz** mezőben van tárolva, hogy hány 32 bites szóból áll a TCP fejrész. Erre azért van szükség, mert az „Opciók” mező hossza nem mindig állandó, így a vevő nem tudhatja pontosan a fejrész hosszát.
Ezután egy használaton kívüli 6 bites mező következik.
Ezt 6 egybites mező követi:

- az **URG** mező értéke 1, ha használja a „Sürgősségi mutatót”
- az **ACK** mező értéke 1, ha a „Ráültetett nyugta” érvényes, azaz a keret használja a nyugta mezőjét
- a **PSH** mező értéke 1, ha a vevőnek nem lehet pufferben tárolnia az adatokat, hanem késedelem nélkül kell továbbítania
- az **RST** mező értéke 1, ha valamilyen probléma lépett fel az adásban
- a **SYN** mező összeköttetés kezdeményezésénél használt bit (SYN=1 és ACK=0 jelzi az összeköttetés kezdetét)
- a **FIN** mező értéke 1, ha vége az összeköttetésnek.

Az „**Ablakméret**” mezőben lévő érték az összeköttetés alatt forgalomban lévő adatok mennyiségét határozza meg. Nem lenne szerencsés, ha minden egyes datagram elküldése előtt meg kellene várni az előző nyugtáját, mert így a forgalom rendkívüli mértékben lelassulna. Másrészt viszont nem lehet folytonosan küldeni az adatokat, hiszen például egy gyorsabb számítógép adatárama elárasztaná a lassabb gépeket. Ennek megoldására mindkét oldal az Ablakméret mezőben elhelyezett bájtok számával közli, hogy éppen mekkora adatmennyiséget képes még befogadni. Az adatok vételével ez a szám, azaz az ablak mérete, folyamatosan csökken. Amikor eléri a nullát a küldőnek szüneteltetnie kell az adatok továbbítását. A vevő ablakmérete az adatok feldolgozása során nő, ami jelzi, hogy kész további adatok fogadására.

Az **Ellenőrzőösszeg** olyan szám, amelyet a datagramban lévő bájtok összeadásával kapunk. Az eredmény aztán bekerül a TCP fejlécbe. A vevő oldali TCP is kiszámítja a fentiek szerint az ellenőrzőösszeget. Ha a kettő nem egyezik, akkor a datagrammal az átvitel közben valahol valami baj történt, és azt a protokoll eldobja.

Gyakran ugyanaz a datagram használható az újabb adatok engedélyezésére és nyugtázásra is (aktualizált ablak segítségével). A **Sürgősségi mutató** mezőben lévő érték beállításával bármelyik oldal utasíthatja a másikat arra, hogy a feldolgozást egy adott bájttal folytassa. A gyakorlatban többek között ez az aszinkron eseményekkel kapcsolatban használatos, például amikor vezérlőkarakter vagy más, a kimenetet megszakító parancs kerül továbbításra.

Az **Opciók** mezőt a szabályos fejrészben nem szereplő lehetőségek megvalósítására tervezték. A legfontosabb lehetőség, hogy a hosztok meghatározhatják a TCP adatmező maximális hosszát. Ezt az összeköttetés sebességének és megbízhatóságának függvényében választhatják meg.

1.2. Az UDP protokoll

A fentiekből kiderül, hogy a TCP keretforma eléggé bonyolult. Előfordul olyan eset, amikor ez a keretforma felesleges. Elegendő egy jóval egyszerűbb keretforma is. Ilyen eset pl. a névfeloldási kérelem, mikor egy internetböngészőbe beírunk egy webcímet, azt át kell alakítania IP-címmé. Ha a kérésre rövid időn belül nem jön válasz, a kérést elveszettnek tekintjük, és megismételjük.
Ilyen egyszerű protokoll az UDP (user datagram protocol – felhasználói datagram-protokoll). A következő ábra szemlélteti a felépítését:

Forrásport	Célport
UDP szegmens hossza	UDP ellenőrző összeg

58. ábra
Az UDP felépítése

Az UDP a forrás és a cél azonosítására portszámot használ. Ezáltal egyszerre többen is használhatják a protokollt.
Az UDP szegmens 8 bájtos fejrészből, és az azt követő adatmezőből áll. Az ellenőrző összeg használata opcionális, van, mikor csak 0 szerepel helyette.

1.3. Az Internet hálózati rétege: az IP

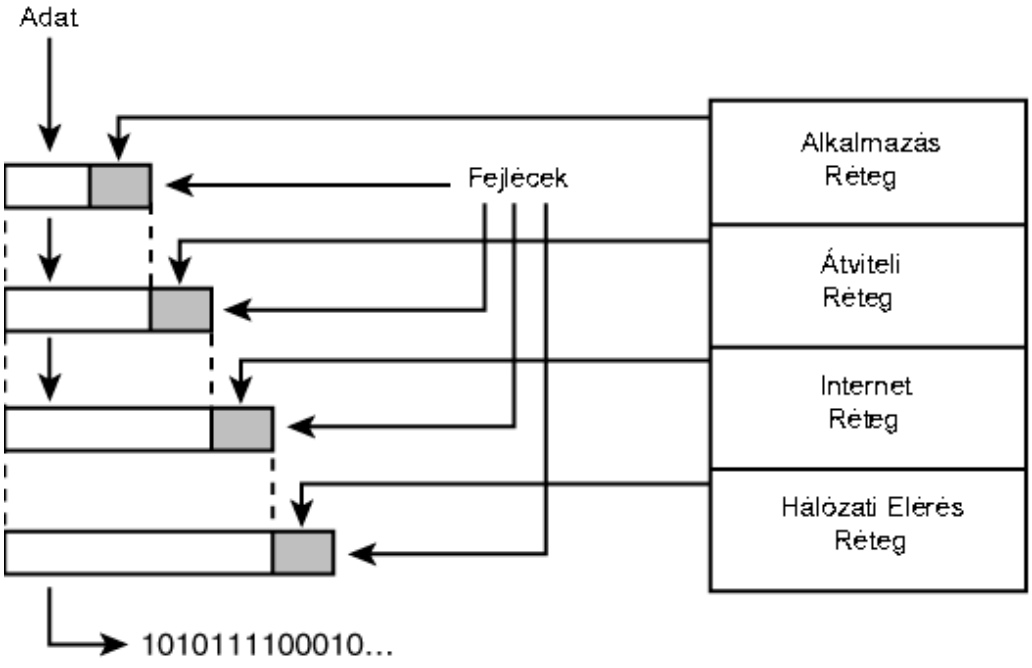
A TCP az általa feldolgozott datagramokat átadja az IP-nek (internetprotokoll). Persze ezzel együtt közölnie kell a rendeltetési hely internetcímét is. Az IP-t ezeken kívül nem érdekli más: nem számít, hogy mi található a datagramban, vagy hogy hogyan néz ki a TCP fejléc. Az IP feladata abban áll, hogy a datagram számára megkeresse a megfelelő útvonalat, és azt a másik oldalhoz eljuttassa. Az útközben fellelhető átjárók és egyéb közbülső rendszereken való átjutás megkönnyítésére az IP a datagramhoz hozzáteszi a saját fejlécét. A fejléc fő részei a **forrás** és a **rendeltetési hely internetcíme** (32 bites címek, pl. 128.6.4.94), a **protokollszám** és egy **fejrész-ellenőrzőösszeg**. A forrás címe a küldő gép címét tartalmazza. (Ez azért szükséges, hogy a vevőoldal tudja, honnan érkezett az adat.)
A rendeltetési hely címe a vevőoldali gép címét jelenti. (Erre pedig azért van szükség, hogy a közbenső átjárók továbbítani tudják az adatot.) A protokollszám kijelöli, hogy a datagram a különböző szállítási folyamatok közül melyikhez tartozik. A TCP egy biztos választási lehetőség, de léteznek egyebek is (pl. UDP). Végül az ellenőrzőösszeg segítségével bizonyosodik meg a vevő oldali IP arról, hogy a fejléc az átvitel során nem sérült-e meg.
A TCP és az IP különböző ellenőrzőösszegeket használ. Az IP-nek meg kell győződnie a fejléc sértetlenségéről, különben rossz helyre küldhet el adatot. A TCP és az IP a biztonság és a hatékonyság növelése miatt tehát külön ellenőrzőösszegeket használ. Az IP fejléc a következőképpen néz ki:

Verzió	IHL	Szolgálat típusa	Teljes hossz
Azonosítás		DF MF	Darabeltolás
Élettartam	Protokollszám		Fejrész ellenőrző öszege
Forrás címe			
Cél címe			
Opciók (0 vagy több szó)			

59. ábra
Az IP fejrész felépítése

Nem esett szó a fejlécben lévő többi mező jelentéséről, mert a legtöbbjük a jelen tananyag keretein túlmutat. A **darabeltolás** és a **DF**, **MF** mezők a datagramok részeinek nyomon követésére használatosak. Egy datagramot például akkor kell széttördelni, amikor az egy olyan hálózaton halad keresztül, amely számára túl nagy méretűnek mutatkozik.

Az **Élettartam** mezőben lévő szám mindig csökken, amikor a datagram egy rendszeren halad keresztül. Amikor eléri a nullát, a datagram megsemmisül. Ezt az eljárást a rendszerben esetleg felépülő végtelen ciklusok miatt építették a protokollba. Persze ezek felléptének valószínűsége az ideális esetben nulla, de a jól megtervezett hálózatoknak a bekövetkezhetetlen eseményekkel is el kell tudniuk bánni. Amikor a hálózati réteg összerak egy teljes datagramot, tudnia kell, hogy mit tegyen vele.
A **Verzió** mező azt mutatja meg, hogy az IP-nek melyik verzióját használta a küldő (pl. IPv4 vagy IPv6).
Az **IHL** mező tartalmazza az IP fejrész hosszát. Erre azért van szükség, mert a TCP-hez hasonlóan az IP hossza sem állandó.
A **Szolgálat típusa** mező tartalma utal arra, hogy a küldő mit kíván továbbítani, és ennek megfelelően milyen szolgálatot szeretne. Más szolgálatot igényel ugyanis a hangátvitel, a képátvitel, az adatátvitel stb.
Ezután a **Teljes hossz** mező következik, melybe a datagram minden része beleértendő, a fejrész és az adatrész is. A maximális hossza 65535 bájt lehet.
Végül essen szó az **Azonosítás** mezőről, ami ahhoz kell, hogy a célhoszt meg tudja állapítani, hogy egy újonnan érkezett csomag melyik datagramhoz tartozik. Egy datagram minden egyes darabja ugyanazzal az Azonosítás-mezőértékkel rendelkezik. Az adatátvitelhez minden rétegben kapcsolódik egy fejrész. A fentiekben tárgyalt TCP-, valamint az IP-réteg mellett természetesen ez szükséges az alkalmazási és a hálózati rétegben is, mint ahogy ezt a következő ábra mutatja.



60. ábra.
Fejrészekkel történő kiegészítés a rétegekben

1.4. Címzési rendszer

Ha egy hálózat számítógépei a kommunikációhoz a TCP/IP protokollt használják, minden számítógép minden adaptere (hálózati kártyája) egyedi azonosítóval rendelkezik, mely egyedi azonosítók alapján a számítógépet az IP protokoll megtalálja a hálózatban. A számítógéphez rendelt azonosítót IP-címnek (IP address) nevezzük, mert az IP protokoll alapvető feladata, hogy a TCP szállítási szintű csomagokat a fejrészben megadott című állomáshoz továbbítsa, akár nagy kiterjedésű hálózaton keresztül is.

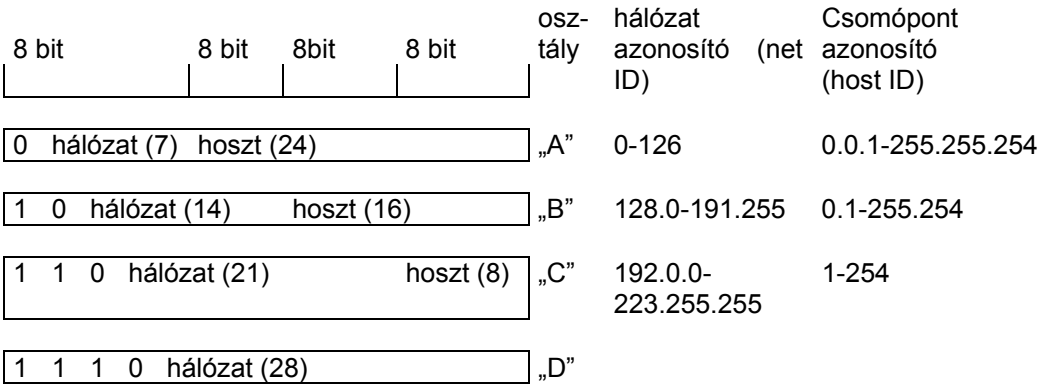
A címzési rendszer kialakításánál azt a valóságos tényt vették figyelembe, hogy a címzés legyen hierarchikus: azaz vannak hálózatok (network), és ezen belül gépek (hosztok). Így célszerű a címet két részre bontani: egy **hálózatot azonosító** (net ID – network identification), és ezen belül egy, a **csomópontot azonosító** címre (host ID – host identification).

A hálózati csomópontok IP-címe 32 bites szám, amelyet a leggyakrabban az úgynevezett pontozott tízes formában (dotted decimal form) írunk le, azaz négy darab 0 és 255 közötti decimális számmal, egymástól pontokkal elválasztva, például 192.168.67.4. Ez a négy tízes számrendszerbeli szám úgy keletkezik, hogy a 32 kettes számrendszerbeli számot nyolcasával átváltjuk tízes számrendszerbe, és egymás mögé írjuk. Elválasztásként pedig pontot használunk.

Pl.:	11000000	10101000	01000011	00000100
	192	.168	.67	.4

Az IP-cím két részből áll: az első a csomópontot tartalmazó helyi hálózatot azonosítja, a másik a hálózaton belül a csomópontot. Az, hogy az IP-címből hány bit a hálózat és hány a csomópont azonosítója, elsősorban attól függ, hogy az összekapcsolt hálózatok rendszerében mennyi hálózatra, illetve hálózatonként mennyi csomópontra van szükség. A hálózatazonosító az összekapcsolt hálózatok között, a csomópont-azonosító a hálózaton belül egyedi. Ha a hálózat az Internethez csatlakozik, a hálózatazonosítónak az egész Interneten belül egyedinek kell lenni. Ezért az Internethez csatlakozó hálózatok azonosítóit (a számítógépek IP-címeinek első néhány, azaz 8, 16, vagy 24 bitjét) külső szolgáltató határozza meg. Ezt központilag az InterNIC (Inter-Network Information Center) végzi különböző régiók, különböző szervezeteinek bevonásával.

Az IP-címeket, címtartományokat és így a hálózatokat különböző osztályba sorolják. A címzési rendszerben 4 címzési osztályt alakítottak ki.



61. ábra
Az IP-címformátumok

Az első három címforma a következő:

a) 27=128 hálózatot, hálózatonként 224=16 millió hoszttal (A osztályú cím),
b) 214=16 384 hálózatot, 216= 64 K-nyi hoszttal (B osztályú cím),
c) illetve 221=2 millió hálózatot, (amelyek feltételezhetően LAN-ok), egyenként 28=254 hoszttal azonosít.

A „D” osztályú címtartományt olyan esetekben használják, ha az IP-címtartományt vásárló cégnek nincs szüksége egy teljes „C” osztályú tartományra, hanem csak néhány IP-címet szeretne. Napjainkban már annyira kevés szabad IP-cím van, hogy még ha lenne rá elegendő pénzünk, akkor sem tudnánk egy teljes, összefüggő címtartományt vásárolni.
Viszont egyre több Internetre kapcsolt gép van a világon. Ezt az ellentmondást úgy oldják fel a vállalatok, hogy csak néhány IP-címet vásárolnak, ezeket az IP-címeket a szervereiknek és a routereiknek osztják ki. A vállalat többi gépe pedig ún. belső IP-címet használ. Ezekkel a belső IP-címekkel kapcsolódnak a szerverekhez, és a szerver fordítja át a kérést a külvilág felé.

Az IP-címben a hálózat és a csomópont azonosítója az alhálózati maszk (netmask) segítségével választható szét, ezért amikor egy hálózati csomópontot konfigurálunk, az IP-cím mellett az alhálózati maszkot is meg kell adni. Az IP protokoll számára az IP-cím és az alhálózati maszk csak együtt értelmes, mert az IP-cím mindig két részből áll. Az alhálózati maszk hiányában a csomópont nem tudja meghatározni az őt tartalmazó hálózat címét, amely az útválasztáshoz elengedhetetlen.

Az alhálózati maszk is 32 bites szám, amelyben 1-esek jelzik a hálózat, 0-k a csomópont azonosítójának IP-címbeli helyét. Az alhálózati maszk 1-esekből álló sorozattal kezdődik, és 0-sorozattal ér véget.

Példa alhálózati maszk használatára:

IP-cím: 196.225.15.4
Alhálózati maszk: 225.225.255.0

Kettes számrendszerben:

IP-cím: 11000100 11100001 00001111 00000100
Alhálózati maszk: 11111111 11111111 11111111 00000000

A két szám között az ÉS (AND) műveletet bitenként elvégezve a hálózat címét kapjuk:

11000100 11100001 00001111 00000000
(tízes számrendszerben: 196.225.15.0)

Ha az ÉS műveletet a cím és az alhálózati maszk inverze között végezzük el, az állomás hálózaton belüli címét kapjuk.

A címzésnél bizonyos címtartományok nem használhatók.

A 127-el kezdődő címek a "**loopback**" (visszairányítás)-címek, nem használhatók a hálózaton kívül, a hálózatok belső tesztelésére használható. (Pl. az 127.0.0.1 IP-cím a saját gépünket jelenti. Tehát ha a gépünkön fut egy webkiszolgáló, ezen a címen meg lehet nézni a kezdőoldalunkat.)

A hoszt címrészbe csak 1-eseket írva lehetséges az adott hálózatban lévő összes hosztnak üzenetet küldeni (**broadcast**). Például a 195.13.2.255 IP-címre küldött üzenetet a 193.13.2 című hálózatban lévő összes gép megkapja. (Ha a szerverről az összes munkaállomásra szeretnénk egy üzenetet küldeni, melyben például a szerver leállítását közöljük, akkor az a legegyszerűbb módszer, ha erre az x.x.x.255 címre küldünk egy üzenetet.)

Ha a hoszt címrésze 0, az az aktuális hálózatot jelöli. Például a saját gépről 0.0.0.0 címre küldött üzenet a saját gépre érkezik.

Az Internet esetében a rétegeknek megvan az egyedi azonosítója a címzéshez:

Réteg	Címzési módszer
Alkalmazási	Hoszt neve, portja
Internet	IP-cím
Hálózatelérési	Fizikai cím

Amikor egy program adatokat küld a TCP/IP-hálózaton keresztül, az elküldendő adatokhoz mellékeli a saját és a címzett IP-címét is. Ha a címzett címében a hálózat azonosítója más, mint a küldőt tartalmazó hálózat címe, a címzett csak útválasztó(ko)n keresztül érhető el.

Ezért a küldő számítógépen futó IP protokollnak először azt kell megállapítania, hogy az elküldendő csomag címzettje helyi hálózatban van-e, ezt pedig a következőképpen teszi.:

- a küldő IP-címéből a küldő alhálózati maszkja segítségével előállítja a hálózatazonosítót (éppúgy, mint a fenti példában),
- a címzett IP-címéből a küldő alhálózati maszkjával előállítja a hálózatcímet (a címzett alhálózati maszkjával nem rendelkezik),
- a kapott két számot összehasonlítja.

Ha a két szám egyezik, megkeresi a helyi hálózatban, ha pedig nem, a csomagot az alapértelmezés szerinti átjárónak (amely nem más, mint egy útválasztó berendezés) küldi el.

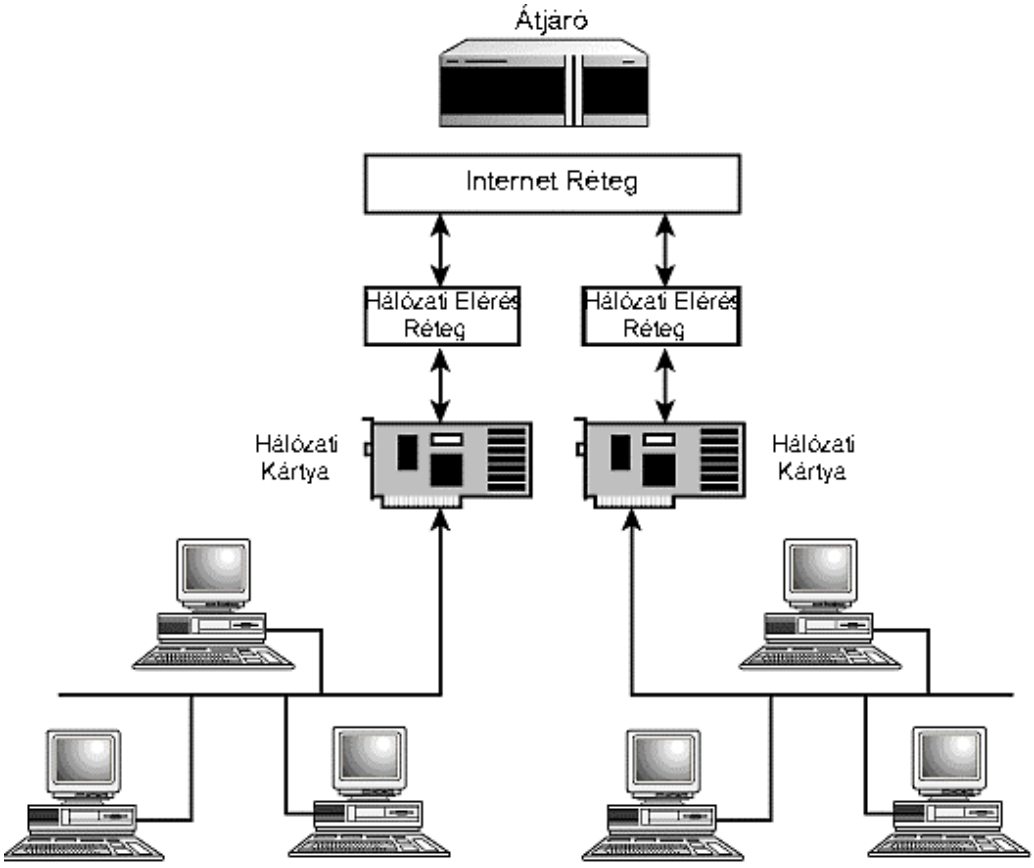
Az IP-cím nem a számítógépet, hanem annak csak a hálózati illesztőjét azonosítja. Ha a számítógépben több hálózati kártya van, minden illesztőnek külön IP-címet kell adni.

A hálózatokban lehetőség van arra, hogy a számítógépek IP-címeit (és a TCP/IP használatához szükséges egyéb paramétereket) egy vagy több kiszolgáló automatikusan ossza szét. Az ilyen kiszolgálók használata esetén a számítógépeken az IP-címet, az alhálózati maszkot, az alapértelmezés szerinti átjárót és a többi paramétert nem a rendszergazda (vagy a felhasználó) állítja be. A számítógépek az operációs rendszer betöltésekor a hálózatokban elérhető valamelyik címkiszolgálóhoz fordulnak, amely a rendelkezésre álló címtartományból ad nekik IP-címet. A számítógépek IP-címe így dinamikusan változhat: ezért hívják azt a protokollt, amely segítségével a címkiosztás történik, dinamikus csomópont-konfiguráló protokollnak (Dynamic Host Configuration Protocol, DHCP).

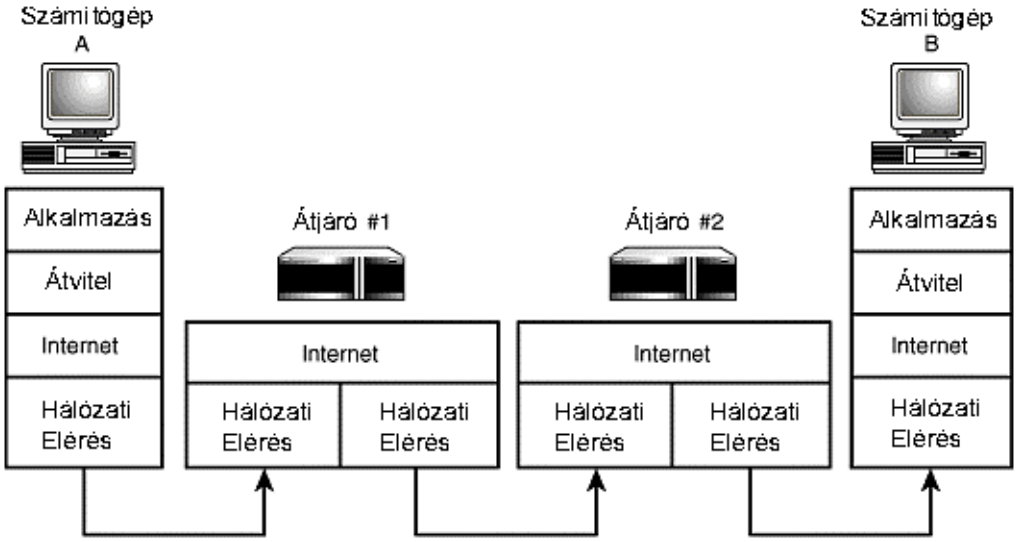
A számítógépeket alacsonyabb (fizikai, adatkapcsolati) szinten nem az IP-cím azonosítja, hiszen a sok közül ez csak egyetlen (bár kétségkívül a legelterjedtebb) megállapodás a számítógépek címzésére. Azonban minden hálózati hardverelemnek az egész világon egyedi azonosítója van: ez a hálózatikártya-azonosító (NetCard ID) vagy hardvercím (hardware address). Egy hálózati kártya tehát a vele elektromosan összekapcsolt más hálózati kártyának célzottan tud jeleket küldeni a címzett kártya hardvercíme alapján, az IP-cím alapján azonban nem. Feladat tehát, hogy a címzett állomás eléréséhez az IP-címhez meg kell találni az adott IP-címmel rendelkező hálózati kártya hardvercímét. Ez a művelet a címfeloldás (address resolution). A címzett állomásnak az IP-cím alapján való megtalálása a hálózatban az IP protokoll feladata. Amikor el kell küldenie egy csomagot, először megállapítja, hogy a célállomás címe a helyi hálózat valamelyik gépéé-e. Ha a címzett számítógép külső hálózatban van, a csomagot az alapértelmezés szerinti – vagy az útválasztó táblázatban megadott – átjáróhoz (útválasztóhoz) kell továbbítani. Ha cím helyi hálózatbeli cím, az **ARP** (Address Resolution Protocol – címfeloldó protokoll) megpróbálja megtalálni az IP-címhez tartozó hardvercímet. Ezért a helyi hálózat gépeinek elküld egy alacsony szintű szórt üzenetet (úgynevezett ARP broadcastot), amelyben megadja a küldő IP-címét és harvercímét, valamint a címzett IP-címét. Ha a címzett számítógép be van kapcsolva, és működik rajta a TCP/IP protokollkészlet, a rajta futó ARP fogadja az üzenetet, és célzottan (a küldő hardvercíme alapján) válaszol rá, a

válaszban megadva saját hardvercímét. A fizikai kapcsolat ettől kezdve lehetséges. Azonban nincs mindig szükség szórt üzenetekre, mivel az ARP fenntart egy átmeneti tárolót, amelybe felveszi az éppen feloldott IP-címeket és a hozzájuk tartozó hardvercímeket. Az ARP-tábla egy bejegyzése tehát egy IP-címből és a hozzá tartozó hardvercímből áll.

Az útválasztás az a művelet, amelynek során a rendszer egy helyi hálózat valamely számítógépétől az adatcsomagokat különböző vonalszakaszokon keresztül eljuttatja azokhoz a címzettekhez is, amelyek nem részei a helyi hálózatnak. A TCP/IP protokollt használó rendszerek számára az a helyi hálózat, amelynek csomópontjai azonos hálózatcímet használnak. Ha egy csomag elküldésekor a címzett csomópont IP-címében a hálózatcím más, mint a küldőé, az IP protokoll megpróbálja a csomagot egy útválasztóhoz (router) eljuttatni, amelynek az a feladata, hogy a kapott csomagot továbbítsa a címzett hálózat felé. Az útválasztó olyan berendezés, amelynek több hálózati csatolója van, és mindegyik más (helyi) hálózathoz csatlakozik. Az útválasztó csomagokat fogad az egyes, hozzá csatlakozó hálózatok számítógépeitől, és továbbítja őket egy másik hálózati csatolóján. Az, hogy melyik hálózati csatolót kell használni a csomag elküldéséhez, a memóriájában lévő útválasztási tábla (routing table) alapján dönti el. A tábla bejegyzései hálózatok felé vezető útvonalakat (route) képviselnek. Két hálózat összekapcsolását és két gép közötti kapcsolatot mutatja be a következő két ábra.



62. ábra
Két hálózat összekapcsolása átjáróval



63. ábra
Két gép közötti kapcsolat átjárókkal

1.5. A domainnév-rendszer (DNS – Domain Name System)

A számítógépek IP-címeit nehéz megjegyezni és könnyű elgépelni. Természetes tehát a felhasználóknak az az igénye, hogy a számítógépeket az IP-címek helyett könnyen olvasható és megjegyezhető nevek megadásával érjük el. Azonban a TCP/IP protokollkészlet használata esetén a számítógépeket csak az IP-cím alapján lehet elérni, név alapján nem. Ezt a műveletet **névfeloldás**nak (name resolution) nevezik.

A névfeloldás alkalmazásával az Interneten lévő szolgáltatógép vagy valamelyik csomópont eléréséhez a számítógépeket csomópontnévvel (host name) is megadhatjuk. A csomópontnév tetszőleges, legfeljebb 256 karakterből álló szöveg lehet. Az Interneten az úgynevezett teljes tartománynévvel (Fully Qualified Domain Name – FQDN) hivatkozhatunk rá. A tartománynév pontokkal (.) tagolt csomópontnév (host name), amelynek egyes részei a számítógépet tartalmazó szervezetet, illetve a számítógép helyét határozzák meg. Minden csomópontnévhez egyetlen IP-cím tartozik, de egy csomóponthoz (azaz IP-címhez) több név is rendelhető.

A névfeloldás alapvetően háromféleképpen történhet:

1. Szórt üzenettel: a küldő számítógép szórt üzenetben elküldi a címzett nevét a hálózatban elérhető számítógépeknek. Erre az üzenetre – az IP-címet is tartalmazó csomaggal – csak az a számítógép válaszol, amelynek a neve egyezik a szórt üzenetben megadott névvel.

2. Táblázatban való kereséssel: a küldő számítógép egy, a helyi merevlemezen lévő táblázatban megkeresi a címzett nevét, és az ott található IP-címet használja a kommunikációhoz.

3. Névszolgáltatóval (name server): ha a hálózat egy számítógépén van olyan adatbázis, amely a hálózat többi csomópontjának nevét és IP-címét tartalmazza, a küldő számítógép ehhez a géphez, a névszolgáltatóhoz (name server) is fordulhat. A küldő gép ide továbbítja a címzett számítógép nevét tartalmazó kérést. A névszolgáltató saját adatbázisában megkeresi a címzett nevét, és ha megtalálta, elküldi a mellette lévő IP-címet a küldőnek. (A küldőnek csak a névszolgáltató IP-címét kell ismernie.)

A DNS rendszer tervezői hierarchiát építettek fel a számítógépek elnevezésében. Ez azt jelenti, hogy a világot (a névteret) tartományokra (domain) osztották. Az egyes tartományokban megnevezett csomópontok és további tartományok is lehetnek. Minden tartományban van legalább egy névszolgáltató (name server), amely

- csomópontok esetén meg tudja adni a csomópont nevéhez tartozó IP-címet,
- a domaineken belüli tartományok esetén a kérést továbbítani tudja az illetékes névszolgáltatóhoz.

A hoszt neve – amely valamilyen szimbolikus név – azonosítja a felhasználó számára a gépet, azaz "így hívja" (pl.: omega). Az Internet használata során két, egymástól akár sokezer kilométerre lévő számítógép között alakul ki kapcsolat. Nyilvánvalóan ezért minden egyes gépet azonosíthatóvá, címezhetővé kell tenni. Erre két, egymással egyenértékű módszer áll rendelkezésre.
Az elsődleges módszer az, amit IP-címzésként már megismertünk, míg a másodlagos – a felhasználók által szinte kizárólagosan használt – módszer az azonosító domain-nevek rendszere.

A címben szereplő egyes címrészeket ma már nem véletlenszerűen határozzák meg, hanem hierarchikusan felosztott földrajzi terület, domainek alapján. Így a cím egyes bájtjai (8 bites csoportjai) a domaint, az ezen belüli aldomaint és hosztot, azaz a címzett számítógép helyét jelölik ki. A domain általában egy ország globális hálózati egysége vagy hálózati kategóriája, az aldomain ezen belül egy különálló hálózatrész, a hoszt pedig az ezen belüli felhasználókat kiszolgáló gép azonosító száma.

A felhasználó számára könnyebben használható a név alapján történő címzés, ahol a sok számjegyből álló IP-cím helyett egy karakterlánc, az FQDN (Fully Qualified Domain Name) használható. Az FQDN, azaz a teljes domain-név, amelyet a DNS (Domain Name System), vagyis a domainnév-rendszer szerint képeznek, ugyanúgy hierarchikus felépítésű, mint az IP-cím, formailag pedig több, egymástól ponttal elválasztott tagból áll.

Például az omega.sziget.hu címben az egyes tagok sorrendben a kiszolgáló gépet, a hosztgépet (ennek neve omega), az aldomaint, azaz hálózati altartományt (sziget.hu), végül pedig a domaint, vagyis az adott ország globális hálózati tartományát (hu) határozzák meg. A hálózati altartomány, az aldomain több tagot is tartalmazhat, de akár hiányozhat is a cím domain-név részéből.

A domain-név egyes részeit néha eltérő kifejezéssel adják meg: a hálózati tartomány domain vagy „network", az altartomány aldomain vagy „subnet", a kiszolgáló gép a hoszt vagy „host-address".

A domain-nevek használata az Internet számára némi járulékos munkát ad, hiszen egy adatcsomag-továbbítás előtt a hosztcímből meg kell határozni a vele egyenértékű IP-címet, és a küldemény hosztcímét ezzel kell helyettesítenie. Az összetartozó IP-címeket és hosztcímeket a hosztgép először a helyi címtáblázatban (host table) keresi. Ha a keresés eredménytelen, a hosztgép az Internet valamelyik speciális gépéhez, a névszolgáltatóhoz fordul, amely az Internet gépeinek adatait tartalmazó, szabályos időközönként frissített címtáblázatot kezeli. A címtáblázatokban a hoszt.aldomain.domain alakú hosztcímhez a vele egyenértékű IP-cím, esetleg hivatkozási (alias) alak is tartozhat.
Pl. ugyanarra az IP-címre hivatkozik a www.pgsm.hu, a www.pannongsm.hu, a www.pgsm.net. Ezek közül csak az egyik az „igazi" domain név, a többi alias név.

Az IP-cím kérésekor azt is közölni kell a névszolgáltatóval, hogy az mire kell. Ha például levelezéshez kérjük, akkor a névszolgáltató a névhez tartozó MX (Mail Exchange) adatrekordot adja vissza, különben a tényleges IP-címet.

Az előbbi példa szerint az omega.sziget.hu cím-meghatározása a következő: a gép internetcímének meghatározásához 4 potenciális kiszolgálót kellene megkérdezni. Először egy központi kiszolgálótól kellene megtudakolni, hogy hol található a „hu" kiszolgáló, amely nem más, mint a hálózatba kapcsolt magyar internethelyek nyilvántartása. A gyökérként szereplő kiszolgáló több „hu" kiszolgáló nevét és internetcímét adná meg. (Minden szinten több ilyen névkiszolgáló van, hogy az esetleges meghibásodások ne okozzanak fennakadást.) A következő feladat lenne a „hu" kiszolgáló lekérdezése a sziget névkiszolgálójáról. Itt is több kiszolgáló nevét és internetcímét kapnánk meg. Ezek közül általában nem mindegyik található az intézmény területén (egy esetleges áramszünet fellépte miatt).

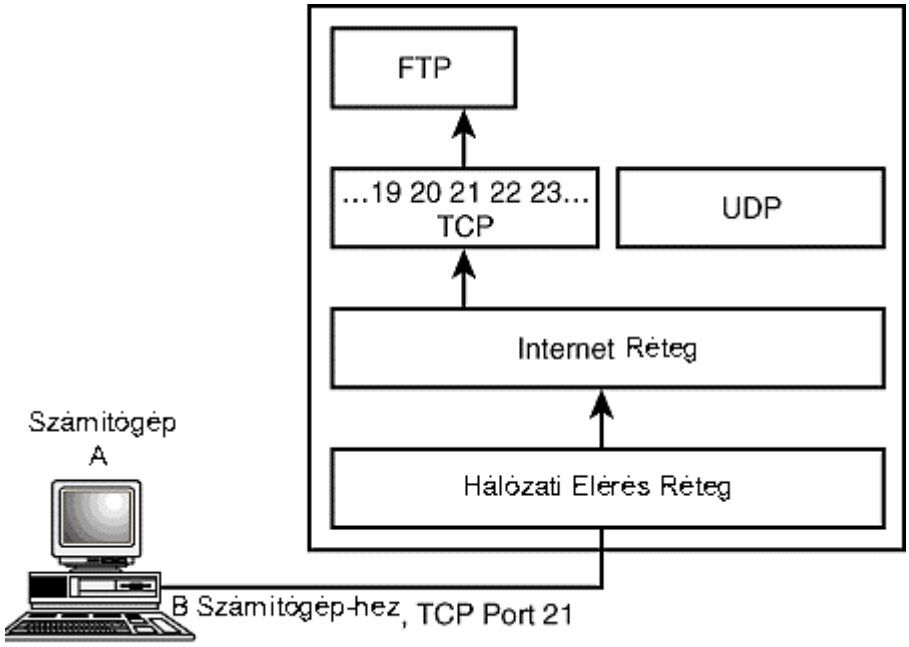
Ez után a sziget névkiszolgálójától kérdeznénk le az omega gép adatait. A végső eredmény az omega.sziget.hu gép internetcíme lenne. A fenti szintek mindegyike egy tartományt (domain) jelöl. A teljes omega.sziget.hu név egy tartománynév (domain name).

Az esetek nagy többségében szerencsére nem kell a fenti lépések mindegyikét végrehajtani. A legfelső kiszolgáló (gyökér) ugyanis egyben a legfelső szinten lévő tartományok (pl. hu) névkiszolgálójaként is

szerepel. Tehát a gyökérkiszolgáló felé irányuló egyetlen kérdéssel a SZTAKI névkiszolgálójához lehet eljutni. Az alkalmazott szoftverek pedig a már feltett kérdésekre kapott válaszokra emlékeznek, az így megkapott domain-név és a hozzá tartozó IP-cím eltárolódik. Persze minden ilyen információnak van egy megfelelő élettartama, ami tipikusan pár napnak felel meg. Az élettartam lejárta után az információkat fel kell frissíteni, amivel az esetleges változások is nyomon követhetők.

Az IP-cím – hosztcím átalakítást a TCP/IP automatikusan végzi, de a „host" operációsrendszer–parancs kiadásával mi is lekérdezhetjük egy ismert felhasználó számát.

Az előbbiek alapján már nyilvánvaló, hogy az egyes hosztgépekhez nemcsak IP-cím vagy az azzal egyenértékű domain-cím tartozik, hanem a hosztgépek a rajtuk futó alkalmazások eléréséhez tartozó portcímet (Application Selection Address) is használnak. Ezért a címeket ki kell egészíteni az alkalmazás elérésére szolgáló portcímmel is (9-10. ábra).



64. ábra
Szolgáltatáselérés portcímen keresztül

Gyakran használt portcím pl. a 80-as vagy a 8080-as port, melyet a proxy szerverek használnak. Másik elterjedt portcím a 21-es, az FTP szerverek által használt port.
A tűzfalak egyik védelmi lehetősége az ún. portszűrés. Ennek az a célja, hogy csak azok a portok legyenek „nyitva", melyeket a rendszergazda szándékosan definiál. Így kiszűrhetők az ismeretlen, vagy véletlenül nyitva lévő portokon keresztül érkező támadások.

Míg az egyes hosztokat a hosztcímük egyértelműen meghatározzák, addig másokat több felhasználó használ, tehát őket is meg kell

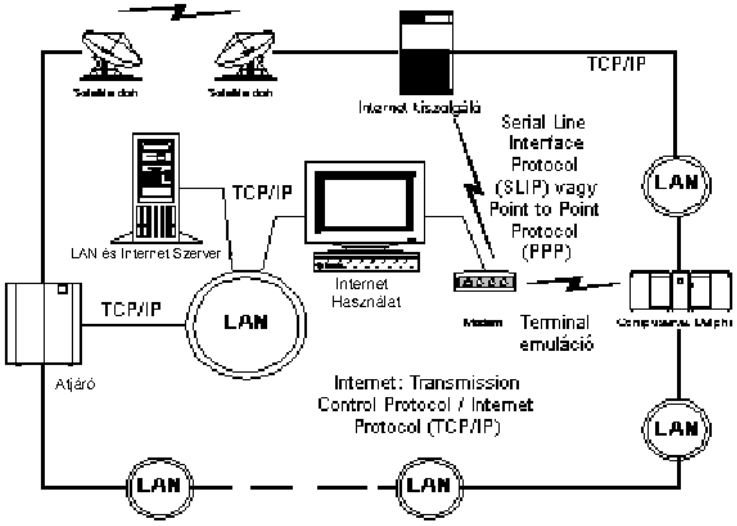
különböztetnünk egymástól : erre felhasználói nevük (login-nevük), vagyis az adott hoszton az egyedi azonosítónév szolgál.

Egy személy levelezési (e-mail) címe tehát két fő részből áll, és így épül fel:

felhasználónév@hoszt.aldomain.domain

2. Kapcsolódás az Internetre

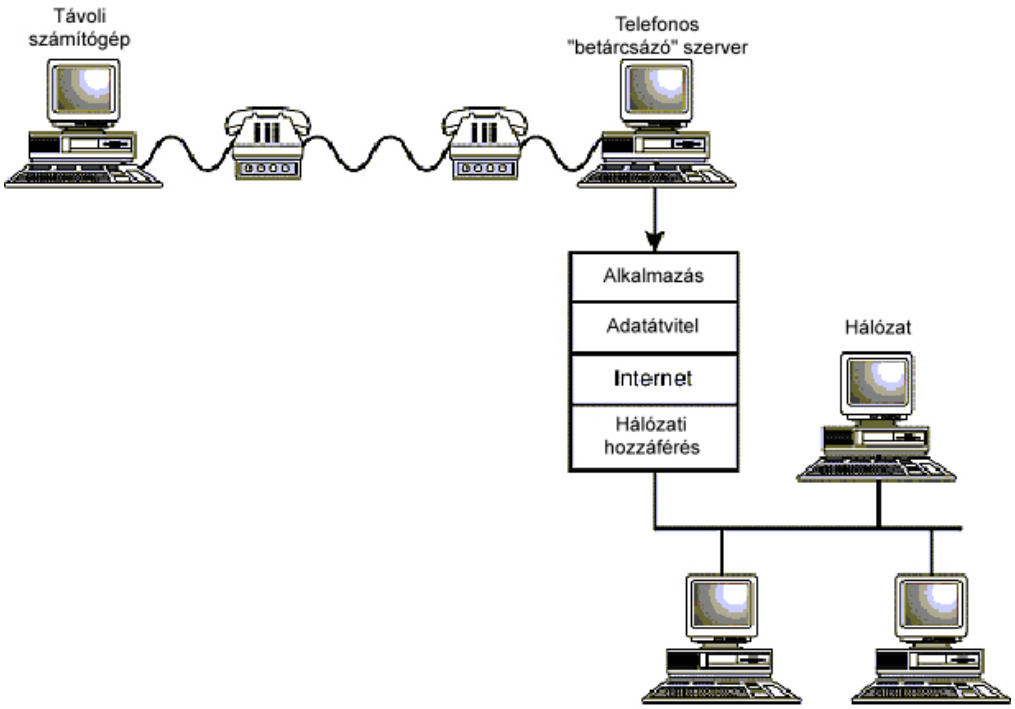
Az Internet felépítését a következő ábrán láthatjuk. A legfontosabb része a nagy adatátviteli sebességű, általában optikai kábelekből és műholdas kapcsolatokból álló gerinchálózat (bone), amely az ide kapcsolódó hálózatok információit szállítja.



65. ábra
Az Internet hálózat felépítése, kapcsolódási lehetőségek

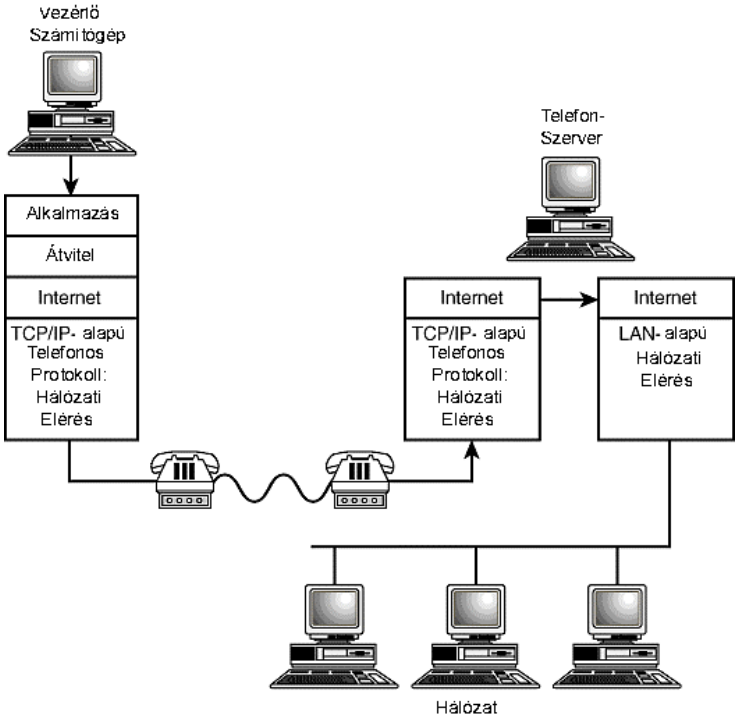
A csomagokat routerek irányítják a különféle útvonalakon. Azonban kevés felhasználónak adatik meg a gerincre történő csatlakozás közeli lehetősége, általában a “főúttól messze”, mellékutak mentén, vagy csak egy kis ösvény végén laknak. Ez a hasonlat itt azért is találó, mert valóban tükrözi az adatátviteli sebesség csökkenését, amit például egy telefonos kapcsolat jelenthet. A felhasználó által elérhető adatátviteli sebességet a gerincig vezető alhálózatok adatátviteli sebessége közül a legkisebb fogja meghatározni.

A megfelelő hálózati teljesítmény eléréséhez csak nagy teljesítményű gépekkel lehet a gerincvonalakra csatlakozni. Az átlagos felhasználók ezért a helyi hálózati kapcsolataikat használhatják fel, míg egyéni felhasználók számára az Internet szolgáltatók (providerek) által üzemeltett nagyteljesítményű gépeken keresztül való csatlakozás a megoldás. Ennek megfelelően a következő kapcsolódási megoldások lehetségesek:



66. ábra
Telefonvonalon való kapcsolódás

A hálózati kapcsolódás feltétele: a helyi hálózaton a TCP/IP protokoll használata. Egy routeren keresztül az Internetre küldött csomagok eljuthatnak a célokig.
Telefonvonalon keresztüli kapcsolódás esetében egy modem és a telefonvonalon TCP/IP-szerű kapcsolatot megvalósító SLIP/PPP (SLIP - Serial Line Interface Protocol: soros vonali kapcsolati protokoll, PPP - Point to Point Protocol: pont-pont kapcsolati protokoll) protokoll szükséges. Számítógépünk a vonal másik végén egy Internetre kapcsolódó kiszolgáló számítógépen keresztül egy IP-címet hordozó hálózatra kapcsolt géppé válik. On-line szolgáltatón keresztül (terminálemulációval) az Internetre kapcsolódó gépen fut az a program, amelyet a telefonvonalon keresztül a számítógépet terminálként használva kezelünk.



67. ábra
Terminálszerver-megoldás

A felhasználót általában nem ez, hanem az elérhető szolgáltatások érdeklik. A szolgáltatások alapvetően két csoportba sorolhatók: közvetlen hálózati kapcsolatot nem igénylő (off-line) szolgáltatás – ilyen a levelezés –, és azt igénylő (on-line) szolgáltatások. Ennek megfelelően több megoldás is lehetséges.

A legegyszerűbb szolgáltatás a levelezés: ez lényegében hálózati kapcsolatot nem igényel. Általában egy internetszolgáltató számítógépén elhelyezett postaládát használunk: ennek tartalmát modemes kapcsolaton keresztül kezelhetjük; UUCP- (Unix to Unix Copy – Unixos gépek közti fájlmásolás) protokoll segítségével. Unixot futtató géppel modemen keresztül kapcsolódunk a szolgáltató gépére és a leveleket egy menetben fel-, illetve letöltjük; shell-számlát nyitunk: terminálként (vagy a szolgáltató speciális szoftverén keresztül) bejelentkezünk a szolgáltató gépére, és arról böngésszük a hálózatot; SLIP vagy PPP számlát nyitunk, amelyen keresztül gyakorlatilag minden böngésző, levelező és kommunikációs Internet-alkalmazást futtathatunk.

Ha megvan a lehetőség rá, helyi hálózatunkat az Internetre köthetjük. TCP/IP-t és internet-segédprogramokat telepítünk a hálózaton, majd a LAN-t valamilyen hálózati kapcsolattal (X.25, azaz nagy sebességű, bérelt ISDN-telefonvonal) routeren keresztül csatlakoztatjuk az Internetre.

3. Internetszolgáltatások

Az Interneten – mivel eltérő felépítésű hálózatokat köt össze –, szükségessé vált a rajta keresztül történő kommunikáció közös szabványainak kidolgozása, amelyet az RFC (Request for Comments – leírási ajánlások) dokumentumok tartalmaznak. A szabványok közös alapjául a UNIX operációs rendszerben megvalósított megoldások szolgáltak, mivel elsőként ilyen operációs rendszerű gépeket kötöttek össze.
Az Internet fontosabb alkalmazási protokolljai a következők:

- SMTP (Simple Mail Transfer Protocol) egy alkalmazási protokoll, amely a hálózati felhasználók egymással való kommunikációját teszi lehetővé; leveleket tud küldeni és fogadni.
- TELNET: terminálemuláció segítségével – a saját gépet terminálként használva – egy távoli hosztra felhasználóként lehet bejelentkezni.
- FTP (File Transfer Protocol): e fájlátviteli eljárás segítségével a felhasználó számára lehetővé teszi az általános könyvtár- és fájlműveletek végrehajtását saját gépe és egy távoli hoszt lemezegysége között (pl. fájlátvitel, -törlés, -átnevezés).
- Gopher: hierarchikusan felépített információban kereső protokoll.
- HTTP: HyperText Transport Protocol – hypertext átviteli protokoll.

A következőkben ezen protokollok és az ezeket használó szolgáltatások közül a legfontosabbakat tárgyaljuk.

3.1. Elektronikus levelezés

Az egyik legalapvetőbb szolgáltatás az elektronikus levelezés. Ez az alkalmazás az SMTP-re (Simple Mail Transfer Protocol – egyszerű levéltovábbítási protokoll) épül.

A levelezés, illetve a levelezést megvalósító protokoll működését a következőkben tekintjük át.
Tegyük fel, hogy a TOPAZ.RUTGERS.EDU nevű számítógép szeretné az alábbi üzenetet elküldeni.

Date: Sat, 27 Jun 87 13:26:31 EDT
From: hedrick@topaz.rutgers.edu
To: levy@red.rutgers.edu
Subject: meeting

Menjünk holnap vacsorázni!

Az üzenet formátumát egy internetszabvány (RFC 822) írja le. A szabványban megfogalmazódik, hogy az üzenetet ASCII karakterekként kell továbbítani. Az üzenet szerkezetének az alábbiak szerint kell kinéznie: fejlécsorok, aztán egy üres sor, majd az üzenet szövege következik. Végül a fejlécsorok szintaxisát definiálja részletesen: általában egy kulcsszó, majd egy érték.

A fenti üzenet címzettje LEVY@RED.RUTGERS.EDU. Kezdetben ez úgy nézett ki, hogy csak a címzett nevét és a gépet írták bele: "személy és gép". A szabványok fejlődése azonban ezt sokkal rugalmasabbá tette. Ma már más rendszerek üzeneteinek a kezelésére is vannak előírások. Ezzel lehetővé válik az Internetre nem kapcsolt gépek miatti automatikus átirányítás (forwarding): például az üzenetek egy sor rendszer számára egy központi (mail server) géphez kerülnek. Egyáltalán nem szükséges tehát, hogy létezzen a RED.RUTGERS.EDU névvel jelölt számítógép. A névkiszolgálókat úgy is be lehet állítani, hogy az üzenetek címzettet jelentő mezőjébe tanszékeket írunk, és minden egyes tanszék üzeneteit egy megfelelő számítógéphez irányítjuk. Az is lehetséges, hogy a @ jel előtti részbe ne egy felhasználónak a nevét írjuk, hanem valami mást. Egyes programokat fel lehet készíteni az üzenetek feldolgozására. A levelezési listák, illetve az olyan általános nevek, mint "postmaster" vagy "operator" kezelésére is felkészült a rendszer.

Az üzenet küldésének módját az RFC 821 és 974 dokumentumok tárgyalják. A küldést végző program párszor lekérdezi a névkiszolgálót, hogy meghatározza a célállomást. Az első lekérdezés alkalmával arról tájékozódik, hogy mely gépek kezelik a RED.RUTGERS.EDU gépnek szóló leveleket. Ebben az esetben a kiszolgáló válasza, hogy a RED.RUTGERS.EDU saját maga kezeli az üzeneteit. Ezután a program a RED.RUTGERS.EDU címét kéri le, ami 128.6.4.2. Ezt követően a levelező program egy TCP kapcsolatot nyit meg a 128.6.4.2 gép 25-ös portjára. A 25-ös port a leveleket fogadó foglalatnak felel meg. Miután a kapcsolat létrejött, a levelező program megkezdi a parancsok küldését.

Az alábbiakban álljon itt egy tipikus kommunikáció. A sorok előtt az szerepel, hogy az a TOPAZ vagy a RED nevű géptől származik-e. A példában TOPAZ kezdeményezte a kapcsolatot:

RED 220 RED.RUTGERS.EDU SMTP Service at 29 Jun 87 05:17:18 EDT
TOPAZ HELO topaz.rutgers.edu
RED 250 RED.RUTGERS.EDU - Hello, TOPAZ.RUTGERS.EDU
TOPAZ MAIL From: <hedrick@topaz.rutgers.edu>
RED 250 MAIL accepted
TOPAZ RCPT To: <levy.red.rutgers.edu>
RED 250 Recipient accepted
TOPAZ DATA
RED 354 Start mail input; end with <CRLF>.<CRLF>
TOPAZ Date: Sat, 27 Jun 87 13:26:31 EDT
TOPAZ From: hedrick@topaz.rutgers.edu
TOPAZ To: levy@red.rutgers.edu
TOPAZ Subject: megbeszélés
TOPAZ
TOPAZ Találkozzunk holnap 1 órakor!
TOPAZ .
RED 250 OK
TOPAZ QUIT
RED 221 RED.RUTGERS.EDU Service closing transmission channel

A parancsokban mindenütt normál szöveg szerepel: ez az Internet szabványokra tipikusan jellemző. A protokollok többsége szabványos ASCII parancsokat használ, ami arra is jó, hogy követhessük, éppen mi történik, és a problémákat diagnosztizálni lehessen. A levelezőprogram például minden ilyen beszélgetést egy állományban naplóz. Ha valami nem a megfelelő módon történik, akkor az állományt elküldhetjük a postmasternek. Ez ASCII formátumú, tehát látható, hogy mi történt. A dolog arra is jó, hogy közvetlenül a levelezést kiszolgáló géppel lépjünk kapcsolatba tesztelés céljából. Második észrevételként említjük, hogy a válaszok mindegyike számmal kezdődik: ez is az internetprotokollok jellemző vonása. A megengedett válaszokat a protokollok definiálják. A számok segítségével a felhasználói programok egyértelműen kommunikálhatnak. A válaszok maradék része szöveg, amely a könnyebb olvashatóság miatt szerepel, és nincs semmiféle kihatása a programok működésére. Maguk a parancsok arra használatosak, hogy a levelező program a kiszolgálóval közölje azokat az információkat, amelyek az üzenet továbbítása miatt szükségesek. A fenti kiszolgáló az információt az üzenetből is kiolvashatja. Bonyolultabb esetekben azonban ez nem lenne biztonságos. Minden kommunikáció a HELO paranccsal kezdődik, amit a kapcsolatot kezdeményező rendszer nevének kell követnie. Ezek után következik a küldő és a címzett meghatározása. (Lehet több RCPT-parancsot) is kiadni, ha több címzett van.)
Végül maga az üzenet jön: a szöveget egy olyan sorral fejezzük be, amelyben csak egy pont szerepel. (Ha a szövegben is szerepel ilyen sor, akkor a pont megduplázódik.) Miután az üzenet fogadása megtörtént, a küldő másik üzenetet küldhet, vagy befejezheti a kommunikációt, mint ahogy a fenti példában is történt.

Egy levelezőprogram (mail) segítségével szöveges állományt küldhetünk az Internet bármelyik felhasználójának. Ehhez az kell, hogy minden levelezőnek egyedi címe legyen, és a címzés is szabványos legyen. Egy felhasználó e-mail címe általánosan a következőképpen épül fel:

Felhasználói_név @ gépnév . domain_név . subdomain_név . ország(intézmény)azonosító

Általánosan fogalmazva egy felhasználóinév-részből (username) és egy címrészből (domain) áll, a kettő között a @ jel található. Ez a "kukac" az angol "at" szót jelenti, vagyis arra utal, hogy ez a felhasználó HOL (melyik gépen) található meg. A felhasználói_név egy rövid azonosító, amely nem tartalmazhat speciális karaktereket. A @ (kukac) jel a felhasználói nevet választja el a gépet leíró, utána lévő résztől. A cím hierarchikus felépítésű, a legutolsó jelöli a legmagasabb szintet, így szűkítve a kört. Ha ezt értelmezni akarjuk, akkor célszerű hátulról kezdeni. Az utolsó azonosító egység az országra vagy az intézmény jellegére utal.
(Országra utal pl.: .hu: Magyarország, .nl: Hollandia, .fr: Franciaország … Intézmény típusra utal: .com: comersional – kereskedelmi, .edu: education – oktatás, .mil: military – katonai, .gov: government – kormányzati, .net: network – hálózatkiszolgáló …)

Ha nem ASCII (0-127-es kódú) karaktereket tartalmazó üzeneteket kívánunk küldeni, hanem olyat, amelyben található olyan karakter, aminek a 8. bitje 1, azt a rendszer levágja, elvész.
Így közvetlenül bináris fájlok átvitele nem lehetséges. Több megoldás létezik erre a problémára, a legelterjedtebb átkódoló program az UUENCODE/UUDECOD, amely a bájtokat olyan bájtokra konvertálja, amelyben csak ASCII karakterek szerepelnek, majd az átvitelt követően visszakonvertálja eredeti formára.

Egy másik megoldás esetén már van lehetőség nem ASCII karakterek, képek, hangok küldésére levélben is. Ezt az eljárást MIME-nek (Multi-purpose Internet Mail Extensions) nevezik. Amelyik levelezőprogram ismeri ezt, azzal írható, illetve olvasható akár magyar ékezeteket tartalmazó levél is.

A levelek küldését és fogadását ténylegesen egy folyamatos hálózati kapcsolattal rendelkező számítógépen futó program, a Mail-szerver (levelezéskiszolgáló) végzi. A felhasználók ennek a programnak küldik, illetve ettől kapják meg leveleiket. Az elküldött és kapott leveleket ez a program tárolja, és a címek alapján végzi a hálózaton keresztüli kézbesítést.
Lényeges megkülönböztetni a hálózati internetcímeket a levélcímektől. A levelek címrésze határozza meg annak a gépnek az internetcímét, amelyen a levelezés kiszolgáló program fut, és ezen címrész alapján a gépre küldött leveleket egy olyan lista segítségével kézbesíti, amely a levelezésbe bevont felhasználókat azonosítja.

A papíralapú levelezésnél papírra írjuk a levél szövegét, az elektronikus levelezésben erre a célra szövegszerkesztőt használunk. A hagyományos levelet borítékba tesszük, a borítékra felírjuk a feladót és a címzettet a megfelelő szabályok betartásával. Ezt a funkciót az elektromos levelezésben a levél fejléce látja el.
Mindkét esetben a megírt és megcímzett levelet egy közvetítő mechanizmusra bízzuk, ami az egyik esetben a Posta, a másik esetben internetes levéltovábbító programok. A címzett mindkét módszer esetén rendelkezik egy postaládával, ahová a beérkezett leveleit várja. Levélküldéskor a címzett megadásakor ennek a postaládának a címét használjuk. A papíralapú levelezés esetén a postaláda egy doboz, melybe a postás a leveleket teszi. A tulajdonos a kulcsa segítségével kinyitja a postaládát és megnézi, hogy jött-e levele, és amennyiben jött, elolvassa. Az elektronikus levelezés postaládája egy elektronikus háttértárolón található, amely fizikailag a legtöbb esetben egy fájl, ritkábban egy könyvtár. A tulajdonos ehhez általában számítógép-hálózaton keresztül fér hozzá a kulcsa segítségével. Itt is rendszeresen ellenőrizni kell, hogy érkezett-e levél. A beérkezett levelek letöltődnek a felhasználó gépére, és ezeket ott elolvashatja. A hagyományos levelezésnél a levelekhez való hozzáféréshez egy kulcsra van szükség, melynek a jelszó az elektronikus megfelelője.

A levelek címzettjének megadása is hasonló: mindkét esetben rögzíteni kell, hogy a címzett postaládája hol található, és az ottani sok-sok postaláda közül melyik a címzetté.

Természetesen a sok hasonlóság mellett különbségek is vannak, amelyek általában az elektronikus változatot dicsérik. Az e-mail sokkal gyorsabban jut el a címzetthez, bárhol is legyen. Nem ritka, hogy a levél másodperceken belül már meg is érkezik a címzett postaládájába. A levelek küldése ugyanannyiba kerül, bárhol legyen a címzett, vagyis költség szempontjából mindegy, hogy a címzett a szomszéd lakásban él, vagy egy másik kontinensen. A több példányban elküldött levelek nem emelik a levélküldés költségét, nem kerülnek többe egy fillérrel sem, ha nem egy embernek küldünk levelet, hanem pl. a volt osztálytársainknak.

Levelezni valamilyen levelező programmal lehet.
E programok mindegyik megvalósítja az alábbi funkciókat:

- levél küldése közvetlenül, vagy egy listán szereplő címzetteknek (send),
- kapott levelek tartalomjegyzékszerű listázása a „levél témája” (subject) mezőket mutatva,
- válasz adott levélre (reply),
- levél továbbküldése (forward),
- levél tárolása különböző irattartókba (folderekbe),
- levél törlése (delete).

Levelezőprogramok:

- Pegasus Mail: önálló levelezőprogram. Bonyolult, sokrétű, ingyenes program.
- Eudora Pro: önálló levelezőprogram.
- Pine, Elm, Mutt, XF-Mail: főként Linuxon és más UNIX-okon használt szabad, ingyenes, önálló levelezőprogramok.
- MS Outlook Express: egy egyszerű otthoni levelezésre tervezet program, a MS operációs rendszereinek részét képezi.
- MS Outlook: a Microsoft vállalati levelezésre tervezett kliensprogramja, igen sok funkcióval, lehetőséggel. Az előbb említett programokhoz képest „túlbonyolított”. Rendelkezik a levelező kliens melett csoportmunka végzésére alkalmas összetevőkkel is. Az Office programcsomag részét képezi.
- Lotus Notes: az MS Outllok-hoz hasonló komplex keretrendszer, mejnek természetesen az e-mail kliens is része.

A levél megírása a feladó levelezőprogramjának szövegszerkesztőjében történik, melyet a feladó elküld. A címzett a beérkezett levelet saját levelezőprogramjában olvassa el. A két esemény egy sereg programot hoz működésbe, melyek a levél kézbesítését végzik az Interneten keresztül.

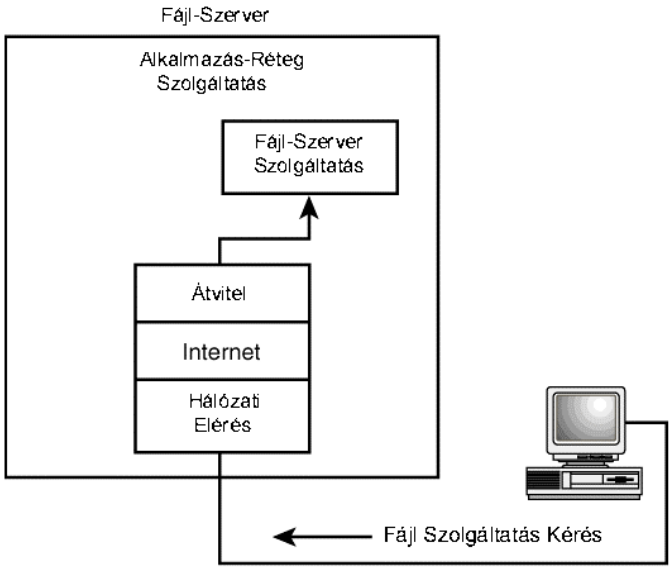
A levelezőprogram a megírt levelet általában nem közvetlenül a címzettnek adja, hanem egy úgynevezett SMTP szervernek adja át.
Előfordulhat, hogy a messzi túloldal felé vezető út zsúfolt, esetleg műszaki probléma akadályozza a gyors, azonnali kézbesítést. Ez esetben szerencsés, ha a levél nem a felhasználó gépén várakozik – növelve ezáltal a felhasználó költségeit –, hanem egy állandóan Internetre kötött számítógépen. Ez az SMTP szerver. Az ő feladata, hogy a levelet továbbítsa, vagy az esetleg sikertelen levélküldést újra-újra próbálja. Ahhoz, hogy levelezőprogramunk az SMTP szolgáltatást igénybe vehesse, vagyis képes legyen levelet küldeni, be kell állítanunk az SMTP szerver címét. Általában ez a mail.sajatzonank.hu nevű gépet jelent. Az SMTP szerver kikeresi a neki átadott levélből a címzett e-mail címét és átadja a levelet a célcímen működő SMTP szervernek, amely beteszi azt a címzett személy postaládájába..

A felhasználó gyanítja, hogy új levele érkezett, ezért megnézi a postaládáját, az ott lévő új leveleket letölti a saját gépére. A postaláda általában nem a felhasználó saját gépén található, hanem az internetszolgáltatónál egy erre a célra üzembe helyezett számítógépen. Vállalati levelezés esetén is általában egy központi számítógép tárolja a beérkezett leveleket.

A beérkező leveleinket tároló szerver címét az internetszolgáltatónktól vagy a rendszergazdánktól kell megkapnunk. Tudnunk kell azt is, hogy melyik protokollt használhatjuk. A POP3 vagy IMAP szerver nevét és a levelesládánk nevét a levelezőprogramunkban be kell állítanunk.

3.2. Állományok átvitele - FTP - File transfer protokoll

Az FTP protokoll a hálózatban lévő gépeken megtalálható fájlok átvitelére használható. Használata az e-maillel szemben már folyamatos hálózati kapcsolatot igényel. Adatátvitelisebesség-igénye is jelentősebb, hiszen elfogadható időn belül kell átvinnünk esetleg több száz kilobájtnyi adatot. Néhány kbit/s-os átviteli sebesség már elfogadható. A szolgáltatás szintén szerver-kliens modellen alapul, azaz egy szolgáltatószerver és a felhasználó gépe közötti fájlok átvitelét biztosítja.



68. ábra
Az FTP kliens-szerver modell

Az FTP protokoll két átviteli módban működhet: ASCII és binary. Az előbbi – mivel 7 bites kódokat használ – szövegállományok átvitelére alkalmas, az utóbbi bármilyen általános fájlra. Fontos továbbá, hogy egyes rendszerek (pl. Unix) különbséget tesznek kis és nagybetűk közt, azaz a fájl nevében ezeket nyugodtan alkalmazhatjuk.

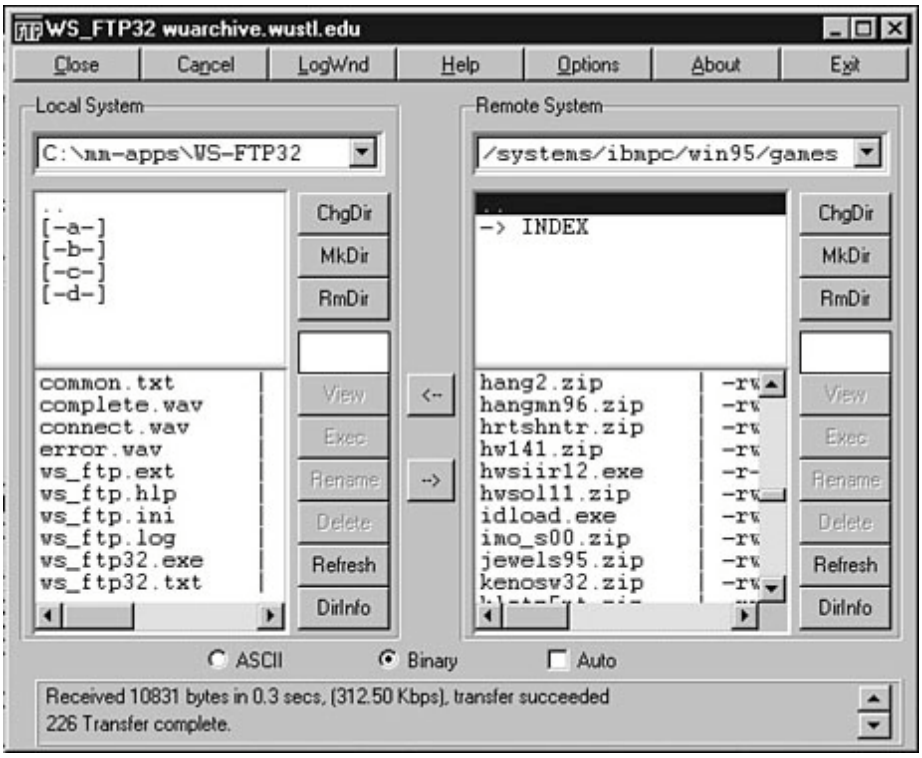
A felhasználó általában akkor tud egy távoli gépről/gépre másolni, ha a távoli gépen is rendelkezik felhasználói jogosultsággal (account-tal).

A kapcsolat egy FTP programmal lehetséges, ott kell megadni a célgép nevét, ami egy internetcím. Ha a kapcsolat létrejött, a rendszer kéri az azonosítót és a jelszót. Ha a belépés sikeres, akkor a következő legalapvetőbb parancsokat használhatja:

- **dir** paranccsal listázhatja a célgép könyvtárszerkezetét,
- **cd** paranccsal válthat a könyvtárak között,
- **get** paranccsal hívhat le fájlokat a távoli gépről,
- **mget**-tel egyszerre több fájlt hívhat le a távoli gépről,
- **put** paranccsal tölthet fel fájlt a távoli gépre,
- **mput**-tal egyszerre több fájlt tölthet fel a távoli gépre,
- az ASCII- és binary-üzemmódok közt az **asc**, illetve **bin** paranccsal lehet váltani.

Vannak mindenki számára elérhető ún. nyilvános elérésű gépek, amelyekre természetesen nem kell account-tal rendelkezni, ez az ún. anonymous ftp. Az ilyen gépekre bejelentkezve bejelentkező (login) névként az "anonymous" szót kell begépelni. A rendszer ekkor arra kér, hogy jelszóként a saját e-mail címünket adjuk meg, ez sokszor gyakorlatilag nem kötelező, kizárólag statisztikai célt szolgál. Ezek után a távoli gépet, pontosabban annak nyilvánosan elérhető könyvtárait láthatjuk, és az összes fenti FTP parancs használható.

A Windows operációs rendszerekben alkalmazhatunk kényelmes grafikus felületet a fájlok átvitelére. Erre mutat példát a következő ábra.

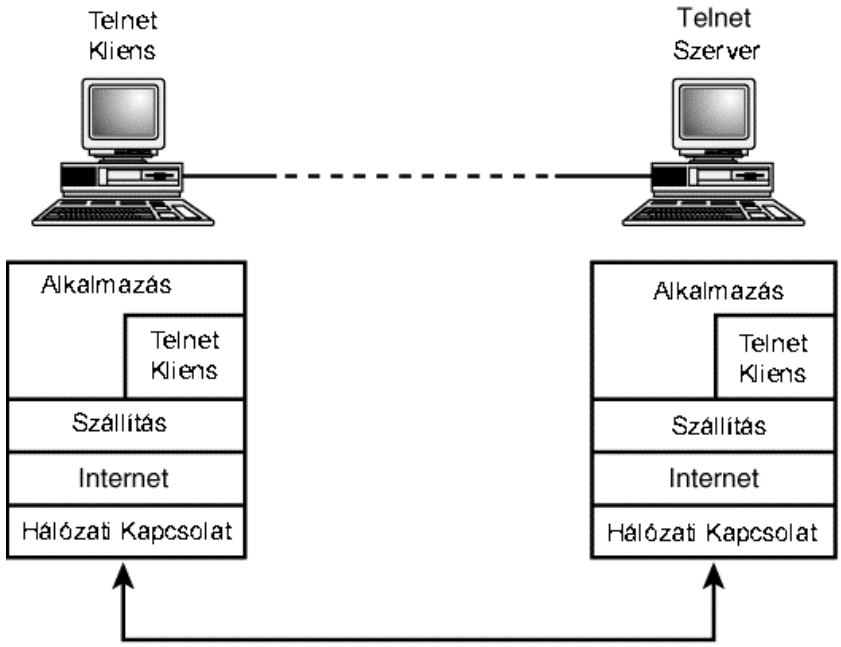


69. ábra
Az FTP kliens program felhasználói felülete

Azok részére, akik csak e-mail-kapcsolattal rendelkeznek, létezik a levéllel történő off-line FTP, az FTPMAIL. Ennek az a lényege, hogy vannak olyan, hálózatra kötött számítógépek, amelyek az FTPMAIL szerverprogramot futtatják. Ez fogadja a leveleket, és feldolgozza bennük az FTP-vel elérni kívánt gép címét és az FTP parancsokat tartalmazó utasításokat. Az FTPMAIL program végrehajtja a kijelölt FTP kapcsolatot, letölti a megadott fájlt, UUENCODE-olja – azaz visszaalakítja a PC számára is érthető 8 bites formába a 7 bites formáról –, majd elküldi levélben a feladónak. Ez egy nem túl kényelmes, de jól használható módszer fájlok letöltésére, ha nincs más mód. Természetesen ehhez pontosan ismerni kell a letöltendő fájl pontos útvonalát is.

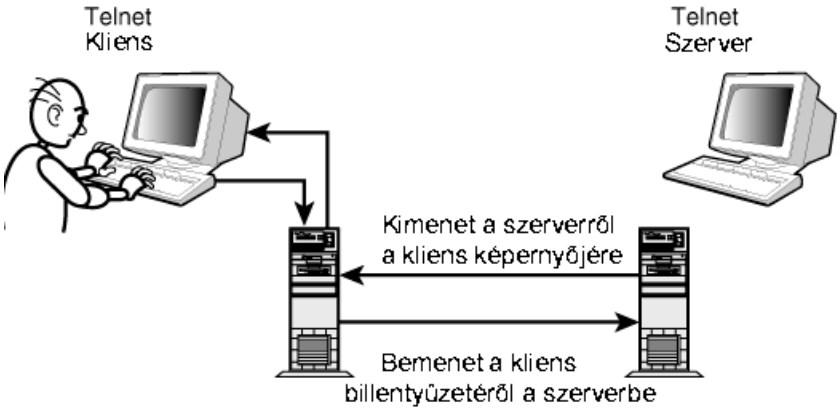
3.3. TELNET

Egy távoli gépre úgy lehet belépni, mintha egy terminálja előtt ülnénk, azaz a TELNET a gépek közti távoli bejelentkezést lehetővé tevő protokoll neve. Ez is folyamatos (on-line) hálózati kapcsolatot igényel, és sebességigénye hasonló az FTP-hez (persze csak ha azt szeretnénk, hogy egy leütött billentyű ne 10 másodperc múlva jelenjen meg...). TELNET-tel csak akkor tudunk egy másik gépre belépni, ha azon a gépen is van accountunk.



70. ábra
A TELNET kliens-szerver szolgáltatás

Bejelentkezés után a rendszer úgy viselkedik, mintha ott ülnénk a távoli gép előtt, azaz a távoli gép operációs rendszerének konvenciói érvényesek.
Parancsainkat a TELNET protokoll adja át a távoli gép operációs rendszerének, és az hajtja végre. Így e távoli gépen programokat futtathatunk, megnézhetjük oda érkezett leveleinket stb.



71. ábra
A kliens-szerver kommunikáció

Ezen lehetőség a hálózati gépek biztonságának egy sebezhető pontja. Ha ugyanis egy távoli gépre rendszeradminisztrátori jogokkal be tudunk lépni (felhasználói név: root, a jelszót pedig próbálkozási módszerrel “kitaláljuk”), akkor a géppel mindent megtehetünk. Az ilyen behatolás módot nyújt arra is, hogy a távoli gépet felhasználva (a TELNET-et ott elindítva) belépjünk egy “kényesebb” gépre. E behatolás felderítésekor a bejutó címe az erre használt gép címe, és ha az oda történő behatolás nyomait eltüntetjük, akkor nem lehet az „elkövető” nyomára bukkanni.

3.4. Gopher

A Gopher nevét a University of Minnesota hivatalos kabalaállatáról, egy kis rágcsálóról (a Gopher szó magyar megfelelője: pocok) kapta, mert itt fejlesztették ki ezt a kliens-szerver filozófiájú rendszert.
A Gopher szervere többfeladatos (multitasking), rendszerint több felhasználós (multiuser) operációs rendszer alatt futó információgyűjtő, információszolgáltató alkalmazás.

A Gopher képes

- hierarchikus struktúrában információkat (lapokat) tárolni,
- kliensektől kiinduló kapcsolatkérésre kapcsolatokat létesíteni,
- kliens kérésére "lépni" föl/le a hierarchikus struktúrán,
- kliens kérésére lapot (fájlt) letölteni a kliens számára,
- kliens kérésére "szolgáltatást" biztosítani.

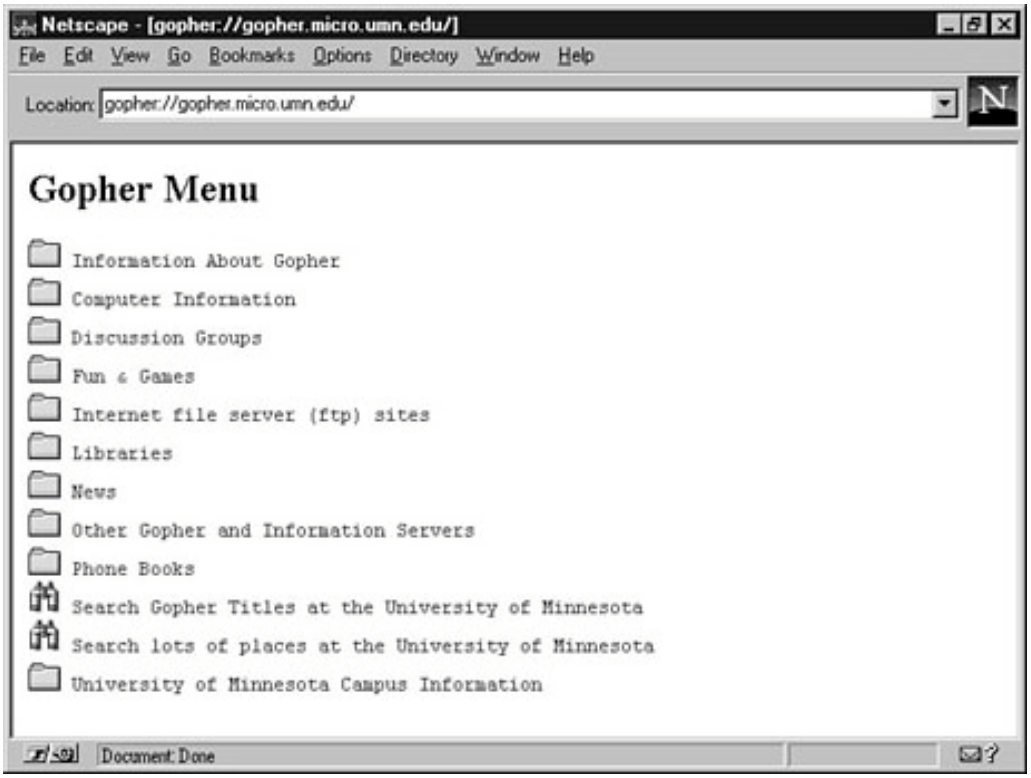
A Gopher kliens szinte minden operációs rendszer alatt futhat, és a felhasználó indíthatja.
A Gopher kliens alkalmas

- kapcsolat létesítésére adott Gopher szerverrel,
- egy-egy "lap" fogadására és "kezelésére".

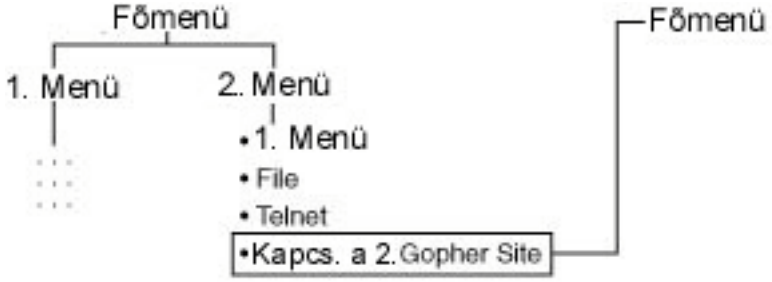
Egy "lap" lehet:

- egy "menü", ami nem más, mint az adott szinten egy jegyzéklista: föl/le léphetünk rajta, kiválaszthatjuk egy elemét (azaz egy további lapot);
- egy szolgáltatás, pl. keresés, átkapcsolás másik Gopher szerverre, egy processz elindítása stb.;
- egy fájl, ami a gopher fájlrendszrének hierarchiájában az utolsó elem. Lehet szöveg vagy kép: ekkor a kliens megjelenítheti (viewer), egy hangfájl, melyet egy lejátszóval meghallgathat (ha a kliens képes rá), egy video stb. A fájlokat le is töltheti a helyi fájlrendszerbe, esetleg postázhatja is.

A kapcsolat a kliens és a szerver között csak addig él, amíg a "lap letöltődik". A letöltött lapot a kliens tárolja, megjeleníti. Nem terheljük a hálózatot, ha egy-egy lapot sokáig nézegetünk, csakis a kliens gép erőforrásait használjuk ez alatt. A TELNET-es kapcsolatépítéshez képest ez nagy különbséget jelent, hiszen ott a kapcsolat addig él, amíg a távoli géppel az összeköttetés fennáll.



72. ábra
Egy Gopher-menü



73. ábra
Kapcsolattípusok egy Gopher-menüben

3.5. A World Wide Web

A WWW az Internet világában forradalmi változást hozott. Hatására az Internet akadémiai, kutatói hálózatból üzleti és hobbihálózattá vált, szerepet kapott a szórakoztatás világában, a tájékoztató média körében; a pénzforgalom és kereskedelem, a reklám világában az üzleti alkalmazások motorjává vált. Hatása akkora, hogy sokan, mikor az Internet kifejezést meghallják, csakis a WWW világára gondolnak.

A WWW koncepciójában a már jól ismert kliens-szerver koncepció mellett három – tulajdonképpen eddig szintén ismert – paradigma fonódik össze:

- a hypertext paradigmája,
- a hypertext utalások kiterjesztése IP-hálózatokra
- a gondolat és a multimédia paradigmája.

A hypertext paradigma lényege olyan szövegmegjelenítés, melyben a lineáris vagy a hierarchikus rendszerű, rendezett szövegolvasás korlátja megszűnik. Elektronikus szövegek lineáris olvasásához elegendő egy egyszerű szövegnézegető (viewer). Már a legegyszerűbb szövegszerkesztő is megfelel, melynek segítségével előre, hátra lapozhatunk a szövegben, sőt, egy esetleges kereső (search) funkcióval már-már átléphetünk egy szinttel feljebb, közelíthetjük a rendezett szövegek olvasásához. Rendezett olvasást biztosítanak a szótárprogramok, adatbázis-lekérdezők. A hypertext jellegű rendszerekben a szövegdokumentumokban valamilyen szövegrészekhez rögzítettek a kapcsolódó dokumentumok is. A megjelenítő valamilyen módon kiemelten mutatja meg ezeket a szövegrészeket. Ezek a kiemelt részek utalások (kapcsolatok, linkek) más dokumentumokra, más szövegekre, szövegrészekre. A hypertext böngésző nem csak kiemelten jeleníti meg a szövegrészeket, hanem lehetőséget ad azok kiválasztására is (pl. mutatóval rákattinthatunk). A kiemelt rész kiválasztásával az utalt, hivatkozott (linked) dokumentum betöltődik a nézegetőbe, folytatható az olvasás, természetesen itt ugyancsak lehetnek utalások, akár közvetlenül, akár közvetetten már előzőleg nézegetett dokumentumra is. Az így biztosított információs rendszer jellegzetesen hálós szerkezetű. Léteznek hypertext szövegeket létrehozó, azokat kezelni tudó információs rendszerek, bár jelentőségük a WWW terjedésével egyre szűkebb.

A hypertext IP hálózatra való kiterjesztése megszünteti azt a korlátozást, hogy az utalások csak ugyanarra a helyszínre, számítógéprendszerre vonatkozhatnak. Egy-egy kapcsolódó dokumentum helye a hálózaton "akárhol" lehet, ha az utalások megfelelnek az Uniform Resource Locator (URL: egységes forrásazonosító) szabványnak.

Végül a multimédia-paradigma megszünteti a szövegekre való korlátozást: nemcsak hypertext-háló, hanem hypermédia-háló alakulhat ki. Hivatkozott dokumentum lehet kép, hanganyag, mozgókép, adatfájl, szolgáltatás stb. is. Ráadásul a képdokumentumokban könnyű elhelyezni további utalásokat is, onnan tovább folytatható a láncolás.

3.5.1. URL (Uniform Resource Locator – Egységes forrásazonosító)

Az egységes forrásazonosító megadja a megjelenítő program számára, hogy az adott szövegrészhez képhez, grafikához kapcsolt dokumentumot milyen módszerrel lehet megjeleníteni, milyen típusú kapcsolatot kell felépíteni, illetve hogy ez a forrás hol, az Internetre kapcsolt gépek közül melyiken található. Ilyen azonosítás a következő:

http://helios.date.hu:70/web/inf/index.htm

A kapcsolt állomány az index.htm nevet viseli a helios.date.hu gépen lévő web/inf nevű könyvtárban. A kiszolgáló a HTTP protokollal érhető el, amely a webszolgáltatáshoz az alapértelmezésként szereplő 80-as port helyett a 70-es portot használja. Az URL a következő információkat tartalmazza:

- a protokollt, amelyet az adott forrás eléréséhez használunk. Ezt az URL első tagja adja meg: ilyen protokollok például az FTP, HTTP, GOPHER stb.;
- annak a kiszolgálónak az internetnevét (domain név) vagy címét (IP-cím) amelyen az adott forrás található. Ez az információ két perjellel (//) kezdődik és egy (/) zárja le;
- a kiszolgáló portjának a számát. Ha ez nem szerepel, akkor a megjelenítő-program az általánosan használt alapértelmezést feltételezi. Ha nem a WWW-hez javasolt 80-as portcímet használják, akkor ezt az URL-ben a kiszolgáló nevéhez vagy címéhez kettősponttal kapcsolva kell megadni;
- a forrás helyét a kiszolgáló lemezegységének hierarchikus állományrendszerében (könyvtár/fájlnév).

Egy adott HTML-kapcsolaton belül az azonos könyvtárban lévő állományok eléréséhez nem kell a teljes keresési útvonalat megadni. Ha egy dokumentumot elértünk a rendszeren, ez már bizonyos információkat szolgáltat a következő kapcsolat felépítéséhez. Így a szomszédos állományok eléréséhez elegendő egy rész-URL alkalmazása, ami az aktuális dokumentumhoz viszonyítva relatív kapcsolódást biztosít. Azonos könyvtárban lévő dokumentumok esetén először csak a teljes URL-t kell megadni, utána már elég a többi fájlnak csak a nevét beírni. A

http://helios.date.hu/

URL esetén a megjelenítőprogram a megadott kiszolgáló főkönyvtárát keresi. A WWW szerver konfigurálásakor megadható, hogy ilyen esetben melyik legyen az a HTML dokumentum, amelyet a kiszolgáló elküld a felhasználónak. Ez lehet pl. üdvözlés vagy információ a szolgáltatásokról, más URL megadása, tartalomjegyzék, hibaüzenet.

A WWW kiszolgálót futtató gépen a felhasználók a saját könyvtárukban lévő, a rendszer konfigurálásakor definiált speciális nevű alkönyvtárban mindenki számára hozzáférhető, személyes HTML dokumentumokat hozhatnak létre. Ezekre a könyvtárakra való hivatkozás UNIX alapú szerverek esetében pl. a ~ karakterrel kezdődik, és a könyvtári hivatkozás a felhasználó neve. A ~ karakter azt jelzi a kiszolgáló számára, hogy ez nem egy szokásos alkönyvtár, hanem az adott felhasználó alkönyvtárában kell az állományokat keresni. Például a „nagy" felhasználói névhez tartozó személyes dokumentumok a

http://helios.date.hu/~nagy/

URL segítségével érhetők el. A kiszolgáló konfigurálásakor meg kell adni annak az alkönyvtárnak nevét, amelyben a felhasználók létrehozhatják személyes dokumentumaikat. Ez a könyvtárnév a kiszolgáló konfigurációs állományában (a UNIX rendszereknél általában a /etc/httpd.conf) található meg (pl. public_html, wwwhomepage).

Ugyancsak a rendszer létrehozása során definiálható annak az állománynak a neve, amely a rendszerbe való belépéskor, illetve a saját könyvtárak címzésekor jelenik meg a felhasználók képernyőjén. Ezt a HTML dokumentumot általában welcome.html vagy index.html névvel látják el.

3.5.2. A HTTP protokoll


A WWW kliensek a böngészőprogramok, a tallózók. Képesek a Hyper Text Markup Language (azaz hiperszöveg leíró nyelv – HTML) direktíváival kiegészített szövegek megjelenítésére, bennük az utalásokhoz rendelt szövegrészek kiemelt kezelésére, a kiemelt szövegek kiválasztására. Képesek bizonyos képdokumentumok megjelenítésére, ezekben kiemelések kiválasztására, hangfájlok, videók lejátszására, közvetlenül, vagy valamilyen segédprogram aktiválásával. A szerverek pedig képesek szöveg-, kép-, hang- és videófájlokat megkeresni saját fájlrendszerükben, és azokat elküldeni a kliensnek megjelenítésre. A kliens és szerver közötti üzenetváltások jellegzetesen négy lépéses forgatókönyv szerint történnek a Hyper Text Transport Protocol (Hiperszöveg-szállítási protokoll – HTTP) szabályozása alatt:

- Az első lépés a kapcsolatlétesítés (connection): a kliens kezdeményezi, ehhez a legfontosabb információ a szerver azonosítója.
- A második lépésben a kliens kérelmet (request) küld a kapcsolaton a szervernek, ebben közli, hogy milyen protokollal, melyik dokumentumot kéri (csak megemlítjük, hogy az átviteli eljárás – a „method" – is paramétere a kérelemnek).
- Ezután a szerver megkeresi a kért dokumentumot és válaszol (response): a kapcsolaton leküldi a kért dokumentumot.
- Végül a kapcsolat lezárul (close). Mindezek után a kliens felelőssége, hogy mihez kezd a leküldött dokumentummal:

ideiglenesen tárolja a saját memóriájában és/vagy fájlrendszerén; a dokumentum fajtájától függően (pl. szöveges vagy képi információ) megjeleníti azt, esetleg külső lejátszó elindításával (pl. hanganyag esetében), de lehetőséget ad a felhasználónak végleges lementésre (fájl eltöltése esetén) stb.
Már a programozás kérdéskörébe tartozik, hogy ha olyan dokumentumot kap a böngésző, melyet közvetlenül nem tud megjeleníteni, lejátszani (futtatni), milyen segédprogramot hívjon meg a megjelenítésre. A felhasználó a MIME-szabványoknak megfelelő lejátszókat beállíthat, rendszerint a böngésző konfigurációs menüjében.

A HTTP ügyfélkiszolgáló protokollt hypertext dokumentumok gyors és hatékony megjelenítésére tervezték. A protokoll állapotmentes, vagyis az ügyfélprogram több kérést is küldhet a kiszolgálónak, amely ezeket a kéréseket egymástól teljesen függetlenül kezeli, és minden dokumentum elküldése után le is zárja a kapcsolatot. Ez az állapotmentesség biztosítja, hogy a kiszolgáló mindenki számára egyformán elérhető és gyors.

3.6. A WWW-alkalmazások fejlesztésének eszközei: HTML, CGI, XML, PHP, JAVA

3.6.1. A HTML

A dokumentumok logikai struktúráját a HTML (Hyper Text Markup Language) jelölései segítségével lehet szabályozni. A HTML arra készült, hogy segítségével a dokumentumok szokásos, sorban egymás utáni olvasása helyett, a szövegben elhelyezett kapcsolatok alapján az egész dokumentum könnyebben legyen áttekinthető és elolvasható. Segítségével logikusan szervezett és felépített dokumentumokat lehet készíteni, az olvasó által kezelhető kapcsolatokat létrehozni ezeken belül és ezek között. A dokumentum fogalmát itt általánosabban kell értelmeznünk: ezek objektumok, amelyek lehetnek: szöveg, kép (grafika), hang (zene), de akár mozgókép (film) is.

A fenti módon szervezett szöveget hypertextnek hívjuk. A folyamatos, sorokba rendezett szöveg végigolvasása helyett a kereszthivatkozásokat követve könnyen át lehet kerülni a szöveg egy más részére, megnézni más információkat, azután visszatérni, folytatni az olvasást, azután megint egy másik bekezdésre ugrani.
Ilyen szerkezetűek a Microsoft Windows, illetve a Windows alatt futó programok súgó (de ilyen szerkezetű ez a tananyag is).
Amennyiben a szöveg mellett más objektum is megjelenik, akkor hipermédiáról beszélünk.

A hálózaton az objektumok, illetve az ezek részei közötti kapcsolatok magába a szövegbe épülnek be megjelölt szavak és grafikus elemek formájában. Amikor egy ilyenre a felhasználó az egérrel rákattint, a rendszer automatikusan létrehozza a kapcsolatot, és a kapcsolt objektumot megjeleníti a képernyőn (vagy ha például hang, akkor lejátssza). Lényeges, hogy a kapcsolt objektum is tartalmazhat további kapcsolásokat különböző objektumokhoz, amelyek elvileg a hálózaton bárhol lehetnek. A WWW úgy is tekinthető, mint egy dinamikus információtömeg, amelyben a hypertext segítségével létrejött kapcsolatok (linkek) találhatók.
Mindezek eredményeként egy adott információ a hálózat bármely pontjáról megszerezhető, illetve ugyanahhoz az információhoz több úton is el lehet jutni a különböző kapcsolatokon keresztül.

A HTML formátumú fájl valójában egy szöveges fájl, szintén szöveges (olvasható) vezérlőkódokkal. Ezek a vezérlőkódok < és > jelek között szerepelnek, és a szöveg megjelenését, formátumát, például a betűk nagyságát, formáját stb. jelölik. A szöveg egyéb dokumentumokra vagy a dokumentum más részeire való hivatkozásokat is tartalmazhat, amit a vezérlőkódok segítségével adhatunk meg linkek formájában. Ezek a linkek – amelyek a megjelenítéskor általában kék színű, aláhúzott szövegekként, vagy kék keretes ikonokként jelennek meg – hypertext alakúvá teszik a dokumentumot. A legtöbbször minden egyes link hivatkozás egy másik HTML oldalra, amely a világháló bármely pontján lehet.

Főbb HTML elemek

<html>...</html>	*A forráskódot ezen jelek közé kell zárni, ez jelzi a dokumentum elejét és végét.*
<head>...</head>	*A fejléc elejét és végét jelzi, ezen belül helyezhetők el a fejlécelemek.*
<title>...</title>	*A HTML oldal címe, ezt olvashatjuk a Netscape fejlécében, és ezt "jegyzi meg" a Bookmark is.*
<body>...</body>	*A HTML dokumentum "teste". Lehet hátteret is készíteni: body background="nev.jpg" vagy "nev.gif" mozaikként bekerül a háttérbe.*

Body elemek *(ezek az elemek lehetnek a HTML oldal <body> </body> elemei között)*

<hn>...</hn>	*Címsorstílusok n=1..6*
* ... *	*Hagyományos szöveg egy bekezdése. A megjelenítő mindig az aktuális ablakszélességének megfelelően tördeli.*
...	*Vastagított (Bold) szöveg.*
...	*Dőltbetűs (Italic) szöveg.*
* ...</br>*	*Sortörés.*
<hr>...</hr>	*Vízszintes választóvonal.*
* ... *	*A közte levő szöveg, kép mindig középre rendezve látszik. Alapértelmezésben minden balra rendezett.*
<blink>...</blink>	*A közöttük lévő rész villog.*
* ... *	*Nem sorszámozott lista. Az egyes listaelemek ... jelek közé kerülnek. Lehet több ilyen lista egymásba ágyazva is.*
* ... *	*Sorszámozott lista. Az egyes listaelemek itt is ... jelek közé kerülnek.*
<table>...</table>	*Táblázat. Az egyes sorok <tr>...</tr> jelek között, azon belül az egyes cellák tartalma <td>...</td> közé zárva. Szerepelhet itt kép is, újabb táblázat nem.*

Linkek létrehozása

...	*A link létrehozásának általános formája. a "..." helyére több dolog írható, például:*
...	*Hivatkozás egy másik HTML fájlra.*
...	*FTP szolgáltatás meghívása.*
...	*Hivatkozás Gopher-menüre.*
...	*TELNET kapcsolat létrehozása.*
...	*Közvetlen levélküldés beépített levélszerkesztővel a címzettnek.*
...	*News szolgáltatás linkelése.*

Színek beállítása

Lehetőség van a háttér, a normál szöveg, a linkszöveg és a már "megjárt" link színeinek egyedi beállítására.
<BODY bgcolor="#....." text="#....." link="#....." vlink="#.....">

A hat pont helyére 3 db kétjegyű hexadecimális szám kerül az RGB szabvány három csatornájának megfelelően. Ezek a számok 00-ff tartományba eshetnek, ami megfelel a decimális 0-255 intervallumnak.

Kép elhelyezése

<img src="könyvtárnév\filenév" [align="..."] [alt="..."] [border=...]>	*Kép elhelyezésének általános formája, a zárójelek közti utasítások opcionálisak.*
align="..."	*A szöveg képhez igazítása: top; middle; bottom.*
alt="..."	*A képhez tartozó alternatív szöveg, karakteres böngészők (pl. Lynx) ezt jelenítik meg a kép helyén.*
border= n	*A képkeret vastagsága n=1…6*

3.6.2. A CGI

A manapság legismertebb WWW-böngészők nem csak a HTTP-protokollt ismerik, hanem más protokollok segítségével nemcsak WWW-szolgáltatókkal tudnak kapcsolatot létesíteni, azoktól szolgáltatásokat kérni. Hogy csak a legfontosabbakat említsük, rendszerint képesek FTP protokollon keresztül állományátvitel-szolgáltatások igénylésére, TELNET protokollal távoli elérésre, Gopher-protokollal Gopher-szolgáltatás és böngészés végzésére, POP3 protokollal levélszekrények vizsgálatára, letöltésére, SMTP vagy MIME protokollal levelek feladására (kapcsolat levéltovábbító szolgáltatóhoz), a USENET NEWS (hírolvasó rendszer) levelek olvasására.
Mindezekhez viszonylag egységes felhasználói felületet biztosítanak, innen adódik tehát az a téveszme, hogy az Internet a WWW, vagy fordítva: hiszen egy jó WWW-tallózó szinte minden szolgáltatást biztosít, amit az Interneten elérhetünk.

Amit eddig elmondtunk a WWW világról, az még mindig nem biztosítja igazán a programozhatóságot. A WWW szolgáltatóknak rendszerint van még további szolgáltatásuk is. A legegyszerűbb "programozási" lehetőség az, hogy bizonyos HTML-utasításokat a szolgáltató szervere dolgoz fel, így gyakorlatilag a szerveren fog futni a HTML-be ágyazott „program”. A szerver elindítja a parancsértelmezőt, végrehajtatja a parancsot, az eredményeit pedig szövegfájl-válaszként elküldi a kliensnek megjelenítésre.
A Common Gateway Interface (CGI) protokoll szerint akár paramétereket is küldhetünk a kliensből a CGI programnak, amely akár bele is írhat az utoljára megjelenített dokumentumba. Maga a CGI program pedig akármilyen nyelv is lehet, gyakran egyszerű burokprogramok (shellcript), többnyire lefordított és szerkesztett futtatható fájlok. (Ne feledjük: a CGI nem egy programnyelv, hanem egy interfész, azt szabályozza, hogy kap és ad információkat, paramétereket.)
Leggyakoribb alkalmazási területük a számlálók és vendégkönyvek elhelyezése a WWW-nyitólapokon, pontosidő-szolgáltatás, keresések a helyi vagy akár távoli WWW-rendszerekben, átjárók adatbázis-lekérdező rendszerekhez, kérdőívek, szavazólapok kitöltetése, de egyéb programozási megoldásokra is alkalmasak.

3.6.3. Az XML

Az XML (Extensible Markup Language) egy leíró nyelv, a strukturált információkat tartalmazó dokumentumok számára.
A strukturált információk kétféle dolgot foglalnak magukba: egyrészt tartalmat (szöveg, képek) másrészt információkat a tartalom struktúrájáról (például hogy az adott helyen lévő szöveg fejléc, lábléc vagy fejezetcím).
A leíró nyelv egy mechanizmus arra, hogy ezeket a struktúrákat azonosítsuk a dokumentumban.
Az XML specifikáció azt definiálja, hogy milyen módon írható le egységesen a dokumentum. A "dokumentum" szó mögött itt nem a hagyományos értelemben vett dokumentumot kell érteni, hanem más XML-adatformátumok sokaságát. Ilyenek lehetnek például vektorgrafikák, matematikai egyenletek stb. Az XML rövid idő alatt az Internet egyik alapvető építőelemévé vált. A világon egyre több vállalat használja különböző e-business alkalmazásoknál.

3.6.4. A PHP

Hivatalos nevén "PHP: Hypertext Preprocessor" (hypertext-előfeldolgozó) azonban már régen kinőtte ezt az utótagot. Mára már a PHP a legelterjedtebb tartalomgenerátor a HTML-oldalakhoz, a PHP-t használó weboldalak száma több millióra tehető. A népszerűség oka abban keresendő, hogy a nyelvet (amint azt a neve is jelzi) kezdettől fogva a HTML-oldalakba ágyazásra tervezték, a fejlesztőkörnyezetek is eleve úgy vannak kialakítva, hogy webszerverhez kapcsolódnak, és a programot ezen keresztül futtatják, az eredményt pedig weboldalként jelenítik meg.
A széleskörű használat következményeként rengeteg kiegészítése készült, adatbázis-kezeléstől képkonvertáláson át grafikus kezelőfelületig szinte mindent tudunk készíteni a segítségével.

A nyelvhez leírást és sok fontos kiegészítést találhatunk a http://www.php.net/ weboldalon.

A HTML-be ágyazottságból kifolyólag alapvetően weboldalak forrásába írunk PHP programot, így meg kell különböztetnünk a dokumentum egyéb részeitől. Ez többféle módon is megtehető; az alábbiakban e módszerekről ejtünk szót.

3.6.5. A JAVA programozási nyelv

Mint említettük, a WWW-böngészőkkel egységes, felhasználóbarát felületet kapott a WWW, ezzel részben az Internet is. A programozás eszközeit – korlátozottan – igénybe lehet venni. A CGI-programokkal, melyek a szerveroldalon futnak, bizonyos feladatokat megoldhatunk, bizonyos alkalmazásokat készíthetünk.

A Sun Microsystems (informatikai nagyvállalat) fejlesztői – felismerve az eddigi programnyelvek korlátozásait – egy tejesen új programnyelvet dolgoztak ki a WWW-programozáshoz, a Java nyelvet. Ezzel párhuzamosan a WWW-tallózók fejlesztői olyan böngészőt készítettek, amelyik a Java nyelven írt programokat képes értelmezni és futtatni. Az ilyen tallózók Java virtuális gépként viselkednek.
A HTML-dokumentumokban a Java programokra való hivatkozások ugyanúgy megtalálhatók, mint pl. a képhivatkozások, és a dokumentum letöltése során akár ezeket is megkapjuk. Az a tény, hogy a program nem a szerveroldalon fut (mint ahogy a CGI-programoknál), hanem letöltődik a böngészőhöz, és az hajtja végre, több előnyt is eredményezett. Egyik előny az, hogy tehermentesítik a szervert, esetlegesen a hálózatot. Másik, talán még nagyobb előny, hogy nem kell a különböző operációs rendszerekhez, géptípusokhoz illeszteni az alkalmazást, a "szabványos" Java kódot a Java virtuális gép, a böngésző végre tudja hajtani, és ennek a feladata az adott hardver, operációs rendszer tulajdonságaihoz való illesztés.
Hátrány is jelentkezik azonban, elsősorban biztonsági kérdések merülnek fel a Java alkalmazások (appletek) futtatásánál. Miután a helyi gépen futtatunk akár bizonytalan eredetű programokat, külön gondot kellett fordítani arra, hogy ne legyen lehetséges vírus- vagy féregprogramokat készíteni a Java nyelv segítségével. Ennek következtében a Java programocskák nem képesek a számukra kijelölt területen túllépni, maguk a böngészők pedig külön kérésünkre további biztonsági szintként nem fogadnak Java alkalmazásokat (amivel el is vesztjük a programozhatóságot). A Java nyelv könnyen megtanulható, különösen C++ ismeretek birtokában.

A Java az Internet közvetlen tartozékának tűnik. Java programnyelven viszonylag egyszerű alkalmazásokat írni, az Interneten keresztül más gépek felé adatokat továbbítani, grafikákat, interaktív weboldalakat, felületeket létrehozni. Természetesen minden más olyan feladat is megoldható, amire a klasszikus programozási nyelvek képesek.
Nem szokás új programozási nyelvet fejleszteni azonban anélkül, hogy ne használnának fel korábbi nyelvekkel kapcsolatos tapasztalatokat. Mi lehet vajon a kapcsolat a korábbi programnyelvekkel?

A Java erősen támaszkodik a C++ nyelvre. Ennek oka a C++ objektumorientáltságában, gyorsaságában és teljesítményében keresendő, de a nyelv korábbi jelentősége az Interneten sem elhanyagolható. Ugyanakkor a Java nyelvet megtisztították rengeteg szükségtelen dologtól, ami a C++ nyelv használatát megnehezítette. Ez a tisztítás a nemcsak a programozók tehermentesítését szolgálja, hanem a helyesen működő programok fejlesztését is garantálja. Egy internetnyelv esetén ez különösen fontos, mivel szakszerűtlenül programozott alkalmazások – amelyek nem megfelelően működnek vagy összeomlanak, lefagynak – nagy kockázatot jelentenek egy számítógép-hálózatban.

Ezek után tekintsük át, mely tulajdonságok jellemzik leginkább e programnyelvet:

- Egyszerű. A Java egy olyan programozási nyelv, amelynek szintaktikája a C++ mintáját követi. Ezzel dolgozik ma a legtöbb objektumorientált szoftvert fejlesztő programozó. A Sun mérnökei e nyelvet a nehezen érthető, és bizonyos esetekben fölöslegesen bonyolult dolgoktól is megtisztították.
- Objektumorientált. A Java objektumorientált nyelv. Ez egy olyan fejlesztési módszert jelent, amelyben újrafelhasználható adatobjektumok megfelelő összekapcsolásával hozzuk létre a kívánt programot.
- Biztonságos. Mivel a Java appletek a felhasználó gépén futnak a hálózatról való letöltés után, lényeges, hogy a letöltött kód ne tartalmazzon hibákat és vírusokat.
- Rendszerfüggetlen és hordozható. A Java egyik legnagyobb erősségét az a képessége jelenti, hogy ugyanaz a kód különböző számítógép-platformokon is futtatható. A fejlesztőknek nem kell a már megírt programot minden olyan platformra külön-külön átírni, lefordítani és hibamentesíteni, amelyen futtatni szeretnék. Bármely gép, amelyen van Java interpreter (értelmező), képes a Java appletek futtatására. A Javat nem érdekli, hogy milyen operációs rendszer van a gépen.
- Párhuzamosságot támogató. A Java lehetőséget ad arra, hogy a végrehajtás egyszerre több szálon fusson (multitasking). Ez rendkívül fontos tulajdonság egy webet megcélzó programnyelvnél, mert így jobb interaktív tulajdonságok és nagyobb valósidejű teljesítmény érhető el.

Appletek és programok

A Java segítségével önálló programokat is lehet írni, amelyek a C++ nyelvű programokhoz hasonlítanak, továbbá olyan appleteket is készíthetnek, amelyek egy böngészőn belül futtathatók. A legtöbb Java kód (a szövegszerkesztőben megírt forrásszöveg, amit a Java lefordít), amellyel találkozunk, böngészőben futtatható applet, és nem önálló program.

- **Appletek**. Az applet olyasmit jelent, mint “kis alkalmazás”, ami alatt a következőt kell értenünk: az appletek nem önálló programok, hanem mindig egy meghatározott környezetet igényelnek, amelyben képesek létezni és végrehajtódni. Ezt a környezetet a WWW-böngészők jelentik, amelyeknek természetesen “Java-képesnek” kell lenniük. Ma már minden modern böngésző biztosítja ezt. Ha egy webdokumentumot egy applettel szeretnénk gazdagítani, akkor egy hivatkozást kell elhelyeznünk a HTML-dokumentumban az appletre. Ha egy internetfelhasználó ezek után kapcsolatba lép a dokumentummal, akkor a böngésző először magát a dokumentumot tölti le, majd mikor a felhasználó arra a helyre jut, ahol az appletnek meg kell jelennie, a böngésző automatikusan végrehajtja azt.

A dokumentum olvasóinak az applet úgy jelenik meg, mint az oldal szerves része, így nekik egyáltalán nem tűnik fel, hogy a háttérben épp egy program fut.

- **Alkalmazások (programok)**. Ellentétben az appletekkel, az alkalmazások teljes értékű önálló programok, amelyek végrehajtásához nincs szükség böngészőre vagy más speciális környezetre (leszámítva a Java-értelmezőt). Az önálló programok futtatásához a Java interpreterét kell használnunk, ez egy olyan speciális program, amely a Java virtuális kódját processzorspecifikus bináris utasításokká fordítja. Az így futtatott alkalmazásoknak olyan képességeik vannak, amelyekkel az appletek nem rendelkeznek, például fájlműveleteket tudnak végezni.

Java fejlesztőeszközök

Az első lépésekhez a Java-programozás területén elvben az egyik fejlesztői környezet éppen olyan alkalmas, mint a másik, feltéve, hogy megfelel a Sun által előterjesztett szabványának. A fejlesztőeszközök közötti különbség inkább csak a kezelési komfort, valamint az ár-teljesítmény viszony tekintetében van. Néhány ismertebb Java fejlesztőkörnyezet:

- **Sun JDK** – Egyrészt minden Java-fordítók "őse", másrészt egy állandóan a fejlődés élvonalában lévő alkalmazás. A JDK (Java Developer Kit) a programozáshoz szükséges eszközök (Compiler, Interpreter, Debugger, AppletViewer stb.) gyűjteménye, amely tartalmazza a Javához szükséges alapkönyvtárakat és egy sor példaprogramot. Nem rendelkezik azonban integrált fejlesztői környezettel, ezért némileg nehézkes a használata. (Windows alatt például csak MS-DOS ablakból használható). A minimális kezelési komfortot kompenzálja a nagy teljesítőképesség, a Java-standard optimális támogatása, kedvező ára (Internetről ingyenesen letölthető), valamint az, hogy mindig friss verzió áll rendelkezésre.
- **Visual J++** - Színes Microsoft-fejlesztőkörnyezet.
- **Borland Java-Add on** – a Borland C++ számára készített bővítés, jelenleg még csak a régi Java-szabványt támogatja, de hamarosan megjelenik az új verzió.
- **JBuilder** – a JBuilder egyesíti a nagy teljesítőképességet a Java-standard optimális támogatásával, valamint az integrált fejlesztői környezetet.

Összefoglalás

Mivel napjaink egyik legfontosabb hálózata az Internet, ebben a fejezetben csak az Internet lehetőségeivel foglalkoztunk.
A „hálózatok hálózatának" kialakulásáról indultunk, összehasonlítottuk az előzőekben tanult ISO OSI ajánlással, majd megvizsgáltuk részletesebben.

Az Internet TCP/IP protokollpárost használ, így a napjainkban használt hálózatba kötött gépek többségén is megtalálható ez a protokollpáros. Ezért rendkívül fontos az ismerete. A protokollok részletes elemzése után az IP címzési rendszerét is megvizsgáltuk.
Ezután már elegendő ismerettel rendelkeztünk ahhoz, hogy a hálózatok összekapcsolásával is foglalkozzunk egy keveset.
A fejezet végén pedig megvizsgáltuk az Internet szolgáltatásait. Természetesen ez a rész is túlságosan összetett ahhoz, hogy mindenre ki tudjunk térni, ezért néhány szolgáltatást bővebben ismertettünk, néhányat pedig csupán érintőlegesen.
Reméljük, hogy elegendő ismerethez jutott mindenki ahhoz, hogy a gyakorlati életben felmerülő problémákat könnyebben megoldja, az Internet nyújtotta lehetőségeket jobban kihasználja.

Ellenőrző kérdések

1. Milyen rétegekből épül fel a TCP/IP protokollcsaládra épülő Internet hálózati modell?
   Válasz
2. Melyek a TCP protokoll feladatai?
   Válasz
3. Melyek az IP feladatai?
   Válasz
4. Mit jelent az A, B, C osztályú címtartomány?
   Válasz
5. Mi a címfeloldás?
   Válasz
6. Mi a domain-név?
   Válasz
7. Mi a név feloldás?
   Válasz
8. Mi az SMTP? Mire szolgál?
   Válasz
9. Mire szolgál a dinamikus hoszt konfiguráló protokoll?
   Válasz
10. Mi az SMTP?
    Válasz
11. Mi az FTP?
    Válasz

12.Mi az a TELNET és milyen biztonsági problémákat okozhat?
Válasz

13. Mi az a Gopher?
Válasz

14.Milyen fő 3 elvet valósít meg a WWW?
Válasz

15.Mi az egységes forrásazonosító?
Válasz

16.Mi a HTML?
Válasz

17.Mit jelent és milyen célt szolgál a CGI?
Válasz

18.Mi az XML?
Válasz

19.Mi a PHP?
Válasz

20.Mi a JAVA?
Válasz

21.Mi az APPLET?
Válasz

Terminológiai szótár

Alhálózati maszk: az IP-címben a hálózat és a csomópont azonosítója az alhálózati maszk (netmask) segítségével választható szét, ezért amikor egy hálózati csomópontot konfigurálunk, az IP-cím mellett az alhálózati maszkot is meg kell adni. Az IP protokoll számára az IP-cím és az alhálózati maszk csak együtt értelmes, mert az IP-cím mindig két részből áll. Az alhálózati maszk hiányában a csomópont nem tudja meghatározni az őt tartalmazó hálózat címét, amely az útválasztáshoz elengedhetetlen. Az alhálózati maszk is 32 bites szám, amelyben 1-esek jelzik a hálózat, 0-k a csomópont azonosítójának IP-címbeli helyét.

ARPANET: az amerikai hadsereg projektje volt a hidegháború idején. Célja az volt, hogy egy olyan elosztott számítógép-hálózatot hozzanak létre, amit nem bénít meg, ha az egyik központja kiesik. Így egy esetleges háború esetén is működőképes irányítási rendszert tudnak biztosítani.

ASCII: American Standard Code for Information Interchange – amerikai szabványos kód a kölcsönös információcserére. Klasszikus 7 bites adatábrázolás: az angol abc betűinek, a számoknak és speciális írásjeleknek bináris kódját tartalmazta. Nem szerepelt benne jónéhány nemzeti karakter.

Átjárók (gateway): a hálózatok egyik fizikai eszköze az összekötetések megvalósítására. Átjárókat (gateway) akkor használnak, ha olyan hálózathoz csatlakoznak, amely felépítése nem követi az OSI-modellt.

Befulladás: a torlódás egy szélső esete, amikor a csomagok annyira feltorlódnak, hogy a csomópont már képtelen feldolgozni.

Broadcast: a hoszt-címrészbe csak 1-eseket írva lehetséges az adott hálózatban lévő összes hosztnak üzenetet küldeni (például a 195.13.2.255 IP-címre küldött üzenetet a 193.13.2 című hálózatban lévő összes gép megkapja).

Broadcasting: a hálózat összes hosztjának szóló üzenet.

CCITT: Comité Consultatif International Télégraphique et Téléphonique – nemzetközi telekommunikációs szabványügyi hivatal. Ma már az ITU (International Telecommunications Union – Nemzetközi Telekommunikációs Egyesület) végzi a telekommunikációban a szabványok kialakítását, a szabványosítással kapcsolatos koordináló feladatokat.

Célhoszt: kommunikáció során ennek a hosztnak küldenek adatokat. Általános értelemben adónak szokták nevezni.

Címfeloldás (address resolution): a számítógépeket alacsonyabb (fizikai, adatkapcsolati) szinten nem az IP-cím azonosítja, hanem a hálózatikártya-azonosító (NetCard ID) vagy hardvercím (hardware address). Feladat tehát, hogy a címzett állomás eléréséhez az IP-címhez meg kell találni az adott IP-címmel rendelkező hálózati kártya hardvercímét. Ez a művelet a címfeloldás (address resolution). A címzett állomásnak az IP-cím alapján való megtalálása a hálózatban az IP protokoll feladata.

Common Gateway Interface (CGI): HTTP szerver bővítését külső programokkal lehetővé tevő első, és mai napig legelterjedtebb felület. A CGI-protokoll szerint paramétereket is küldhetünk a kliensből a CGI-programnak, amely akár bele is írhat az utoljára megjelenített dokumentumba. Maga a CGI-program pedig akármilyen nyelv is lehet, gyakran egyszerű burokprogramok (shellscript), többnyire lefordított és szerkesztett futtatható fájlok. A CGI nem egy programnyelv, hanem egy interfész, azt szabályozza, hogy kap és ad információkat, paramétereket.

CRC: hibajavító eljárás. Cyclic Redundancy Check, ciklikus redundancia-ellenőrzés. A kódot elosztják egy előre meghatározott bitsorozattal, és a maradékot a kóddal együtt továbbítják. A vevő szintén elvégzi az osztást, és a kapott eredményt összehasonlítja a kóddal együtt megkapott ellenőrző bitsorozattal. Ahol a két sorozat nem egyezik, ott hiba volt.

CSM/CD (Carrier Sense Multiple Access with Collision Detection): vivőjel-érzékelés többszörös hozzáféréssel, ütközésérzékeléssel. Az adók figyelik a csatornát, és csak akkor kezdeményeznek adást, ha a csatorna üres. Ekkor is előfordulhat ütközés, ami az adások azonnali megszakítását eredményezi.

Csomag: az üzenetet kisebb egységekre kell darabolni a könnyebb átvitel érdekében. A hálózati réteg által elvégzett darabolás eredménye az adatcsomag.

Datagram: a TCP/IP protokollban az információ datagramban terjed. A datagram (csomag) az üzenetben elküldött adatok összessége. Minden datagram a hálózatban egyedi módon terjed. Ezen csomagok továbbítására két protokoll, a TCP és az IP szolgál. A TCP (Transmission Control Protocol) végzi az üzenetek datagramokra darabolását, míg a másik oldalon az összerakást. Kezeli az esetlegesen elvesző csomagok újrakérését és a sorrendváltozást. Az IP (Internet Protocol) az egyedi datagramok továbbításáért felelős.

DB: decibel, például az üvegszálakban a fény csillapításának mértékegysége.

Dielektromos állandó: azt a számot, amely megadja, hogy hányszorosára nő egy kondenzátor kapacitása, ha a lemezek közti teret

vákuum helyett szigetelővel töltjük ki, az illető szigetelőanyag dielektromos állandójának nevezzük.

Disztribúció: a Linuxos operációs rendszerek általában ugyanarra a magra (kernel) épülnek. Viszont ezt a magot lehet finomítani: más grafikai kezelővel lehet ellátni, különleges funkciókkal lehet kiegészíteni, optimalizálni lehet bizonyos feladatok ellátására stb. A kereskedelemben beszerezhető vagy Internetről letölthető Linux operációs rendszerek már rendelkeznek ezekkel a tulajdonságokkal. A disztribúciók határozzák meg, hogy a csomag összeállításánál mire fektettek nagyobb hangsúlyt. A csomagok könnyebb megkülönböztetése érdekében a disztribúcióknak külön nevet szoktak adni.

DNS-rendszer: a rendszer tervezői hierarchiát építettek fel a számítógépek elnevezésében. Ez azt jelenti, hogy a világot (a névteret) tartományokra (domain) osztották. Az egyes tartományokban megnevezett csomópontok és további tartományok is lehetnek. Minden tartományban van legalább egy névszolgáltató (name server), amely csomópontok esetén meg tudja adni a csomópont nevéhez tartozó IP-címet, illetve a benne lévő tartományok esetén a kérést továbbítani tudja a megfelelő tartomány névszolgáltatójához.

Dynamic Host Configuration Protocol (DHCP): a hálózatokban lehetőség van arra, hogy a számítógépek IP-címeit (és a TCP/IP használatához szükséges egyéb paramétereket) egy vagy több kiszolgáló automatikusan ossza szét. A számítógépek az operációs rendszer betöltésekor a hálózatokban elérhető valamelyik címkiszolgálóhoz fordulnak, amely a rendelkezésre álló címtartományból ad nekik IP-címet. A számítógépek IP-címe így dinamikusan változhat: ezért hívják azt a protokollt, amely segítségével a címkiosztás történik, dinamikus csomópont-konfiguráló protokollnak (Dynamic Host Configuration Protocol, DHCP).

EBCDIC: binárisan kódolt decimális ábrázolás. A nagy számítógépeken használt régi jelkészlet.

Forgalomirányítás: a csomópontok csomagtovábbításának vezérlése.

Forráshoszt: a kommunikációban ez a hoszt a kezdeményező. A forráshoszttól indítunk csomagokat a célhoszt felé. Általános értelemben vevőnek szokták nevezni.

FTP (File Transfer Protocol): a protokoll a hálózatban lévő gépeken megtalálható fájlok átvitelére használható. Adatátvitelisebesség-igénye is jelentősebb. Az FTP protokoll két átviteli módban működhet: ASCII és binary. Az előbbi – mivel 7 bites kódokat használ – szövegállományok átvitelére alkalmas, az utóbbi bármilyen általános fájlra.

Fürtözés: nagyvállalati környezetben szükséges lehet az egyes egységek szervereit összekötni egymással. Ezt a technikát nevezik fürtözésnek. Előnye, hogy így megbízhatóbb hálózatot tudnak

kialakítani, illetve a felhasználók több szervert igénylő munkája is egyszerűsíthető. A jelentősebb hálózati operációs rendszerek már fürtbe köthetőek.

Gopher: a hálózaton való hierarchikus keresésre szolgáló kliens-szerver szolgáltatás. A Gopher-szerverek többnyire könnyen kezelhető menürendszert adnak a kezünkbe, így menüszinteken keresztül lehet az információt megkeresni. Előfordulhat, hogy a Gopher adatbázisa több gépen helyezkedik el, ekkor a Gopher program automatikusan kapcsolja azt a gépet, amely a kért információt tartalmazza

Hamming-távolság: bithibák esetén az a szám, ahány bitben különbözik két kódszó.

Holtpont: a torlódásnak az a különleges este, amikor két csomópont egymásra vár, azaz A addig nem tudja feldolgozni a csomagot, míg B fel nem szabadul, de B addig sem tudja feldolgozni a csomagot, míg A fel nem szabadul.

Hoszt: egyik jelentése szerint gazdagép, a hálózati kommunikációban résztvevő számítógép. Másik (tágabb) jelentése szerint az a berendezés, amely a hálózati kommunikációban részt vesz. Nem feltétlenül számítógép, hálózati kártya, router … E megközelítés szerint, ha egy számítógépben 2 hálózati kártya van, 2 IP-címmel, akkor az 2 hosztnak tekintendő.

HTTP (HyperText Transport Protocol): A Wold Wide Web-szolgáltatások kommunikációs protokollja alapján a böngészőprogramok, a tallózók képesek a Hyper Text Markup Language (HTML) direktíváival kiegészített szövegek megjelenítésére, bennük az utalásokhoz rendelt szövegrészek kiemelt kezelésére, a kiemelt szövegek kiválasztására. Képesek bizonyos kép dokumentumok megjelenítésére, ezekben kiemelések kiválasztására, hangfájlok, videók lejátszására, vagy közvetlenül, vagy valamilyen segédprogram aktiválásával. A HTTP ügyfél-kiszolgáló protokollt hypertext-dokumentumok gyors és hatékony megjelenítésére tervezték. A protokoll állapotmentes, vagyis az ügyfélprogram több kérést is küldhet a kiszolgálónak, amely ezeket egymástól teljesen függetlenül kezeli, és minden dokumentum elküldése után le is zárja a kapcsolatot. Ez az állapotmentesség biztosítja, hogy a kiszolgáló mindenki számára egyformán elérhető és gyors

Hyper Text Markup Language (HTML): a dokumentumok logikai struktúráját jelölések segítségével lehet szabályozni. A HTML azért született, hogy a dokumentumok szokásos, sorban egymás utáni olvasása helyett a szövegben elhelyezett kapcsolatok alapján az egész dokumentum könnyebben legyen áttekinthető és elolvasható. Segítségével logikusan szervezett és felépített dokumentumokat lehet készíteni, olyan módon, hogy a nyelv alkalmas logikai kapcsolatok létrehozására a dokumentumon belül és dokumentumok között, amit a dokumentum olvasója kezelhet. A dokumentum fogalmát itt

általánosabban kell értelmeznünk: ezek objektumok, amelyek lehetnek: szöveg, kép(grafika), hang (zene), de akár mozgókép (film) is.

Hypertext: a hypertext lényege olyan szövegmegjelenítés, melyben a lineáris vagy hierarchikus rendszerű, rendezett szövegolvasás korlátja megszűnik. A hypertext jellegű rendszerekben a szövegdokumentumokban valamilyen szövegrészekhez rögzítettek a kapcsolódó dokumentumok is. Ezek a kiemelt részek utalások (kapcsolatok, linkek) más dokumentumokra, más szövegekre, szövegrészekre. A kiemelt rész kiválasztásával az utalt, a hivatkozott (linked) dokumentum betöltődik, ebben folytatható az olvasás. Természetesen itt ugyancsak lehetnek utalások, akár közvetlenül, akár közvetetten, már előzőleg nézegetett dokumentumra is. Az így biztosított információs rendszer jellegzetesen hálós szerkezetű.

I/O cím: az I/O cím egy háromjegyű hexadecimális szám, mely a processzor és az eszköz közötti kommunikációs csatornát biztosítja.

IMAP Internet Mail Access Protocol: megengedi a kliensek számára a levelek szelektív letöltését a szerverről, ahol a levelek egy hierarchikus struktúrában tárolódnak.

Internetcím, IP-cím: a számítógéphez rendelt azonosítót IP-címnek (IP address) nevezzük. A hálózati csomópontok IP-címe 32 bites szám, amelyet a leggyakrabban az úgynevezett pontozott tízes formában (dotted decimal form) írunk le, azaz négy darab 0 és 255 közötti decimális számmal. Az IP-cím két részből áll: az első a csomópontot tartalmazó helyi hálózatot azonosítja, a másik a hálózaton belül a csomópontot. Az, hogy az IP-címből hány bit a hálózat és hány a csomópont azonosítója, elsősorban attól függ, hogy az összekapcsolt hálózatok rendszerében mennyi hálózatra, illetve hálózatonként mennyi csomópontra van szükség.

InterNIC: hálózatazonosító, amelynek az egész Interneten belül egyedinek kell lennie. Ezért az Internethez csatlakozó hálózatok azonosítóit – a számítógépek IP-címeinek első néhány, 8, 16, vagy 24 bitjét – külső szolgáltató határozza meg. Ezt központilag az InterNIC (Inter-Network Informatition Center) végzi különböző régiók szervezeteinek bevonásával.

IP: az Internet hálózati rétege. A szállított csomagok a datagramok, amely a forráshoszttól a célhosztig kerülnek továbbításra, esetleg több hálózaton is keresztül. A hálózati réteg megbízhatatlan összeköttetés-mentes szolgálatot biztosít, így az összes megbízhatósági mechanizmust a szállítási rétegben kell megvalósítani, ami biztosítja a két végállomás közötti megbízható összeköttetést.

IRQ: az IRQ egy logikai kommunikációs vonal az eszköz és a CPU között, melyet egy – a 0–15 intervallumba eső – egész számmal azonosítunk.

ISA: (Industry Standard Architecture – ipari szabványfelépítés): alaplapi csatlakozó felület. A régebbi típusú számítógép alaplapok legfontosabb csatlakozótípusa. Ma már „kiment divatból”, de a legtöbb alaplapon még megtalálható egy ilyen típusú csatlakozó is.

Java appletek: az applet olyan “kis alkalmazás”, amely nem önálló program, hanem olyan meghatározott környezetet igényel, amiben képes létezni és végrehajtódni. Ezt a környezetet a WWW-böngészők jelentik, amelyeknek természetesen “Java-képesnek” kell lenniük. Ha egy webdokumentumot egy applettel szeretnénk gazdagítani, akkor egy hivatkozást kell elhelyeznünk a HTML-dokumentumban az appletre. Ha egy Internet-felhasználó ezek után kapcsolatba lép a dokumentummal, akkor a böngésző először magát a dokumentumot tölti le, majd mikor a felhasználó arra a helyre jut, ahol az appletnek meg kell jelennie, a böngésző automatikusan végrehajtja azt. A dokumentum olvasói számára az applet úgy jelenik meg, mint az oldal szerves része.

Java: a Sun Microsystems által kifejlesztett programozási nyelv a WWW-programozáshoz. Ezzel párhuzamosan a WWW-tallózók fejlesztői olyan böngészőt készítettek, amelyik a Java nyelven írt programokat képes értelmezni és futtatni. Az ilyen tallózók Java virtuális gépként viselkednek. A HTML-dokumentumokban Java programokra való hivatkozások ugyanúgy megtalálhatók, mint más (pl. kép-) hivatkozások, és a dokumentum letöltése során akár ezek is letöltődnek.

Keret: az adatkapcsolati réteg a felette lévő rétegtől adatokat kap, és az alatta lévő réteg felé adatokat továbbít. A két szomszédos réteg nem képes azonos méretű adatblokkokat feldolgozni, ezért az adatkapcsolati réteg a hálózati rétegtől kapott csomagokat kisebb darabokká, keretekké tördeli, és ezeket a kereteket továbbítja a fizikai réteg felé.

LED: Light Emitted Diode – fényt kibocsátó dióda.

Léptetőregiszter: egy olyan regiszter (átmeneti tároló), ami a feldolgozásra váró biteket tárolja és feldolgozásra „adagolja” bitenként a számítógépnek.

Lézerdióda: lézerfényt kibocsátó eszköz.

Loopback (visszairányítás): az IP-címekben a 127-el kezdődő címek a “loopback” (visszairányítás) címek, nem használhatók a hálózaton kívül, csak a hálózatok belső tesztelésére.

Marker: jelzőbit, amely két állapotot tud felvenni: 1 vagy 0. 1 azt jelenti, hogy a keret foglalt, 0 pedig, hogy a keret szabad.

MIME (Multi-purpose Internet Mail Extensions): olyan kódkonverziós eljárás, amely alapján elektronikus levélben lehetőség van nem ASCII karakterek, képek, hangok küldésére is. Amelyik levelezőprogram ismeri ezt, azzal írható, illetve olvasható akár magyar ékezeteket tartalmazó levél is. A MIME többcélú internet-levelezési bővítés; szabvány bináris

fájlok ASCII-formátumba való konvertáláshoz szükséges eljárás, azért, hogy azok továbbíthatóak legyenek e-mail-ben.

Multicasting: csoportcímzés. A hálózat hosztjainak egy adott csoportjához szóló üzenet. Előnye, hogy egy üzenettel tudunk tetszőleges bekapcsolt hoszthoz szólni.

Nagygép: a számítógépek csoportosítása során használatos kategória. Ezek a számítógépek nagy teljesítményű, nagy tároló kapacitású gépek, melyeket adatfeldolgozásra szoktak használni.

Névfeloldás (name resolution): a számítógépek IP-címeit nehéz megjegyezni és könnyű elgépelni. Természetes tehát a felhasználóknak az az igénye, hogy a számítógépeket az IP-címek helyett könnyen olvasható és megjegyezhető nevek megadásával érjük el. Azonban a TCP/IP protokollkészlet használata esetén a számítógépeket csak az IP-cím alapján lehet elérni, név alapján nem. Ezt a műveletet, amikor a név, vagyis a gép hívatkozási neve alapján keressük meg a névhez tartozó IP-címet, névfeloldásnak (name resolution) nevezik

Paritásbit: hibajelző módszer. Az adó megszámolja, hogy hány 1-es bit van az adatban, és ennek függvényében vagy 1-el, vagy 0-val egészíti ki a kódot (pl. 1, ha páros számú az egyesek összege, 0, ha páratlan). A vevő ugyancsak elvégzi a műveletet és összehasonlítja a kapott és a számított paritásbitet. Ha megegyezik, nincs hiba (vagy páros számú hiba van).

PCI: alaplapi csatlakozó felület. Manapság az általános célú perifériákat általában ilyen típusú csatlakozóval szerelt kártyákon hozzák forgalomba.

Perzisztencia: CSMA/CD módszer esetében ettől függ, hogy az ütközés után mennyi idővel kezdi el a hoszt az adás megismétlését.

PHP (Hypertext Preprocessor): mára a PHP lett a legelterjedtebb tartalomgenerátor HTML-oldalakhoz, a PHP-t használó weboldalak száma több millióra tehető. A HTML oldalakba ágyazásra lett tervezve, a fejlesztőkörnyezetek is eleve úgy vannak kialakítva, hogy webszerverhez kapcsolódnak, és a programot ezen keresztül futtatják, az eredményt pedig weboldalként jelenítik meg.

POP (Post Office Protocol): egy tárolási mechanizmus bejövő levelek számára. Amikor egy kliens csatlakozik egy POP3 szerverhez, a neki szóló összes levelet letölti magának, és nincs lehetőség a levelek szelektálására. A POP protokoll alkalmas levélszekrények vizsgálatára, letöltésére, SMTP vagy MIME protokollal levelek feladására (kapcsolat levéltovábbító szolgáltatóhoz).

Port: a TCP/IP hálózatokban az adatok az egyik portról a másikra továbbítódnak. A portok a gépeken (valamelyik IP-címen) futó

alkalmazásokhoz rendelt adatátviteli címek. A portcímek értéke egy 16 bites szám, értéke 0 és 32767 között lehet.

RCPT: az angol „recipient” (címzett) szó rövidítése.

Redundancia: „szükséges felesleg”. Az az egység képezi a redundanciát, ami nélkül a rendszer ugyan jelen pillanatban működőképes lenne, de előfordulhat, hogy a körülmények megváltozása során már hiányozna a rendszer zavartalan működéséhez. (Például egy oktatóteremben azok a székek, amelyeken az óra alatt nem ül senki, redundanciát jelentenek, mert jelenleg feleslegesek. Viszont egy másik csoport óráján szükség lehet rájuk.) A szünetmentes tápegység is általában redundancia, mert van áramellátás, de egy áramszünet esetén már szükséges a rendszer működéséhez.

Réteginterfész: az OSI modell rétegei különböző funkciókat valósítanak meg. A rétegek által nyújtott funkciók és az átadott információk összessége alkotja a réteginterfészt.

RGB (red-green-blue – vörös-zöld-kék): egyfajta színkeverési modell. A három szín keverési aránya határozható meg a segítségével, és így a megjelenítendő színt adja meg. (Pl.: 0 – 0 – 0 a fekete kódja, 253 – 1 – 1 a piros egy árnyalatának a kódja.)

Routing: útválasztás. A hálózatokban a több csomóponton áthaladó adatcsomag továbbítása esetén a csomópontok az útválasztás segítségével képes eldönteni, hogy melyik kimenetükön küldjék tovább a csomagot, hogy az a lehető legrövidebb idő alatt célba érhessen.

SMTP (Simple Mail Transfer Protocol): alkalmazási protokoll, amely a hálózati felhasználók egymással való kommunikációját teszi lehetővé; leveleket tud küldeni és fogadni.

Szegmens: egy hálózat fizikailag és/vagy logikailag különálló részei.

Szinkronjel: egy állandó időközönként ismétlődő előre definiált impulzus, amit egy ún. jelgenerátor állít elő.

Tárolóregiszter: egy olyan regiszter (átmeneti tároló), amelybe a számítógép behelyezi azokat a biteket, melyeket a csatornára kíván küldeni. Mindaddig itt tárolja, amíg az összes bitet össze nem gyűjtötte.

TCP/IP protokollkészlet: a TCP fogadja a tetszőleges hosszúságú üzeneteket a felhasználói folyamattól, és azokat maximum 64 KB-os darabokra vágja szét. Ezeket a darabokat egymástól független datagramokként küldi el. A TCP feladata az, hogy időzítéseket kezelve szükség szerint újraadja őket, illetve hogy helyes sorrendben rakja azokat össze az eredeti üzenetté. Minden TCP által elküldött bájtnak saját sorszáma van. A sorszámtartomány 32 bit széles, vagyis elegendően nagy ahhoz, hogy egy adott bájt sorszáma egyedi legyen.

Teljes tartománynév (Fully Qualified Domain Name – FQDN): a névfeloldás alkalmazásával az Interneten lévő szolgáltatógép vagy valamelyik csomópont eléréséhez a számítógépeket csomópontnévvel (host name) is megadhatjuk. A csomópontnév tetszőleges, legfeljebb 256 karakterből álló szöveg lehet. Az Interneten az úgynevezett teljes tartománynévvel (FQDN) hivatkozhatunk rá. A tartománynév pontokkal (.) tagolt csomópontnév (host name), amelynek egyes részei a számítógépet tartalmazó szervezetet, illetve a számítógép helyét határozzák meg. Minden csomópontnévhez egyetlen IP-cím tartozik, de egy csomóponthoz (azaz IP-címhez) több név is rendelhető.

TELNET: terminálemuláció segítségével a saját gépet terminálnak használva egy távoli hosztra felhasználóként lehet bejelentkezni. A gépet úgy lehet használni, mintha egy terminálja előtt ülnénk. A TELNET a tehát egy gépek közti távoli bejelentkezést lehetővé tevő protokoll neve. TELNET-tel csak akkor tudunk egy másik gépre belépni, ha azon a gépen is van accountunk.

Torlódás: több csomóponton át futó adatátvitel esetében előfordulhat, hogy két csomópont között ez valamiért lelassul, ezért az újabb csomagoknak sokat kell várni a feldolgozásra. Ekkor a két csomópontot összekötő csatorna „bedugul”, a csomagok feltorlódnak.

Uniform Resource Locator (egységes forrásazonosító – URL): az egységes forrásazonosító megadja a megjelenítő program számára, hogy az adott szövegrészhez képhez, grafikához kapcsolt dokumentumot milyen módszerrel lehet megjeleníteni, milyen típusú kapcsolatot kell felépíteni, illetve hogy ez a forrás hol (az Internetre kapcsolt gépek közül melyiken) található.

Ütközés: üzenetszórásos adatátvitel estén több hoszt egyszerre szeretné használni az adatcsatornát. Ilyenkor a általuk leadott jelek összeadódnak és értelmetlen jelsorozatot eredményeznek.

WWW (World Wide Web – világháló): ez az internetalkalmazás forradalmi változást hozott. Hatására az Internet akadémiai, kutatói hálózatból üzleti és hobbihálózattá vált, szerepet kapott a szórakoztatás világában, a tájékoztató média körében, a pénzforgalom és kereskedelem, a reklám világában, az üzleti alkalmazások motorjává vált. Hatása akkora, hogy sokan, mikor az Internet kifejezést meghallják, csakis a WWW-világra gondolnak.

X.25: egy nyilvános csomagkapcsolt hálózat, melyet a 70-es években fejlesztettek ki. Mára már elavult, de néhol még mindig megtalálható. Általában a maximális sebessége 64 kb/s volt.

XML (Extensible Markup Language): az XML egy leíró nyelv, a strukturált információkat tartalmazó dokumentumok számára. A strukturált információk kétféle dolgot tartalmaznak: egyrészt tartalmat (szöveg, képek) másrészt információkat a tartalom struktúrájáról (például, hogy az adott helyen lévő szöveg fejléc, lábléc vagy

fejezetcím). A leíró nyelv pedig egy mechanizmus arra, hogy ezeket a struktúrákat azonosítsuk a dokumentumban. Az XML specifikáció azt definiálja, hogy milyen módon írható le egységesen a dokumentum. A "dokumentum" szó mögött nem a hagyományos értelmezésre kell gondolni, hanem más XML adatformátumok sokaságára. Ilyenek lehetnek például vektorgrafikák, matematikai egyenletek stb.

XOR: kizáró „vagy”. Igazságtáblázata a következő:

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Az adatátvitel módja alapján megkülönböztethetünk:

- Szimplex adatátvitelt: ebben az esetben egy kommunikációs csatorna van, és csak egy irányba haladhatnak az adatok. A két irányú kommunikációhoz két adatcsatornára van szükség. Ilyen pl. a TV, rádió.
- Félduplex adatátvitelt: ebben az esetben már két irányú kommunikáció is megvalósítható, de a csatornán egyszerre csak az egyik irányba tudnak az adatok haladni. Két irányú kommunikáció ezért csak úgy jöhet létre, ha a csatornát a két fél felváltva használja. Ilyen pl. a CB rádió.
- Duplex kommunikációt: ebben az esetben ugyanazon a csatornán egyidőben mindkét irányba haladhatnak adatok. Ilyen pl. a telefon.

Ajánlott weboldalak

Dr. Kónya László Számítógép hálózatok
Érdekességek a távközlésről

Felhasznált és ajánlott irodalom

1. Alexis Ferrero: *Az örök Ethernet* (Szak Kiadó, 2001.)
2. Andrew S. Tanenbaum: *Számítógép-hálózatok* (Panem Kft., 1999.)
3. Babócsy László, Varga Szabolcs, Wágner Péter Antal: *NetWare 5 hálózatok* (NeTeN Bt., 1999.)
4. Ed Tittel, Kurt Hudson, James Michael Stewart: *Hálózati ismeretek* (Kiskapu Kft., 1999.)
5. Fred Butzen – Christopher Hilton: *Linux hálózatok* (Kiskapu Kft.1999.)
6. Kelley J. P. Linberg: *NetWare 5 Adminisztrátorok kézikönyve* (Novell Press, 1999.)
7. Morten Strunge Nielsen, MCSE: *WINDOWS 2000 és az Active Directory* (Kiskapu Kft., 2000.)
8. Othmar Kyas: *Számítógépes hálózatok biztonságtechnikája* (Kossuth Kiadó Rt., 2000.)
9. Peter Norton, Mike Stockman: *A hálózati biztonság alapjairól* (Kiskapu Kft., 2000.)

Jegyzetek: