Számítógép Hálózatok

E rövid segédlet célja, hogy a "***Számítógép Hálózatok***" című tantárgy előadásain elhangzottak felidézését megkönnyítse, valamint a vizsgára történő felkészülés során vázlatul szolgáljon a hallgatók felkészüléséhez. Ennek megfelelően a segédlet csupán vázlatosan tartalmazza az előadások tananyagát, de felépítésében követi azt. A segédlet a terjedelmi korlátok adta keretek között tartalmazza az előadásokon bemutatott fontosabb ábrákat, táblázatokat, hogy az egyéni jegyzetelést megkönnyítse. A kurzus anyagát az előadások rendszeres látogatásával, és a segédlet gondos áttanulmányozásával célszerű elsajátítani.

Ez a kurzus – és a hozzátartozó segédlet – nem vállalkozhat arra, hogy alapos számítógép hálózati ismeretekkel lássa el a hallgatókat, csupán a szakterület főbb folyamatainak felvázolására célozhatja meg. Ennek megfelelően a téma iránt részletesebben érdeklődők számára segítséget az irodalomjegyzékben felsorolt könyvek jelenthetnek.

Tartalomjegyzék

1. Hálózati alapismeretek

1.1. Célok, történet, nyílt rendszerek, szabványosítás

A számítógép hálózatok létrejöttének legfőbb céljai az ***erőforrások megosztása***, az ***üzembiztonság fokozása***, valamint a ***takarékoskodás*** volt. Az erőforrás megosztás azt jelenti, hogy a hálózatba kapcsolt számítógépeken tárolt programok, adatok a hálózatból bárhonnan elérhetők, a háttértárolókhoz, nyomtatókhoz, egyéb berendezésekhez távolról is hozzáférhetünk. Az üzembiztonságot egy számítógép hálózat megléte azáltal növelheti, hogy valamely egység meghibásodásakor annak kieső funkcióit egy – a hálózatba kapcsolt másik – számítógép veheti át. Számítógép hálózat alkalmazása azáltal lehet takarékos, hogy az egyenként viszonylag olcsó egységek összekapcsolásával létrehozott rendszer olcsóbb – sokszor lényegesen olcsóbb – lehet, mint egy hasonló teljesítményű (szuper)nagyszámítógép.

A számítógép hálózatok érdemben az 1970-es évek közepén az USA-ban jelentek meg. Kezdetben döntően katonai, illetve tudományos célokat szolgáltak, azonos tipusú berendezések között teremtettek kapcsolatot. Széleskörű elterjedésük az 1980-as évekre tehető. Az 1990-es években a lezajlott szabványosítási folyamatok, valamint az olcsó és üzembiztos technológiák elterjedésének eredményeként teljesen általánossá váltak.

A hálózatokkal kapcsolatos elméleti kérdések megoldására, valamint az egymással kompatibilis technológiák kidolgozásának megkönnyítésére szükség volt egy egységes fogalomrendszer kialakítására. Ebbe az irányba tett döntő lépést a ***Nemzetközi Szabványügyi Szervezet*** – ***International Standards Organization***, ***ISO*** [valójában ***International Organization for Standardization***, de ennek rövidítését nehéz kiejteni] – az 1970-es évek legvégén, amikor közreadta az ***Open Systems Interconnection*** – ***OSI*** – névre keresztelt ***ajánlást***, amely olyan egységes alapelveket fogalmazott meg, amelyek nélkül a hálózatok fejlődése mára elképzelhetetlen lenne. A referencia-modell létrejötte megalapozta a későbbi szabványosítási törekvéseket, az egységes terminológia kialakításával jelentősen felgyorsította a számítógépes hálózatok fejlesztését.

Annak érdekében, hogy különböző gyártók különböző típusú berendezései együttműködhessenek, közösen elfogadott ***szabványokra***, és/vagy ajánlásokra van szükségünk. Ha két gyártó eszközei azonos szabvány/ajánlás előírásait figyelembe véve készülnek, akkor ezek az eszközök – szerencsés esetben – képesek az együttműködésre. Amennyiben egy szabványt vagy ajánlást annak kidolgozói szabaddá tesznek – vagyis publikálják, és alkalmazásáért nem, vagy csak minimális licensz-díjat kérnek – akkor ***nyílt szabványról*** beszélünk. Ha az alkalmazott eszközök együttműködése csupán ilyen nyílt szabványok betartását igényli, akkor az adott eszközt ***nyílt rendszerű*** berendezésnek tekintjük.

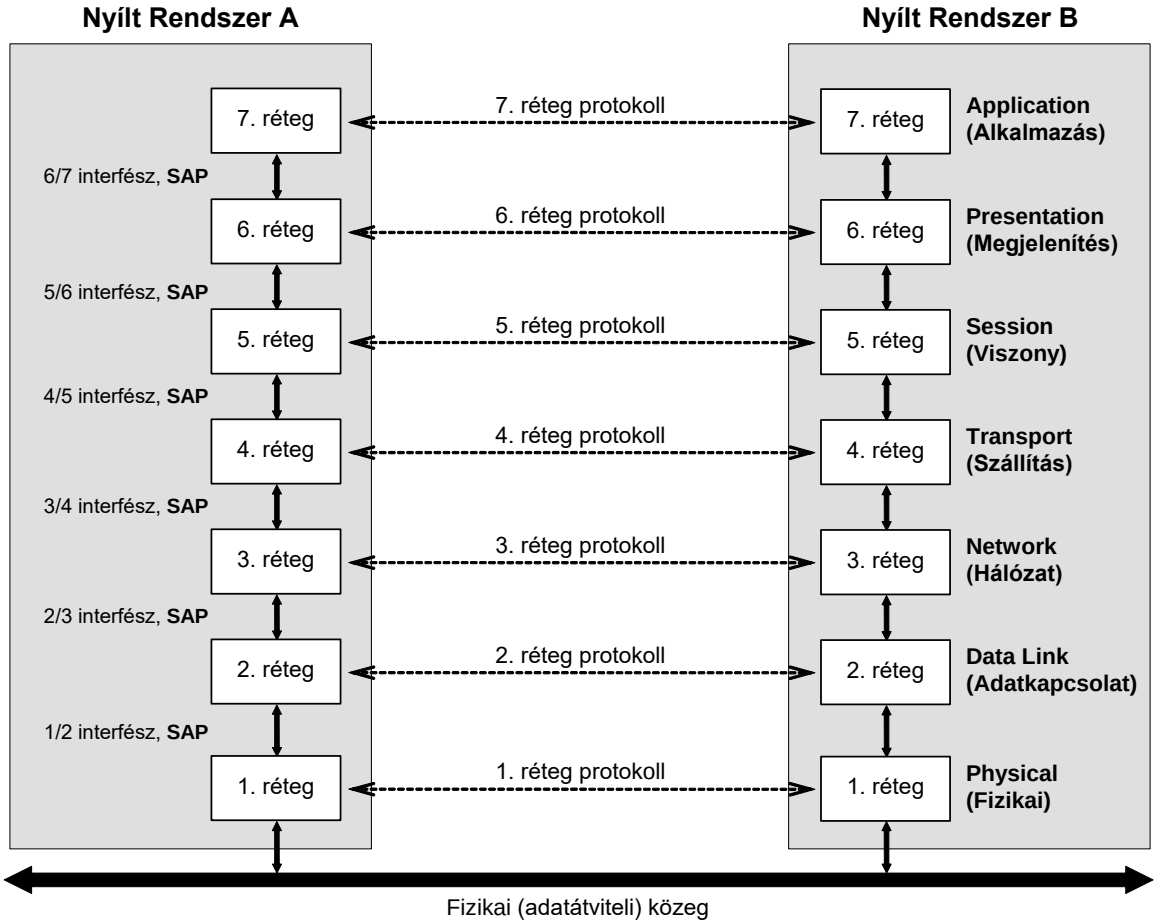
Nyílt rendszeri szabványokat gyakran nemzetközi, non-profit intézmények dolgoztak, és még ma is dolgoznak ki. A hálózati szabványosításban jelentős szerepet játszik a már említett ISO, valamint az ENSZ keretei között létrejött ***CCITT*** – ***Comité Consultatif International Télégraphique et Téléphonique*** [***Consultative Committee on International Telephony and Telegraphy***] – amelyet manapság ***ITU*** – ***International Telecommunication Union*** – néven neveznek. Ugyancsak fontos szabványosítási feladatokat lát el az amerikai ***IEEE*** – ***Institute of Electrical and Electronic Engineers*** – szervezet.

A számítógépes technika és a hírközlés összefonódásával kialakult számítógép hálózati technika mára az ipar egyik legdinamikusabban fejlődő ágazatává vált. A hálózat a mindennapi élet szinte minden területére betört már, gyökeresen átformálva az információ feldolgozásról, tárolásról és szétosztásról – gyakran évszázadokkal – korábban kialakult elképzeléseket.

1.2. Referencia modell, nyílt rendszerek alapfogalmai

Manapság a számítógép hálózatokban döntően az előző fejezetből ismert nyílt rendszerű berendezések üzemelnek. Ezen eszközök fejlesztésének elméleti alapjait az említett OSI referencia modell teremtette meg. Az ISO OSI ***referencia-modell*** értelmében két nyílt rendszer összekapcsolása során végrehajtandó feladatok összessége különféle ***rétegekbe*** – ***layer*** – csoportosítható. A modell értelmében az egymástól távol elhelyezkedő berendezések fizikailag valamely adatátviteli közegen keresztül kapcsolódnak, miközben e kapcsolat kialakításához és fenntartásához szükséges feladatokat a különféle rétegeket megvalósító áramkörök/programok végzik. A modell értelmében az egyes rétegek – kivéve a tényleges kapcsolatban álló 1. rétegeket

– a két távoli nyílt rendszerben egymással csupán ***logikai*** kapcsolatban állnak, miközben tényleges adatátviteli igényeik lebonyolítására a közvetlenül alattuk elhelyezkedő réteg ***szolgáltatásait*** használják fel. E logikai kapcsolat során a rétegek szigorúan meghatározott szabályhalmaz – úgynevezett ***protokollok*** – segítségével érintkeznek (más szavakkal: az adott réteg egy ***protokollgépet*** képez). Miközben egy réteg logikai kapcsolatban áll távoli társával, a közvetlenül felette elhelyezkedő másik réteg számára szolgáltatásokat nyújt. Ezt a szolgáltató felületet ***interfésznek*** – más néven ***Service Access Point***, ***SAP*** – nevezzük. A referencia-modell meghatározza az egyes rétegek által végrehajtandó feladatokat, vagyis a rétegek célját, valamint a réteg – célra jellemző – nevét. (Az egyes rétegek szerepéről más fejezetekben szólunk.)

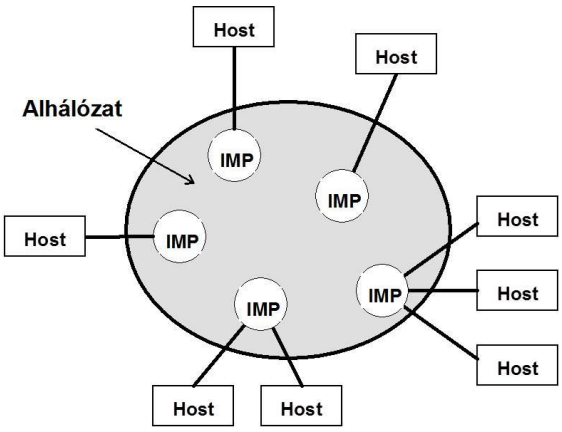


A számítógép hálózatokat különféle csoportokba sorolhatjuk. A leggyakoribb csoportosítási kritérium a ***fizikai kiterjedés.*** A hálózat fizikai kiterjedése ugyanakkor erősen befolyásolja az alkalmazott adatátviteli technológiát is. A csoportok, és főbb jellemzőik a következők:

- ***Személyi hálózat – Personal Area Network: PAN*** – Az áthidalt távolság általában kevesebb mint 10m, az adatátviteli sebesség néhány Mbit/sec. Általában egy (néhány) ember felügyelete alatt álló, közeli gépek összekötését biztosítja (pl. mobil telefon és fülhallgató).
- ***Helyi hálózat – Local Area Network: LAN*** – Az átívelt távolság tipikusan 10…10 000 m, az adatátvitel sebessége 10…1 000 Mbit/sec. Egy LAN többnyire teljes terjedelmében egyetlen tulajdonos fennhatósága alá tartozik, tipikusan homogén adatátviteli technológiát alkalmaz.
- ***Városi hálózat – Metropolitan Area Network: MAN*** – E csoportba tartozó hálózatok tipikus kiterjedése az 10…100 km tartományba esik, sokszor egyetlen városra korlátozódik, azon belül néhány intézményt, egyetemi központot, stb. kapcsol össze. A hálózat több tulajdonos fennhatósága alá is tartozhat, az összekapcsolt számítógépek gyakran eltérő adatátviteli technológiát alkalmaznak. Tipikus adatátviteli sebességnek a 2…155 Mbit/sec tekinthető.
- ***Nagyterületű hálózat – Wide Area Network: WAN*** – WAN-nak nevezzük az országokon belül, illetve országokat (kontinenseket) összekötő hálózatokat. Tipikusan több tulajdonos/szolgáltató felügyelete alá tartozik, gyakran nagymértékben különböző, teljesen eltérő adatátviteli technológiák együttműködését igényli. Manapság tipikus adatátviteli sebességként 28…2 000 Kbit/sec sebesség adódik, de kontinensek közti nagyteljesítményű gerincvezetékek esetében 34…600 Mbit/sec értékek is előfordulnak.

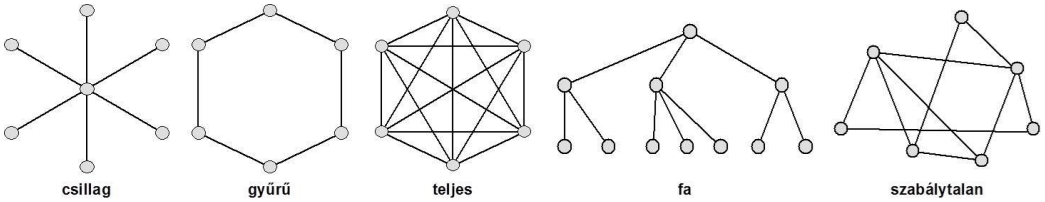
1.3. Topológia, fizikai közegek

A számítógép hálózatokban számítógépeket – más néven ***hoszt***okat (***host***) – kapcsolunk össze. A hosztokat összekötő rendszert gyakran ***alhálózatnak*** is nevezik. Egy-egy alhálózat ***adatátviteli csatornákból***, és az ezeket kezelő speciális egységekből áll. Ez utóbbiakat gyakran ***Interface Message Processor-nak – IMP*** – nevezik. Az alhálózat hosztjai az IMP-khez kapcsolódnak.

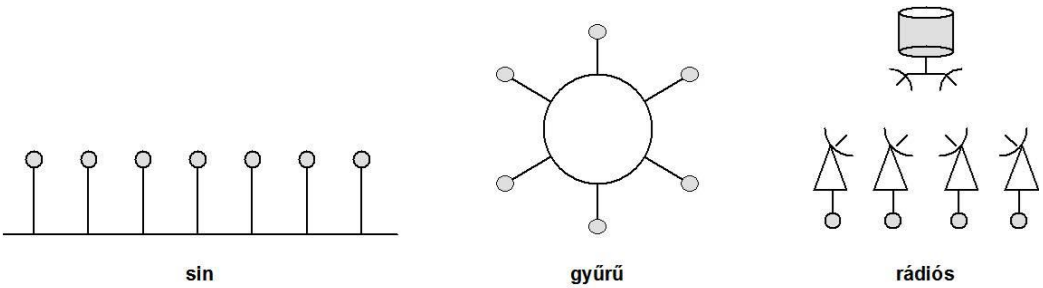


Egy alhálózaton belül az IMP-k különféle módon kapcsolódhatnak egymáshoz. Alapvetően kétféle alhálózati kapcsolattípust különböztetünk meg, ezek a ***pont-pont*** kapcsolatú, illetve ***üzenetszórásos*** alhálózatot.

- A pont-pont kapcsolatú alhálózatban egy IMP egy másik IMP-vel pont-pont kapcsolatban áll (természetesen egy IMP egy időben több pont-pont kapcsolatot is fenntarthat). Az alhálózatot alkotó pont-pont kapcsolatok topológiája alapján az alhálózatot tovább csoportosíthatjuk. ***Csillag***, ***gyűrű***, ***teljes***, ***fa***, illetve ***szabálytalan*** alhálózatokról beszélhetünk.



- Az üzenetszórásos alhálózatban valamennyi IMP egyetlen adatátviteli csatornára kapcsolódik. Ilyenkor az adást minden IMP egyszerre hall(hat)ja. Az ilyen alhálózatok speciális probléma megoldását igénylik, nevezetesen annak vezérlését, hogy az egyes IMP-k közül egy adott pillanatban melyik használhatja adásra a közös adatátviteli csatornát.



Az adatátviteli csatornákat osztályozhatjuk annak alapján, hogy a csatorna végpontjain elhelyezkedő berendezések egyszerre adhatnak és vehetnek-e adatokat, vagy sem. E csoportosítás alapján három kategóriát különböztethetünk meg:

- ***Szimplex*** adatátvitelről beszélünk, ha adatok csak az egyik végpontból a másikba áramolhatnak.
- ***Félduplex – half duplex*** – adatátvitel az, amikor adatok az egyik végpontról a másikra és viszont is áramolhatnak, de egy adott időpontban csak az egyik irányban.
- ***Duplex – full-duplex*** – adatátvitelről akkor beszélünk, ha az adatok mindkét irányban egyszerre haladhatnak az adatátviteli csatornában.

Osztályozhatjuk az adatátviteli csatornát a fizikai közeg típusa szerint is. Leggyakrabban ***vezetékes átvitelt*** alkalmaznak. Ilyenek a ***sodrott érpár*** – más néven ***UTP*** (***Unshielded Twisted Pair***) – illetve a ***koaxiális*** kábel. Mind gyakrabban találkozhatunk ***üvegszálas*** hálózatokkal is. ***Vezeték nélküli*** hálózatok esetén fény (lézer), illetve rádiós átvitelt használnak. Ez utóbbi esetben kistávolságú átvitelre sokszor ***szórt spektrumú*** rendszereket alkalmaznak, míg nagy(obb) távolságra ***mikrohullámú***, illetve ***műholdas*** megoldásokat.

1.4. Adatátviteli csatornák felhasználása, multiplexálás

Két pont között adatokat sokféleképpen továbbíthatunk. Beszélhetünk ***párhuzamos*** – ***parallel*** – átvitelről, amikor egy-egy elemi adatcsoportot – több vezeték felhasználásával – egy időben továbbítunk. Ez a fajta adatátvitel költségessége, és gyakorlati korlátai miatt a számítógép hálózatokban nem terjedt el (felhasználására tipikus példa a számítógép és nyomtató közötti kapcsolat). Számítógép hálózatokban kizárólag ***soros*** – ***serial*** – átvitelt alkalmaznak, amikor a továbbítandó digitális adat bitjeit az egyetlen csatornán időben egymás után továbbítják.

Soros átvitel esetén beszélhetünk ***aszinkron*** – ***asyncronous*** –adatátvitelről. Ekkor az adó- és vevőáramkörök időbeli stabilitása csak rövid időre – csak egy-egy karakter átvitelére – biztosítható. De ehhez is az szükséges, hogy az átvitel során úgynevezett ***start/stop biteket*** alkalmazzanak, amelyek átvitele viszont csökkenti az átvitt hasznos információ mennyiségét.

Szinkron – ***syncronous*** – átvitel esetén az adó és vevőáramkörök a kapcsolat teljes ideje alatt összehangoltan működnek. Ehhez az időzítések igen szigorú betartása szükséges. Ez csak úgy biztosítható, ha az adatátvitel során alkalmanként úgynevezett ***szinkron karakterek*** átvitele is megtörténik. Ezek átvitele azonban lényegesen kevesebb időt igényel, mint a start/stop biteké.

Egy adatátviteli csatorna – más néven ***adatátviteli közeg*** – létrehozása és üzemeltetése gyakran jelentős költséggel jár. A kiadások csökkentése érdekében igyekeznek maximálisan kihasználni minden egyes rendelkezésre álló adatátviteli csatornát. Ezért a hosztok közti összeköttetést biztosító ***gerincvezetékeket*** – ***back-bone*** – úgy használják, hogy azokon egy adott időben több adatátviteli kapcsolat is fenntartható legyen, vagyis az adatátviteli közeget ***alcsatornákra*** osztják, ***multiplexelik***. Az alcsatornákra osztásra elvileg több lehetőség is kínálkozik:

- Létrehozhatunk alcsatornákat úgy, hogy ***frekvenciaosztást*** – ***Frequency Division*** – használunk. Ekkor egy-egy alcsatorna jelét különböző ***vivőfrekvenciákra*** "ültetjük rá", majd a modulált vivőfrekvenciákat egyetlen szélessávú átviteli csatornán keresztül továbbítjuk. Ekkor egy-egy alcsatorna a vonal sávszélességének egy (kis) részét folyamatosan elfoglalja. E módszerre legismertebb példaként a rádióadók rendszere említhető.

- Az adatátviteli csatornát (szinkron) ***időosztással*** – (Syncronous) ***Time Division*** – is feloszthatjuk. Ilyenkor minden alcsatorna egy (előre kiosztott) ***időszeletet*** kap, amely ideje alatt az eredeti csatornát egyedül használhatja. Az adatátviteli csatorna mindkét végén szinkronban dolgozó kapcsolók végzik az alcsatornák kapcsolását. Ilyen rendszerre példa a telefontechnikában alkalmazott ***PCM*** – ***Pulse Code Modulation*** – rendszer. (8 kHz mintavételi frekvencia [125µsec], logaritmikus 8 bites mintavételezés [64 kbit/sec], 32[E1]/24[T1] csatorna [2 048/1 544 kbit/sec])

  A hatékonyság további növelésére alkalmazzák még a ***statisztikus multiplexelés*** módszerét, ahol a gerincvezeték alcsatornáinak tényleges száma kisebb, mint a kiosztott alcsatornák száma. Ezt az teszi lehetővé, hogy az alcsatornák egy része időszakosan kihasználatlan lenne.

  A távoli hosztok közti kapcsolat kiépíthető úgy, hogy a kapcsolat kezdetén a hosztok között elhelyezkedő gerincvezetékek alcsatornáit felhasználva kiépül egy adatátviteli út, majd az így létrejött kapcsolaton zajlik az adatcsere. Ezt ***vonalkapcsolás***nak – ***circuit switching*** – nevezik. A kiépült "vonalat" aztán a két végponton elhelyezkedő hoszt kizárólagosan használhatja mindaddig, amíg a "vonal" él. Tipikus vonalkapcsolt rendszer a telefonhálózat.

  A vonalkapcsolás során az összekapcsolt hosztok fix sávszélességű csatornát kapnak, még akkor is, ha azt nem tudják teljesen kihasználni. Ez rontja a vonal gazdaságosságát. Megoldásként adódik a ***csomagkapcsolás*** – ***packet switching*** – alkalmazása. Ekkor az átviendő adatokat (kis) részekre – csomagokra, packet – osztják, majd ezek a csomagok utaznak az adatátviteli hálózaton. ***A csomagkapcsolás jól alkalmazkodik a változó adatátviteli igényekhez, ezért a számítógép hálózatokban manapság egyeduralkodónak tekinthető.***

  A csomagkapcsolás alkalmazása azonban problémákat is felvet. Elsőként gondoskodni kell a csomagok kialakításáról. Ennek érdekében szabályokat – bitsorrend, min/max csomagméret, csomag kezdet- és végjel, stb. – kell alkotni. E szabályok összessége része az adatátviteli protokollnak. A sikeres adatátvitel érdekében e szabályokat minden az adattovábbításban résztvevő hosztnak és IMP-nek be kell tartania.

1.5. Keretképzés, hibajavítás, forgalomvezérlés

A digitális adatátvitel során a folyamatosan áramló információt ***keretekbe*** – ***frame*** – tördelve viszik át az adatátviteli csatornán. A keretek használata lehetővé teszi az adatátvitel során óhatatlanul fellépő átviteli hibák felismerését. A keretképzés, és felhasználása azonban összetett feladat, számos kérdés megoldását igényli. Ezek közül a legfontosabbak a következők:

- ***Keretképzés***. Ha elhatároztuk, hogy a digitális adatokat keretekbe tördelve továbbítjuk, megoldást kell találni arra a kérdésre, hogyan lehet a keret elejét és végét felismerni. Egy lehetséges megoldás a ***karakterszámlálás***. Ekkor minden keret elején a keretet alkotó karakterek száma áll, majd a karakterek következnek. Ez egyszerű megoldás, de ha a vevő elvéti a számolást – "kiesik a szinkronból", ami az átviteli hibák miatt előbb-utóbb biztosan bekövetkezik – akkor többé nem képes a soron következő keretek elejének – és így hosszának – felismerésére. Sokkal jobb megoldás, ha a keretek elejét valamilyen speciális jelsorozattal jelöljük meg, ami nem fordulhat elő az átvitt adatok között. Ha ASCII karaktersorozatot továbbítunk, akkor e célra ***vezérlőkarakterek*** használhatók. A keret elejét pl. a szabványos **DLE** – ***Data Link Escape*** – **STX** – ***Start of Text*** – karakterpár, míg végét a **DLE**, **ETX** – ***End of Text*** – páros jelölheti. Ez a módszer akkor kerül bajba, ha az átviendő információban előfordulnak az említett kódpárosok, mert az a keret téves kezdet/vég-detektálását eredményezné. Ez ellen úgy védekezhetünk, hogy a kritikus karaktersorozatokat ***karakterbeszúrással*** – ***character stuffing*** – átalakítjuk az adóban, majd visszaalakítjuk a vevőben. A karakterbeszúrás hátránya, hogy erősen kötődik az ASCII kódrendszerhez, előnytelen bináris és/vagy nem 8 bites adatok esetén. Manapság inkább ***bitbeszúrást*** – ***bit stuffing*** – alkalmaznak. Ekkor a keret elejét és végét jelölő speciális jel a "**01111110**"– ***flag*** – bitsorozat. Az adó elkerülendő a kétértelműséget az átvitt adatban az 5 egymást követő "**1**"-es után beszúr egy "**0**"-át. A vevő viszont törli a vett adatból az 5 "**1**"-est követő "**0**"-át, helyreállítva így az eredeti adatot. Ez a módszer azon túl, hogy nem feltételezi az adatok 8 bites kódolását, az adó és vevő közti szinkronizálás problémáját is megoldja. Egyrészt minden csomag elején/végén biztosan előfordul a flag, valamint az esetleg kihasználatlan vonalon folyamatosan flag-eket küldenek.
- ***Hibajavítás***. Az adatátvitel során nem bízhatunk abban, hogy a vevő (hibátlanul) vette az elküldött adatokat. Ezért a szokásos eljárás a következő:

1. Az adó – az alkalmazott protokoll által kijelölt szabályok szerint – keretekbe tördeli az átviendő információt, de eközben minden keret elejére egy sorszámot, végére pedig egy – az átvitt információ tartalmára jellemző – ellenőrző számot – ***CRC***, ***Cyclic Redundancy Check*** – illeszt, majd elküldi az így kiegészített keretet, egy időben elindítva egy ***időzítőt*** – ***timer*** – is.
2. A vevő észlelve a keretet, annak tartalmából – az adóoldalival azonos szabályok szerint – ugyancsak elkészíti a CRC-t, majd összeveti azt a keretben érkezett CRC-vel. Ha a két érték eltér, ez átviteli hibát jelez. Ezután a vevő ***nyugtázó keretet*** – ***ACK***, ***acknowledgement*** – küld az adónak, amelyben az adott sorszámú keret helyes/helytelen vételét jelzi.
3. Az adó a beérkező nyugtából értesül a sikeres/sikertelen vételről. Ha hiba történt, az adó újra elküldi a sérült keretet. Ha a keret elküldésekor elindított időzítés úgy telik le, hogy nem érkezik vissza nyugta, ez azt jelenti, hogy a keret – vagy a nyugta! – az átvitel alatt elveszett. Ekkor az adó az adott keretet ismét elküldi.

   Az itt leírtak csupán a valóságos megoldások elvi vázlatát adják. A gyakorlatban kifinomult – és sokféle – módszert alkalmaznak a keretek sorszámozására, a CRC előállítására, a nyugták feldolgozására, valamint az átviteli hibák hatását kiküszöbölő újraküldésre.

- ***Forgalomvezérlés***. A számítógép hálózatokban különféle teljesítményű adatátviteli vonalak és számítógépek együttműködése valósul meg. Ez felveti annak problémáját, mit tegyünk, ha gyors adó lassú vevővel kényszerül együttműködni. Ekkor az adó előbb-utóbb elárasztaná adatcsomagokkal a vevőt. Ennek megakadályozására a vevő és az adó közötti alkalmas visszacsatolással forgalomvezérlést – ***flow control*** – kell alkalmazni, amellyel a vevő az adó sebességét a számára elfogadható mértékre csökkentheti. A hibajavításhoz hasonlóan számos különféle módszer ismert, ezek elve azonban hasonló: az adó csak korlátozott számú keretet küldhet – ezt a korlátot gyakran ***ablaknak***, ***window*** nevezik – a vevő felé, és e korlát kimerülésekor le kell állnia mindaddig, amíg a vevőtől a további adást engedélyező jelzést nem kap. A különféle forgalomvezérlő módszerek erősen eltérő vonali hatékonyságot eredményezhetnek.

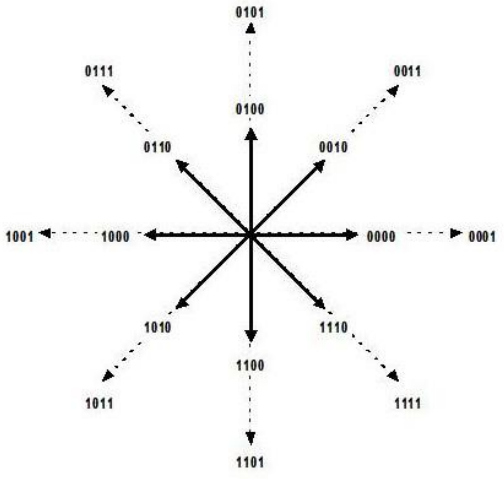
2. Adatátviteli módszerek

2.1. Adatátvitel nyilvános analóg telefonhálózaton: modemek

A 100 éves történelemre visszatekintő telefonhálózat napjainkban a legelterjedtebb vezetékes adatátviteli rendszer. Megléte jelentősen befolyásolta a számítógép hálózatok kialakulását, és ma is segíti azt. Ezt technikailag az alapozta meg, hogy bizonyos korlátok mellett a telefonhálózat felhasználható adatátvitelre is. Ahhoz, hogy az analóg átvitelre épített rendszer digitális adatátvitelre legyen képes, speciális berendezések alkalmazása szükséges. Ezeket modulator-demodulátornak, röviden ***modemnek*** nevezik.

A modern digitális telefonközpontok üzembe állítása mellett is a központ és a telefonkészülék közti vonalszakasz – az "***előfizetői hurok***" – a legutóbbi időkig nem sokat változott. Manapság is igaz, hogy sávszélességét mesterségesen a ***300…3 400 Hz*** tartományra korlátozzák, és alapsávi egyenáramú átvitelt használnak. A telefonmodemeknek ehhez a környezethez kell alkalmaz-kodniuk. Annak érdekében, hogy különböző típusú modemek is együtt tudjanak működni, számos nemzetközi szabványt alakítottak ki. Ezek a szabványok elsősorban a modemek ***funkcionális protokolljait*** definiálják. A korszerű telefonmodemek egy időben három funk-cionális protokollt is használnak az adatátvitel során. Ezek a következők:

- Kötelezően alkalmazandó a ***modulációs protokoll***. Segítségével alakítjuk át a digitális jelet analóg jellé. ***Amplitúdó***-, ***frekvencia***-, ***fázis***- és ***kombinált*** modulációt ismerünk. Manapság leginkább kombinált (fázis + amplitúdó) modulációt alkalmaznak. A modulációs eljárások esetén az információt rendre a vivő-frekvencia amplitúdója, frekvenciája, illetve fázisa hordozza. A kombinált moduláció eredményeként a vonalon egyetlen jelváltozással 1-nél több bitnyi információt továbbítanak, (a másodpercenkénti jelváltozások sebességét ***baud***-ban mérik), vagyis 1 baud≠1 bit/sec. Az ábrán 1 baud=4 bit/sec fázis + amplitúdó modulált jel ***Nyquist-diagramja*** látható (0-45-90-135-180-225-270-315° fázis, két amplitúdó érték). Két modem csak akkor képes egymással kapcsolatba lépni, ha van ***közös*** modulációs protokolljuk. A korszerű telefonmodemek az alkalmazható szűk sávszélesség ellenére viszonylag nagy átviteli sebességet érnek el (általában 28 800 bit/sec). Egy manapság elterjedten használt szabványos modulációs protokoll pl. a CCITT V.34 (28 800 bit/sec).



- A modemek közötti adatátvitel során (gyakran) fellépő hibák kijavítását szolgálja az opcionális ***hibajavító protokoll***. Ha a kapcsolatba kerülő két modem rendelkezik közös hibajavító proto-kollal, akkor hibamentes adatátvitelt biztosítanak. A vonali hibák változatlanul jelen vannak, de a két modem – az adatok újraküldésével – kiszűri azokat. A hibajavítás egyrészt rontja az adatát-vitel teljesítményét – plusz adatok átvitele – másrészt viszont fokozza a teljesítményt, mert az adatokból úgy képeznek csomagokat, hogy azokban nincs szükség start/stop bitek átvitelére. Egy manapság elterjedt hibajavító protokoll pl. a CCITT V.42 (LAP-M).

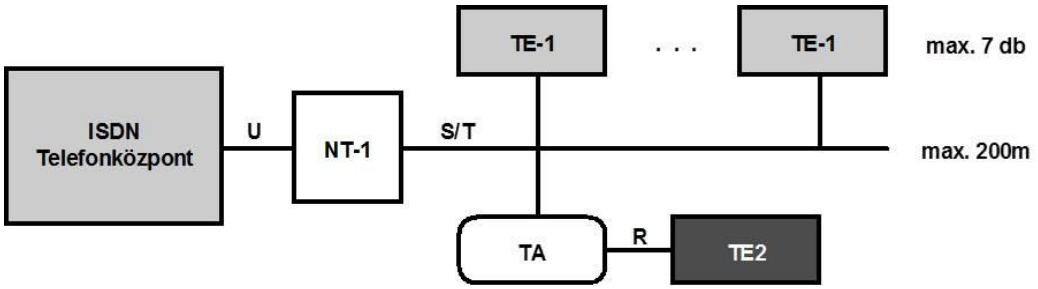
- A harmadik – ugyancsak opcionális – funkcionális protokoll az ***adattömörítő protokoll***. Alkalma-zása csak akkor lehetséges, ha a két modem egyidejűleg hibajavító protokollt is alkalmaz. Az adattömörítő protokoll segítségével az adó-modem az átvitt információt valós időben kom-presszálja, majd a vevő-modem dekompresszálja azt. Fontos megjegyezni, hogy a modemekben alkalmazott valós idejű tömörítés kisebb hatékonyságú, mint az ismert tömörítő programok teljesítménye. Egy manapság elterjedt adattömörítő protokoll pl. a CCITT V.42bis, amely elméletileg max. 4:1 tömörítést eredményezhet. (már tömörített adatok átvitelekor a modem tömörítő protokollja valójában még ronthatja is az átviteli sebességet).

  A személyi számítógépek és az Internet elterjedésével egyre több telefonmodem üzemel. A számítógép és a modem közti kapcsolat többnyire a PC soros portján – vagy azzal funkcionálisan egyező – áramkörön keresztül zajlik. A kapcsolatban a PC a ***DTE – Data Terminal Equipment*** – , a modem a ***DCE*** – ***Data Circuit-terminating Equipment*** – szerepét tölti be.

2.2. Digitális adatátvitel: ISDN

A nyilvános telefonhálózatban a központ és az előfizető közti vonal – az előfizetői hurok – csavart érpárú rézkábel, amelyen az átvitt analóg jel sávszélességét mesterségesen 300…3 400 Hz tartományra korlátozzák. Ha eltávolítjuk a ***sávszűrőket***, és a jeleket digitálisan továbbítjuk, akkor a meglévő vezetéken mintegy ***2Mbit/s*** átviteli sebesség érhető el. Az ilyen alapelven működő rendszert ***ISDN*** – ***Integrated Service Digital Network*** – nevezzük.

Az ISDN tervezésekor célul tűzték ki, hogy egyszerre legyen képes hang, kép, és számítógépes adatátvitelre. Ennek érdekében az ISDN több csatlakozási felületet – ***interface*** – definiál, amelyeken keresztül a digitális adatátvitel lebonyolítható. Egy ISDN kapcsolat elvi vázlata az alábbi ábrán látható:



A központ és az előfizetői végberendezés – az ***NT-1, Network Termination*** – között az ***U*** interface teremt kapcsolatot. Itt ***144kbit/s*** sebességű, kéthuzalos, full-duplex, digitális adatátvitel zajlik. Az NT-1 egység feladata, hogy az U interface-t ***S/T*** interface-re alakítsa. Ez utóbbi már 4-huzalos, és ide maximálisan 7 db ***TE-1*** – ***Terminal Equipment*** – egység kapcsolható. A TE-1 egységek telefonok, fax-készülékek, stb. lehetnek. Egy speciális TE-1 egység a ***TA*** – ***Terminal Adapter*** – amely az S/T buszt ***R*** busszá alakítja. Az R busz nem más, mint egy hagyományos telefonvonal, így ide a megszokott telefon és fax berendezések – ***TE-2*** – kapcsolhatók. Mindezek az egységek akár egyetlen készülékbe is építhetők, de lehetnek különállóak is.

Az S/T felületen két, egyenként ***64 kbit/s*** sebességű ***B*** – ***bearer, hordozó*** – csatornát, valamint egy ***16 kbit/s*** sebességű ***D*** – ***delta*** – csatornát találhatunk. A B csatornák szolgálják – akár egyidejűleg is – az adatátvitelt, míg a D csatorna szerepe a hívások felépítéséhez, és a TE-1 egységek közötti adatcseréhez szükséges kommunikáció biztosítása (***alácímzés!***). Az S/T buszra tetszőleges funkciójú TE-1 egységek kapcsolhatók, így a B csatornák több célra is felhasználhatók. A B csatornákat külön-külön használhatjuk (de szükség esetén össze is vonhatjuk, 1*128 kbit/s), így egy időben akár két telefonbeszélgetés, vagy pl. egy fax- és egy adatátvitel bonyolódhat, természetesen más-más hívószámokra is. Az S/T buszt úgy is tekinthetjük, mint egy kis lokális hálózatot, ahol az adatok a B csatornákon, a címzéshez és vezérléshez szükséges információk a D csatornán áramolnak.

- A leírt konfigurációt ***BRI***-nek ***– Basic Rate Interface*** – nevezik. Létezik egy nagyobb teljesítményű változat is, ezt ***PRI***-nek ***– Primary Rate Interface*** – nevezik, ebben ***30 B*** csatorna, és egy – de 64 kbit/s sebességű – D csatorna található (ekkor az U interface sebessége 1 984 kbit/s).

Egy ISDN központ – a régi analóg központokhoz hasonlóan általában – ***vonalkapcsolást*** végez, vagyis a hívás felépítése után közömbös számára, hogy a kiépült csatornán milyen típusú adatátvitel zajlik. Ha a kapcsolásban csak ISDN központok vesznek részt, úgy nincs is probléma, de régi – analóg – központok vonalba lépése átviteli problémát okoz. Ezért a digitális átvitel előnye csak homogén ISDN kapcsolat alatt használható ki. Az ISDN rendelkezik olyan szabványos megoldásokkal, amelyek segítségével a hívás-felépítés során jelezhetjük a kiépítendő csatorna célját, így a központ erre figyelemmel választhat az alternatív útvonalak között (csak hang átvitel, vagy tényleges adatátvitel).

A leírtaknak megfelelően az ISDN előnyösen alkalmazható pl. két lokális hálózat összekötésére, a megszokottnál lényegesen gyorsabb fax-átvitelre, video konferencia lebonyolítására, stb. Fontos megjegyezni, hogy az ISDN ***hívásfelépítés*** igen gyors, így pl. egy közönséges modemes híváshoz viszonyítva a kapcsolás akár 10…20-szor gyorsabb is lehet (a szokásos 20…30 sec helyett, tipikusan 1…2 sec).

2.3. Digitális adatátvitel: Ethernet

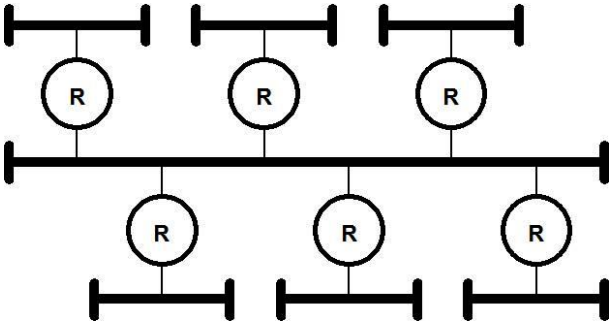
Napjaink leggyakrabban alkalmazott lokális hálózati technológiája az ***Ethernet***. Előnye, hogy szolgáltatás/ár viszonya más megoldásokhoz képest a legjobb, valamint lényegében minden alkalmazott hálózati protokoll továbbítására felhasználható. Az Ethernet valójában egy korlátozott hosszúságú, ***sin*** topológiájú kábelrendszer, amelyhez korlátozott számú hoszt kapcsolható. Az eredeti – ***IEEE 802.3***, 1983 – specifikáció szerint a kábelen az adatátvitel sebessége ***10 Mbit/s***. Az alkalmazott kábelfajta szerint több különféle Ethernet-típust különböztetünk meg:

- ***Vastag*** – ***thick, 10BASE5***: Vastag Ethernetet elsősorban gerinchálózatok kialakítására használnak, amelyre több kisebb hálózat csatlakozhat. A kb. 1.5cm átmérőjű – általában sárga színű – kábel ***maximum 500m hosszú*** lehet, és ***legfeljebb 100 hoszt csatlakozhat rá***. A kábelen 2.5m-enként jelzés található, a hosztok – a kábel megszakítása nélkül – itt csatlakozhatnak egy ***vámpír fognak*** – ***vampire tap*** – nevezett eszköz segítségével. A kábel hullámimpedanciája 50 Ω, és mindkét végén egy-egy 50 Ω-os ***lezáró ellenállást*** kell használni. A vastag Ethernet előnye a viszonylag hosszú kábel és az üzem közbeni bővítés lehetősége, míg hátránya – a nehezen hajlítható kábel következményeként – a nehézkes telepítés.

- ***Vékony*** – ***thin, 10BASE2***: A vékony Ethernet szegmens kb. ½ cm átmérőjű, könnyen hajlítható, 50 Ω hullámimpedanciájú, legalább 50 cm hosszúságú, végein ***úgynevezett BNC csatlakozóval*** ellátott, egymáshoz ***T-csatlakozóval*** kapcsolt kábelek összessége. A szegmens ***maximális hossza 185 m*** lehet, és ***legfeljebb 30*** kábeldarabból állhat. A szegmens két végén a vastag Ethernet-hez hasonlóan 50 Ω-os lezáró ellenállásokat kell alkalmazni. Egy-egy hoszt a T-csatlakozóknál kapcsolódhat a kábelre. A vékony Ethernet előnye a könnyű telepítés és a kis költség, míg hátránya a korlátozott hossz, valamint az, hogy a bővítés megzavarja a már üzemelő hosztokat (bővítéskor a szegmenset rövid időre meg kell szakítani).

- ***Csavart érpárú*** – ***Twisted Pair, 10BASE-T***: A csavart érpárú Ethernet ***csillag*** topológiájú rendszer, ahol a középpontban egy ***hub***-nak nevezett aktív egység található. A hub-hoz csatlakoznak az – egyenként ***max. 100 m*** hosszúságú – kábelek, míg a hosztok a kábelek másik végén találhatók. Különféle UTP kábelek léteznek, ezeket ***2…5 szintűeknek*** nevezik. A magasabb szintű kábel nagyobb átviteli sebességet tesz lehetővé. 4-es szintű kábel szükséges a 10Mbit/s sebességhez, de 5-ös szintű kábellel már ***100 Mbit/s*** sebesség is elérhető. Ezt nevezik ***100BASE-TX***-nek. 1997-ben jelentette be a ***Lucent Technology*** a 6-os szintű kábelt – és a hozzá tartozó illesztőkártyát – amellyel ***1.2 Gbit/s*** sebesség érhető el. A csavart érpárú Ethernet előnye az egyszerű és olcsó telepítés, az üzem közbeni bővítés lehetősége, valamint a megnövelt sebesség, hátránya viszont a hub szükségessége és a viszonylag kis kábelhossz.

- ***Üvegszálas*** – ***Fiber Optic, 10BASE-FL***: Az üvegszálas Ethernet nagyon hasonló a csavart érpárúhoz, de adatátviteli közegként ***üvegszálat*** használunk. A kábelek ***maximum 2 km*** hosszúak lehetnek. Itt is megnövelhetjük az átviteli sebességet ***100 Mbit/s***-ra, ezt ***100BASE-FX***-nek nevezik. Az üvegszálas Ethernet előnye a megnövelt kábelhossz, valamint az elektromágneses zavarok elleni védelem. Hátránya viszont a telepítés magas költsége, a drága hub szükségessége, valamint az üvegszálas kábel mechanikai sérülékenysége.

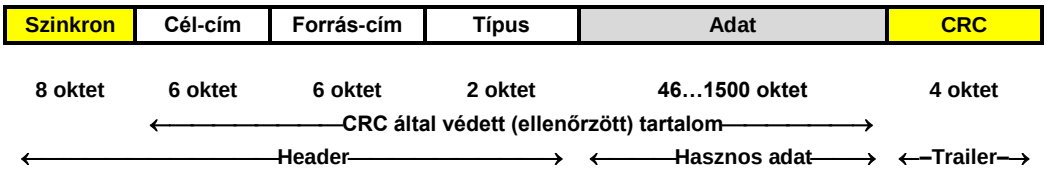


Bármelyik említett Ethernet típust használjuk is, a hálózat méretét ***repeater***-ek és ***bridge***-ek segítségével még tovább növelhetjük. A ***repeater*** olyan aktív eszköz, amely erősítőként és jelformálóként működve lehetővé teszi több szegmens összekapcsolását. (A korábban említett hub lényegében egy sokkapus repeater.) Felhasználásuk további előnye, hogy egy szegmens meghibásodásakor a hiba arra a szegmensre korlátozható. Repeaterek alkalmazásakor ügyelni kell az "***5-4-3***" szabály betartására. Eszerint két hoszt között legfeljebb 5 szegmens állhat, a jel 4 repeater-nél többen nem haladhat át, és csak 3 szegmenshez kapcsolódhatnak hosztok (a másik 2 szegmens csak repeatereket kapcsolhat össze). Az ábrán egy lehetséges hálózat vázlata látható.

Az említett korlátozások oka, hogy a jel a vezetékeken ugyan ***csaknem fénysebességgel terjed***, de mégis időre van szüksége, amely késleltetéseket az Ethernet hálózat tervezésekor figyelembe vettek. Ha a tervezett időzítések nem teljesülnek, ez a rendszer hibás működéséhez vezethet.

A ***bridge*** olyan eszköz, amely a jelformáláson túl az Ethernet szegmenseket – a címek vizsgálatára alapuló ***forgalom-szűrés*** felhasználásával – ***logikailag szétválasztja***. A bridge valamely hoszttól érkező adatot csak akkor enged át egy másik szegmensre, ha az adatcsomag formailag hibátlan, és a megcímzett hoszt a másik szegmensen található. A bridge általában automatikusan "megtanulja", hogy melyik hoszt melyik szegmensen található.

- Az Ethernet hálózat – bármilyen legyen is fizikai topológiája – úgy működik, mint egy sin, vagyis egy hoszt csomagokat küldhet ugyanazon Ethernet hálózat bármely másik hosztjának, miközben a hálózat minden hosztja egyszerre hallja az adást. A kiküldött csomag – más néven ***frame*** – 72…1 526 oktet hosszú lehet, és alakja a következő:



A ***szinkron*** bitsorozat a csomag elejének felismerését segíti, tartalma: **10101010…10101011.** Az Ethernet ***címek*** (cél-, forrás-) 48 bitesek, amelyet többnyire 6 hexadecimális számként olvasnak ki: **aa:bb:cc:dd:ee:ff**. Minden Ethernet kártya rendelkezik egy saját címmel, amelyet adáskor a forrás-cím helyére illeszt, illetve csomag vétele során összehasonlít a cél-címmel. A ***típus*** mező célja a csomagban szállított adat típusának megjelölése. Az ***adat*** mező tartalmazza a hasznos információt. A mező mérete 46…1 500 oktet között változhat. A ***CRC*** mezőt az adó számítja ki, és illeszti a csomag végére, tartalma a kiküldött oktetek tartalmától függ. A vevő a CRC ismételt kiszámításával, majd a számított- és a beolvasott érték összevetésével felismerheti a csomag tartalmának átvitel alatti esetleges sérülését.

Amint említettük, minden Ethernet kártya **– *központilag kiosztott*** – egyedi címmel rendelkezik, amely ***egyértelműen azonosítja*** (az utolsó bit 0, vagyis a cím páros). Ha a cél-cím páratlan, ez azt jelzi, hogy az adó egyszerre több vevőnek szánja a kiküldött csomagot. Ezt ***multicast*** címzésnek nevezik. A multicast speciális esete, amikor a cél-cím minden bitje 1. Ezt ***broadcast*** címzésnek nevezik, és az így feladott csomagot a hálózat ***összes*** hosztja beolvassa.

Mivel az Ethernet hálózatban egy időben csak egyetlen adó működhet, ezért biztosítani kell az adásra várakozó hosztok ***ütemezését***. Fontos jellemzője az Ethernetnek, hogy e feladatot nem egy központi, kitüntetett egység látja el, hanem ***valamennyi hoszt*** szabályozott együttműködése biztosítja. Az alkalmazott módszert ***CSMA/CD***-nek – ***C***arrier ***S***ense ***M***ultiple ***A***ccess / ***C***ollision ***D***etection – nevezik, és vázlatos működése a következő:

1. Egy adásra várakozó hoszt köteles "belehallgatni" a vonalba, és mindaddig – illetve még további 9.6μs ideig (***Minimal Packet Spacing***) – várakozni, amíg a vonalon egy másik hoszt adását érzékeli.
2. Ha a vonal már "csendes", a hoszt megkezdheti az adást, amit mindaddig folytathat, amíg a csomag végére nem ér, illetve összeütközést – a vonalra küldött, és az egyidejűleg visszaolvasott jel különbségét – nem észleli. Összeütközéskor még néhány oktetet – ***collision jam*** – köteles kiküldeni, majd saját adását meg kell szakítania.
3. Összeütközés után a hoszt köteles egy véletlen hosszúságú – ***backoff*** – ideig várakozni, majd az 1. pont szerint ismét próbálkozni. Ha egymás után 16 összeütközés történt, véglegesen vissza kell vonulnia (magasabb szintű beavatkozás szükséges).
4. A backoff idő ***n****51.2 μs – 512 bit-idő –, ahol ***n*** értéke az ismételt próbálkozások száma.

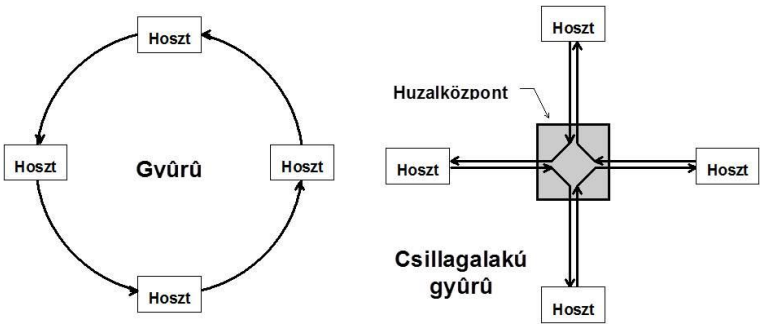
Mindezek a szabályok együttesen ***nagy valószínűséggel*** biztosítják, hogy központi irányítás nélkül is a hálózaton egy időben csak egy adó legyen aktív. Mindamellett extrém körülmények között előfordulhat, hogy egy adásra várakozó hoszt "nem jut szóhoz". Ennek valószínűsége ugyan kicsi, de nem lehetetlen. Ha ezt nem engedhetjük meg, más közeg-hozzáférési módszert kell alkalmaznunk.

2.4. Digitális adatátvitel: Token Ring, FDDI

Amint említettük, bizonyos esetekben az Ethernet technológia a hosztok számára nem biztosít azonos prioritású, kiszámítható hozzáférést a hálózathoz. Ilyen igény kielégítésére sikeresen alkalmazható az alább ismertetett ***Token Ring*** és ***FDDI*** megoldások valamelyike. Közös tulajdonságuk, hogy az úgynevezett ***vezérjel továbbadás*** – ***token passing*** – közeg-hozzáférési technikát, valamint gyűrű topológiát alkalmaznak.

A ***Token Ring*** elnevezés márkanév, az ***IEEE 802.5*** szabványban definiált vezérjeles gyűrű technológiát alkalmazó – az ***IBM*** által kifejlesztett – lokális hálózati rendszer neve. A Token Ring-ben általában UTP kábelt alkalmaznak, az adatátviteli sebesség 4 Mbit/s (létezik 16 Mbit/s sebességű változat is, ekkor ***STP*** – ***Shielded Twisted Pair*** – kábelt kell alkalmazni). A gyűrűre legfeljebb ***72*** hoszt csatlakozhat (STP esetén ***260***).

A gyűrű topológiának megfelelően egy-egy hoszt csak két közvetlen szomszédjával áll kapcsolatban. Az állomások között kifeszülő kábelek végül ***zárt gyűrűt*** alkotnak, amelyben az adatok egy ***előre kikötött irányban*** körbehaladnak. A gyűrű működése során nincsenek olyan szigorú időbeli megkötések, mint amilyenekkel az Ethernet technikánál találkoztunk, így a kábelhosszak sem annyira kritikusak (valójában a túl "kicsi" gyűrű okoz problémát). Egy kábelszakadás azonban a teljes rendszer működésképtelenségét eredményezi. Ez ellen egy "***huzalközpont***"-nak nevezett eszközzel védekezhetünk, amely a gyűrűt ***"csillag alakú gyűrű***"-vé alakítja.



A gyűrűben alaphelyzetben egy speciális csomag, a ***vezérjel*** – ***token*** – kering. Ha egy hoszt adni kíván, ezt mindaddig nem teheti meg, amíg hozzá nem kerül a token. Amikor ez bekövetkezik, először el kell távolítani a tokent – más ne szerezhesse meg azt –, majd megkezdődhet a csomag adása. A kiküldött csomag egymás után áthalad a soron következő hosztokon, míg eléri a megcímzettet, amely lemásolja azt magának, de továbbküldi a soron következőhöz. Így a csomag végül visszajut a feladóhoz, amely eltávolítja azt a gyűrűből. (A sikeres vételről is értesítés küldhető. A csomagban egy mezőt a feladó töröl, majd útjára indítja a csomagot. A vevő sikeres vétel esetén úgy küldi tovább a csomagot, hogy ezt a mezőt beállítja. Mivel a csomag visszajut a feladóhoz, a mező ellenőrizhető.) Mindezek után az adó újabb csomag küldésébe kezdhet – ha van még kiküldésre várakozó, és nem járt le a "***token birtoklási idő***" –, illetve továbbadhatja a tokent a soron következő hosztnak. (Ha lejárt a token birtoklási idő, köteles azt továbbadni.)

- Az ***FDDI*** – ***Fibre Distributed Data Interface***, ***ANSI X3.139*** – nagyon hasonló a Token Ringhez, de itt az adatátviteli közeg két – egy ***elsődleges*** és egy ***másodlagos*** – üvegszálas gyűrű. A gyűrűk hossza maximum ***200 km***, az alkalmazott átviteli sebesség ***100 Mbit/s*** (ennél gyorsabb változatok is működnek már), a hosztok száma maximum ***1 000*** lehet. Az adatátvitel normálisan az elsődleges gyűrűn zajlik, a másodlagos gyűrű – amelyben az adatáramlás iránya ellentétes az elsődleges gyűrűhöz képest – csak biztonsági tartalékot képez.

  A közeghozzáférés módja csaknem teljesen azonos a Token Ringnél megismerttel, a különbség mindössze annyi, hogy egy csomag kiküldése után az adó a tokent ***azonnal*** visszaküldi a gyűrűbe, még mielőtt a feladott csomag – a gyűrű körbejárása után – visszatérne a feladóhoz.

  Az említett technikák azon túl, hogy ***felülről korlátos időközönként garantálják*** az adás lehetőségét, egyszerű tervezést, és teljesen digitális működést biztosítanak. Nincs szükség minimális csomagméret bevezetésére sem, továbbá nagy terhelés esetén a hatékonyság akár 100% is lehet (kis terhelésnél viszont romlik). A rendszer legnagyobb hátránya – a gyűrűbeli token karbantartásáért felelős – ***felügyelő állomás szükségessége***. (Általában bármely hoszt lehet felügyelő is, hogy egy meghibásodás ne okozza a gyűrű teljes leállását. Kezdetben a hosztok ***versennyel*** döntik el, ki legyen az aktuális felügyelő állomás.)

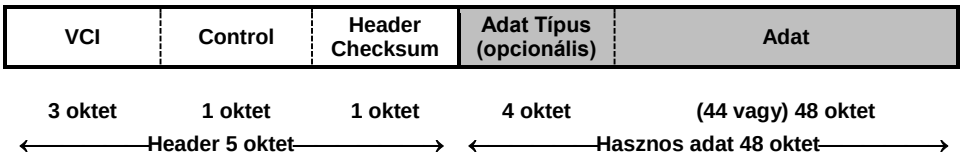
2.5. Digitális adatátvitel: ATM

Ahogy a digitális integrált áramkörök ára csökkent, úgy vált lehetségessé mind szélesebb körben a digitális adatátvitel alkalmazása. A digitalizálás elérte a telefon technikát is, mára a központok közötti – úgynevezett ***trunk*** – kapcsolatok szinte mind digitális összeköttetések. Ezeken a vonalakon az alkalmazott átviteli módszernek – ***STM, Syncronous Transfer Mode*** – köszönhetően egy időben számos telefonbeszélgetés bonyolítható le. Az STM működési elve a következő: Az analóg telefonjelet a központokban mintavételezve először digitális jellé alakítják (125 µsec, 1 oktet), majd több független vonal digitalizált jelét ***egyetlen*** nagysebességű vonalon továbbítják. Ha pl. két központ között 32 független vonalat akarunk, akkor 125 µsec-onként 32 oktetet kell továbbítani (32*8*8 000 = 2 048 000 bit/sec, E1). Az adatköteget szokásosan ***vonat***nak – ***train*** – nevezik. A vonaton belül az első vonal digitalizált jele az 1. oktetbe, a második vonal jele a 2. oktetbe, s.i.t. kerül, vagyis a vonat felépítése akkor sem változik, ha egy vonal éppen kihasználatlan. Mindez a sávszélesség egy részének elvesztését jelenti.

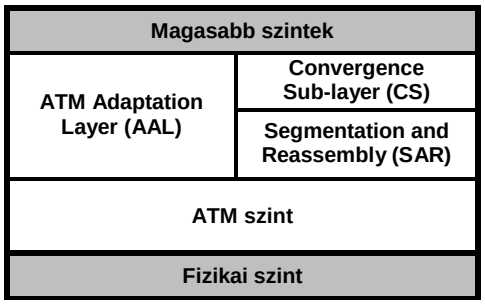
Az országok/kontinensek között kiépített telefon ***gerincvezetékek*** többnyire Gbit/sec sebességű optikai kábelek, amelyek beruházási költségei tetemesek. Ugyanakkor mind több más célú adatátviteli igény – pl. kábel-TV, számítógépes adatátvittel, stb. – is felmerül, amelyek szintén nagysebességű vonalakat kívánnak. A költségek csökkentésére kézenfekvő megoldás lenne különféle célú hálózatok jeleinek ***egyazon vezetéken*** történő továbbítása.

Az eltérő felhasználások azonban eltérő követelményeket támasztanak. A hang- és képátvitel fix sávszélességet igényel, eltűr némi adatvesztést, de nem tolerálja a változó késleltetést (real-time). Egy ilyen csatornát ***rögzített sávszélességűnek*** – ***CBR***, ***Constant Bite Rate*** – neveznek. Másfajta felhasználások – pl. file transzfer – viszont változó sávszélességet igényelnek, elviselik a késleltetés változását, de nem tolerálják az adatvesztést. Az ilyen fajta átviteli csatornákat ***változó sávszélességűnek*** – ***VBR***, ***Variable Bit Rate*** – nevezik.

- Amint láttuk, az STM technika lehetővé teszi, hogy egy átviteli vonalon több csatorna jelét továbbítsuk, de a csatorna-kiosztás nem változtatható. E hátrány kiküszöbölésére dolgozták ki az ***ATM*** – ***Asyncronous Transfer Mode*** – technikát, amely igen nagy sebességű ***alkalmazásfüggetlen*** adatátviteli közegként – vagyis fizikai szintű entitásként – üzemelhet. Az ATM a továbbítandó információt kis csomagokba – ***cella*** – tördeli. Egy cella a következő alakú:



A cella fejlécből és hasznos adatból áll. Az ATM specifikáció nem szól a szállított adat típusáról, csak a fejlécet értelmezi. Eszerint a ***VCI – Virtual Channel Identifier*** – a két szomszédos ATM kapcsológép közti ***pont-pont kapcsolat*** azonosítására szolgál, a ***Control*** mező különféle vezérlési célokra szolgál, míg a ***Header Checksum*** a fejléc esetleges meghibásodásának felderítését szolgálja. Az alacsony – kb. 10^{-9} – hibaarány miatt az adatot nem védi CRC. A cella mérete azért kicsi, hogy cellavesztéskor kevés adat vesszen el, valamint az ATM gép felépítésének egyszerűsítése érdekében.



← Telefon, Ethernet, Token Ring, stb.

← Cellavesztés kompenzálása

← Magasabb szintű protokoll-elemek cellákra tördelése, majd összeállítás

← Virtuális áramkörök kialakítása, cellák irányítása

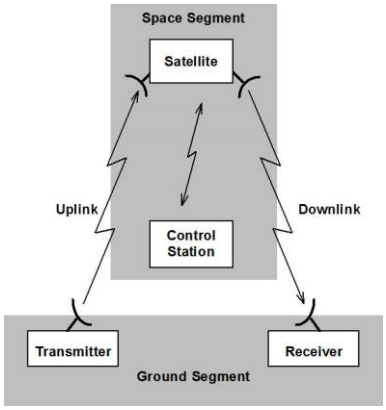
← Szállító közeg (többnyire üvegszál)

Az ATM hálózat felhasználásához úgynevezett ***adaptációs szinteket*** – ***Adaptation Layer*** – használnak, amelyek a különféle típusú protokollokat az ATM-re ***illesztik***. Maga az ATM hálózat a vonalkapcsolt hálózatokat imitáló ***virtuális áramköröket – Virtual Circuit*** – alkalmaz, vagyis egy kapcsolat kiépítésének kezdeti szakaszában történik meg a követendő útvonal kijelölése, ezután az adott virtuális áramkörhöz tartozó cellák mindig ezt az utat követik.

2.6. Digitális adatátvitel: Űrtávközlés

Napjainkban – elsősorban a nagytávolságú – adatátvitel jelentős része távközlési műholdak felhasználásával bonyolódik le. Bár a műholdak működésének részletei jelentősen meghaladják e kurzus kereteit, néhány alapelv ismerete azonban mégis hasznos lehet.

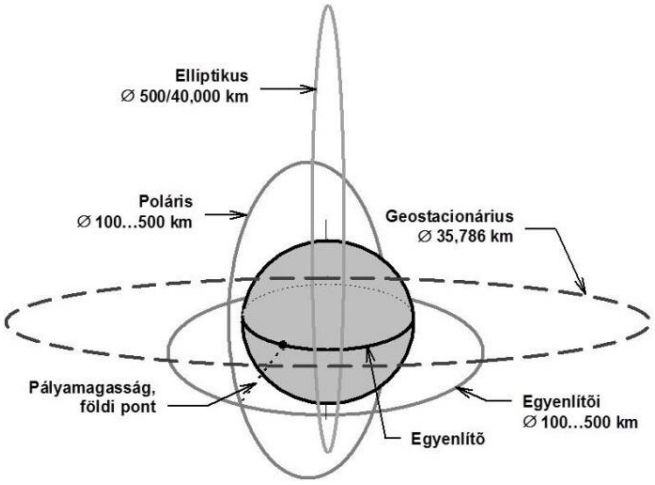
Egy távközlési műholdat alkalmazó rendszer vázlata a mellékelt ábrán látható. A ***rendszer földi szegmensre*** – ***Ground Segment*** – és ***űrszegmensre*** – ***Space Segment*** – bontható. Az űrszegmens részének tekintjük a földi ***irányító központot*** – ***Control Station*** – is. A földi vevőállomás különálló ***adó*** – ***Transmitter*** – és ***vevő*** – ***Receiver*** – részre bontható, amely azonban egy egységbe épül. Egy ilyen állomás lehet fix, vagy mobil kialakítású. A Földről a műholdra irányuló sugarat ***Uplink***-nek, míg az ellenkező irányút ***Downlink***-nek nevezik. A műhold szerkezete – az úgynevezett ***bus*** – fogja egybe a tápegységet, a rakétahajtóműveket, a hőszabályzó egységet, a műholdat irányító egységet (***Tracking, Telemetry & Command, TTC***), valamint magát a ***hasznos terhet***, amely az ***antennákat*** és a ***jeltovábbító*** – ***repeater*** – egységet jelenti.



Az űrtávközlésben használt ***vivőfrekvenciákat*** nemzetközi szabványok írják elő, szokásos értékük 3…31 GHz közötti ***mikrohullám***. A jeltovábbító egység feladata, hogy az Uplinken érkező mintegy 10…100 pW teljesítményű jelet vegye, majd azt 10…100 W teljesítményűre felerősítve a Downlinken keresztül visszasugározza a Földre. Így a jeltovábbító egység mintegy 200dB-t (!!!) erősít, miközben a szükséges sávszélesség kb. 0.5…1.5 GHz. Ekkora sávszélességet csak több együttműködő adó/vevő páros – ***transponder*** – együttműködésével lehet elérni. Egy konvencionális műhold esetén a jeltovábbító rendszer csak erősítést végez, de ha a vett jelek regenerálását is elvégzi, akkor ***regeneratív*** műholdról beszélünk (ez utóbbi a modernebb).

Egy műholdat egy időben számos földi állomás használhat. Ez felveti a ***többszörös elérés*** problémáját, amelyet többnyire ***frekvencia-osztással*** – ***Frequency Division Multiple Access***, ***FDMA*** –, ***időosztással*** – ***Time Division Multiple Access***, ***TDMA*** – oldanak meg. Az alkalmazott megoldások hasonlatosak a földi rendszerekben megszokott eljárásokhoz.

Egy műhold antennája által "látott" terület lehet ***globális*** – a Föld max. 42.4% látható – ***zónás*** – pl. egy-egy földrészre korlátozott – illetve ***pontszerű*** – ***spot beam*** –.

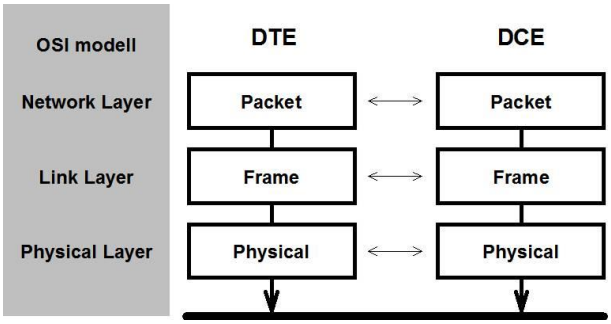


A szóba jöhető műholdpályák alapján ***elliptikus***, ***cirkuláris***, illetve ***geo-stacionárius*** pályáról beszélünk. Az űrtávközlésben elterjedten alkalmazzák a geo-stacionárius műholdakat, mivel ilyenkor a földi állomás a műholdat mindig az ég egy adott pontján "látja", valamint a műhold a Föld csaknem feléről látható. Ugyanakkor a nagy távolság – ~40 000 km – miatt rendkívül nagy – 238…278 ms – késleltetési idő, valamint óriási teljesítmény-veszteség – kb. 200 dB – adódik.

Az űrtávközlés alkalmazása az előnyök – olcsó földi (akár mobil is) állomás, egyszerre a Föld jelentős része belátható, egy adó egyszerre több vevőt is kiszolgálhat (pl. műholdas TV-adás), egyszerű területi re-konfiguráció, távolság független díjak, nagy megbízhatóság – mellett hátrányokkal is terhelt. Ilyen hátrány a geo-stacionárius szatellitek esetén a nagy késleltetési idő (kétirányú hangátvitel esetén lényegében használhatatlan), a lehallgathatóság (titkosítás szükséges), valamint az a tény, hogy a műhold elvesztése a teljes hálózat megsemmisülésével jár. Mindezek a hátrányok azonban számítógépes adatátvitel esetén gondos tervezéssel többnyire kompenzálhatók – földi alternatív útvonalak biztosítása, real-time felhasználások elhagyása –, így a műholdas rendszerek alkalmazása a számítógép hálózatok terén egyre növekszik (pl. Magyarországon néhány bank és benzinkút-hálózat is e technológiára alapozta adatátviteli rendszerét.)

3. Létező hálózatok

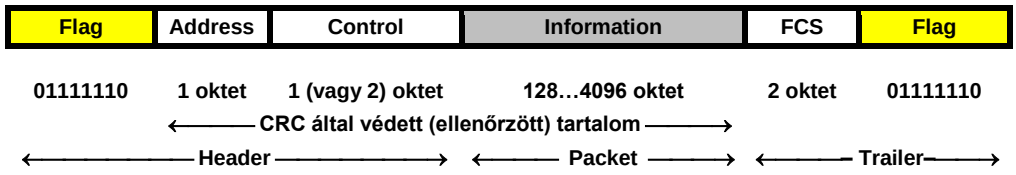
3.1. Kapcsolat-orientált csomagkapcsolt adathálózat: X.25



Az első széles körben elterjedt, az OSI referencia-modell alapján felépített és szabványosított csomagkapcsolt adatátviteli protokoll a CCITT ***X.25*** ajánlása volt. Számos országban építettek olyan ***nyilvános*** csomagkapcsolt hálózatot, amely erre az – 1980-ban megjelent – szabványra épült (Magyarország: 1990). Az X.25 ajánlás olyan rendszert definiál, amely az ***OSI alsó három szintjéhez*** tartozó feladatokat foglalja magába, és meghatározza ***egy felhasználói végberendezés*** – ***DTE***, ***Data Terminal Equipment*** – és a ***hálózat szolgáltatási végpontja*** – ***DCE***, ***Data Circuit Equipment*** – közti áramkörök és adatátviteli módszerek összességét.

- Az X.25 ajánlás fizikai szintje szinkron, duplex adatátvitelt – ***X.21***, vagy ***X.21bis*** – definiál, valamint meghatározza az alkalmazandó csatlakozók típusát, a jelek feszültségszintjeit, valamint pontos szerepüket.

- Az X.25 az adatkapcsolati (Link) szintet ***Frame Layer***-nek nevezi, amelynek feladata a ***sorrendtartó, hibamentes*** adatátvitel biztosítása. A Frame szint az adatátvitel során a ***HDLC*** – ***High-level Data Link Control*** – eljárást alkalmazza, amely a transzparens átvitel érdekében a ***keret***ek – ***frame*** – kialakításakor a ***bitbeszúrás*** – ***bit stuffing*** – módszert használja, és az átviteli hibák feltárására a kereteket ***ellenőrző összeggel*** – ***FCS***, ***Frame Check Sum*** – látja el. A keretek elejét és végét speciális bitsorozat – a ***flag*** – határolja.



Az adatkapcsolati szinten az adatfolyam vezérlését, az esetleges átviteli hibák javítását, valamint a vonal felépítés/bontás feladatait a ***LAPB*** – ***Link Access Procedure Balanced*** – protokoll látja el. A protokoll körbeforgó ablak technikával biztosítja a fenti feladatok ellátását. A LAPB protokoll-elemei a HDLC keret ***Address*** és ***Control*** mezőiben helyezkednek el. A protokoll pont-pont kapcsolatot tart a DTE és a DCE között.

- Az ajánlás 3. szintje a ***csomag*** – ***packet*** – nevet viseli. Ezen a szinten olyan protokollt definiáltak, amely lehetővé teszi, hogy a Frame szint által biztosított kapcsolaton egy időben több ***logikai kapcsolatot*** – ***virtual circuit*** – létesítsünk. Egy ilyen virtuális áramkört először fel kell építeni. Az X.25 hálózat ***kapcsolat-orientált*** adatátvitelt alkalmaz, vagyis két végberendezés – DTE – közti virtuális áramkör kiépítése során alakul ki a tényleges útvonal (hasonlóan a telefon hálózathoz), majd a virtuális áramkörön szállított valamennyi adatcsomag később mindvégig ezt az útvonalat fogja használni. Az X.25 ***SVC – Switched Virtual Circuit*** – és ***PVC*** – ***Permanent Virtual Circuit*** – áramköröket különböztet meg. Egy SVC-vel tetszés szerinti X.25 végpontot lehet megcímezni, míg a PVC előre definiált végpont felé a DTE bekapcsolásakor automatikusan épít ki kapcsolatot.

  Az X.25 hálózat közönséges terminálokkal való felhasználásának elősegítésére az – ***X.3, X.28, X.29*** ajánlásokban – speciális eszközöket – ***PAD***, ***Packet Assembler Disassembler*** – definiáltak, amelyek az egyszerű terminálok helyett ellátják a viszonylag bonyolult protokoll kezelését. A PAD számos ***paraméter*** megadásával sokféle meglévő terminál illesztéséhez adaptálható.

  Az X.25 hálózatban a végpontok ***megcímzéséhez*** – az ***X.121*** ajánlásban definiált – a hagyományos telefon-hálózatban megszokott számokhoz hasonló címzést alkalmaznak, amely ország - körzet - végállomás – alállomás struktúrájú címzést biztosít.

3.2. Kapcsolatmentes csomagkapcsolt adathálózat: Internet

A valóban világméretű számítógép hálózat alapjait egy csaknem 30 évvel ezelőtt, az Egyesült Államok Honvédelmi Minisztériumának Kutató Intézetében – ***DARPA, Defense Advanced Research Project Agency*** – megindított fejlesztés szolgáltatta. 6 éves munka eredményeként létrejött az ***ARPANET*** hálózat, amely a ma ***Internet*** néven nevezett világháló magját képezte.

A DARPA szerteágazó fejlesztéseinek középpontjában egy ***protokoll-család*** kialakítása állt. A fejlesztés eredményeként létrejött protokollokat ma ***TCP/IP*** néven nevezik, és valójában néhány alapvető, valamint számos speciális protokoll összességét jelenti.

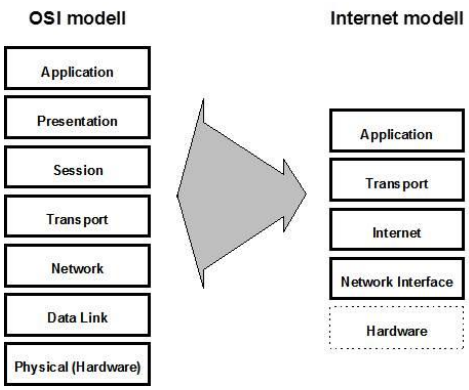
A TCP/IP protokoll-családot tehát nem egy szabványosítási szervezet, vagy egy nagy számítástechnikai cég dolgozta ki, hanem számos egyetemi- és katonai kutató intézet közös munkájának eredménye. Ezek a protokollok valójában a mai napig sem szabványok a szó szoros értelmében, csupán ajánlások. Mint ilyenek, azonban az évek során minden szabványnál szélesebb körben terjedtek el, így tökéletesen illik rájuk a ***de facto szabvány*** elnevezés.

A protokoll-család kidolgozása során a fejlesztők – nyugodtan nevezhetjük őket kutatóknak is – maguk is messzemenően kihasználták a fokozatosan kiépülő hálózat szolgáltatásait. Ennek megfelelően a fejlesztési dokumentumok megírásuk idején az egymástól gyakran ezer kilométerekre dolgozó szerzők között elektronikus formában "utaztak", és az idők során rajtuk ragadt egy név: ***RFC*** – ***Request For Comment, Véleménykérés***. Az Internet minden de facto szabványa RFC-k formájában ma is bárki által ***szabadon*** elérhető, és lényegében minden Internettel kapcsolatos ismeretet tartalmaznak. Az RFC-ket sorszámozták, máig több, mint 2000 ilyen dokumentumot publikáltak. Ha valamelyik tartalma felett eljárt az idő, akkor a megfelelő RFC-t egyszerűen egy újabbal helyettesítik, de egy már publikált RFC-t ***sohasem*** változtatnak meg. Bármely RFC-hez bárki véleményt fűzhet, és elküldheti azt az RFC-t jegyző szerző(k)nek, így járulva hozzá az Internet további fejlődéséhez. Lássuk mik is voltak a fejlesztés alapelvei!

Egy számítógép hálózat felhasználása során elvileg két különböző módon járhatunk el. A hálózat kezelésének részleteit a ***felhasználói programokba építhetjük be***, de ez azzal a hátránnyal jár, hogy a hálózat változása minduntalan a felhasználói programok teljes köre változtatásának igényét kelti. Sokkal praktikusabb megközelítés az, ha a hálózatkezelés részleteit ***az operációs rendszerbe integráljuk***, amely aztán szabványos eljárások formájában nyújtja e szolgáltatásokat a felhasználói programok számára. Ekkor valamely hálózati változás "csak" az operációs rendszer(ek) módosítását igényli, a felhasználói programok hálózat függetlenek.

Nem ismerünk azonban olyan hálózati technológiát, amely minden célra egyaránt mindenben megfelelő lenne, ezért a számítógép hálózatok kiépítése során rengeteg különféle módszer alkalmazható. Az ***Internet*** kialakításakor a fejlesztők ***legfőbb célja*** az volt, hogy elrejtsék a technikai részleteket, olyan protokollokat hozzanak létre, amelyek bármilyen topológia, átviteli közeg, operációs rendszer, számítógép, stb. esetén azonos módon alkalmazhatók legyenek. Ezért az Internet a következő generális – ***hardware független*** – alapelvekre épül:

- Az Internet ***csomagkapcsolt*** adatátviteli rendszert alkalmazó ***független hálózatok összessége***.
- ***Minden hálózat egyenlő***, függetlenül attól, hogy milyen módszerrel szállítja a csomagokat, milyen késleltetést okoz, milyen fizikai címzést használ és mekkora csomagméretet alkalmaz.
- Az egymástól ***független hálózatok közti kapcsolatot*** olyan berendezések teszik lehetővé, amelyek egy időben két (vagy több) – egymástól akár teljesen eltérő – független hálózathoz is csatlakoznak, és képesek adatok egyik hálózatból a másikba továbbítására. Ezeket az eszközöket ***Gateway***-knek, vagy manapság elterjedtebb nevükön ***Router***-eknek nevezzük.
- Az Internet valamennyi végberendezése és Routere ***egységes címet*** – az úgynevezett ***IP címet*** – alkalmazza. Az IP cím olyan azonosító, amelyet ***központilag osztanak ki***, vagyis a világméretű Internetben két hoszt és/vagy router nem kaphatja meg ugyanazt a címet.
- Az ***Internet protokollok*** az OSI modellhez hasonlóan ***rétegesen épülnek egymásra***, de ellentétben az OSI modellel az Internet modell mindössze ***4 szintet*** definiál. Mindazonáltal a két modell között erős logikai összefüggés tapasztalható.



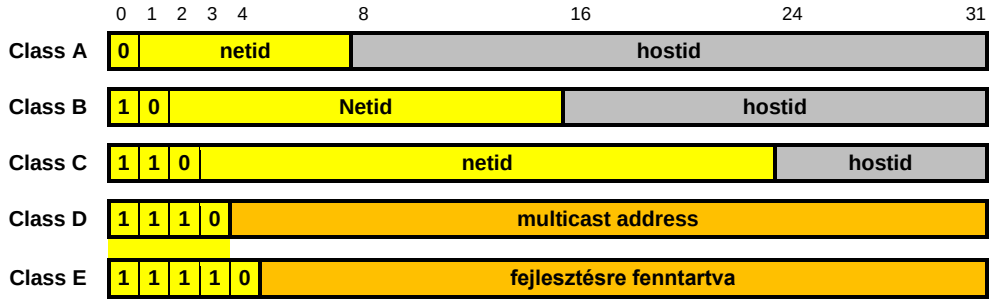
3.3. Internet címek

Az Internetben – az IP protokollt alkalmazó világméretű hálózatban ***– bármely hoszt kapcsolatba léphet bármely másik hoszttal***. Ehhez viszont elengedhetetlen, hogy az Internet minden hosztja rendelkezzen egy egységes szerkezetű, hardware-független, egyedi azonosítóval: ***IP címmel***.

Az IP cím egy ***32 bites***, előjeltelen bináris szám, amely logikailag két részre: ***hálózat-azonosítóra*** – ***netid*** – és ***hoszt-azonosítóra*** – ***hostid*** – tagolódik. A netid úgy tekinthető, mint egy prefix, vagyis egy hálózaton – pl. egy intézmény saját lokális hálózatán – belül valamennyi hoszt netidja megegyezik.

A 32 bites IP címeket igen nehéz lenne binárisan kimondani, ezért a 32 bitet 4 8-bites csoportba osztják, majd az így létrejött 4 bináris szám decimális megfelelőjét egymástól egy-egy pont karakterrel elválasztva írják le. Az IP cím ilyetén ábrázolását ***pontozott decimális címnek*** – ***Dotted Decimal Address*** – nevezik. Pl. **11000000 11010001 01010100 10000001 = 192.225.84.129**

Amint említettük az IP cím logikailag minden esetben netid-re és hostid-re bomlik. Annak függvényében, hogy a 32 bitből mennyi a netid, az IP címeket ***osztályokba*** soroljuk. Ha a netid 8 bites, akkor ***A***; ha 16 bites, akkor ***B***; ha pedig 24 bites, akkor ***C-osztályú*** címről beszélünk. A címtér ilyen felosztása az Internetbe tömörülő különféle hálózatok egyedi méretéhez jól illeszkedő struktúrát eredményez. Az alábbi táblázat a különféle osztályú IP címek felépítését szemlélteti.



Az Internet minden hosztja ***legalább egy IP*** címmel rendelkezik. A routerek azonban egyszerre több hálózathoz is tartoznak, így ezeknek legalább annyi IP címük van, ahány hálózathoz kapcsolódnak. Valójában tehát ***az IP cím*** nem a hoszt/router címe, hanem a ***hálózati kapcsolat címe***. A több hálózathoz kapcsolódó berendezéseket ***multi homed*** eszközöknek nevezzük.

A netid-k és hostid-k kiosztásakor néhány további konvenciót is alkalmaznak. Ha a netid és/vagy hostid = **0**, a cím "***ezt***" – ***this*** – a hálózatot/hosztot jelzi. Ha a hostid, vagy az IP cím minden bitje = **1**, ez az úgynevezett (***limited***, ***directed***) ***broadcast*** cím. A speciális IP címek a következők:

csupa 0	"Ez" a host (csak startup alatt, cél-cím nem lehet)
netid | csupa 0	"Ez" a hálózat (forrás/cél-cím nem lehet)
csupa 0 | hostid	Hoszt "ezen" a hálózaton (csak startup alatt, cél-cím nem lehet)
csupa 1	Limited broadcast [csak erre a hálózatra] (forrás-cím nem lehet)
netid | csupa 1	Directed broadcast [a címzett hálózatra] (forrás-cím nem lehet)
127 | akármi (gyakran 1)	Loopback [127.0.0.1] (soha nem jelenhet meg a hálózaton)

Mivel az Interneten a legkülönfélébb számítógépek működnek együtt, ezért szükséges volt az 1 oktetnél hosszabb adatok esetében alkalmazandó ***byte-sorrend*** definiálása. A különböző számítógépek általában két különféle sorrendű byte-ábrázolást alkalmaznak, ezeket ***Big Endian*** – pl. Sparc, Motorola CPU – és ***Little Endian*** – Intel CPU – néven nevezik. Az Internet szabványos byte-sorrendje – ***Network Standard Byte Order*** – a ***Big Endian***, vagyis először mindig a ***legmagasabb*** helyértékű biteket tartalmazó byte kerül átvitelre, s.i.t.

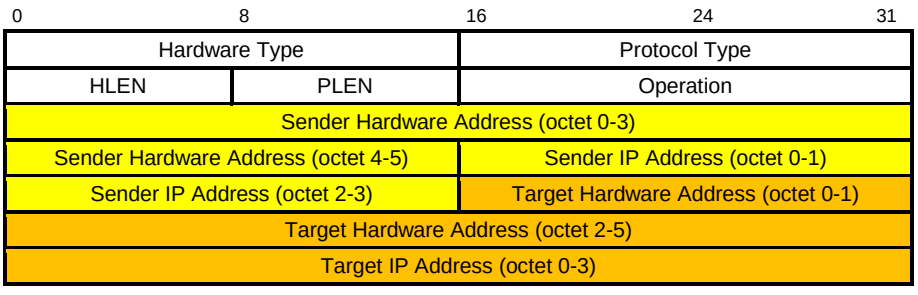
3.4. IP-címek és fizikai-címek egymáshoz rendelése

Az Interneten a hosztok egymást IP címeikkel azonosítják. De a konkrét hálózatok vizsgálata során láttuk, hogy egy adott technológiát – pl. Ethernet – alkalmazó hálózaton az adó a vevőt ***csak annak fizikai címével*** azonosíthatja. Mindez olyan eljárást igényel, amelynek segítségével az Internet egy hosztja az IP cím birtokában meghatározhatja az ahhoz tartozó fizikai címet.

Miért is szükséges, hogy az IP cím és a fizikai cím különböző legyen? Mint tudjuk, a különféle hálózati technikák különféle fizikai címeket használnak. Az Internet tervezésekor viszont alapelv volt a technológiai különbségek elfedése, vagyis az IP cím egyetlen valódi fizikai címre sem hasonlít. Az IP cím ezen tulajdonsága azonban inkább előny, mint hátrány, mert így ***az Internet Protocol Stack***-et – vagyis az Internet kompatibilis protokollokat megvalósító rendszerprogramokat – döntő hányadában a ténylegesen alkalmazott hardware részletektől függetlenül alakíthatjuk ki. De a teljes rendszer csak akkor működhet, ha megtaláljuk annak módját, hogy az IP címhez hozzárendeljük a tényleges fizikai címet.

Az leírtakat ***címfeloldási problémaként*** – ***address resolution problem*** – ismerik, és a különféle hálózatokban eltérő módon oldják meg. Lényegében kétféle módszert: a ***direkt leképezés*** – ***direct mapping*** – és a ***dinamikus leképezés*** – ***dynamic mapping*** – alkalmaznak.

- ***Direkt leképezés***. Ha a hálózatunkban alkalmazott ***hálózati illesztő kártya – Network Interface Card, NIC*** – rövid, és szabadon beállítható címeket használ, akkor lokális hálózatunkat felépíthetjük úgy, hogy az egyes hosztok illesztőkártyáin fizikai címként az adott hoszt IP címének hostid-ját állítjuk be. Ezt a megoldást alkalmazhatjuk pl. a Token Ring hálózatokban, ahol a hosztok 8 bites fizikai címe az illesztő kártyán állítható be. Ez a megoldás igen ***hatékony*** címfeloldást eredményezhet, mivel az IP címből a fizikai cím meghatározása csupán néhány gépi utasítás végrehajtását igényli. A leírtak ugyanakkor ***hátrányokat*** is magukban rejtenek. Ezek abból adódnak, hogyha meg kívánjuk – vagy meg kell – változtatni egy hoszt IP címét, ez a gép fizikai címének változtatási kényszerét is jelenti. Ugyancsak gondot okozhat annak biztosítása, hogy a különböző időpontban üzembe állított gépek eltérő fizikai címeket használjanak. Hibás konfigurálás következtében könnyen előfordulhat hogy két gép azonos fizikai címet kap, ugyanakkor az ilyen jellegű hiba feltárása sem feltétlenül egyszerű feladat.
- ***Dinamikus leképezés***. Hogy megértsük, a címfeloldás bizonyos esetekben miért is bonyolult feladat, emlékezzünk vissza az Ethernet hálózatra. Ott az illesztő kártyák mindegyike ***48 bites*** – a kártya gyártója által meghatározott – fizikai címet használ. Mivel az IP cím 32 bites, elvileg sincs módunk arra, hogy a 48 bites fizikai címet a 32 bites logikai címre képezzük le. E probléma feloldására az Internet protokoll-család tervezői igen szellemes megoldást dolgoztak ki, amely Ethernet – és hasonló címzési rendszert alkalmazó – hálózatokban alkalmazható.
- ***Azt a protokollt, amellyel a hoszt egy másik hoszt fizikai címét csupán annak IP címét ismerve megállapíthatja, ARP-nek – Address Resolution Protocol – nevezik.*** Ethernet hálózaton az ARP azt használja ki, hogy a hálózatban létezik broadcast cím. Egy másik hoszt IP címét kereső hoszt a hálózatra olyan Ethernet csomagot küld, amelynek adatmezejébe egy ARP csomagot helyez, továbbá cél-cím mezejébe a broadcast címet, forrás-cím mezejébe saját Ethernet címét írja:



Ethernet hálózat esetén a **Hardware Type** mező = **1**. A **Protocol Type** mező = **0800H**, jelezve, hogy a csomagban IP címek feloldását kérjük. Az **Operation** mező azonosítja az adott csomag jelentését, lehetséges értéke **1=ARP Request**, **2=ARP Response**, **3=RARP Request**, illetve **4=RARP Response** lehet. A **HLEN** és **PLEN** mezők az IP- és fizikai címek hosszúságát határozzák meg, Ethernet hálózat esetén ezek értéke rendre **4**, illetve **6**. A csomag feladója kitölti még a **Sender Hardware Address** és **Sender IP Address** mezőket megfelelő saját címeivel, majd útjára bocsátja a csomagot.

Mivel ***minden*** hoszthoz megérkezik a broadcast címmel kiküldött csomag, így az adott IP címet birtokló is megkapja azt. Felismerve, hogy a csomagban saját IP címe áll, ***válaszcsomagot*** küldhet a feladónak, amelyben a megfelelő mezők kitöltésével informálhatja a kérdezőt.

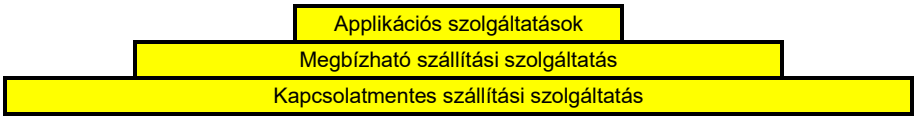
A leírtak szerint működő címfeloldás csak akkor lehet hatékony, ha a hoszt – a későbbi felhasználásra számítva – egy ideig még tárolja a megismert IP – fizikai címpárokat. Ha ezt nem tenné, akkor ***minden*** csomag kiküldése előtt újabb ARP Request / ARP Response kommunikációra lenne szükség, amely a hálózati forgalom jelentős növekedését eredményezné. A címpár tárolót ***ARP Cache***-nek nevezik. E tároló segítségével a hoszt csak akkor küld ARP Request-et, ha a cache-ben nem találja a keresett IP – fizikai címpárt.

- A címfeloldásnak létezik egy ***ellenkező irányú*** felhasználása is. Az Internetre csatlakozó háttértároló nélküli berendezések esetén felmerül a kérdés, hogy bekapcsoláskor ezek honnan ismerik meg saját IP címüket. A megoldást szolgáló protokollt ***RARP***-nak **– *Reverse Address Resolution Protocol*** – nevezik, és a következőképpen működik:

  A saját IP címét kereső gép összeállít egy – az ARP kéréshez hasonló – ***RARP Request*** csomagot, amelyben saját IP címét ***üresen hagyja***, majd ismételten kiküldi azt a hálózatra mindaddig, amíg választ nem kap. A hálózaton egy – vagy több – hoszt ***RARP server***-ként üzemelve várakozik az ilyen kérésekre. A RARP szerver saját háttértárolóján egy ***táblázatot*** tárol, amelyben a hálózat összes ***fizikai – IP címpárosa*** megtalálható. Amikor megérkezik a kérés, e táblázatban megtalálható a keresett IP cím, így az egy válaszcsomagban visszaküldhető. Több RARP server működését az üzembiztonság fokozása indokolja.

3.5. Kapcsolatmentes adatátvitel, az IP datagram

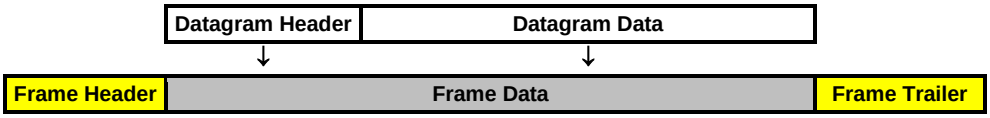
Habár a világméretű Internetben a legkülönfélébb típusú számítógépek üzemelnek, a felhasználók mégis az egész hálózatot ***egyetlen virtuális hálózatnak*** tekinthetik. Mindezt az Internet szabványos protokolljai biztosítják. Ezek a protokollok különféle **– *egymásra épülő*** – szolgáltatásokat nyújtanak, amelyek logikailag a következő ábrával szemléltethetők:



A réteges felépítés lehetővé teszi, hogy a különböző szinteken elhelyezkedő protokollok egymástól nagymértékben ***függetlenek*** lehessenek, egyik a másik változtatása nélkül módosulhasson.

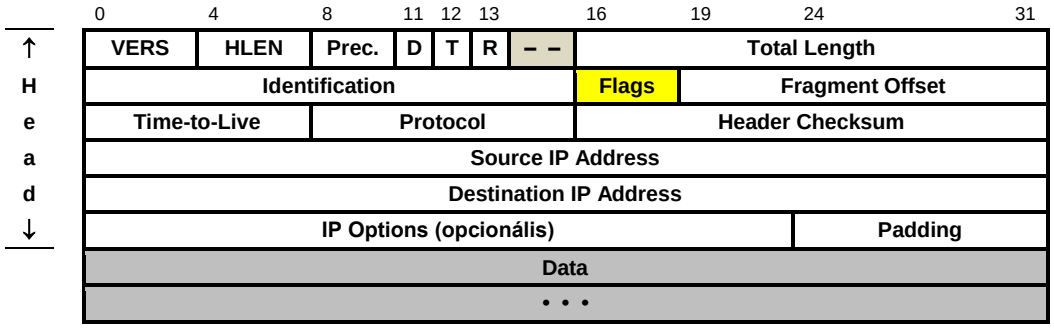
A legalapvetőbb Internet szolgáltatás a ***kapcsolatmentes szállítási szolgáltatás – Connectionless Delivery Service*** – néven ismert. Ez a szolgáltatás csomagok – az Internet terminológia szerint nevük ***datagram*** – szállítását végzi, mégpedig "***felelősség nélkül***", vagyis a szolgálat nem garantálja, hogy a feladott csomagok eljutnak a címzetthez, sorrendben jutnak el, esetleg megduplázódnak, illetve hibátlanul érkeznek-e meg. A ***kapcsolatmentes*** jelző arra utal, hogy minden egyes datagram – még ha összetartozó adatokat szállít is – a hálózat szempontjából teljesen ***független*** a többitől, vagyis esetleg más útvonalon, más késleltetést szenvedve járja be a feladó és a címzett közötti távolságot, mint egy másik datagram. A "felelősség nélkül" kitétel arra utal, hogy a hálózat mindent megtesz a datagram célba juttatása érdekében – vagyis ok nélkül nem dobja azt el –, de ha ez mégis bekövetkezik, akkor a csomagvesztés tényéről sem a feladót, sem a címzettet nem értesíti. A fent leírt szolgáltatást nyújtó protokollt ***Internet Protokollnak*** – röviden ***IP*** – nevezik. A protokoll egyrészt definiálja a datagram formátumát, másrészt meghatározza, hogy az IP hálózat berendezései milyen szabályok szerint kezeljék azokat.

A datagramok a tényleges adatátvitel során a fizikai szállító közeg csomagjaiban – pl. Ethernet esetén egy-egy ***frame***-ben – "utaznak". Ezt "***csomagolásnak***" – ***encapsulation*** – nevezik, és a következő ábrával szemléltethető:

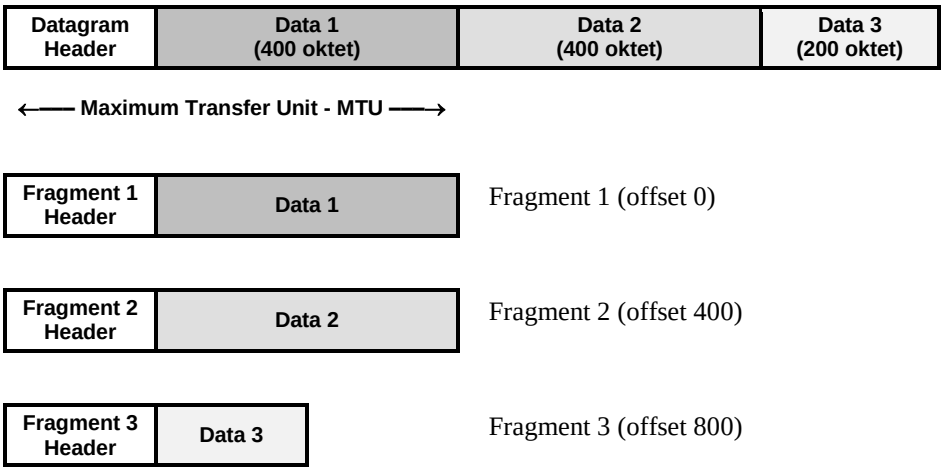


Az IP datagram számos ***mezőt*** tartalmaz. A **VERS** az alkalmazott IP protokoll verziószámát jelzi (jelenleg **4**), a **HLEN** a datagram fejlécének 4-oktetben mért hosszát jelzi (ha nincs opció, akkor **5**), A **Precedence** mező a datagram "fontosságát" jelzi, **7** a legmagasabb, **0** a legalacsonyabb. A **D**, **T**, és **R** bitek az átvitel jellegére utalnak, **D=1** alacsony késleltetésű átvitelt igényel, **T=1** nagy sávszélességet kér, **R=1** nagy megbízhatóságra tart igényt. Természetesen ezek olyan kérések,

amiket az IP ha teheti, figyelembe vesz, ha nem, figyelmen kívül hagy (a datagram által bejárt útvonal függhet ezektől az értékektől). A **Total Length** a datagram teljes – header + data – hosszát jelzi oktetekben mérve (max. **65 535**).



Amint látjuk, a datagram teljes mérete 64 Koktet lehet. A tényleges fizikai közeg azonban általában nem engedi ekkora csomagok szállítását (Ethernet = max. 1 500 oktet). Ha a datagram tényleges mérete meghaladja a fizikai szállító közeg maximális csomagméretét – a ***Maximum Transfer Unit***-ot, ***MTU*** –, akkor a datagramot fel kell ***tördelni*** – ***fragmentation*** –. Az eredeti datagram így keletkező darabjait ***fragment***-eknek nevezik. Tördelés után egy datagram például a következőképpen nézhet ki:



A széttördelt datagram ***strukturálisan*** megegyezik az eredeti datagrammal, csupán néhány mező tartalma változik. Az **Identification** azonosítja az eredetileg egyazon datagramhoz tartozó töredékeket, vagyis e mező értéke minden fragmentben azonos (egy következő datagram ezen mezeje viszont már biztosan eltérő értéket tartalmaz). A **Fragment Offset** adja meg a fragmentben található adat eredeti datagramon belüli ofszetjét. A tördeléssel kapcsolatos utolsó mező a 3 bites **Flags**, amelynek első bitje az úgynevezett "***do not fragment***" bit. Ha ez **1**, akkor a datagram nem tördelhető. Ha mégis tördelésre lenne szükség, akkor az ezt detektáló berendezés ***eldobja*** ezt a datagramot, de erről ***értesítést küld*** vissza a feladónak. A **Flags** mező utolsó bitje a "***more fragments***" bit. Ez a bit minden fragmentben **1**, kivéve az utolsót.

Ha széttördeltük a datagramot, azt előbb-utóbb össze is kell állítani. Ezt a feladatot a datagram ***címzettje*** hajtja végre. A tevékenység angol elnevezése: ***reassembling***.

A datagram **Time-to-Live** mezeje az Internetbe küldött datagram ***maximális élettartamát*** korlátozza. A mező értékét minden – a datagramot továbbító – router **1**-el csökkenti. Ha az **0**-ra változik, akkor a router ***eldobja*** a datagramot, de erről ***értesítést küld*** a feladónak.

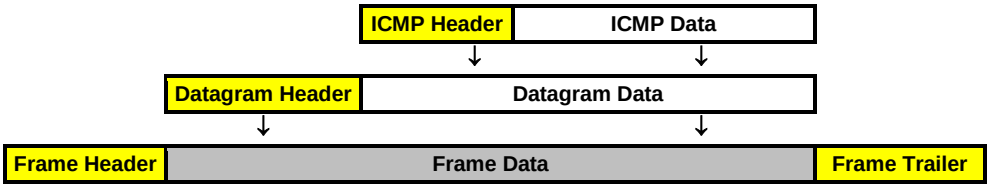
A **Protocol** mező a datagram **Data** mezejében szállított adat ***típusát*** – a magasabb szintű protokoll azonosítóját – tárolja. A megfelelő értékeket az Internetben ***központilag osztják ki***.

A **Header Checksum** a fejlécben esetleg bekövetkező hibák felderítését segíti, míg a **Source-** és **Destination IP Address** mezők rendre a datagram feladójának, és címzettjének IP címét tárolják. Végezetül a **Data** mező a ***hasznos adatot*** – egy magasabb szintű ***protokoll-elemet*** – tartalmazza.

3.6. Az IP hálózat vezérlése, ICMP

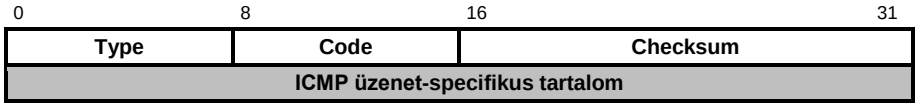
Az Internetben a datagramok a feladó hoszttól az ***útba eső routerek közreműködésével*** jutnak el a címzett hosztig. Ha ezen az úton valamelyik router olyan körülményeket észlel, amelyek lehetetlenné teszik a datagram továbbítását, akkor ezt ***jeleznie kell*** a datagramot útjára bocsátó hoszt IP protokoll-gépének, lehetővé téve ezzel, hogy az intézkedhessen a hiba elhárításáról.

Azt a protokollt, amely a fent leírt célok megvalósítását biztosítja**, *Internet Vezérlő Üzenet Protokollnak*** – ***Internet Control Message Protocol***, ***ICMP*** – nevezik. Az ICMP a hibajelzésen túl bizonyos hálózat-ellenőrzési funkciók ellátására is felhasználható. Az ICMP integráns része az IP specifikációnak, így megvalósítása minden IP-csomagban kötelező. Egy-egy ICMP protokoll-elem IP datagramok adatmezejébe csomagolva utazik a hálózatban.



Egy ICMP üzenet generálásakor annak feladója az üzenettel csupán jelzi egy bizonyos állapot bekövetkeztét, de maga semmilyen egyéb tevékenységet nem végez. Más szavakkal: az ICMP csak ***jelzi*** a hibát, de ***nem javítja*** azt. Ez utóbbi az ICMP üzenet címzettjére – a jelzett állapotot kiváltó datagram feladójára – hárul.

Számos különféle ICMP üzenet létezik, amelyek formája eltérő, de valamennyi ICMP üzenet a következő mezőkkel kezdődik:



A **Type** mező jelzi az adott ICMP üzenet típusát, a **Code** mező ezen belül finomítja az üzenet jelentését, a **Checksum** pedig az ICMP üzenet tatalmában esetleg bekövetkezett hiba felderítését szolgálja. Ha az ICMP üzenet valamely hibát jelez, akkor minden esetben tartalmazza a hibát kiváltó datagram fejlécét, és a datagramban szállított adat első 8 oktetét. A különféle ICMP üzenetek vázlatosan a következők:

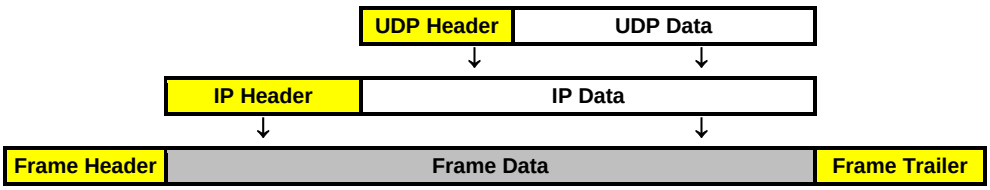
- ***Echo*** – Ezen üzenet segítségével ellenőrizhetjük, hogy egy adott IP című hoszt ***elérhető-e*** az Interneten. Bármely hoszt/router, amely ilyen ICMP üzenetet vesz, válaszjelzést küld vissza a feladónak. A közismert ***ping*** parancs erre az ICMP üzenetre alapozva működik.
- ***Host/Port Unreachable*** – Ha egy routerhez/hoszthoz olyan datagram érkezik, amelyet nem tud végső címzettjéhez – illetve a címzett porthoz (lásd UDP) irányítani, ezt az ICMP üzenetet generálja. Az üzenet többnyire az adott hoszt/port pillanatnyi működésképtelenségére utal.
- ***Source Quench*** – Amennyiben egy router/hoszt valamely oknál fogva nem képes a beérkező datagramokat az adott ütemben fogadni, ***torlódásról*** – ***congestion*** – beszélünk. A source quench – ***forrás lefolytás*** – üzenettel a datagramok túl szapora feladója tempójának mérséklésére kényszeríthető. (“Gyorsító” üzenet nincs, a feladó a lassítás után lassan ismét “gyorsít” majd.)
- ***Redirect*** – Ha a hoszt egy datagramot olyan routerhez küld továbbításra, amely ennél jobb útvonalat is ismer, redirect üzenetben értesíti a datagram feladóját, miközben megadja ezen jobb útvonalat kezelő – ***egyazon lokális hálózaton lévő*** – másik router IP címét is.
- ***Time Exceeded*** – Ha egy router a datagram továbbítása során a Time-to-Live mező értékét 0-ra változtatta, és ezért az adott datagramot eldobta, akkor erről az eldobott datagram feladóját ezzel az üzenettel értesíti.
- ***Time Stamp*** – Az üzenet segítségével lekérdezhető egy távoli gép órájának aktuális értéke.
- ***Subnet Mask Request*** – Az üzenet segítségével megszerezhető egy adott hoszt által használt ***subnet-mask*** aktuális értéke.
- ***Parameter Problem*** – A datagramot továbbító valamely router a datagram fejlécében hibát talált.

3.7. Az IP hálózat felhasználása: UDP

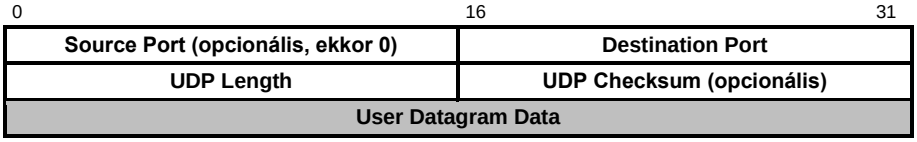
A modern számítógépek hatalmas teljesítménnyel rendelkeznek. Ez lehetővé teszi, hogy feldolgozó kapacitásukat egy időben ***több*** feladat megoldására is felhasználjuk. Ehhez természetesen az is elengedhetetlen, hogy az effajta működést a gép operációs rendszere is támogassa. Ha megteszi, akkor ***többfeladatú*** – ***multitask*** – operációs rendszerről beszélünk.

- Az applikációs programok, és az operációs rendszer mind teljesebb szétválasztása érdekében a hálózat kezelésének részleteit az operációs rendszer tartalmazza. Ennek értelmében a hálózat szempontjából a végső címzett nem is egy felhasználói program – más néven ***process*** –, hanem annak operációs rendszerbeli absztrakciója. Ezt a képzelt elemet az Internet hálózatban ***kapunak*** – ***port*** – nevezzük. Egy hoszton belül a portokat ***16 bites előjeltelen bináris*** számokkal azonosítjuk, amit ***port-címnek*** nevezünk.

  Mint láttuk az Internet hálózatban a hosztokat IP címükkel azonosíthatjuk. Egy adott IP címre feladott datagramot az IP hálózat a hosztig szállítja, de nincs eszköze arra, hogy a multitask rendszerben üzemelő gép portjait is megcímezze. Azt a protokollt, amely lehetővé teszi, hogy egy adott hoszton belül az egyes portokat megcímezzük, ***UDP***-nek – ***User Datagram Protocol*** – nevezzük. Az UDP szolgáltatás az IP szolgáltatásra épül, ahhoz képest csak a port-címzés lehetőségét nyújtja. Más szavakkal***: az UDP kapcsolatmentes, felelősség nélküli szállítási szolgáltatást nyújt***, amely azonban a hoszton belüli portok megcímzésére is képes. Egy-egy UDP protokoll-elem IP datagramok adatmezejébe csomagolva utazik a hálózatban.

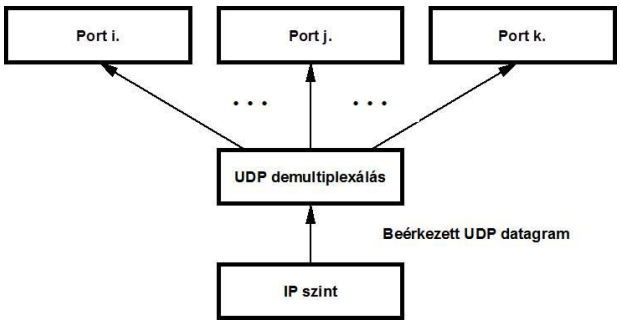


Az UDP protokoll-üzeneteket ***felhasználói datagramnak*** – ***user datagram*** – nevezzük. A user datagram ***fejlécre***, és ***adatmezőre*** oszlik. A user datagram felépítése a következő:



A **Source Port** és **Destination Port** mezők rendre az UDP datagramot feladó process hoszton belüli port-címét, illetve a címzett process – címzett hoszton belüli – port-címét jelölik. (A **Source Port** megadása opcionális. Ha nem használjuk, értékét **0**-ra kell állítani.) Az **UDP Length** az UDP datagram – header + data – oktetekben mért hosszát adja meg. Az opcionális **UDP Checksum** mező az UDP datagramban esetleg bekövetkező hibák felderítésére szolgál (pseudo header!).

Mindezek után látható, hogy az UDP protokoll feladata lényegében a beérkező UDP datagramok megfelelő porthoz irányításában merül ki. Ezt ***demultiplexálásnak*** nevezzük, és vázlatos működése az ábrán követhető.



Egy hoszton belül a portok, mint láttuk, végső soron programokhoz kapcsolódnak. Ahhoz, hogy egy távoli ***kliens*** kapcsolatba léphessen a ***server*** egy adott programjával, a server IP címén túl ismernie kell annak port-címét is. Ennek megkönnyítésére vezették be a ***jól ismert port*** – ***well-known port*** – fogalmát. Ennek értelmében bizonyos applikációk minden Internet hosztban egy ***központilag meghatározott*** – 1…1 023 közötti – port címen érhetők el. Az így le nem foglalt port-címek kiosztása a hosztok joga, és a szabadon felhasználható port-címek kiosztása a beérkező ***dinamikus*** igények alapján történik.

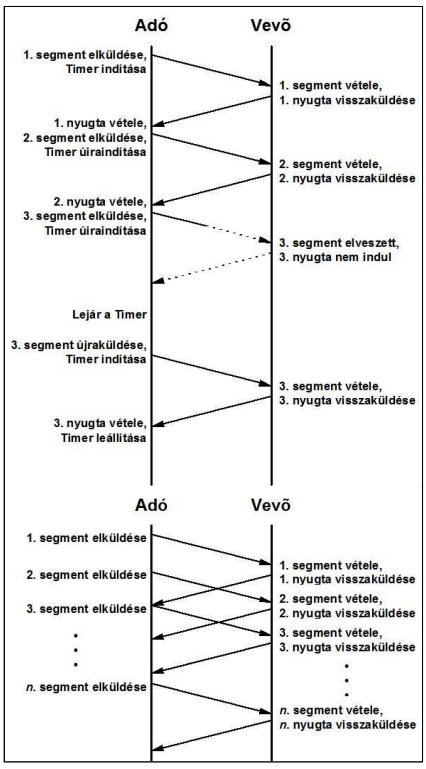
3.8. Az IP hálózat felhasználása: TCP

Ez idáig kapcsolatmentes, felelősség nélküli szállítási szolgáltatásokról beszéltünk. A hálózat gyakorlati felhasználásainak jelentős hányada azonban nem képes feladatát helyesen betölteni, ha az átvitel alatt adatvesztés következik be (gondoljunk csak egy file átvitelre). Mindez szükségessé tette olyan protokoll kidolgozását, amely ***megbízható szállítási szolgáltatást*** nyújt. Ezt a protokollt ***szállítás vezérlési protokollnak*** – ***Transmission Control Protocol***, ***TCP*** – nevezik. A TCP maximálisan kihasználja a már megismert IP tulajdonságait, de attól valójában ***független***, más datagram szállítási rendszer felett is üzemeltethető. Az Internetben – az IP-vel együtt – olyan fontos szerepet tölt be, hogy a két különálló protokollt gyakran közös néven – ***TCP/IP*** – említik.

A TCP megléte ***mentesíti*** az egyedi applikációkat attól a tetemes feladattól, hogy maguk gondoskodjanak a hibamentes átvitel megvalósításáról. Ezután az applikáció és a TCP közti kapcsolat ***5*** szempont szerint jellemezhető:

1. ***Bitfolyam szemlélet –*** A TCP kapcsolaton átvitt adat ***bitsorozat***, vagyis semmilyen belső struktúrát – még ha rendelkeznek is ilyennel – nem tulajdonítunk az adatoknak. Az egyetlen betartandó szabály, hogy az átvitt adatmennyiségnek ***egész számú oktetet*** kell tartalmaznia.
2. ***Kapcsolat orientált – Virtual Circuit Connection*** – A tényleges adatátvitel megkezdése ***előtt*** a két végpontnak virtuális kapcsolatba kell lépnie, erőforrásokat kell allokálnia a kapcsolat idejére, és az átvitel alatt mindkét félnek a sikeresen vett adatokról ***nyugtákat*** kell küldenie partnerének.
3. ***Tárolt továbbítás – Buffered Transfer*** – A TCP protokoll a szállításra átadott adatokat a maga igényei szerint tördelheti, de azokat ***sorrendhelyesen*** adja át az adatot feldolgozó – ***consumer*** – applikációnak. Ez bizonyos esetekben – pl. távoli terminal-echo – problémát okozhatna, ezért lehetőséget biztosítottak arra, hogy az applikáció utasíthassa a protokollt az azonnali szállításra. Ezt a parancsot ***push***-nak nevezik.
4. ***Részenkénti szállítás – Unstructured Stream*** – A feldolgozó applikáció nem számíthat arra, hogy a beérkezett adatokat logikai határokra illesztve kapja meg (ellentmondana az 1. pontnak). Ezért az esetleges rekord összeillesztésekről az applikációnak kell gondoskodnia.
5. ***Független kétirányú kapcsolat – Full Duplex Connection*** – Egy kiépített TCP kapcsolatra úgy tekinthetünk, mintha két ellentétes irányú, ***független*** csatornánk lenne. Arra is lehetőségünk van, hogy az egyik irányú kapcsolatot a másik előtt szakítsuk meg. (A tényleges kapcsolat azonban egy időben szakad meg, mert a csatornák nemcsak a hasznos adatok, de az ellenkező irányú nyugták szállítását is ellátják. Ezt a megoldást ***piggybacking***-nek nevezik.)

- A TCP a hibamentes átvitelhez a ***pozitív nyugtázás újraküldéssel – positiv acknowledgement with retransmission*** – néven ismert eljárást használja. A módszer működése vázlatosan a következő:

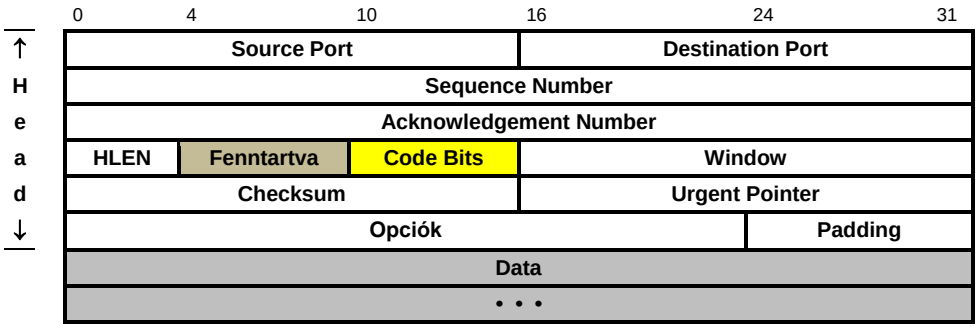


  Az adó elküld egy csomagot – ezzel egy időben elindít egy időzítőt is –, majd pozitív nyugtára vár. A nyugta időbeni megérkezése jelzi, hogy a vevő hibátlanul vette az adást. Ha az időzítés a nyugta visszaérkezése előtt jár le, az adó feltételezi, hogy az adat elveszett – meghibásodott – ezért újra elküldi a tárolt csomagot. Annak érdekében, hogy az esetleg megduplázódott csomagokat/nyugtákat felismerhessük, mindegyiket ***sorszámmal*** kell ellátni.

  Amint az első ábrán is látható rengeteg idő telik el várakozással, lényegében csak fél-duplex átvitel zajlik. Jelentősen javítható a helyzet, ha egy időben több nyugtázatlan csomag lehet úton. Ezt a megoldást ***csúszó***, vagy ***körbeforgó ablak*** – ***sliding window*** – technikának nevezik. Az egy időben úton lévő csomagok maximális számát az ***ablakméret – window size*** – határozza meg. A módszer hatékonysága az ablakmérettől és a hálózat sebességétől függ, optimális esetben elérhető a szaturáció (a vonali kapacitás teljes kihasználása). A működést az alsó ábra szemlélteti. A TCP a sliding window technika egy speciális – és különösen hatékony – változatát használja.

Mivel a TCP kapcsolat-orientált protokoll, így a tényleges adatcsere előtt ***kiépül*** az adó és vevő között egy logikai kapcsolat – ***virtuális áramkör***, ***virtual circuit*** –. Az adó a sliding window technika alkalmazása érdekében három ***mutatót*** – ***pointer*** – használ, amelyeket a virtuális kapcsolat létrejöttekor 0-ra állít, majd az átvitel során folyamatosan növel. A pointerek valójában a virtuális áramkörön átküldött adatfolyam ***ofszetei***, vagyis az elküldött oktetek sorszámai. A baloldali pointer a csúszó ablak alsó határát – a virtuális áramkörön már kiküldött, és már ***pozitívan nyugtázott*** legmagasabb ofszetű oktetet – jelöli. A jobboldali pointer a csúszó ablak felső határát – egy további pozitív nyugta beérkezése előtt ***még elküldhető*** legmagasabb ofszetű oktetet – jelöli. A középső pointer az ablak belsejébe – a ***következőként elküldendő*** oktetre – mutat. A vevő hasonló pointer-készlettel rendelkezik, amelyet a nyugták generálásához használ. A bal- és jobb pointer közötti "távolság" az ***ablakméret***, vagyis a nyugta nélkül elküldhető oktetek száma. A TCP egyik érdekessége, hogy az ablakméret dinamikusan változhat. A vevő a visszaküldött nyugtában jelzi, hogy hány további oktet fogadására készült fel. Ezt az adatot "***ablakméret ajánlatnak***" – ***window advertisement*** – nevezzük. Az ajánlat változtatásával a vevő finoman szabályozhatja az adó által kibocsátott adat mennyiségét.

A TCP protokoll adatelemét ***szegmensnek*** – ***segment*** – nevezik, és felépítése a következő:



A **Source Port** és **Destination Port** mezők – az UDP-nél megismert módon – a virtuális áramkör címzett gépeken belüli végpontjait jelölik ki. A **Sequence Number** az adott szegmens **Data** mezejében érkező oktetek adatfolyamon belüli ofszetét határozza meg. Az **Acknowledgement Number** az ***ellenkező irányú*** adatfolyam átvitelére vonatkozó nyugtát – a vevő által elvárt következő oktet ofszetét – jelzi. A **HLEN** a szegmens fejlécének 4-oktetben mért hosszát adja meg (ha nincs opció, akkor **5**). A **Code Bits** 6 bites mező – balról jobbra haladva – a következő biteket tartalmazza:

URG ***Urgent Pointer Valid*** – **1** értéke jelzi, hogy a szegmens "***sürgős***" – ***urgent*** – adatot tartalmaz, ekkor az **Urgent Pointer** a sürgős adat ablakon belüli végcímét jelöli.

ACK ***Acknowledgement Valid*** – **1** értéke jelzi, hogy az **Acknowledgement Number** értéke érvényes. Ez az érték a vevő által eddig a logikai kapcsolaton hibátlanul vett legmagasabb ofszetű oktetre mutató pointer (***kumulatív nyugta***). A kumulatív nyugtázás előnye, hogy egy nyugta elvesztése nem okoz felesleges újraküldést.

PSH ***Push*** – A szegmens azonnal elküldendő adatot tartalmaz, ezért a felhasználó arra utasítja a TCP protokollt, hogy ne várakozzon további adatokra ezen oktetek elküldése előtt.

RST ***Reset Connection*** – **1** értéke a virtuális áramkör mindkét irányú, azonnali megszakítását – abnormális kapcsolatbontást – eredményez.

SYN ***Synchronize Sequence numbers*** – a virtuális áramkör kiépítéséhez szükséges kezdeti műveletekhez szükséges adatcserét különbözteti meg a normális adatfolyamtól.

FIN ***End of Stream*** – Virtuális áramkör – ***egyik irányú*** – lezárása.

A **Window** a vevő által javasolt ablakméretet adja meg. A **Checksum** a szegmens fejlécének ellenőrző összegét tárolja.

A TCP protokoll az elküldött szegmensekre visszaérkező nyugták beérkezési idejének mérésével szélsőségesen változó késleltetési idejű, illetve a fellépő torlódások miatt adatokat vesztő hálózatokhoz is képes hatékonyan igazodni. A TCP protokoll az UDP-nél megismert "***jól ismert portokhoz***" – ***well known port*** – hasonló port kiosztást alkalmaz.

A TCP számos – magasabb szintű – Internet protokoll számára nyújt sorrendtartó, hibamentes adatátviteli szolgáltatást.

4. Irodalomjegyzék

David J. Wetherall, Andrew S. Tanenbaum
Számítógép Hálózatok (Harmadik, bővített, átdolgozott kiadás)
Panem Könyvkiadó Kft. Budapest, 2013, ISBN: 978-963-545-529-4

Petrényi József: TCP/IP alapok I. kötet
http://mek.oszk.hu/08300/08374/

Néhány további – többnyire angol nyelvű – ajánlott irodalom:

Petrényi József: TCP/IP alapok II. kötet
http://mek.oszk.hu/08300/08374/

IBM Redbooks: TCP/IP Tutorial and Technical Overview
http://www.redbooks.ibm.com/redbooks.nsf/RedbookAbstracts/gg243376.html

Charles M. Kozierok: The TCP/IP Guide (On-line verzió)
http://www.tcpipguide.com/free

Connected: An Internet Encyclopedia (On-line verzió)
http://www.freesoft.org/CIE/

Stephen A. Thomas: IP kapcsolás és útválasztás
Kiskapu Kft. 2002. (ISBN 963-9301-41-8)

W. Richard Stevens: TCP/IP Illustrated, Volume 1 The Protocols
Addison Wesley Longman, Inc. 1994 (ISBN 0-201-63346-9)

Eric A. Hall: Internet Core Protocols: The Definitive Guide
O'Reilly & Associates, Inc. 2000 (ISBN 1-56592-572-6)