2014

Számítógép-hálózatok

ÖSSZEFOGLALÓ JEGYZET

Disclaimer

Jelen jegyzet a Budapesti Műszaki és Gazdaságtudományi Egyetem **BMEVIHI215 – Számítógép-hálózatok** tárgyának hallgatók számára elérhető, 2013/2014 2. félévi diasorainak felhasználásával, Dr. Simon Vilmos beleegyezésével készült.

A diasorok kizárólagos tulajdonosa a Budapesti Műszaki és Gazdaságtudományi Egyetem.

A jegyzet a saját munkám, a jegyzetben található esetleges logikai és egyéb hibákért a tárgy oktatói nem tartoznak felelősséggel, ugyanakkor nagyra értékelem, ha valaki a javító szándékú megjegyzéseit elküldi nekem emailben – amint az időm engedi – értsd: nyáron ☺–, javítani fogom a jegyzet korábban megírt részeit.

A jegyzet **egyetemen belül** szabadon felhasználható, az egyetemen kívüli felhasználásra a diasorokban foglaltak vonatkoznak!

A dokumentum utolsó frissítése: **2014.05.29.**

Tartalomjegyzék

1 2014.02.11 – Bevezető előadás 1.

1.1 Az internet „atyjai”

- J. C. R. Licklider
  - ARPA/DARPA: (Defense) Advanced Research Projects Agency
  - összekapcsolt számítógépet víziója
  - szerves ember-gép kapcsolat
- Paul Baran
  - decentralizált hálózatok elve
- Leonard Kleinrock
  - csomagkapcsolt hálózatok elve
  - ARPA már csomagkapcsolt elven működött – az ő hatására
  - hierarchikus routing megalkotója
  - csomagkapcsolás lényege:
    - vonalkapcsolás addigi használata helyett (pl. telefonhálózat)
    - kis adagokban továbbított adatok
    - csomópontról csomópontra („store-and-forward” elv)
    - posta analógiájára (csomagok vándorolnak egyik „bázisról” másikra)
    - a számítógép közti kapcsolat jellemzően burst-ös
- ARPANET
  - 1969-1970-ben próbálták ki először a működését
  - 1980-ban USA-ban széles elterjedtség (kutatóegyetemek, nagyobb városok közt)
  - először NCP (Network Control Protocol) használata, ezt később felváltotta az a TCP/IP protokoll
- Robert Kahn, Vinton Cerf
  - IP: Internet Protocol
    - csomópontok viselkedését szabályozza
    - **best effort** szolgáltatást biztosít
  - TCP: Transmission Control Protocol
    - végpontok közti kommunikációt szabályozza
    - **megbízható** átvitelt biztosít
    - később UDP megjelenése
  - intelligencia a hálózatok szélén (végpontokon)
  - router kezdeti koncepciója
- Vannevar Bush
  - asszociatív keresés alapjai
  - emberi agy működését modellezi a kereső, nem könyvtári keresőnél megszokott elven működik
  - hiperlinkek, stb.
- Sir Tim Berners-Lee
  - a WWW megalkotója
  - CERN: első WWW oldal (info.cern.ch)
  - HTML: Hypertext Markup Language
  - HTTP: Hyerptext Transfer Protocol
  - újabb kutatási terület: szemantikus web

1.2 Az internetről általánosságban

- Jelenleg Web 2.0 és Web 3.0 között vagyunk

- Internet általánosságban
    - Tier-1: legnagyobb ISP-k (Internet Service Provider) pl. AT&T, Sprint etc.
        - NAP: Network Access Point-on keresztül a nyilvános hálózatokhoz is kapcsolódnak
        - egymással is kapcsolódnak
    - Tier-2: kisebb ISP-k, akik a Tier-1-ben lévőkkel szerződést kötve férnek hozzá az internethez
        - Tier-2-k egymás között is kicserélik a forgalmat + NAP-on keresztül nyilvános hálózatokhoz hozzáférnek
    - Tier-3 és helyi ISP-k
        - ez a „hozzáférési hálózat”, amivel általában a felhasználó szerződést köt
        - Tier-3 és helyi ISP-k ügyfelei magasabb szintű ISP-knek, akiken keresztül tudják biztosítani az internetet
    - egy csomag sok hálózaton halad át (több ISP-n)
- GEANT
    - európai kutatói gerinchálózat
    - Magyarország is része, méghozzá egy 10 Gbit/s-es vonalon
- Helyi hálózat (LAN – Local Area Network)

2 2014.02.13 – Bevezető előadás 2.

2.1 Hálózati és internet-szabványok

- szabványosítás szempontjából fontos a **konformancia**: biztosíték arra, hogy az eszköz tudja, amit szeretnénk – ez még azonban nem biztosítás
- a szabványban specifikált dolgok egy **részhalmazát** tudja a termék
    - ha két termék azonos részhalmazt ismer, akkor képesek lesznek együttműködni
- gyári és nyilvános szabványok is léteznek
- **de facto**[1] és **de jure**[2] szabványok
- szabványosító szervezetek általánosságban
    - **IEEE – Institute of Electrical and Eletronics Engineers**
    - **IETF – Internet Engineering Task Force**
    - ISO – International Standardization Organization
    - ANSI – American National Standards Institute
    - ITU – International Telecommunication Union
        - -T: távközlés
        - -R: rádióhírközlés
    - ETSI – European Telecommunication Standard Institute
- **IETF – Internet Engineering Task Force**
    - nagy szervezet, gyártók, szolgáltatók, kutatók is tagjai
    - **Working Group**-okban, levelezőlisták segítségével
    - dokumentumai az **RFC**-k (Request For Comments)
    - TXT formátumúak az RFC-k
    - lassabban dolgozik, mint pl. az ISO (SMTP vs. Internet Standard)
    - vicces RFC-k
    - publikusan hozzáférhetőek
- **IEEE – Institute of Electrical and Eletronics Engineers**
    - USA központú, világon a legnagyobb villamosmérnök-egyesület
    - különböző munkacsoportok
        - 802.3 Ethernet
        - 802.11 WLAN
        - 802.15 WPAN[3]
        - 802.16 BWA[4]
    - de facto-ból de jure szabványt csinálnak

2.2 Jellegzetes internet-alkalmazások és igényeik a hálózattal szemben

- **connectivity** - az összekőthetőség alapvető elvárás
    - úthálózat analógiáján
        - el lehessen jutni A városból B városba
        - jó utak legyenek, megfelelő útjelzések, kereszteződések
        - ha valahol dugó van, azt kezelni kell a hálózatban (információ, elterelés stb.)
    - számítógép-hálózatban:
        - A és B végpont összeköthető legyen
        - linkek és hálózatok csomópontok sorozatán keresztül

[1] **de facto**: szó szerint „a tények felöl nézve”, gyakorlatban létező, de nem rögzített szabvány, szabály
[2] **de jure**: szó szerint „a jogi helyzet szerint”, vagyis itt rögzített szabványról, szabályról van szó
[3] Wireless Personal Area Network
[4] Broadband Wireless Access

        - **megfelelő útválasztó** és **irányító protokollokon** keresztül
        - kellően **gyors** és **megbízható linkjeik** legyenek
        - ha torlódás van, el kell terelni a forgalmat, szabályozni a forrást…
    - önmagában az eljutás nem elégséges
        - eltévedés – csomagok eltévedhetnek, eldobódnak
        - kell kommunikációs eljárás és protokoll a végpontok közti adatcseréhez
    - ezek már elegendőek pl. webszerver eléréséhez, fájlátvitelhez, levelezéshez, P2P kapcsolathoz
- a **weblap objektumokat** tartalmaz (HTML fájlok, képek, videók, stb.)
- minden objektum címezhető a saját URL-jével
- **HTTP – HyperText Transfer Protocol**
    - kliens-szerver modellre épül
    - kliens kéréseit a szerver szolgálja ki
    - **független az átviteli rétegtől** (de általában TCP/IP van fölötte)
    - HTTP 1.1 újdonságok
        - perzisztens kapcsolat – már stream is továbbítható felette
        - pipelining – több kérés együttes elküldése válaszvárás nélkül
        - byte serving – a kért erőforrásnak csak a megadott részét küldi el a szerver
    - HTTP 2.0 az IETF fejlesztésében
        - multiplexált stream-ek TCP kapcsolaton belül
        - Google SPDY (de facto?) protokolljának lényeges elemeit használja
        - kérések priorizálása
        - kérés és válasz fejrészek tömörsítése
        - statikus tagokat nem küldi többször
        - push és hint funkció szerver oldalon
- **FTP – File Transfer Protocol**
    - file átvitel hostról hostra
    - kliens-szerver modellre épül
    - két párhuzamos kapcsolat: egy vezérlési ill. egy adatátviteli csatorna (más porton)
        - **out-of-band** (FTP) vs. in-band protokoll (HTTP)
        - **out-of-band:** a vezérlés és az adatátvitel mindenképp két külön szálon fut
    - **aktív átvitel:** a **kliens** csatlakozik a szerverhez egy porton (vezérlő kapcsolat) és megmondja, melyik porton csatlakozzon rá a **szerver**, ezt követően a szerver csatlakozik az adott portra
    - **passzív átvitel:** a **kliens** csatlakozik egy porton a **szerverrre**, majd válaszul megkapja a szervertől, hogy melyik másik portjára csatlakozzon rá adatátvitel céljából
- **Elektronikus levelezés**
    - három fő eleme van
        - **User Agent** – levelek írása, szerkesztése, olvasása (pl. Thunderbird)
        - **Levélszerverek** – postaládában a bejövő levelek, kimenő levelek várólistája szintén szerveren
        - **Levelező protokoll (SMTP)** – a szerverek között email továbbításra
            - tárolja és továbbítja a leveleket a címzett szerverének
    - levelek eléréséhez:
        - POP: Post Office Protocol – **letölti** a levelet a felhasználó gépére
        - IMAP: Internet Mail Access Protocol – összetettebb működés, több kliens működését is képes kezelni, figyeli a flageket, olvasottságot, stb.
        - HTTP: Gmail, Hotmail, Yahoo Mail stb.

- **P2P fájlcsere**
    - 1. generáció: eredeti „Napster” típusú, centralizált
        - központi szerverhez kapcsolódnak a felhasználók
        - tudatják IP címüket, megosztani kívánt tartalmat
    - 2. generáció: teljesen elosztott, nincs központi szerver (pl. Gnutella)
        - előnye a robosztusság, hátránya az exponenciális forgalomnövekedés
    - 3. generáció: práhuzamos kommunikáció egyszerre több csomóponttal (pl. BitTorrent)
        - seed, tracker, leech csomópontok
    - FTP-vel ellentétben mindenki lehet szerver és kliens is
    - előnye: jobban skálázódik, robosztusabb
    - hátránya: nagy sávszélesség igénye van
        - P2P forgalom blokkolása, rosszindulatú szeletek injektálása szolgáltatók részéről
- itt „csak” **megbízható kapcsolat szükséges** és **valamekkora sávszélesség** – a TCP/IP elég, de a TCP mindenképp kell a végpontok közti kommunikáció szabályozására
- nagyobb igénnyel bíró alkalmazások
    - audió és videó streaming (tárolt, élő, interaktív stb.)
    - VoIP (Voice over IP)
    - videókonferencia
- **Streaming**
    - forrásnál tárolt média, hálózaton továbbításra kerül a kliensnek
    - a kliens azelőtt megkezdi a lejátszást, hogy az egész tartalom megérkezne
    - időbeli követelmény – időben érkezzen meg a kijátszáshoz
    - interaktivitáshoz komoly késleltetési követelmények vannak
    - élő streaming esetén lejátszási puffer (playback buffer)
- VoIP – IP felett megy a hang, mert az IP feletti multimédia-forgalom annyira elterjedt
    - késleltetés (delay) – max. 150 ms
    - késleltetés-ingadozás (packet delay variance) – néhány 10 ms
    - csomagvesztés – maximum néhány százalék, de csak akkor, ha:
        - a kiesett részek rövidek, 10 ms nagyságrendűek
        - véletlenszerűen oszlanak meg az időben
    - Video over IP még szigorúbb
- a multimédia alkalmazások **érzékenyek a késleltetésre**, de **tűrik az adatvesztést**
    - nem kell (sőt, általában nem is szabad) megbízható kommunikáció a végpontok között (TCP protokoll)
- az első csoport egyszerű alkalmazási **tűrik a késleltetést**, de **nem tűrik az adatvesztést**

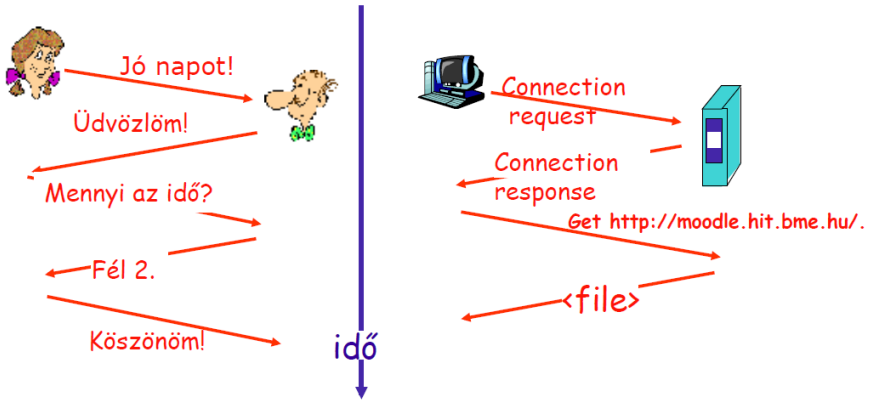
2.3 Csomagvesztés és késleltetés

- Mik a csomagvesztés és a késleltetés okai?
    - sorban állnak a csomóponti gépek (routerek) tárolóiban
    - a beérkezési sebességük meghaladja a kimenő link átbocsátóképességét
    - ha nincs hely a tárban, a csomagok eldobásra kerülnek
- csomagok késésének négy fő oka
    - feldolgozás a csomópontban
        - hibaellenőrzés
        - kimenő link meghatározása
    - sorbanállás

        - várakozás továbbításra
        - router terheltségétől függ
    - adási idő
        - R = a link adatátviteli sebessége (bit/s)
        - L = csomaghossz (bit)
        - **adási idő = L/R**
    - terjedési idő
        - d = a link fizikai hossza
        - s = terjedési sebesség az átviteli közegben (cca. $2x10^8$ m/s)
        - **terjedési idő = d/s**
    - csomóponti késleltetés összefoglalva:
        - $\mathbf{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$
        - d_{proc} = feldolgozási (processing) idő, pár us
        - d_{queue} = sorbanállási idő/késleltetés, forgalomfüggő
            - forgalom intenzitása: L*a/R
                - L*a/R ≈ 0: nincs sorban állás, kis késleltetés
                - L*a/R → 1: nagy késleltetés
                - L*a/R > 1: végtelen késleltetés, több igény jön, mint amennyit ki tudunk szolgálni
        - d_{trans} = adási idő, L/R
        - d_{prop} = terjedési idő, d/s
- csomagvesztés oka, hogy a sor kapacitása véges, betelik
- ha tele van a puffer, a csomag eldobásra kerül
- újra küldésre kerülnek az elveszett csomagok (vagy nem)
- ha a csomagellenőrzés a csomagot hibásnak találja, akkor is eldobásra kerül
    - zajok, zavarok, bithibák miatt lehet ilyen
- szükség van QoS (Quality of Service – szolgáltatásminőség) biztosító módszerekre és protokollokra
    - pl. foglaljunk csomóponti képességeket és link-kapacitásokat egy adott session számára vagy jelöljük meg a csomagokat, hogy prioritásuk szerint kerüljenek kiszolgálásra – a beszédcsomagok magasabb prioritást élveznek stb.

3 2014.02.18 – Protokollarchitektúrák

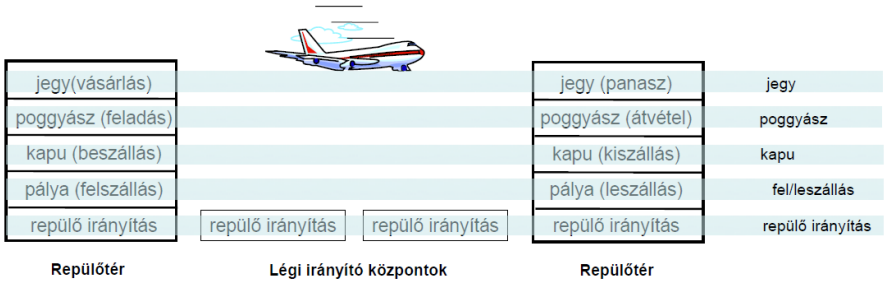
- protokollok lefektetése fontos része a számítógép-hálózatoknak
  - szintaxis: formátum meghatározása
  - szemantika: jelentés meghatározása
  - viselkedés: mit kell tenni valamely üzenet vételekor, küldésekor
- módszer → algoritmus → protokoll
  - pl. routingnál linkállapot-módszer → Dijkstra-alg. → OSPF protokoll



1. ábra Protokollok az emberi és gépi kommunikációban

  - ábrázolási forma: message sequence chart
- protokoll két elkülöníthető része:
  - továbbított információ alakja, formája
    - **statikus** rész, **Protocol Data Unit** (PDU) vagy protokoll-adategység
  - mi történik továbbításkor
    - **dinamikus** rész
    - mit kell tenni a vételkor
    - rendellenes esetek kezelése szintén
- egyetlen protokoll nem megoldás, mert a feladatokat részekre kell bontani
  - egyszerűbb a tennivalók elvégzése
  - részenként önálló szabályok alakíthatók ki
  - könnyebb, részenkénti továbbfejlesztést biztosít
  - részekre bontás
    - tipikusan **rétegekre** tagolja a teendőket
    - önálló, jellegzetes gyakori feladatok
      - illeszkedés a jelátvivő közeghez
      - összeköttetés során megbízható adattovábbítás
      - végpontok azonosítása
      - kedvezőtlen forgalmi állapotok kezelése
    - hibák kezelése

2. ábra Légi közlekedés szervezése



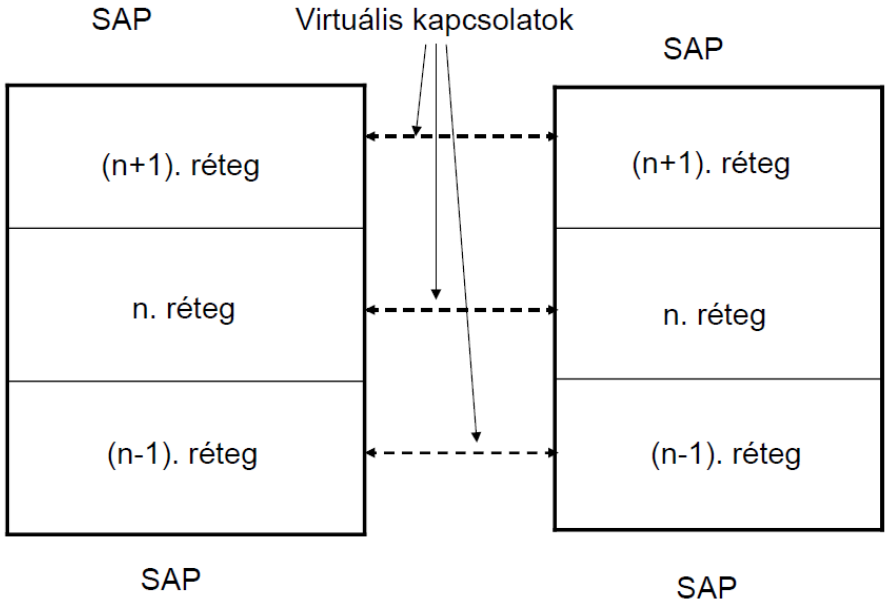
3. ábra Légi közlekedés rétegzett szervezése

- **rétegek**: minden rétegben egy-egy szolgáltatás
  - saját rétegen belüli tevékenység
  - támaszkodva az alatta lévő rétegek által nyújtott szolgáltatásra
- hálózati architektúra – protokollrétegek egymásra épülése
  - számítógép-architektúra, szoftver-architektúra
- referenciamodell – csak a rétegek és azok feladatai, a protokollok **nem**

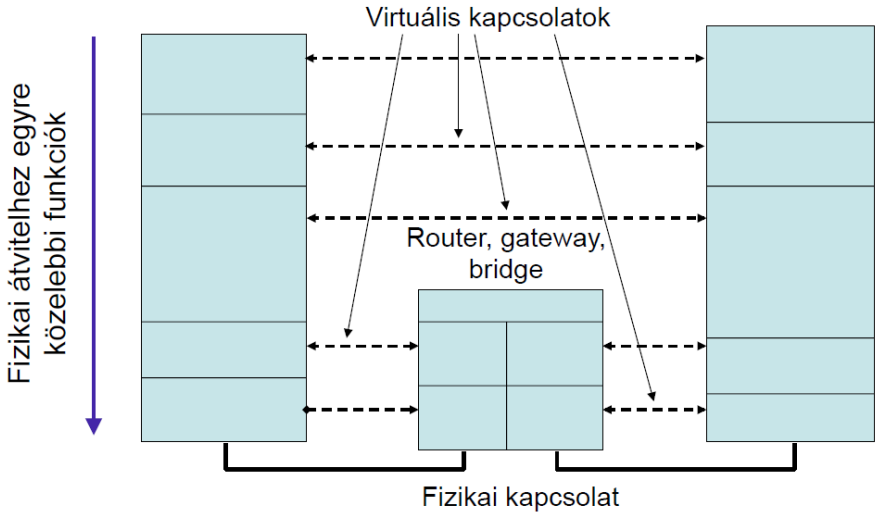
3.1 OSI referenciamodell

- ISO-ban kidolgozott modell
- Open System Interconnection – nyílt rendszerek összekapcsolása
- 7 feladatcsoport, 7 réteg
- protokollok nem részei ugyan, de kidolgoztak és szabványosítottak protokollokat is az egyes rétegekhez
- a rétegzett protokollarchitektúrák előnyei és hátrányai
  - előnyei:
    - kezelhető részekre bonthatóak a feladatok
    - az egyes részek függetlenek egymástól, így könnyen módosíthatók
    - fentebb lévő feladatoknak közös kiszolgálást nyújthat egy lentebbi rész
  - hátrányai:
    - réteg működéséhez szükséges információk másik rétegben → sérül a rétegstruktúra
    - feladatok duplikálása (pl. hibavédelem) elengedhetetlen, hatékonyságot rontja

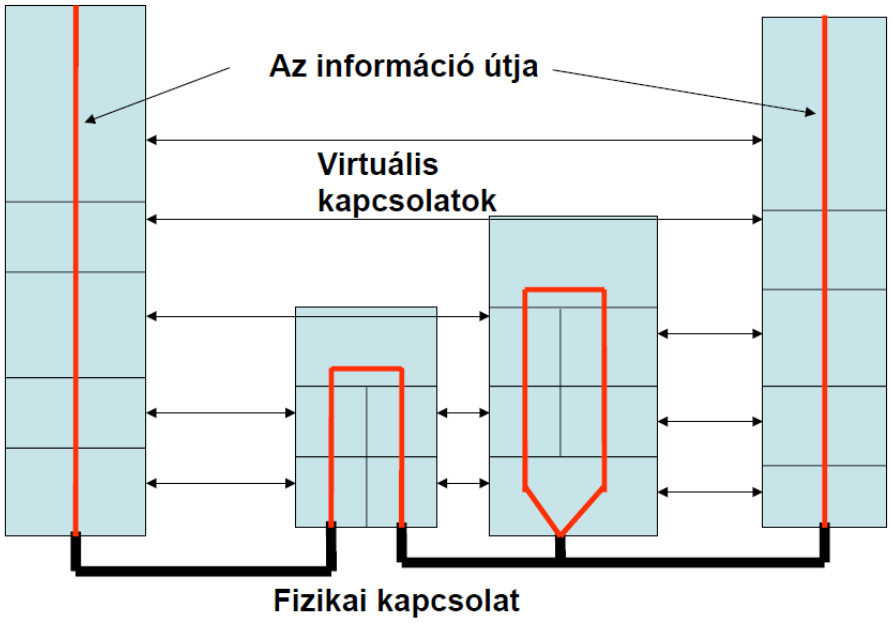
- rétegek SAP-pal (Service Access Point – szolgáltatás elérési pont) rendelkeznek és interfész-elven kommunikálnak, elrejtve a funkció tényleges megvalósítását egymástól
- azonos rétegek között virtuális kapcsolat van



- nem minden rétegre van szükség minden csomópontban



4. ábra Protokollrétegek



5. ábra Információ útja az egyes rétegekben

- PDU
  - ezek kerülnek továbbításra a szomszédos rétegek között
  - PDU-k általános felépítése
  - Fej (header) – Adat (payload) – Farok (trailer)
    - fej- és farokrészben találhatók kiegészítő információk, hibavédelem, egyéb opcionális paraméterek
  - rétegek közt lefele haladva beágyazás (encapsulation), rétegek közt felfelé haladva kibontás (decapsulation) történik

3.1.1 ISO OSI referenciamodell részei

1. Alkalmazási réteg (**Application** layer)
2. Megjelenítési réteg (**Presentation** layer)
3. Viszonyréteg (**Session** layer)
4. Szállítási réteg (**Transport** layer)
5. Hálózati réteg (**Network** layer)
6. Adatkapcsolati réteg (**Data link** layer)
7. Fizikai réteg (**Physical** layer)

- PDU elnevezések az OSI modellben
  - szigorúan véve
    - az adott réteg PDU-ja
    - vannak/voltak „szabványos” elnevezések
      - Data link – keret (frame)
      - Network – csomag (packet)
      - Transport – szegmens (segment)
      - feljebb (főként Application layer) – üzenet (message)
  - az egyszerűség kedvéért gyakran minden **csomag**nak nevezünk

3.1.1.1 Fizikai réteg (Physical layer)

- fizikai közeg specifikációja

    - érintkezők kiosztása, használatos feszültségek, kábelek specifikációja stb.
- bitek, bitcsoportok továbbítása **fizikai** csatornán
    - vonali kódolás: szimbólumreprezentáció
    - moduláció: vivő viszi át az információt
- bitszinkron biztosítása
- csatlakozók típusának és méretének rögzítése

3.1.1.2 Adatkapcsolati réteg (Data link layer)

- bitsorozatok keretezése → ebből lesznek a csomagok
- hibaellenőrzéshez szükséges adatok előállítása (vételnél: ellenőrzése) és esetleges javítása
- **fizikai címek** kezelése (MAC címek)
- **közeghozzáférés** kezelése (Media Access Control)
- forgalomszabályozás itt is lehet, de manapság már ritkán
- fontos különbség **fizikaihoz** képest: ez a réteg megmondja több készülék esetén az interakció módját egy osztott médiummal
- tipikus elemei: **bridge, switch**

3.1.1.3 Hálózati réteg (Network layer)

- Egyedi linkek logikai összefűzése **végpontok között csatornává**
    - meghatározott QoS-t biztosítva, akár több hálózaton keresztül
- **logikai címzés**: hálózati eszközök közötti kapcsolásokhoz
- **útvonalkeresés** a hálózaton belül
- forgalomirányítás és csomagjavítás
- fontos különbség **adatkapcsolatihoz** képest: az adatkapcsolati réteg egy hálózaton belül teremt kapcsolatot, a hálózati réteg több hálózaton keresztül
- tipikus elemei: **routerek**

3.1.1.4 Szállítási réteg (Transport layer)

- végpontok közti megbízható kommunikációért felelős
- csomagtovábbításon túl:
    - **hibamentes** összeköttetés létrehozása
        - hibás csomagok ismétlése
        - duplikált csomagok eldobása
        - csomagsorrend helyreállítása
    - **forgalomszabályozás**
- posta-analógia: csak a levél borítékát látja, tartalmát nem

3.1.1.5 Viszonyréteg (Session layer)

- kapcsolat **irányának** kezelése (duplexitás – kétirányúság kezelése)
- összeköttetés kezelése – **kapcsolat felépítése és bontása**
- adatfolyam-szinkronizáció – az összefüggő folyamok összehangolása
- **manapság** általában **a szállítási rétegben** implementálják

3.1.1.6 Megjelenítési réteg (Presentation layer)

- az előző rétegekkel ellentétben (amik fejben lévő meta-adatokat kezelnek), ez a réteg **felhasználói adatokat kezel**
- adatábrázolásért felelős
    - szintaktikai ellenőrzés
    - adatábrázolás
    - operációs rendszerek miatti különbségek kezelése

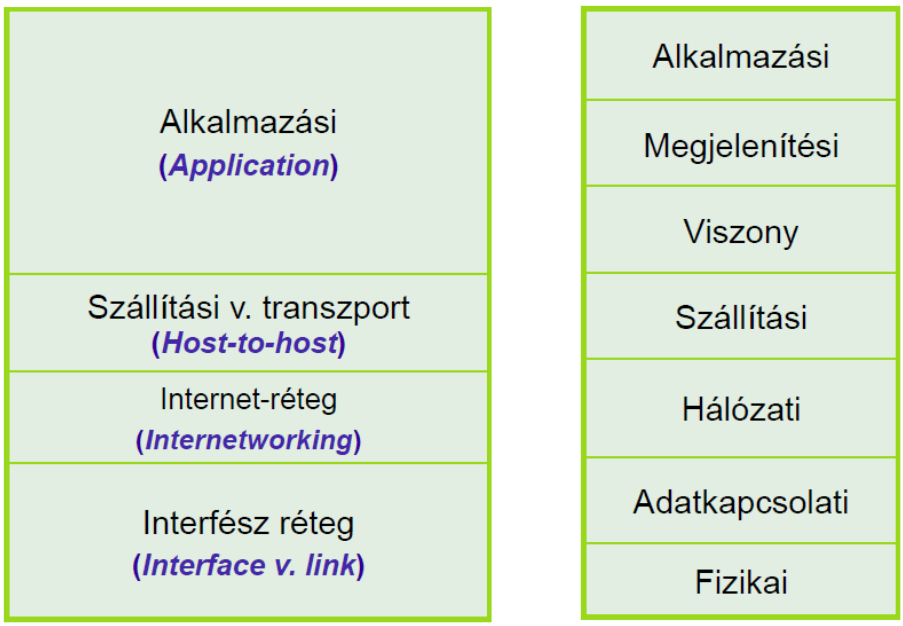
- adattömörítés és titkosítás
- általában az **alkalmazási rétegben** van implementálva

3.1.1.7 Alkalmazási réteg (Application layer)

- végpontokon futó alkalmazási programok helye
- felhasználják az alsóbb rétegek szolgáltatásait, de ők maguk csak a felhasználót szolgálják ki
- fő feladatai:
  - kommunikációs partnerek meghatározása
  - hálózati erőforrások elérhetősége
  - szinkronizáció (felhasználók közt)
  - formátum és biztonsági egyeztetés
- tipikus implementáció: **HTTP, FTP, SMTP**

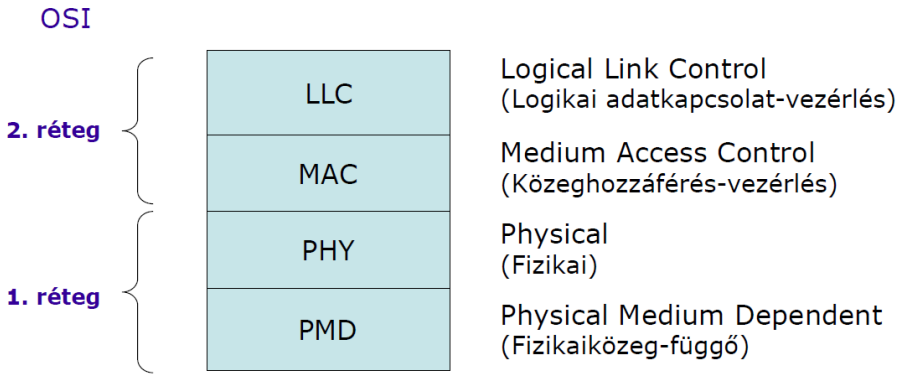
3.1.2 Nem OSI szerinti protokollarchitektúrák

- az OSI mint architektúra nem valósult meg soha teljesen
- referenciamodellként hasznos – **mérőléc** egyéb architektúrák megértéséhez
- példa nem OSI-szerinti protokollarchitektúrákra
  - TCP/IP – túl sok réteg a felső szinteken
  - IEEE – LAN – MAN – túl kevés réteg az alsó szinteken



6. ábra TCP/IP és OSI rétegek megfeleltetése

- interfész réteg – kommunikáció egy hálózati szegmensen **belül**
- internet réteg – **független hálózatok** közti összekötés (címzés, routing)
- transzport réteg – **process-process** kommunikáció
  - hibaellenőrzés, forgalomszabályozás, portkezelés
- alkalmazási réteg – **interfész a felhasználókhoz**
- már az OSI előtt is használták – internet világát tükrözi
- összeköttetésmentes csomagkommunikációt tesz lehetővé
- operációs rendszerek **többsége** mind TCP/IP-t használ

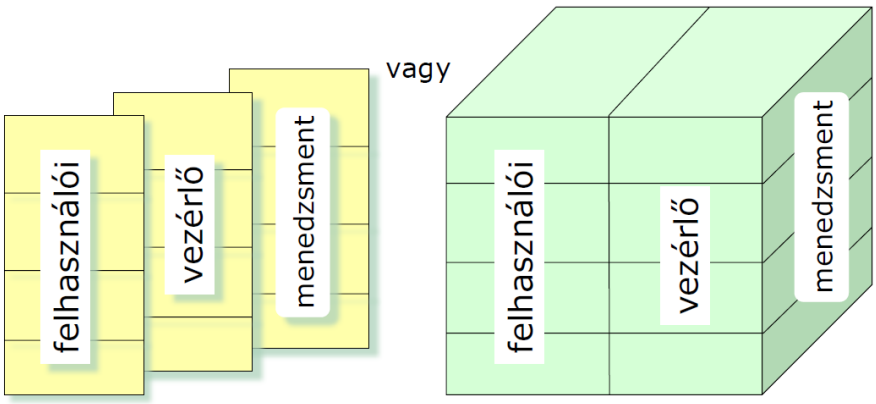


7. ábra IEEE szerinti LAN architektúra és OSI megfeleltetése

- a gyakorlatban jól használható kombinált referenciamodell
  - 5 réteg – felső 3 TCP/IP-nek, alsó 2 OSI-nak felel meg

3.1.3 Többsíkú protokollarchitektúrák

- bizonyos feladatok egymásra épülnek, de vannak eltérő jellegűek
  - pl. adatforgalom és forgalomszabályozás
- felhasználói-, vezérlő- és menedzsment feladatcsoportok szerinti **síkok**
  - felhasználói információ továbbítása
  - felhasználói információ továbbításához szükséges ismeretek cseréje
  - felhasználói információ továbbításához való képesség fenntartása, ellenőrzése



8. ábra Feladatcsoport-síkok felépítése

- cross-layer megközelítés
  - azt az elvet sérti, ami szerint a **nem szomszédos** rétegek közt nincs közvetlen kommunikáció

- példa információ lefelé való haladására
    - TCP nem tudja, hogy torlódás vagy hibás csomagok miatt van-e csomagvesztés
    - mondja meg ezt a szállítási rétegnek az alsóbb réteg
- példa információ felfelé való haladására
    - alkalmazási réteg megmondja a fizikai rétegnek, hogy milyen késleltetést vár el
- megvalósítása
    - új interfész rétegek közötti közvetlen kommunikációhoz
    - megosztott adatbázis rétegeknek
    - teljesen új absztrakció – itt már nincsenek klasszikusan vett protokoll rétegek

4 2014.02.20 – Fizikai szintű kommunikáció 1.

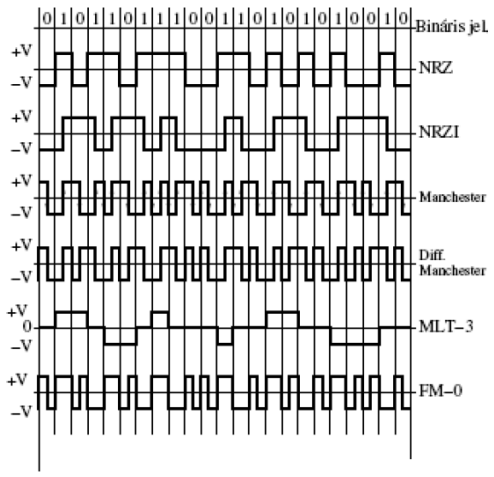
- fizikai szintű átvitel
  - a hálózatnak az a funkcionalitása, amely két csomópont között bitek (szimbólumok) továbbításával foglalkozik
  - az **adó** szimbólumokból **jelalakokat** készít, majd ezeket továbbítja a csatornán
  - a jelek **torzulnak**, **zajok** és **zavarok** adódhatnak hozzájuk
  - a vevő feladata, hogy a kapott jelből olyan szimbólumsorozatot állítson elő, ami a lehető legjobban megfelel az adó által küldöttnek
- **bit, bitsebesség** – bit/s, **információátviteli** sebesség
- **szimbólum, szimbólumsebesség** – baud (Bd), **jelzési** sebesség
- hány bitnyi információt hordoz egy szimbólum?
  - bináris átvitel – a két sebesség megegyezik
  - többszintű átvitel – a bitsebesség többszöröse lehet a szimbólumsebességnek
    - pl. 4 szintű átvitelnél (00, 01, 10, 11 miatt) kétszerese a bitsebesség a jelzési sebességnek
- elvileg korlátlanul növelhetők a szintek, de
  - a gyakorlatban sávkorlátosak – ezért torzítanak, elkenik a jeleket
  - zajok és zavarok forrásai – döntéskor hibát okoznak
  - minél több szint van, annál közelebb kerülnek egymáshoz az egyes szintek
- szemábra – oszcilloszkópos megfigyelésnél az egymás utáni jelalakok egymásra rajzolódnak, a szem tisztasága jelzi, hogy mekkora esélyünk van adott időpontban a helyes döntésre
- átvitel zajos csatornán – additív (hozzáadódó) zaj miatt minél nagyobb a zaj, annál többször van tévesztés
- az átvitelnél fontos
  - ne legyen DC tartalma a jelnek – emiatt bipoláris kódolás (AMI – Alternate Mark Inversion) szükséges: -1, 0, +1 jelszintek
  - bitszinkron biztosításához órajel kinyerése – ne legyenek hosszú és egyforma bitsorozatok
  - vonali kódolás: **digitális alapsávi moduláció**
    - a forrás szimbólumsorozatához olyan jelsorozatot ad, amely az átvitel során a legkevésbé torzulhat

4.1 Vonali kódolások

4.1.1 Return-to-zero kódolás

- két impulzus között mindig visszatér a 0 szintre
- szinkron biztosított, de kétszer akkora sávszélesség kell ugyanakkora adatátviteli sebesség eléréséhez
- van DC komponens hosszú 0 vagy 1 sorozatok esetén

4.1.2 Vonali kódolások összehasonlítása



non-return-to-zero (NRZ)

NRZ inverted (NRZI)

Manchester

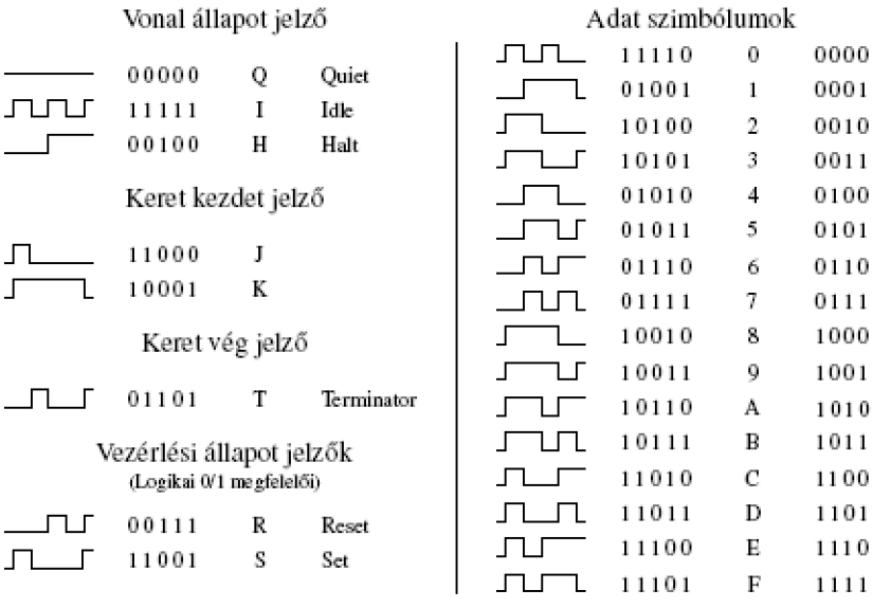
Diff. Manchester

más kód 1.

más kód 2.

- **RZ** (return-to-zero)
  - \+ megoldott a szinkron
  - \- kétszeres sávszélesség kell, mint NRZ-nek
- **NRZ** (non-return-to-zero)
  - \+ fele akkora sávszélesség, mint RZ-nél vagy Manchesternél
  - \- hosszú 1-es/0-s sorozat esetén szinkronvesztés
- **NRZI** (non-return-to-zero-inverted)
  - \+ ugyanaz, mint NRZ
  - \- 1-eseknél mindig van váltás a bitidő közepén, hosszú 0 itt is problémás – megoldás: bit stuffing
- **Manchester**
  - 1-esnél felfelé, 0-nál lefelé váltás bitidő közepén, egymás utáni azonos biteknél bithatáron mindig vált
  - pl. Ethernetnél, RFID-nél használják
  - \+ jól szinkronizálható, nincs DC
  - \- kétszeres bitsebesség, mint NRZ esetében
- **Differenciális Manchester**
  - \+ hibavédettebb (nem számít a polaritás, csak az átmenet)
  - 0-nál bitidő elején és közepén is váltás van
  - pl. optikai tárolásra, token ring esetében
- **MLT** (Multi-level Transmit)-3
  - 3 feszültségszint – ciklikusan mozog, ha nem 0, egyébként marad ugyanazon a szinten
  - \+ kevesebb interferencia, kisebb sávszélesség
- **FM-0**
  - minden bithatáron váltás, 0-nál középen is
  - \+ szinkron, nincs DC komponens
  - \- több sávszélességet igényel
- 4B/5B kódolás
  - hatékonyabb kódolás – összeegyezteti a jól szinkronizálhatóság és a kis sávszélesség-növekedés feltételeit
  - a bemeneti bitfolyam minden bit-négyeség 5 bites kódokkal helyettesítjük
  - a 16 helyett 32 kombinációnk van, amiből

- csak azokat használjuk, amik jó tulajdonságúk – legalább két váltás a sorozaton belül
- a többiből 8-at speciális jelzésekre használunk, a maradék 8-at nem használjuk fel



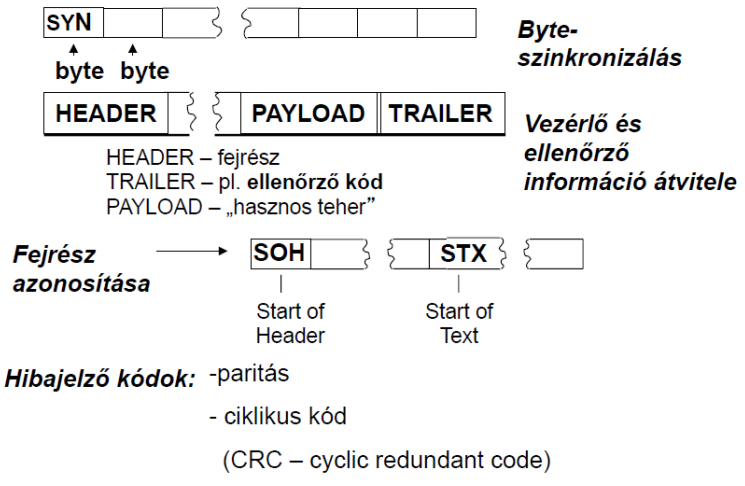
9. ábra 4B/5B kódolás illusztrációja NRZI alkalmazása mellett

4.2 Bitfolyamok strukturálása

- keretezésre van szükség **hibavédelmi** és **szinkronizálási** célokból
- szinkronizálás – bitszinkron, byte-szinkron, keretszinkron
- hibajelzés – paritáskezelés, ciklikus redundancia-ellenőrzés (CRC)

4.2.1 aszinkron és szinkron átvitel

- aszinkronnál **start** és **stop** bitek – órajel nem kerül továbbításra
  - jól használható text továbbításához (pl. telex), csak kis sebességre
- szinkronnál bitbeszúrás, keret alkalmazása



10. ábra Szinkronizálás, hibejelzés

4.2.2 Modulációs eljárások, digitális moduláció

- **moduláció** – a szimbólumsorozat továbbítására egy – a jelzési sebességnél általában nagyobb frekvenciájú – szinuszos vivőt használunk
- a vivő jellemzőjét változtatjuk a szimbólumoknak megfelelően
    - amplitúdóját – **ASK** (amplitude-shift-keying)
    - frekvenciáját – **FSK** (frequency-shift-keying)
    - fázisát – **PSK** (phase-shift-keying)

4.2.2.1 ASK

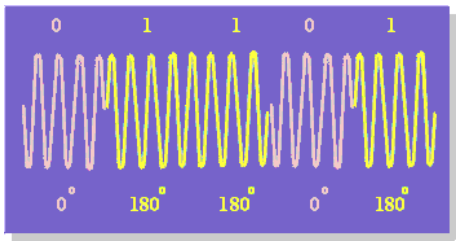
- előnye: ASK moduláció és demoduláció olcsó, egyszerű
- hátránya: érzékeny a zajokra, különböző terjedési utak körülményeire
- felhasználása: digitális adat átvitele optikai kábelen

4.2.2.2 FSK

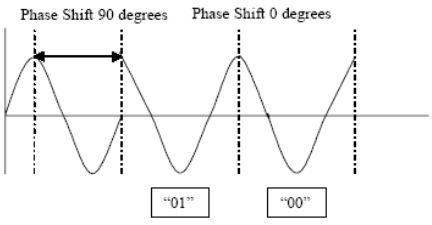
- 0 szimbólumnak f_1 frekvencia, 1 szimbólumnak f_2 frekvencia felel meg
- a vivő frekvenciáját változtatjuk a moduláló jel szerint
- előnyei:
    - egyszerű moduláció és demoduláció
    - érzéketlenség a nem-lineáris erősítők torzítására
        - jó hatásfokú erősítők alkalmazása
        - gazdaságos akkumulátorhasználat a mobil készülékekben
- hátránya: sávszélesség kihasználása nem olyan jó, mint PSK esetében
- népszerű változatai:
    - AFSK (audio FSK) – modemek, alapsávban
    - GMSK (Gaussian Minimum Shift Keying) – GSM
- felhasználása: távoli mérőóra leolvasás

4.2.2.3 PSK

- Bináris PSK (BPSK)



- többszintű PSK esetén 90 fokonként van forgatás



- konstellációs diagramon komplex számokkal ábrázoljuk az átviendő szimbólumokat
- kvadratúra vivők: modulálva a valós és képzetes résszel a sin illetve cos vivőt, mindkettőt átvisszük ugyanazon a frekvencián
- vevő arra a szimbólumra választ, amihez a vett szimbólumnak legkisebb az euklidészi távolsága

- BPSK – QPSK – 8PSK
    - QPSK megduplázza a BPSK adatsebességét
        - viszont nagyobb adóteljesítményre van szükség ugyanazon jel-zaj viszonyhoz
        - Bluetooth 2 (DQPSK)
    - BPSK
        - hibatűrő, de alacsony sávszélességet tud csak
        - RFID (biometrikus útlevelek, bankkártyák…)
    - 8PSK
        - legmagasabb szintű PSK
        - 8 szint fellett már magas a hibaarány
        - EDGE, 2.5G

4.2.2.4 QAM (Kvadratúra amplitúdó manipuláció)

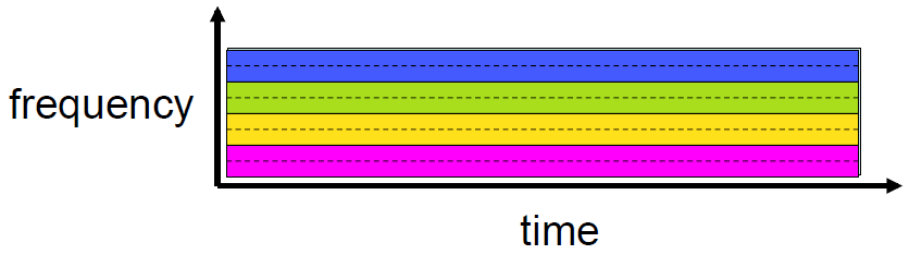
- két, 90°-kal eltolt vivőnek egyszerre moduláljuk az amplitúdóját, majd összeadva átküldjük a csatornán
- leggyakoribb a 16-, 64-, 128- és 256-QAM – egyre nagyobb sebesség, de rosszabb hibaarány
- 8PSK felett már érdemes áttérni 16-QAM-re
- 64-QAM és 256-QAM: digitális kábel és földfelszíni TV adás

4.2.2.5 Adaptív moduláció

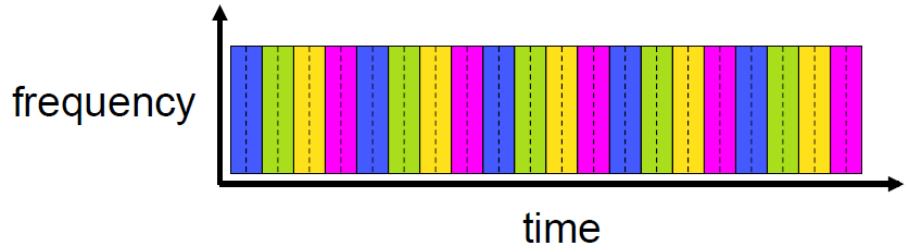
- a rádiós link állapotától függően vált modulációt ← szükséges hozzá a csatornainformáció
- 3G mobilrendszereknél a HSPA (High-Speed Packet Access)
    - zajos csatornánál QPSK
    - tisztább csatornánál 16-QAM

4.2.2.6 Multiplexálás

- több csatorna jelét egy jellé alakítani egy osztott csatornán
- egy kommunikációs csatornát magasabb szintű logikai csatornává
- demultiplexálás – vevő oldalon elő lehessen állítani az eredeti csatornákat
- **FDM – Frequency Divison Multiplexing**



- **TDM – Time Division Multiplexing**



- **WDM – Wavelength Divison Multiplexing**
    - több eltérő hullámhosszú fényt viszünk át – több csatorna alakítható ki
    - végülis FDM, csak üvegszálon így hívjuk

    - népszerű megoldás, nem kell új optikai kábeleket lefektetni
- **CDM – Code Division Multiplexing**
    - kódosztású nyalábolás
    - analógia: egyszerre beszél több ember különböző nyelveken
    - adatokhoz csatornánként speciális kód (chip kód)
    - minden felhasználónak saját chipkódja van
    - szórt és széles sávú pszeudozajjal kerül átvitelre a jel, a túloldalon XOR-ozzák vele
- szórt spektrum előnyei-hátrányai
    - előnyei:
        - nehezen lehallgatható, chip kód miatt zajnak tűnik
        - nehezen blokkolható
        - nem érzékeny a keskenysávú interferenciára és a többutas terjedésre
        - frekvencia újra felhasználható
    - hátrányai:
        - nagyon pontos teljesítmény-szabályozás kell
        - interferencia a chip-határoknál szinkron szétcsúszásakor
        - bonyolultabb adó-vevő

5 2014.02.25 – Fizikai szintű kommunikáció 2.

- hírközlő csatornék a gyakorlatban
    - (réz)vezetékes csatornák
    - fényvezetős (üvegszálas) csatornák
    - vezetéknélküli csatornák
        - szabadtéri fényátvitel
        - infravörös átvitel
        - rádiós átvitel: földfelszíni, műholdas és ezen belül mobil

5.1 Fémvezetők

- két fémvezető, köztük dielektromos szigetelés – akár levegő is lehet (légvezeték)
    - TEM – transzverzális elektromos-mágneses hullámvezető
- két fő típusa: szimmetrikus érpár (sodrott érpár) és koaxiális kábel

5.1.1 Sodrott érpár

- két szigetelt, egymásra spirálisan felcsavart rézvezeték – UTP: árnyékolatlan (unshielded)
- a csavarás a két ér egymásra hatását és külső zavarok küszöböl ki, pl. elektromágneses interferencia
- minél sűrűbb a csavarás, annál nagyobb az adatátviteli sebesség és a méterenkénti ár is
- „Cat” jelző utal a kábelek minőségére, fejlettségére
- Cat-5 elterjedt (pl. Fast Ethernet)
- kábelek esetén jelölések
    - Shielded (S) – árnyékolt, fémhálóval
    - Foiled (F) – árnyékolt, fémlemezzel
- Cat-6 és Cat-7 nem terjedt el
- előnyei
    - vékony, rugalmas kábel: könnyű szerelés
    - mérete miatt sok fér belőle egy kábelcsatornába
    - a legolcsóbb LAN kábel
- hátrányai
    - komoly előírások szereléskor az EMI[5] miatt
    - videó átvitelnél különböző hosszak miatt késleltetés – szellemkép, színtorzulás, amit kompenzálni kell – bonyolult

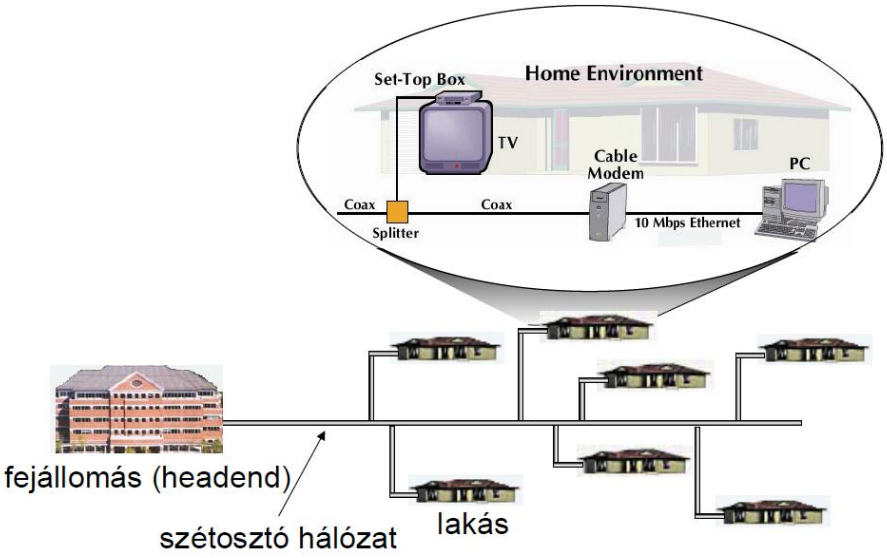
5.1.1.1 Struktuált kábelezés

- döntően UTP-kábelezést használ
- központi elosztókból (rendezők) minden végponthoz külön kábel megy
- általános célú telefon- és számítógép-hálózat kialakítására egyaránt alkalmas
- a végpontra bármi csatlakoztatható
- könnyen (át)konfigurálható
- fő elemei
    - főrendező – az épület központi rendezője, ahova a gerinchálózat csatlakozik
    - alrendezők – szintenkénti kábelezés elosztói
    - vízszintes kábelezés – többnyire sodrott érpáras, de lehet vezetéknélküli is
    - csatlakozók

[5] elektromágneses interferencia

5.1.2 Koaxiális kábelek

- amennyiben egy vezeték magas frekvenciájú áramot visz át: antenna viselkedés
- az egyik vezető körülöleli a másikat, nincs kimenő rádióhullám a belsőből, a külső le van földelve
- két végén lezárás hullámimpedanciával (Ethernet – 50 Ohm, Kábel TV – 75 Ohm)
- jó zavarvédettség és nagy sávszélesség
- számítógép-hálózatokban ma már kevésbé használják, az optika kezdi kiszorítani TV-knél, internetelérésben pedig a sodrott érpár



11. ábra Kábeltévé-hálózat

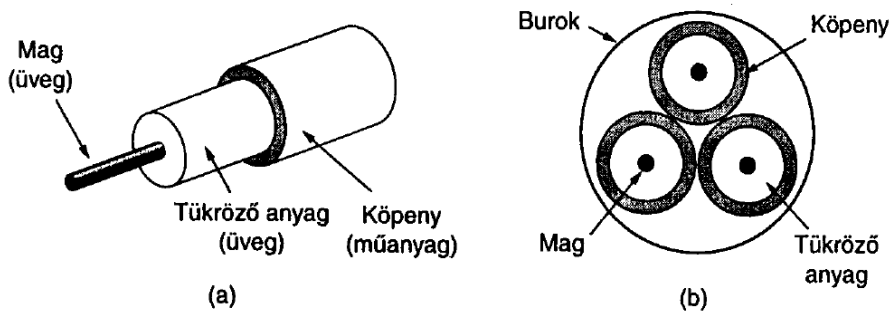
5.1.3 Optikai kábelek, optikai ablakok

5.1.3.1 Optikai ablakok

- **850 nm**
  - 1970-es évek
  - 2-3 dB/km-es szálcsillapítás
  - 100 Mbit/s modulációs sebesség
- **1310 nm**
  - 1980-es évek közepe
  - 0,5 dB/km-es szálcsillapítás
  - átviteli sebesség Gbit/s-os nagyságrendbe nőtt
  - optikai szálnak itt van diszperziós minimuma
- **1550 nm**
  - ideális elnyelés szempontjából
  - 0,2 dB/km-es szálcsillapítás
  - néhányszor 10 Gbit/s-os nagyságrendű sebesség
  - WDM rendszerek használják főként
- veszteségek
  - visszaverődés (reflexió) – határfelületek gondos összeillesztésével minimálisra csökkenthető
  - csillapítás – megfelelő anyagválasztással minimalizálható
  - fénytörés (refrakció) – teljes visszaverődés jelensége

5.1.3.2 Optikai átvitel

- optikai szálban felléphet ugrásszerű törésmutató változás (step-index fiber) vagy fokozatos törésmutató változás (graded-index fiber)
- mag/héj értékek **multimódusú** szálakban
    - 50/125 um
    - 62,5/126 um
    - 100/140 um
- monomódusú szál esetén 9-10/125 um

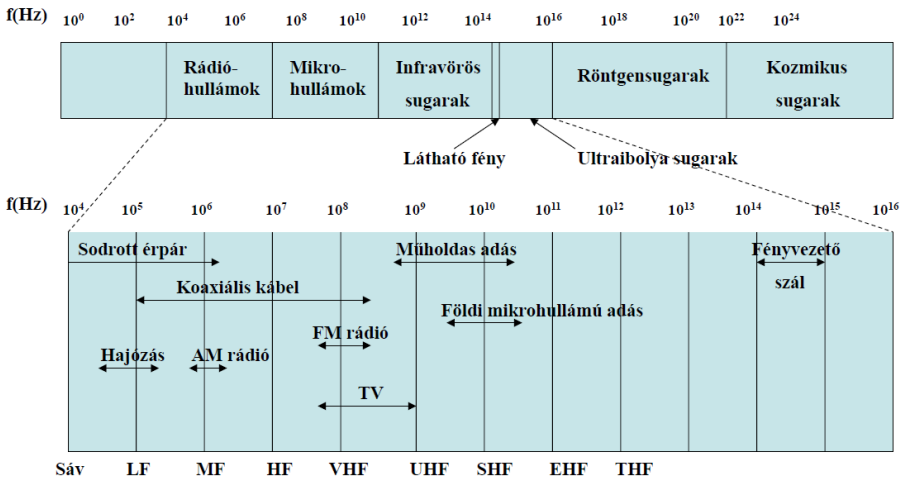


12. ábra Üvegszál és üvegszálas kábelek konstrukciója

- **Fiber to the Home (FTTH)**
    - optikai szál a központi elosztótól az otthonig
    - OLT: Optical Line Terminator
    - ONT: Optical Network Terminator
    - magas adatátviteli sebesség érhető el (tv, internet, telefon szolgáltatás)
- mono- vs. multimódusú szálak
    - \+ nincs módusdiszperzió
    - \- kábel olcsóbb, de adó-vevővel drágább
    - \- nagyobb optikai teljesítmény esetén szálégés
    - általában kis távolságokra multi-, nagyobbra monomódusú szál (gerinchálózat)
- optikai átvitel előnyei
    - \+ nagy távolságokra nagy sávszélesség
    - \+ nincs EMI, kisebb csillapítás, kevesebb ismétlőállomás
    - \+ megbízhatóbb, lehallgatása költséges, könnyen felfedezhető
    - \- diszperzió – eltérő sebességgel (kromatikus), eltérő úton (módus) terjedhetnek a nyalábok
        - felső korlát az átvihető sávszélességre és maximális távolságra

5.1.4 Rádiócsatornák

- elektromágneses hullámtartomány: ultrahang és látható fény között
- széles és nem kihasznált tartomány: 10 KHz – 1 THz
- szabadban hullámterjedés
- adó- és vevőantennák alakítják át és vissza a jeleket
- hullámhossz nagyságrendjébe eső antennák kellenek
- antennák fő jellemzője az **antennanyereség**
    - mekkora teljesítménysűrűséggel (térerővel) sugároz egy ugyanakkor teljesítménnyel táplált izotrop antennához képest dB-ben



13. ábra Elektromágneses hullámtartomány

- hullámterjedést befolyásoló jelenségek
  - szabadtéri csillapítás
  - visszaverődés (reflexió)
  - törés (refrakció)
  - elhajlás (diffrakció)
  - szóródás (scattering)
- rádiócsatorna nyitottságából adódóan
  - **zajok**
    - termikus, vevő bementén lévő
    - atmoszférikus, kozmikus, galaktikus
  - **zavarok**
    - ipari zajok, zavarok
    - más rádiórendszerekből származók
- a vett jel nagyságát alapvetően a szabadtéri csillapítás befolyásolja
- adó- és vevőantennák nyereségével javítható, mivel az antennák „irányítottak”
- a zaj döntően a vevőkészülék bemenetén jelentkező termikus zaj
- a rádiócsatorna nyitottsága miatt kritikus a frekvenciasávok felosztása a felhasználók között
- engedélyköteles és engedélyhez nem kötött sávok, kivéve
  - 2,4 GHz és 5,6 GHz – ISM sáv (Industrial, Scientific, Medical)
  - többszintű szabályozás (nemzetközi, regionális – EU, nemzeti – NMHH)
- nagyvárosi és rurális környezetben egyaránt problémát okozhat az árnyékolás, csillapítás, visszaverődés
- többutas terjedésnél a különböző utakon terjedő jelek időben eltolva érkeznek meg – gyengítik egymást
  - ezt hívjuk fadingnek (elhalkulás) RH rádiózásnál
  - kiküszöbölhető többszörös vétellel

6 2014.03.04 – Többszörös hozzáférés

- multiple access: nem csak hálózat, hanem közös átviteli közeg, amelyhez minden végpont hozzáfér
    - lehet vezetékes és rádiócsatorna
- multiplexelés ⇔ többszörös hozzáférés
    - közös csatornát használ mindkettő
    - több forrás van
    - multiplexelésnél mindegyik **egy helyen**, **fizikai** rétegben
    - multiple accessnél **szétszórtan**, **adatkapcsolati** rétegben (hívják még MAC- Medium Access Controllnak is)
- többszörös hozzáférés szerepe – takarékoskodás az átviteli közeggel, rugalmas elérés biztosítása
- ehhez szervezési módszerek kellenek és valamekkora erőforrás, amit ehhez felhasználunk (sebesség, idő, frekvencia)
- FDMA, TDMA, CDMA – megismert multiplexálások access megfelelői
- FDMA
    - ortogonális, de technikailag bonyolult
    - védősávok miatt kevésbé hatékony
    - valós idejű átvitelnél használják elsősorban
- TDMA
    - ortogonális
    - rugalmas megosztáshoz jó (különböző méretű részcsatornák)
- CDMA
    - nem ortogonális, de információelméleti értelemben ez a legjobb
- FDD – FDM – FDMA
    - FDD: duplex kialakítás, csatorna megosztása fel/le irányba
    - FDM: **fizikai** réteben több kisebb sávszélességű csatorna kialakítása
    - FDMA: **adatkapcsolati** rétegben felhasználók hozzáférése különböző helyekről
- **FDMA**
    - előnyei
        - nincsenek időzítési problémák, mint TDMA-nál (streamekhez jó)
        - nincs közel-távol probléma, mint CDMA-nál
        - fix sávon felül van lehetőség még osztani
    - hátrányai
        - magas teljesítményű szűrők hw oldalon
        - interferencia ← védősávok használata
- **TDMA**
    - előnyei
        - egy frekvenciacsatornát oszt meg több felhasználó között
        - nincs közel-távol probléma
        - dinamikusan skálázható (pl. WiMAX, Bluetooth)
    - hátrányai
        - interferencia a frekvencián belül
        - szinkronizációs problémák
        - kötött cellaméret
- közös csatornák
    - koaxiális kábel

- műholdas ismétlő állomás

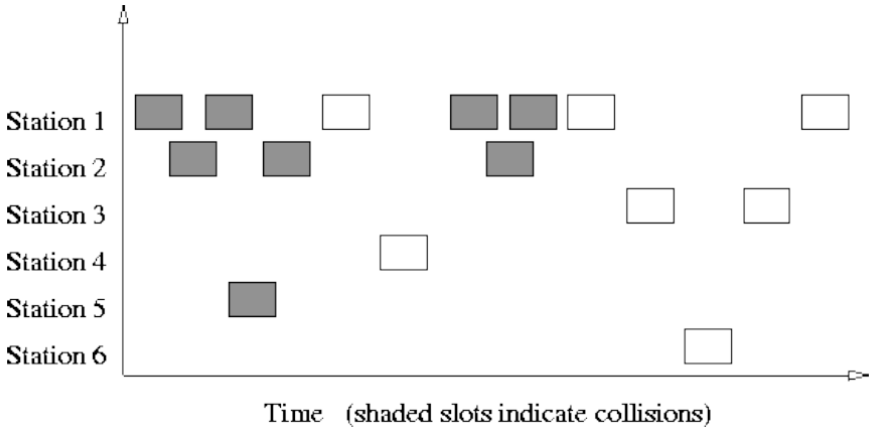
6.1 Hozzáférési módszerek teljesítőképességének jellemzése

- **kihasználtság** (througput): kiszolgált információ a fellépő igény függvényében
- **kiszolgálási késleltetés**: igény jelentkezése és kiszolgálás közti idő
- **igazságosság** (fairness):
  - egyformán részesülnek-e az átbocsátóképességből
  - teljesül-e a jogosultság szerinti kiosztás?
- **stabilitás**: reakció (túl)terhelésre, dinamikus forgalomváltozásra

6.2 ALOHA

- Norman Abramson fejlesztette ki
- első rádiós csomagkapcsolt hálózat
- maga a hálózat már nincs használatban, de az eljárást még mindig használják
- kiinduló ötlet: egyetem távoli részeinek összekötése – közös csatorna hatékony felhasználása
- csillag topológia szerint alakították ki, uplink-downlink struktúrában
  - kimenő csatorna: hub → kliensek; broadcast **csomagban címekkel**, amit csak a címzett fogadott
  - bejövő csatorna: kliens → hub; hub nyugtázta a sikeres vételt
    - ha nem érkezett meg a nyugta, **véletlenszerű idő után** újraadás
    - ütközések felderítése
- minden kliens egy frekvencián kommunikált a hubbal

6.2.1 Egyszerű aloha



- teljesen kötetlen hozzáférés
- mindenki akkor ad, amikor akar
- ha ütközés van, egyik adás sem sikeres
- ütközöttek **véletlenszerű idő múlva** ismét próbálkoznak
- nincs ellenőrzés csatorna foglaltságáról
- nem igényel szervezést
- ütközés = nyugta elmaradása
- nincs csatorna foglaltságáról információ adás előtt → **ütközés**
- véletlenszerű várakozás miatt nincs kezdeményezés → **rossz kihasználtság**
- ütközések után próbálkoznak ismét, visszatartás (backoff) nagyban befolyásolja a protokoll hatékonyságát és a csatorna kapacitását
- rossz kihasználtság
- instabilitás

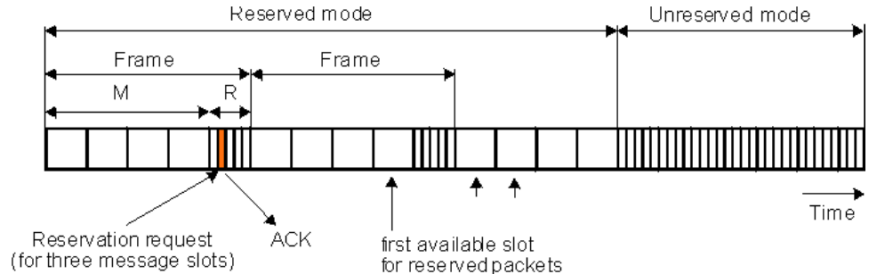
- nem korlátos késleltetés
- hosszú távon egyenlő fairness

6.3 Réselt (slotted) Aloha

- azonos hosszúságú csomagok szinkronjel határon
- csak egy időrésen belül lehet ütközés, nem nyúlik tovább
- jobb kihasználtság
- késleltetés szinte azonos
- instabil
- fairness azonos
- felhasználása:
    - katonai műholdas rendszerek
    - RFID
    - réselt-helyfoglaló aloha – GSM-GPRS

6.4 Helyfoglaló (reservarion) Aloha

- helyfoglalást (reservation) használunk a jellemzők javítására
- csatorna egy részén igénybejelentés – felhasználók „visszahallják”
- egységes döntés a felhasználók közt a használati jogosultságot illetően
- sokkal jobb késleltetés, mint a réselt Alohánál



6.5 Aloha alkalmazása

- növekvő felhasználószám és forgalom mellett rohamosan romlik a kihasználtsága
- burst-ös forgalomnál jobb kiszolgálást eredményez

6.6 Vivőérzékeléses többszörös hozzáférés

- Carrier Sense Multiple Access
- vezérlés a csatorna foglaltságának ellenőrzésére korlátozódik
- szabad csatorna – csatorna igénybevétele
- foglalt csatorna
    - nem figyeli folyamatosan, majd próbálkozik – **nemperzisztens**
    - várakozás a felszabadulásra, felszabaduláskor rögtön igénybe vesszük – **1-perzisztens**
    - p valószínűséggel adunk keretet időrés elején, ha szabad – **p-perzisztens**
- egyszerű megvalósítás – csak a foglaltságot nézzük
- nagy kihasználtság
- késleltetés-érzékeny
- instabilitás túlterheltség esetén
- megvalósítható a fair kiszolgálás

6.6.1 CSMA/CD (CSMA with Collision Detection)

- visszacsatolás a csatornából adás közben – **ütközés**

- észlelt ütközés esetén leállás
- vezetékes csatornán lehetséges az ütközésdetekció
- rádiócsatornán nem tudja detektálni az épp használó az ütközést
- vivőérzékelés – csak akkor ad, ha szabadnak érzékeli a csatornát
- ütközést érzékel – küld egy foglaltsági jelzést, hogy ne legyen több ütközés
- exponenciális backoff után újra ad
- hatékonyság javítható
    - ütközésdetektálás után rögtön leáll az adás
    - csak exponenciális backoff után van újraadás
    - csak fél-duplex esetén van értelme
- RTT (Round Trip Time) – 2×D (D: terjedési idő)
- legrosszabb esetben az ütközés detektálása 2T ideig is eltarthat

6.6.2 CSMA/CA (Collision Avoidance)

- vivőérzékeléses **ütközéselkerülés** – ütközésveszélyes helyzetekre készülünk
- ha szabad a csatorna
    - küld egy jelzést a többi állomásnak, hogy adni készül
    - vagy vár egy további időig és ha utána szabad, akkor ad
- ha foglalt a csatorna
    - exponenciális backoff ideig nem ad
- kevésbé mohó adókban használják
- WLAN-oknál is használják, ahol nincs lehetőség CD-re
- **rejtett terminál** – az árnyékolt felhasználó nem hallott adását akaratlanul is megzavarhatjuk
- **exponált terminál** – túl közeli szomszédok adását is érzékeli és nem meri használni a csatornát
- **RTS-CTS**
    - egy igénybejelentés (RTS) és egy nyugta (CTS) után van adatátvitel
    - átvitel hosszát tartalmazza a két üzenet

6.7 Versenymentes MAC protokollok

- CSMA/CD esetén nincs ütközés, amikor valaki lefoglalja magának a csatornát
- viszont lehet ütközés versenyhelyzet idején → rontja a hatékonyságot

6.7.1 Bit-térkép módszer

- n időrésen j. terminál a j. időrészben jelzi, hogy adni szeretne
- többi terminál nyugtázása után lehet adni, nincs ütközés
- alacsony forgalom esetén nem hatékony, mindig ki kell várni a bittérkép végét, pazarlás

6.7.2 Bináris visszaszámlálás

- azon terminál, amelyik keretet szeretne adni, beírja a címét a fejlécbe binárisan
- az egyes adók, ha magasabb értékű címet érzékelnek, lemondanak az adásról
- sok terminál esetén sérül a fairness elv

6.8 Verseny és versenymentesség, korlátozott verseny

- alacsony forgalom esetén → verseny javasolt, alacsony késleltetés
- növekvő forgalom esetén → egyre kevésbé javasolt a verseny, erőforrásigényes
- ötvözni kell a verseny-alapú és versenymentes protokollokat
    - protokollok korlátozott versennyel
- **cél:** kis késleltetés alacsony terhelésnél, nagy átvitel nagy terhelésnél

6.8.1 Protokollok korlátozott versennyel

- terminálok csoportokba sorolódnak (nem feltétlenül diszjunkt)
- csak a j. csoport tagjai versenyezhetnek a j. időrésért
- ütközés vagy nincs küldés → j+1. csoport j+1. időrésért stb.

6.8.1.1 Adaptív fa protokoll

- mindenki versenyezhet a csatornáért
- ütközés esetén két egyforma csoportra osztódnak a terminálok – csak az egyik versenyez a következő időrésért
- újabb ütközés esetén újabb csoportfelezés
- ha valaki megszerzi közülük – következő időrésért következő csoport versenyez

6.9 Központilag vezérelt többszörös hozzáférési módszerek

- **lekérdezés** (polling): vezérlő kérdése jelenti a csatornahasználati jogot
- **csoportos lekérdezés** (probing): felhasználók nagyobb csoportját kérdezi a vezérlő
- **foglalás** (reservation): a vezérlő az érkező igények alapján csatornahasználati jogot jelöl ki
    - igények gyűjtése **egyedileg** vagy **versenyben**

6.9.1 Lekérdezés (roll-call polling)

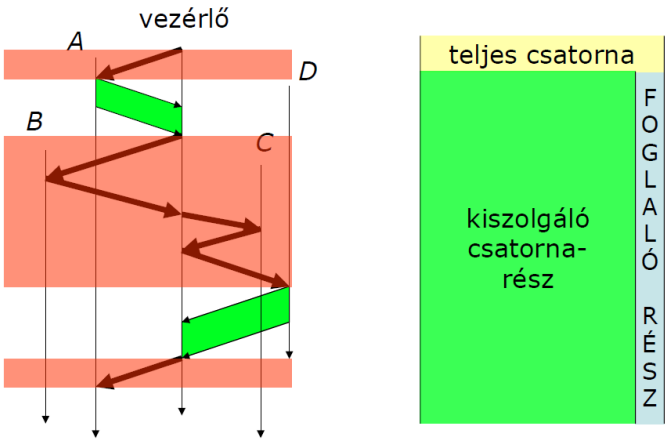
- körbekérdezés – akinek van csomagja, elküldi
- csomagtovábbítás alatt információ-átvitelre használt a csatorna, egyébként hozzáférés szervezésére
- kihasználtság: két időszak arányában
- kiszolg. késleltetés: körbejárási idő fele
- stabil és fair módszer

6.9.2 Csoportos lekérdezés (probing)

- csoportos lekérdezés, ütközés esetén részekre bontás
- javuló kihasználtság
- kiszolg. késleltetés csökkenése
- stabil és fair módszer

6.9.3 Helyfoglalás (reservation)

- rossz RTT → rossz hatékonyság
- **foglalási** és **átviteli** részre érdemes osztani a csatornát
- helyfoglalási részben dedikált csatornarész igénybejelentéshez, verseny alapú eléréssel
- vezérlő a beérkezett igények alapján ad engedélyt a felhasználóknak



6.9.4 Polling elosztott vezérléssel (token passing)

- speciális token birtoklása → csatorna használati joga
- megfelelő szabályok mellett nagyon rugalmas kiszolgálás
- kifinomult működés kell állomások között
- jó kihasználtság és fair kiszolgálás

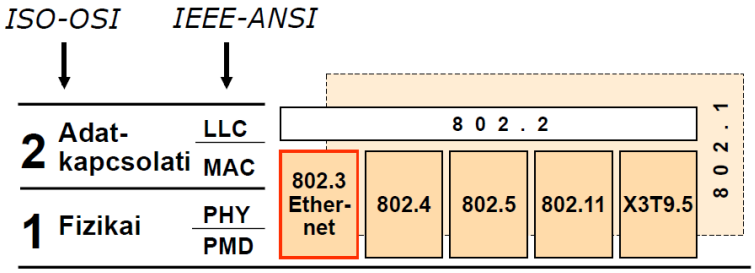
6.9.5 Többszörös hozzáférés előnyei és hátrányai

- előny (függ a használt közegtől)
    - gazdaságosabb (kevesebb vezeték)
    - technikai (jobb teljesítőképesség)
    - rádiócsatornánál szinte nélkülözhetetlen
- hátrány
    - bonyolultabb algoritmusok
    - illetéktelen hozzáférés kezelendő

7 2014.03.06 – LAN

- **WAN** – Wide Area Network – tetszőleges kiterjedésű, akár globális hálózat
- **MAN** – Metropolitan Area Network – városi/nagyvárosi, pár 10 km-es
- **LAN** – Local Area Network – vállalaton, intézményen belüli, lokális, max. pár km-es
- **PAN** – Personal Area Network – személyi hálózat
- **BAN** – Body Area Network – testközeli hálózat

7.1 A lokális hálózatok architektúrája



- **LLC : Logical Link Control – Logikai adatkapcsolati**
- **MAC: Medium Access Control – Közeghozzáférési**
- **PHY: Physical – Fizikai**
- **PMD: Physical Medium Dependent – Fizikaiközeg-függő**
- ***802.1: közös funkciók valamennyi LAN-ra és MAN-ra pl. együttműködés, biztonság (interworking, security)***

Alrétegek (sublayer)

7.1.1 IEEE 802.2 LLC – Logical Link Control

- feladatai – 3. réteg számára megbízható átvitel
  - forgalomszabályozás
  - hibadetektálás, -javítás
- szolgáltatások
  - nyugtázatlan, összeköttetésmentes (datagram)
  - nyugtázott, összeköttetésmentes (datagram)
  - nyugtázott, összeköttetés alapú
- nem mindig van szükség LLC-re, pl. **IP nem igényli**
- protokoll-overhead: MAC-kerethez LLC fejléc adódik

7.1.2 802.3 – az Ethernet

- logikailag (és eredetileg fizikailag is) **busz topológiára épül**
- Ethernetnél CSMA/CD használata
  - előny: mindenki hallott mindenkit
  - hátrány: minimum csomagméret és útvonalhossz összerendelve
- Ethernet jelölésénél: A|B|C
  - A: adatsebesség – 1 = 1 Mbit/s, 10 = 10 Mbit/s…
  - B: alapsávi/szélessávú – Base/Broadband
  - C: átviteli közeg, szegmenshossz
    - T = twisted pair, FX/LX/SX = optikai, T4 = 4 pair twisted pair
    - 2 = 185 m, 5= 500 m
  - pl. 10 Base 5, 1000 Base T

7.1.2.1 10 Base 5 (Vastag Ethernet)

- első kábelezési megoldás, koaxiálisra épül

- állomások vámpír csatlakozóval csatlakoztak
- CSMA/CD
- max. 100 adó, 2,5 méterenként

7.1.2.2 10 Base 2

- BNC csatlakozók, T elosztók, vékony koax kábel
- max. 30 adó/szegmens
- sok csatlakozó miatt sérülékeny – ha egy leáll, az egész kommunikáció leáll
- olcsóbb, mint a hubos megoldás, de nehezebb a hibakeresés
- kiváltotta a 10 Base T

7.1.2.3 Ethernet jelismétlő (repeater)

- több szegmens összekötése
- 5 szegmens – 4 ismétlő – 3 szegmensen terminálok
- ütközés esetén minden porton jam jelzés
- kábel meghibásodás – csak az adott szakaszon, rosszul csatlakoztatott terminál blokkolása

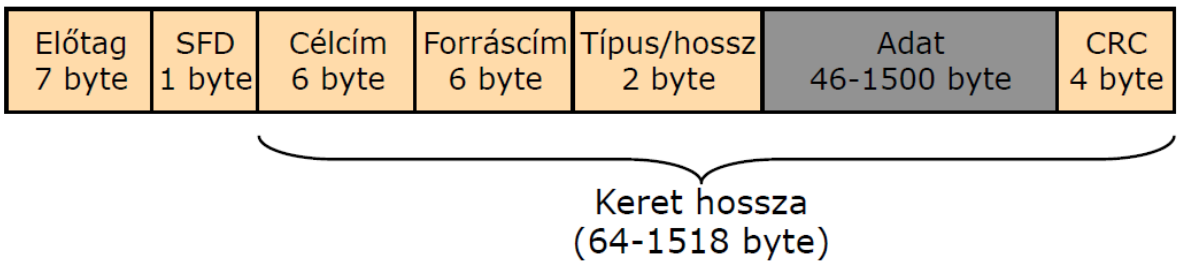
7.1.2.4 10 Base T

- sokkapus ismétlő/hub
- valamilyen porton észlelt jelt továbbadásra kerül portokon
- csavart érpárok (2 pár), RJ45 csatlakozó
- ütközés esetén jam jel minden portra

7.1.2.5 Hubok alkalmazása

- half-duplex: CSMA/CD-re
  - tudja kezelni az ütközéseket és jam jelzést ad, letilt portokat
  - hátránya
    - minden keretet minden portra kiad
    - korlátozott hubszámmal működik
    - egy sebességen és egy keretezéssel működik az összes szegmens (kivéve dual-speed hub)
  - megoldás → bridge és switch adatkapcsolati rétegben
- régi hálózati szegmensek illesztése mai hálózatokhoz
- protokoll analizátorok illesztése
- STP (Spanning Tree Protocol) nélküli switch-ek illesztéséhez

7.1.3 Az Ethernet keretek felépítése



- előtag (preamble) – 7 byte
- SFD (Smart Frame Delimiter) – keret kezedete – 1 byte
- célcím – 6 byte
- forráscím – 6 byte
- típus/hossz – 2 byte – adatmező hossza vagy ítpusa (1500 alatt és 1536 felett)

- adat – min. 46, max. 1500 byte.
- CRC – 4 byte
- résidő számítása – T=2L/C
    - T: résidő
    - L: szegmens hossza
    - C: jelterjedési idő
- **minimális kerethossz – 64 byte**

7.1.4 Medium Access Control – CSMA/CD

- az állomás figyeli a **csatornát**
- ha érzékel adást, elkezdi küldeni a keretet
- ha kettő vagy több állomás ad, abbahagyja az adást, ütközésérzékelés
- valamekkora véletlenszerű késleltetés után újra adja
- **adás előtt** érzékeljük a vivőt – carrier sensing
- **adás alatt** érzékeljük, hogy más is ad – collision detection

7.2 Nagysebességű Ethernet szabványok

- Fast Ethernet (802.3u), Gigabit Ethernet (802.3z), 10Gb Ethernet (802.3ae)

<table>
<tr><th>Jellemzők</th><th>10/100 Mbit/s</th><th>1 Gbit/s</th><th>10 Gbit/s</th></tr>
<tr><td>Fizikai közeg</td><td>▪ UTP
▪ üvegsz.</td><td>▪ koax
▪ UTP
▪ üveg</td><td>▪ üveg
▪ (rézv.)</td></tr>
<tr><td>Résidő (slot time) [byte]</td><td>64</td><td>512</td><td rowspan="3">NINCS CSMA/CD MAC PROT.</td></tr>
<tr><td>Küldési próbálkozás</td><td colspan="2">16</td></tr>
<tr><td>Visszalépési algoritmus korlátja</td><td colspan="2">10</td></tr>
<tr><td>Minimális keretméret [byte]</td><td colspan="3">64</td></tr>
<tr><td>Maximális keretméret [byte]</td><td colspan="3">1518</td></tr>
</table>

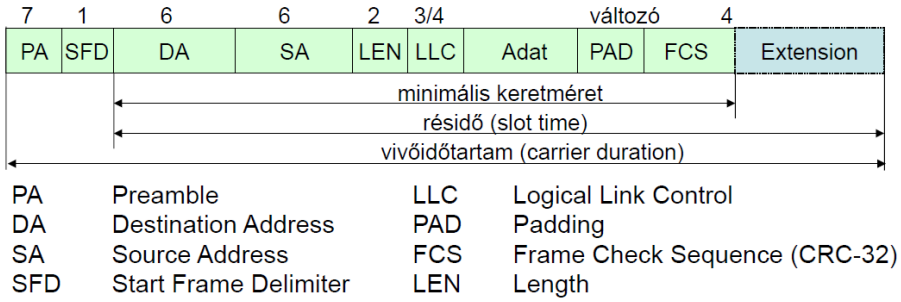
7.2.1 Fast Ethernet – 100 Base X

- különböző médiumokra tervezték
    - Cat-5 UTP, Cat-5 STP, optika
- más-más fizikai médiumfüggő alréteg
- 4B5B bitkódolás van adaptálva rá
- bridge-ek megjelenése különböző sebességű linkek áthidalásához

7.2.2 Gigabit Ethernet

- félduplex üzemmódban működik
    - mint a klasszikus Ethernet, de rövidebb keretidőkkel
    - keretfűzés vagy **vivő kiterjesztéssel (bitek hozzáadása a kerethez)**

- **ma már full duplex switchekkel**



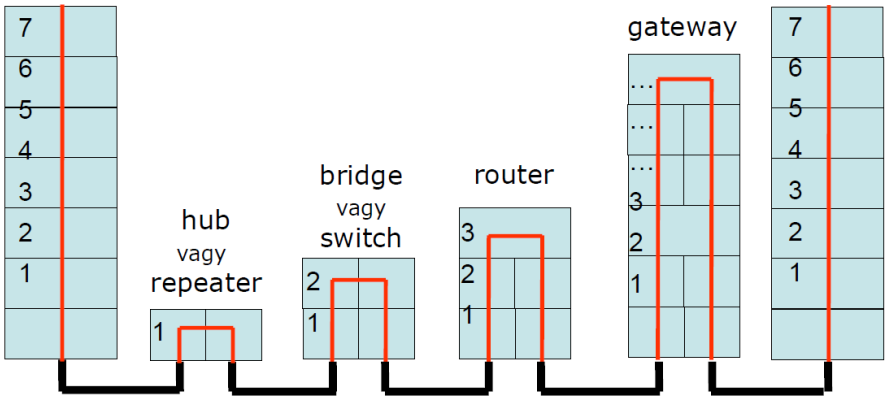
14. ábra Gigabit Ethernet keret felépítése

7.2.3 10 Gb Ethernet

- legújabb tag az Ethernet-családban
- további tízszeres növekedés a Gigabit Ethernethez képest
- kizárólag **full duplex switchekkel**, nincs CSMA/CD
- főként üvegszálon, de van rézvezetékes változata is (min. Cat-6 vagy Cat-7)

7.3 LAN-ok összekapcsolása

- állomások száma, típusa és távolsága korlátozó tényező
- átjátszóval átléphetők a korlátok



15. ábra Különböző képességű átjátszók

Angol név	Magyar név	OSImax. réteg	Tipikus port-szám	Funkcionalitás
repeater	jelismétlő	L1	2	jelerősítés, -továbbítás
hub	többportos jelismétlő	L1	4-16	jelerősítés, -továbbítás minden porton; több eszköz összekapcsolása
bridge	híd	L2	2-8	nem ütköző szegmensek összeköttetése; továbbítás csak a szükséges porton; átviteli közegek közötti konverzió újrakeretezéssel
switch	kapcsoló	L2	4-32	nem ütköző szegmensek összeköttetése; továbbítás csak a szükséges porton, azonos közegen, újrakeretezés nélkül
router	útválasztó	L3	2-10	útválasztás L3 címek alapján
gateway	átjáró	>L3	2-4	protokollkonverzió, -együttműködés

16. ábra Átjátszó eszközök ismertetői

7.3.1 Bridge-ek és switch-ek

- adatkapcsolati réteg
- többféle sebességet képesek kezelni
- bridge: tipikusan LAN-szegmenseket köt össze
- switch: LAN-szegmensek és egyedi állomások összeköthetők
- bridge össze tud különböző LAN-okat kötni
- store and forward elven működnek

7.3.1.1 Switch

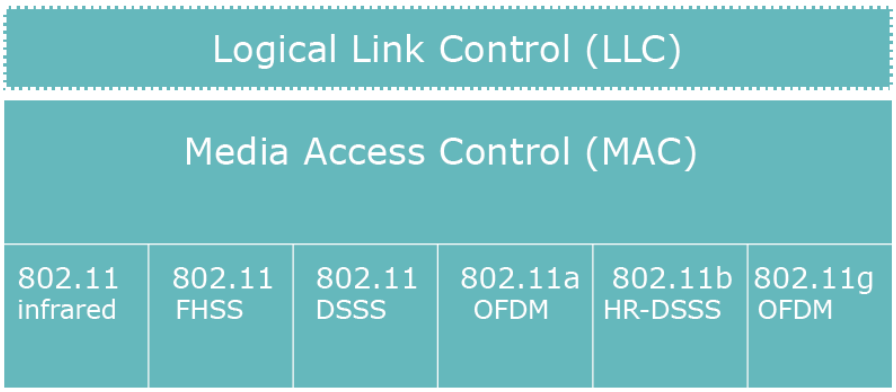
- adatkapcsolati réteg
- tárolja és továbbítja az Ethernet kereteket
- keretfejrész alapján MAC cím szerint szelektíven továbbít
- nem full-duplex esetben CSMA/CD-t használ
- transzparens – végpontok számára látható
- nem kell konfigurálni, öntanuló
- hostok számára dedikált portok a switchen
- full duplex esetben nincs ütközés, minden link egy saját ütközési tartomány
- puffereli a csomagokat
- öntanuláshoz **kapcsolótábla** (switching table)
    - MAC, Interface, Timestamp
    - régi bejegyzések eldobása
    - megtanulja melyik végpontot melyik interfészen lehet elérni
    - amikor vesz egy keretet, megnézi, hogy a MAC cél-címet ismeri-e
        - ha igen, megnézi, hogy onnan jött-e a keret és eldobja
        - ha igen, de nem onnan jött a keret, továbbítja
        - egyébként pedig elárasztja a hálózatot, amíg vissza nem csorog a szükséges MAC cím, amit beír a táblájába, ezzel tanítva magát

8 2014.03.11 – WLAN

- pár száz méteres hatótáv
- 1-2 Mbit/s-tól 100 Mbit/s-ig
- ISM sávban – engedélyköteles
    - 2,4 GHz illetve 5,8 GHz
- épületen belüli LAN-ok részeként, közeli épületek közti kapcsolat, nyilvános internet-elérés
- asztali/hordozható eszközökbe WLAN kártya, elérés biztosításához AP (Access Point)
- 802.11abgn – legelterjedtebb, 802.11ac – 2014. januárjában szabványosított, 1 Gb/s
- 802.11e – QoS, 802.11i – adatbiztonság, titkosítás

8.1 2,4 GHz-es csatorna

- pl. 2,4000 – 2,4835 GHz sávban 13 db 22 MHz-es csatorna, egymástól 5 MHz-re a középfrekvencia
- a frekvencia közepétől 30 dB csillapítás +/- 11 MHz-re (spektrális maszk)
- átlapolás nélkül használható: 1, 6, 11. csatorna
- interferencia más hullámot kibocsátó eszközökkel, közel-távol probléma kezelendő
- 5 GHz sávban 23 nem átlapoló csatorna



17. ábra 802.11 szabványok különböző rétegeknek megfelelően

8.2 DSSS

- korrelációs vevő: detekcióhoz használjuk, kódolásnál alkalmazott álvéletlen jelsorozattal szinkronban
- zajjal zavarokkal nincs korreláció
- adott chip-időn belül maximumot ad, alacsony értékeket más helyen
- chip-frekvencia megválasztása
    - hosszú kód: jó zavarvédettség
    - rövid kód: kisebb sávszélesség-igény
- 802.11 – 11 bites szórás → mérsékelt zavarvédettség, de jó sávszélesség-gazdálkodás
- minimum 10 az ISM sávban

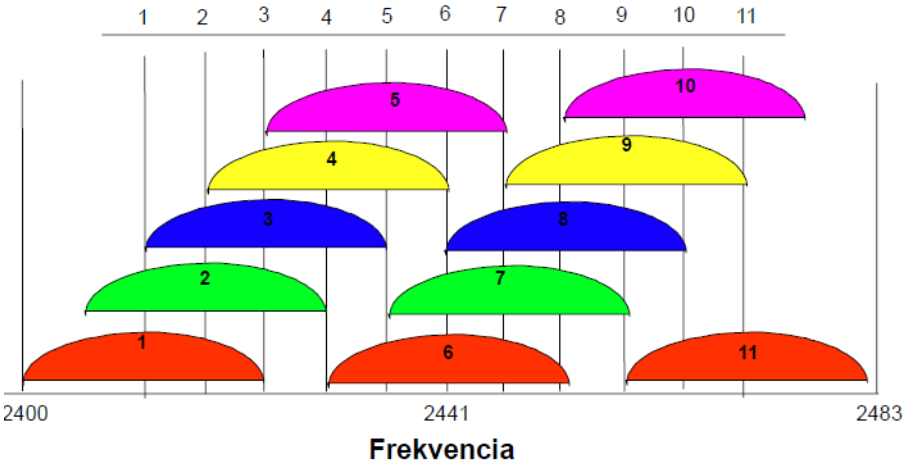
8.3 FHSS

- Frequency Hopping Spread Spectrum
- több frekvenciát használ
- első bit vagy bitcsoport pl. 2,44 GHz-en, második 2,41 GHz-en, harmadik 2,40 GHz-en stb.
- bitsebesség nagyobb vagy kisebb a hopping-sebességnél
- 2,4 GHz sávban minimum 75 frekvencia, max. 400 ms egy frekvencián

- „jó” frekvenciák használata – adaptív működés

8.4 OHDM

- Ortogonal Frequency Division Multiplexing
- sok részsávra osztja a frekvenciát
- párhuzamosított jelfolyam továbbítása
- BPSK/QPSK vagy 16QAM/64QAM
- sávok nem diszjunktak, össze vannak tolva, egymásba átlógnak
- szétválaszthatóak, mert eleget tesznek a Nyquist-elvnek

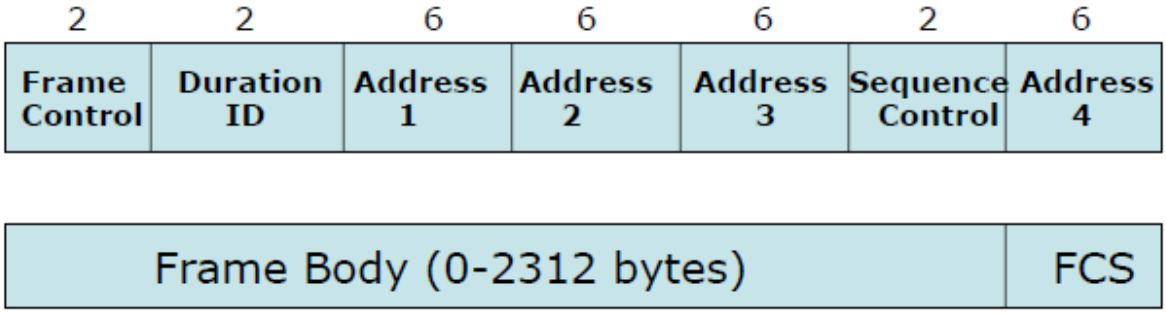


18. ábra Csatornakiosztás

- jól védhető a keskenysávú interferencia és szelektív fading ellen
- magas spektrum-hatékonyság
- érzékeny a Doppler-effektusra és a frekvencia szinkronizációra

8.5 Működési módok, topológia, egyebek

- **ad-hoc** (peer-to-peer működés) vagy **infrastruktúra mód** (access point → kliens irány csak)
- **BSS:** Basic Service Set – egy cella
- **ESS:** Enterprise Service Set – több cella
- **DS:** Distribution System – gerinc
- WLAN ⇔ Bridge ⇔ LAN



19. ábra Keretformátum

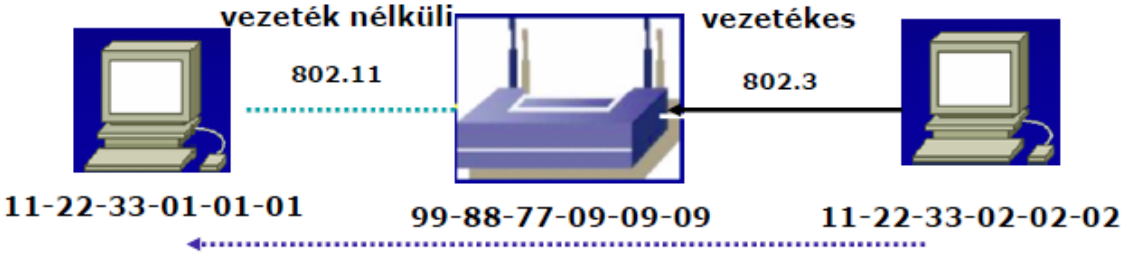
- Frame Control felépítése:

- 2 bit protokoll konverzió
- 2 bit típus
  - ACK, RTS, CTS
  - időbélyeg, SSID, FHSS/DSSS paraméterek
- ToDS-FromDS – ha BSS-en belüli, akkor 00

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	N/A
0	1	DA	Sending AP	SA	N/A
1	0	Receiving AP	SA	DA	N/A
1	1	Receiving AP	Sending AP	DA	SA

DS: Distribution System
BSSID: Basic Service Set ID

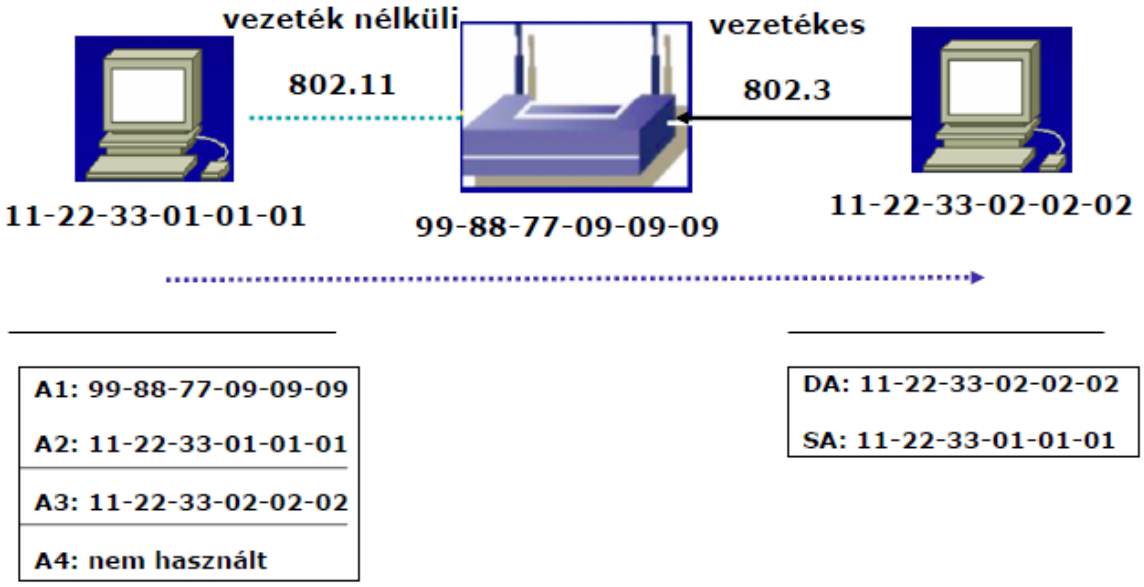
DA: Destination Address (Célcím)
SA: Source Address (Forráscím)



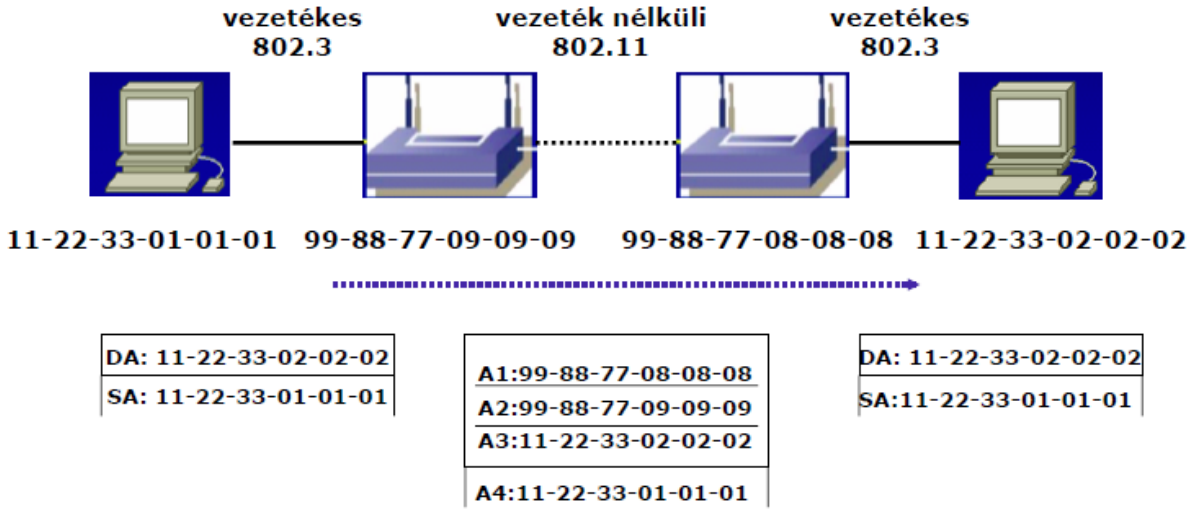
A1: 11-22-33-01-01-01
A2: 99-88-77-09-09-09
A3: 11-22-33-02-02-02
A4: nem használt

DA: 11-22-33-01-01-01
SA: 11-22-33-02-02-02

20. ábra "01" eset



21. ábra "10" eset



22. ábra "11" eset

8.5.1 802.11 MAC

- nem CSMA/CD – CS igen, de nem CD
- vezeték nélküli LAN-okban nem lehet ütközést detektálni

8.5.1.1 DCF – Distributed Coordination Function

- MAC alsó rétege
- ütközés-elkerülés
- különböző IFS-ek (Inter Frame Space) – Short (rövid IFS vezérlőüzenetek), PCF – PIFS, DCF – DIFS (adatkeretekhez)
- **SIFS**:
    - magas prioritású adások esetén ők férnek hozzá először
    - SIFS után ACK, CTS
    - konstans értékű (10/16 us)

    - reduced IFS (RIFS), 802.11n szabványban, még rövidebb
- **PIFS:**
    - SIFS + Résidő
- **DIFS:**
    - SIFS + 2×Résidő
- DCF algoritmus
    - szabad csatorna → DIFS ideig vár az állomás, ha szabad marad, ad
    - ha foglalt (az elején vagy azzá válik), tovább figyeli
    - ha szabaddá válik, DIFS + véletlen késleltetés ideig nem ad, majd ad
    - sikeres adás esetén vevőtől ACK-t vár
- exponenciális backoff algoritmus
    - INT(CW*RND())*Slot_Time
        - CW – Contention Window, kezdetben 31, majd 63, 127, stb.
        - RND() – véletlenszám generátor 0 és 1 között
        - Slot_Time: résidő (20 us / 50 us)
- rejtett állomás problémája itt is fennáll
    - RTS/CTS Handshaking
        - RTS – Request to Send
        - CTS – Clear to Send
        - ACK – Acknowledgement
        - opcionális megoldás
        - előnyös használni, ha nagy a verseny, a rejtett állomás probléma fennállhat
        - jelentős késleltetés növekedést eredményezhet, kisebb adatsebesség
- NAV (Network Allocation Vector)
    - minden RTS keret tartalmazza a csatorna kívánt foglaltságának hosszát
    - NAV: számláló a többi állomásnál, akik RTS/CTS hatására elindítják azt, utána nézik meg, szabad-e a csatorna
    - megoldás a rejtett állomás problémájára
- foglalt közeg: fizikailag vagy virtuálisan lehet foglalt
- exponált állomás problémájt az RTS/CTS nem oldja meg a gyakorlatban
- nincs QoS, prioritások
- sok terminál esetén sok ütközés

8.5.1.2 PCF – Point Coordination Function

- opcionális, DCF „felett”, azzal együttesen valósítják meg
- egyetlen AP vezérli a hozzáférést
- AP által küldött jelzőüzenet hatására állomások beszüntetik a DCF működést
- AP lekérdezi sorban az állomásokat – garantált max. késleltetés
- egy állomás csak akkor ad, ha kérdezik
- prioritás adható állomásokhoz → időérzékeny alkalmazások
- van QoS, de nem elterjedt

8.5.2 QoS biztosítása WLAN-on

- HCF (Hybrid Coordination Function)
    - kétféle MAC-módszer, forgalmi osztályok
        - email – alacsony, Voice over WLAN – magas
    - EDCA (Enhanced Distributed Channel Access)
        - prioritásos kevesebb ideig vár
        - kisebb CW és IFS

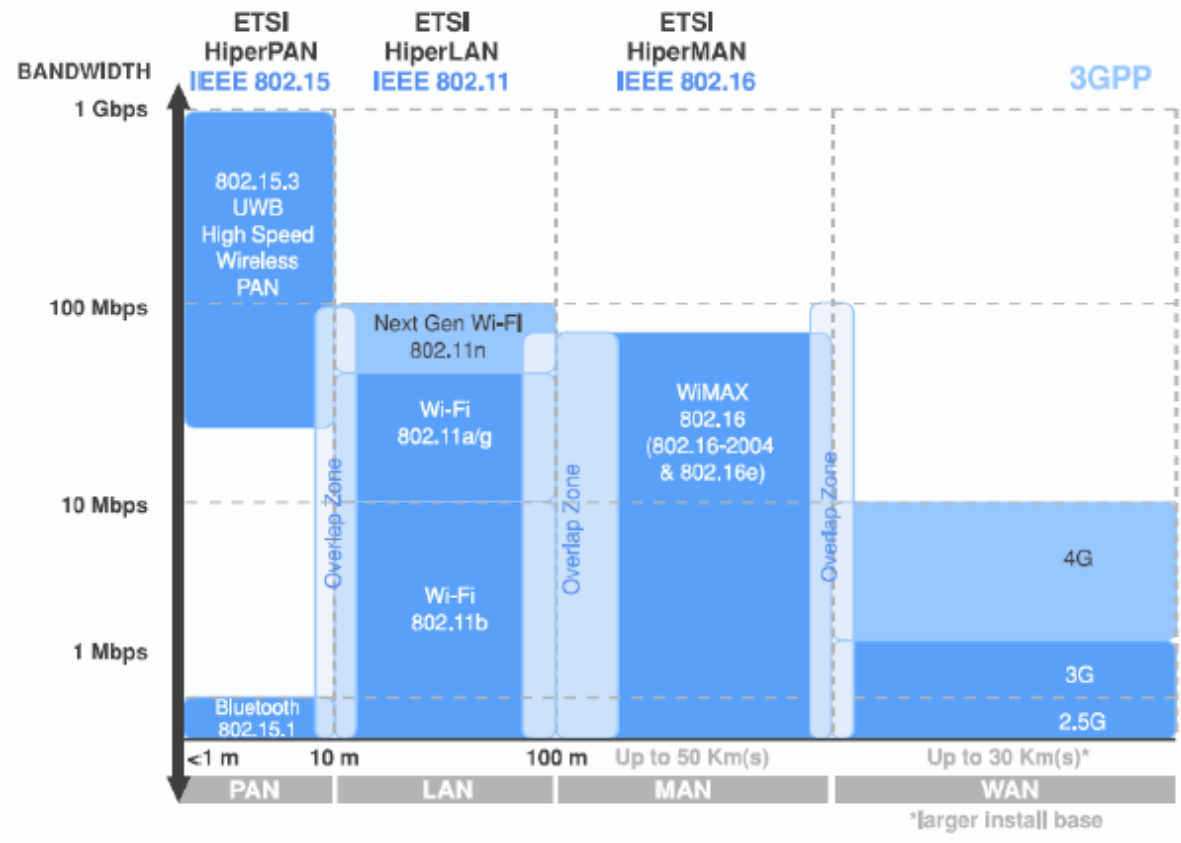
        - hang és video számára versenymentes idő
    - HCCA (HCF Controlled Channel Access)
        - AP bármikor jelezhet versenymentes periódust, egyébként EDCA verseny
        - forgalomosztály-ütemezés is
        - magasabb prioritású osztály előrevétele
        - legbonyolultabb, ritkán valósítják meg

8.5.3 WLAN biztonság

- WEP, WPA, **WPA2**

9 2014.03.13 – BWA

	PAN	LAN	MAN	WAN
Szabványok	Bluetooth	802.11a,b,g Hiperlan2	802.16, MMDS, LMDS	GSM, GPRS, UMTS
Sebesség	< 1 Mbps	2 – 54+ MBPS	22+ Mbps	10-384 kps
Távolság	Rövid	Közepes	Közepes-hosszú	Hosszú
Alkalmazások	Peer-to-peer és eszközök közötti	Magán és vállalati hálózatok	Gyors, „last mile" elérés	PDA, mobiltelefonok, cellás elérés



23. ábra Szélessávú vezetéknélküli hozzáférések összehasonlítása

9.1 Bluetooth

- IEEE 802.15.1 specifikáció az alapja
- olcsó eszközök
- rövidtávú kommunikáció
- max. 1 Mbit/s
- vezeték nélküli csatlakozások széles köre
- iparban is használt (kódolvasók, lokalizáció, szenzorhálózatok stb.)
- 2,4 GHz-es ISM sávban működik
- FHSS-t használ (1600-szor másodpercenként)
- TDD, 625 us-os időrések, adás-vétel egymás után
- master-slave felváltva → párosban master, páratlan slave ad, master órája alapján
- alap adatráta (BDR) – 1 Mbit/s
- megnövelt adatráta (EDR) – 2-3 Mbit/s
- kettő kombinációja (BR/EDR)
- **piconet:**
    - maximum 8 aktív állomás egyszerre
    - 1 master, többi slave
    - scatternet – piconetek hálózata, ad-hoc alapon
    - van node, ami >1 piconethez csatlakozik
    - slave-ek szinkronizáció elején a master adatai alapján órát egyeztetnek
- pairing, szinkron (pl. hang) és aszinkron átvitel (pl. adat)
- aktuális verzió – Bluetooth 4.0 (alacsony fogyasztású, orvosi és sportfelügyelet)

9.2 Nagysebességű PAN-ok

- médiakommunikáció
- új PHY specifikáció 55 MHz-ig
- egyik lehetőség az UWB (Ultra Wide Band)
- Shannon-egyenlet: **C = B*log(1+SNR)**
- Sávszélesség növelésével az adatátviteli sebesség lineárisan növekszik, míg a jel-zaj viszony növelésével csak logaritmikusan → növeljünk sávszélességet
- **UWB**
    - nagy sávszélesség
    - nincs vivő, nanoszekundumos, széles spektrumú impulzusok, kis teljesítmény
    - FCC és ETSI 3,1…10,6 GHz sávban engedélyezi
    - egyelőre nem terjedt el
    - UWB modulációk:
        - time- vagy shape-based
    - Digitális Otthon, vezeték nélküli USB, eddig főleg katonai célokra használták
    - WLAN sávval átlapolódik → rontja a jel/zaj viszonyt kis távolságon

9.3 WiMAX

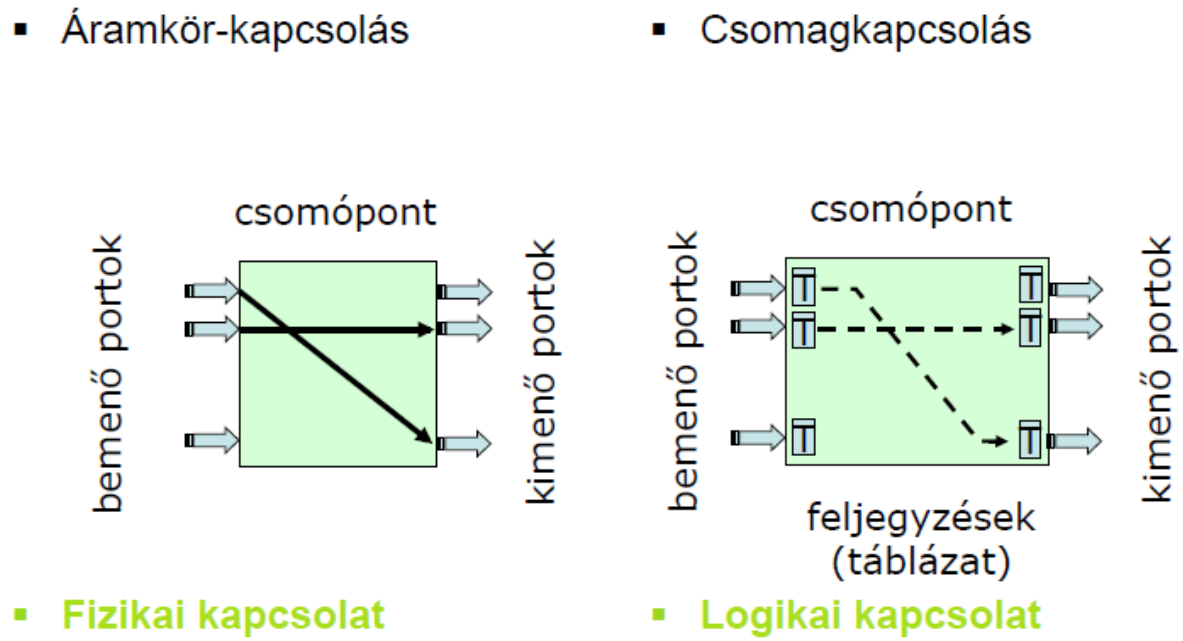
- Worldwide Interoperability for Microwave Access
- sok területen: SoHo, Enterprise, Point-to-Multipoint, Hotspots…
- közvetlen rálátás esetén (LOS – Line of Sight) (nincs takarás a Fresnel-zónában) és takarás esetén (NLOS – No…) is használható (több utas, speciálisan reflektált hullámok)
- Fresnel-zóna sugara: $\boldsymbol{r = 17,32 * \sqrt{\frac{D}{4f}}}$
    - D: távolság km-ben, f: frekvencia

- 802.16d és 802.16e – Fix és mobil WiMAX
- többvivős multiplexálást és modulációt használ
- alvivőkre osztja a rendelkezésre álló sávszélt
- soros-párhuzamos átalakítást után alvivőkön továbbítja az adatokat alacsonyabb szimbólumsebességű modulációval
- WLAN-nal ellentétben **van ütemező algoritmus**
    - egyszer kell versenyezni, utána minden állomásnak a bázisállomás oszt időrést
- QoS biztosítható
- QoS nem biztosítható 5 GHz felett – egyelőre csak katonai célokra
- biztosítja, hogy új adatfolyam ne rontsa a meglévőket

10 2014.03.17 – Kapcsolás, címzés, jelzés

10.1 Kapcsolás

- két nem szomszédos csomópont közti kapcsolat megteremtésének módja
- sokféle kapcsolás létezik (áramkör-, üzenet-, csomag-, …)



- léteznek összeköttetés nélküli hálózatok, ahol az adatátvitel előzetes összeköttetés létrehozása nélkül történik

10.1.1 Áramkörkapcsolás/vonalkapcsolás

- **fizikai** kapcsolat a küldő és a fogadó között
- **fel kell építeni** és az összeköttetés végén **le kell bontani**
- dedikált fizikai útvonalon halad minden információ
- valós idejű átvitelre kiváló
- csomópontok nem tárolják az adatokat az átvitel során
- csak összeköttetés felépítésénél lehet torlódás
- **kapcsolóelemből** és **kapcsolóvezérlőből** áll a kapcsolószerkezet
- blokkoló, blokkolásmentes és többfokú kapcsológépek is léteznek
    - belső blokkolás: szabad az kimenet, de nincs fizikai út
    - külső blokkolás: két bemenet ugyanazt a kimenetet akarja használni
- blokkolással hatékonyság növelhető
- időosztásos kapcsológépek – TDM jelek beírása memóriába, kiolvasás eltérő sorrendben
- **fix sebességen, sorrendhelyesen kell átvinni adatot (pl. telefonvonal)**

10.1.2 Üzenetkapcsolás

- egész üzenet egységes csomagként egyik csomóponttól másikig
- store-and-forward elv
- üzenetekben címrész, csomópontokban átmeneti tárolás
- sorbanállási késleltetés adódhat
- javul a csatorna kihasználtsága
- prioritás rendelhető az egyes üzenetekhez
- támogatja a broadcastot

- késleltetések miatt valós idejű átvitelhez nem jó (hang, videó)
- kevésbé késleltetés-érzékeny adattovábbításra jó

10.1.3 Csomagkapcsolás

- csomagokra tördeli a küldő a küldendő üzenetet
- csomagban: küldő azonosítója/címe, címzett azonosítója/címe, csomag helye az üzenetben
- egymás után küldi a csomagokat a küldő
- közbeeső csomópontok akár egymástól különböző útvonalakon továbbítják a csomagokat a bennük lévő cím alapján
- eredetitől eltérő sorrendben is érkezhet a címzetthez
- címzett a csomagban lévő információ alapján visszaállítja a helyes sorrendet
- csomagkapcsolt hálózatok csomóponti eszközei: **switch és router**
- csomagkapcsolás lehet összeköttetés-mentes (datagram) és összeköttetés alapú (virtuális áramkör kapcsolás)
- **ideális burst-ös és késleltetést jól tűrő adatok átviteléhez**

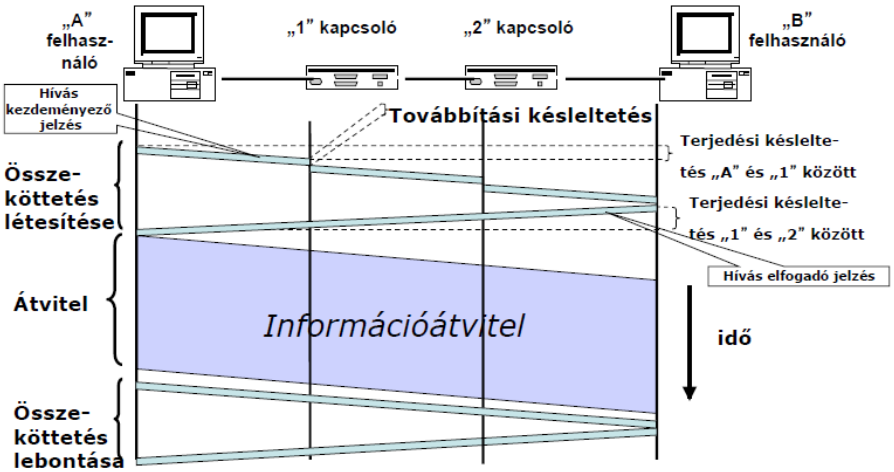
10.1.3.1 Datagram kapcsolás

- minden csomag önálló egység
- minden csomag tartalmazza a rendeltetési hely **globálisan egyedi címét**
- csomópontok megvizsgálják a fejrészt és kiválasztják a következő csomópontot
    - melyik juttatja el leggyorsabban
    - ha található szabad, amely képes a fogadásra
- belső és kimeneti torlódás egyaránt lehet
    - átmeneti és nem látható előre
- túlbiztosítás – belső kapcsolatok gyorsabbak, mint a külsők
- pufferelés – csomagok késleltetése
- visszaduzzasztás – ideiglenesen függessze fel a küldő a küldést
- párhuzamos kapcsolás – több párhuzamos útvonal kialakítása egyidejűleg

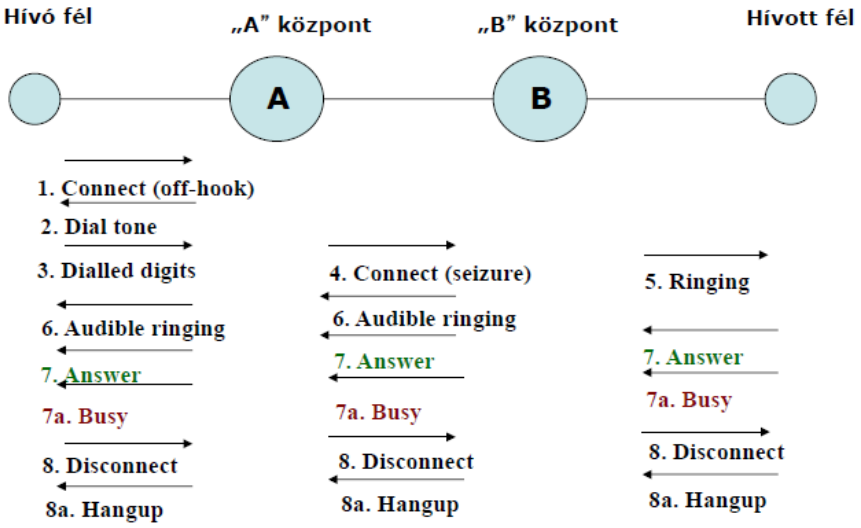
10.1.3.2 Virtuális ármakörkapcsolás

- csomópontokból készült szakaszok alkotnak összeköttetést végpontok közt
- két csomópont közt minden csomag ezt az útvonalat használja
- kapcsolat felépítése – adatátvitel – kapcsolat bontása
- virtuális áramkör azonosító egy adott csomópontban érvényes azonosító

10.2 Jelzés



- „hívások” létrehozása, fenntartása, bontása
- ehhez szükséges jelzések rendszere, protokollok



24. ábra Telefonközpont-példa

- sávon belüli (in-band) és sávon kívüli (out-band) jelzések
  - sávon belüli ma is megvan analóg vonalakban
  - sávon kívülinél külön csatornán vannak a jelzések
- közös csatornás jelzésátvitel – rugalmas, jobb sávszél, ISDN-ben külön digitális csatorna

10.3 Elnevezés és címzés

- **elnevezés** – egyedi név hozzárendelése végpontokhoz
- **címzés** – egyedi cím hozzárendelése a végpontokhoz
- **név → cím átalakítás** – címfeloldás
- hierarchikus névadást kövessünk
- tartományokra (domainek) bontás
- csúcs-szintű tartomány – top-level domain
- címzést is célszerű hierarchikusan szervezni – egyszerűbbé teszi az útvonalválasztást a hálózatban (nem kell minden csomópontba nagy fordítótábla)
- csoportosítás nélküli hiearchia – Ethernet kártya gyári címe
  - 48 bites címek

10.3.1 Névhasználat az interneten

- DNS – Domain Name System
- Top-level domaineket az IANA adminisztrálja (.hu, .org, .edu, .gov…)

10.3.2 Névfeloldás

- DNS végzi, redundánsan
- egyes kérések után cache-eli a feloldást későbbi lekérések esetén

11 2014.03.20 – Routing

- útvonalválasztás, útvonalkijelölés
- ide értjük csomópontokban a csomagok továbbítását is
- mivel a hálózat nem állandó, ezért **adaptív módon** kell és **adott esetben nagy hálózatról** jó döntést hozni
- útvonalválasztó módszerek, algoritmusok és protokollok

11.1 Mi a feladat?

- megismerni és nyilvántartani a hálózat felépítését
- csomópontok, összeköttetések (linkek), lehetséges állapotok
- **összeköttetés alapú eset:** csomópontokban lévő ismeretek alapján útvonal kiválasztása lépésről lépésre, csomópontokban annak feljegyzése és értesítés a kommunikáló végeknek, hogy megkezdhető az átvitel
- **összeköttetés nélküli eset:** nincs előre definiált útvonal, a csomópontok a csomag továbbításával egyidejűleg határozzák meg az útvonalat
- **bridging vs. routing** (bridge vagy switch által)
    - bridging adatkapcsolati rétegben, MAC címek alapján, elárasztással, helyi hálózatban funkcionál
    - routing hálózati rétegben, IP címek alapján, jobban skálázhatóbban
- **áramkörkapcsolt eset:** állandó átviteli csatorna kapcsolat időtartamáig
- **összeköttetés alapú eset:** virtuális csatorna létrehozása csomópontokban feljegyezve
- **összeköttetés-mentes eset:** nincs állandó feljegyzés, folyamatos útválasztás csomagonként
- csomópontok készíthetnek egyénileg is útvonaltáblát (**elosztott**) vagy készíthetünk **centralizált** útvonaltáblát is
    - centralizált: egy ponton gyűjtjük össze az információt, meghatározzuk az útvonalat, továbbítjuk a csomópontoknak ezt az információt
        - minden útvonaltábla „konzisztens” lesz – de nem a valós helyzetet mutatja
        - nagy hálózatnál megváltozhat az állapota, sérülékenység
    - elosztott: minden csomópont egyénileg épít útvonaltáblát
        - kevésbé sérülékeny
        - nincs egységes kép a hálózatról, különböző nézőpontok
- **statikus** és **dinamikus routing**
    - statikus: előre becsült forgalom alapján, centralizált útvonaltáblával – pl. telefonhálózat
    - dinamikus: aktuális forgalom állandó figyelése és táblák alakítása ennek megfelelően – internet esetén elosztott változatban
- gyűjthetünk információt: linkek állapotáról (működik, nem működik), sebességéről, forgalmáról, kiszolgálási díjról, terhelésről stb.
- célszerű a legrövidebb útvonalat meghatározni két felhasználó közt
    - útvonaltáblák meghatározása, kialakítandó útvonalak meghatározása

11.1.1 Routing módszerrel szembeni követelmények

- helytakarékos útvonaltábla, gyors kereséssel
- hibás tábla esélyének minimalizálása
- optimális útvonalak kijelölése – nem rögzített, hogy mi optimális (távolság, költség, stb.)

11.2 Routing telefonhálózatokban

- helyi hívás esetén nincs útvonal, közvetlenül vagy helyközi központon keresztüli kétlépéses kapcsolat
- távolsági hívás esetén helyi központon keresztül, ami egyazon távolságihoz van kötve
- centralizált táblakitöltés a becsült forgalom függvényében
- két útvonal távolsági központ számára
    - közvetlenül a hívott távolsági központjával
    - közvetett utak, tartalékként, ha a közvetlen út foglalt lenne (ne kelljen blokkolni)

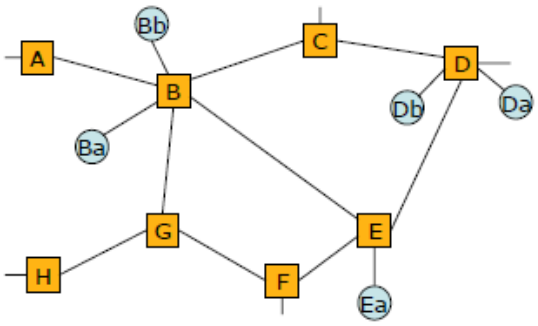
11.3 Routing a számítógép-hálózatban

- összekötöttség maximalizálására
- legyen a módszer elosztott
- két alapvető módszer:
    - távolságvektor – Bellman-Ford algoritmus
        - a csomópontok elmondják a hálózatról alkotott **elképzeléseiket** a **szomszédoknak**
    - linkállapot – Dijkstra-algoritmus
        - a csomópontok elmondják **mindenkinek** a szomszédjaikról nyert **tapasztalataikat**
- különbözőségek:
    - információk gyűjtésének tartalma
    - információk gyűjtésének módja
    - információk terítése

11.3.1 Távolságvektor

- elképzelések: milyen csomópont milyen távol van
- távolság mérése – legegyszerűbb esetben ez lehet a lépések (ugrások) száma
- súlyozni lehet átviteli sebességgel, sorbanállással, költséggel, stb.
    - távolság = költség

A	B,1				
B	A,1	C,1	E,1	G,1	∞
C	B,1	D,1			
D	C,1	E,1			
E	B,1	D,1	F,1		



B-hez megérkezik *C* üzenete:

B	A,1	C,1	D,2	E,1	G,1	∞
C	B,1	D,1				

B-hez megérkezik *E* üzenete:

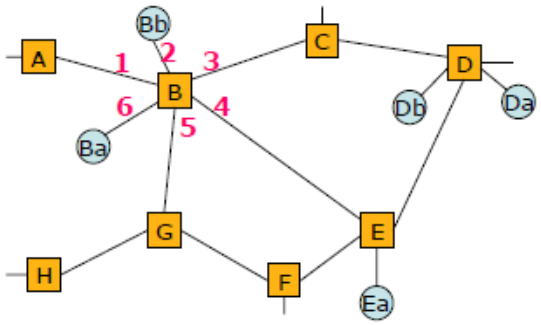
B	A,1	C,1	D,2	E,1	F,2	G,1	∞
E	B,1	D,1	F,1				

(∞ - a hálózat még nem látott része)

- *B* távolságvektora

B	A,1	C,1	D,2	E,1	F,2	G,1	∞

- Sorszámoztuk *B* portjait
- Kitölthetjük *B* útvonaltábláját:



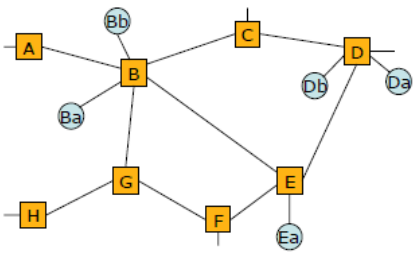
Ba	6	közvetlen
Bb	2	közvetlen
Da	3	
Db	3	
Ea	4	

- problémák: **végtelenig számolás**
  - meghibásodások
  - a csomópontok „routing by rumor” elven korrigálják saját táblájukat, nem tudják, hogy a szomszédok hogy jutottak az adott következtetésre
  - elméletben végtelenig növelhetik a távolságot (gyakorlatban: véges nagy)
  - egymásnak irányítják az el nem érhető csomóponthoz tartó forgalmat, túlterhelik a linket
- **split horizon:** megtiltja a csomópont felé, hogy arra hirdesse az útvonalat, akitől tanulta
- **route poisioning:** hirdeti a csomópontot, hogy elérhetetlen, törölni kell a táblákból
- ezt a kettőt együtt alkalmazzák
- **holddown timer:** indít egy számlálót a csomópont az útvonal elromlásakor, ha adott idő után nem javul meg, új útvonalat keres
- nő a komplexitás
- nem skálázódik jól és nem követi elég gyorsan a topológia változását

11.3.2 Linkállapot módszer

- tapasztalatok: szomszédokhoz vezető linkek aktuális állapota, költsége valamilyen mérték szerint
- elárasztással megkapja **mindenki**
- legrövidebb utak kiválasztása linkállapot információk alapján

B	A	1
B	C	1
B	E	2
B	G	1
B	Ba	0
B	Bb	0



- sorszám és életkor a téves információk ellen, elévülés

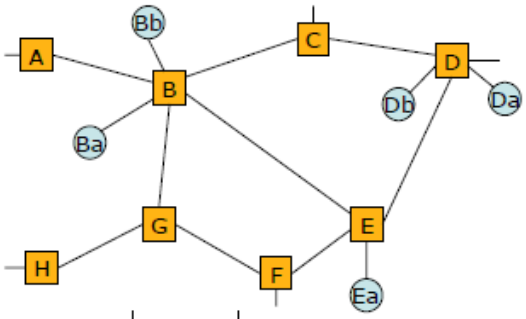
- hibavédő kódolás, autentikáció…

11.3.2.1 Dijkstra algoritmus

- *P* és *T* halmazok

B	C(B,1) E(B,2)
	D(C,2) ~~E~~**(E,3)**

- Ha *T* halmaz üres, akkor vége van



C	B	2
C	D	1
D	C	2
D	E	1
E	B	2
E	D	1
E	F	1

- P: permanent, T: temporary tároló
- P-ben kezdetben csak B szerepel
- T-be betesszük az ismert szomszédokat
  - ha még nincs ott, akkor betesszük
  - ha ott van, de kisebb költséggel is elérhető, akkor cseréljük
- T-ból P-be átvisszük a legkisebb költségűt – ha még nincs ott
- Ha T kiürül, készen vagyunk

11.3.3 Távolságvektor vs. linkállapot

- távolságvektor alapú protokollok
  - rosszabbul skálázódnak
  - lassan konvergálnak
  - hopszámban adnak meg költséget és nem veszik figyelembe az egyéb költségeket pl. linkkapacitás
  - kisebb hálózatokban jók
- manapság linkállapot a domináns módszer

11.4 Routing metrikák

- az adott routing algoritmustól függ
  - linkkapacitás, késleltetés
  - hopszám, megbízhatóság
  - csomagvesztési ráta
  - MTU stb. kombinációja

11.5 Hierarchikus routing

- N csomópont E linkje esetén
  - legrövidebb út: O(E logE)

    - útvonaltábla mérete: O(N)
- megoldás – **autonóm rendszerek**
    - önállóan oldanak meg routing feladatokat
    - külvilág felé egyetlen pontként szerepelnek
    - egységes routing policy van bennük
    - közös hálózatadminisztrátor kezeli
    - gyakran hívják **routing domainnek**
    - globális azonosítója van, IANA állítja ki
- határ routereken keresztül érhetők el más autonóm rendszerek
- routing táblában egyetlen csomópont lesz
- típusai
    - **Multihomed AS:** több, mint egy AS-sel van összekötve, redundánsan kapcsolódik
    - **Stub AS:** csak egy másik AS-sel van összekötve
    - **Transit AS:** csak átmenő forgalom, összes ISP ilyen
- **hot-potato vs. cold-potato elv:**
    - attól függően, hogy az AS mennyi ideig szeretné saját tulajdonban tartani a forgalmat
    - cold-potato elvnél könnyebben nyújt minőségi garanciát

11.6 Mobil végpontok kezelése

- végpontok átmozoghatnak más csomópontok környezetébe, más AS-be
- jó nem tudnia az egyes csomópontoknak, hogy a mobil végpont épp hol van – a hálózat feladata ezt kezelni (ez a mobil routing lényege)

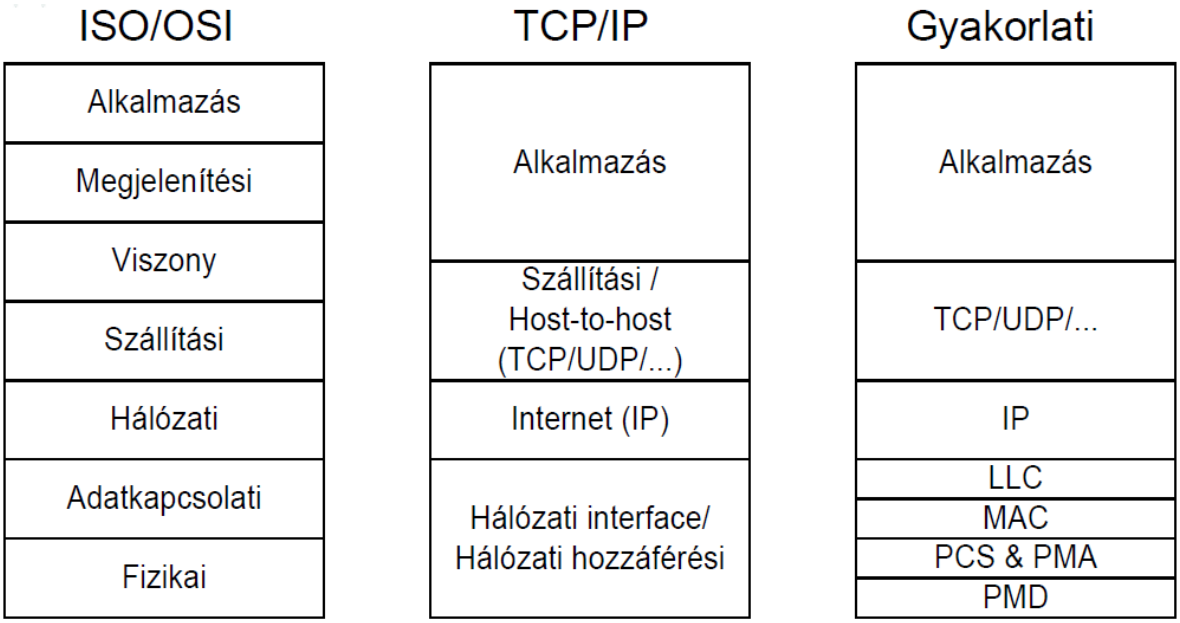
11.7 Multicast routing

- több csomópontnak kell egyidejűleg eljuttatni a csomagokat pl. videó-beszéd-videókonferencia
    - magyarul „többesadás”
- csoportok azonosítása után hálózat végpontjait azonosítják a fizikai és a logikai címek
- csoportok azonosításához – **csoport-cím**
    - legalább egy aktív tagja legyen
- sűrű elhelyezkedés esetén:
    - elárasztás + lemondás (csoport/forráslemondás)
- ritka elhelyezkedés esetén:
    - gerincalapú fa alkalmazása
- elhelyezkedés független: kettő ötvözete

12 2014.03.27 – IP

12.1 IP – bevezetés

- 1974 – IP és TCP alapgondolata
  - ARPANET-ben NCP kiváltására
  - internet kifejezés megjelenése – „Internet Protocol Suite”
- 1980 – IPv4 (Internet Protocol version 4)
  - első, széles körben használt IP és TCP verziók megalkotása
  - amerikai védelmi minisztérium által támogatott szabványok
  - 1983. január 1.: átállás NCP-ről TCP-re
- alapvetően ma is ezt használjuk



IP: Internet Protocol
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
LLC: Logical Link Control

MAC: Medium Access Control
PCS: Physical Coding Sublayer
PMA: Physical Medium Attachment
PMD: Physical Medium Dependent

- OSI 7 rétegét gyakorlatban nem szokták megvalósítani
- IETF szerint az internet protokolljainak architektúrája továbbra sem OSI alapú lesz
- 5 réteg használata, TCP/IP modell, de: interface réteg = adatkapcs. + fizikai

12.1.1 Az IP feladata

- hálózati protokoll
- adattovábbítás a végpontok között
- két fő funkció: **címzés** és **routing**
- tördelés, fragmentálás

12.1.2 Az IP jellemzői

- csomagkapcsolt, datagram típusú
  - összeköttetés-mentes
  - „best-effort” jelleggel működik
    - a csomagok elveszhetnek
    - duplikálódhatnak

        - megváltozhat a sorrendjük
        - meghibásodhatnak
- nem nyújt:
    - ütemezést
    - torlódáskezelést
    - titkosítást és hitelesítést

12.2 IP – címzés

12.2.1 IPv4 cím felépítése

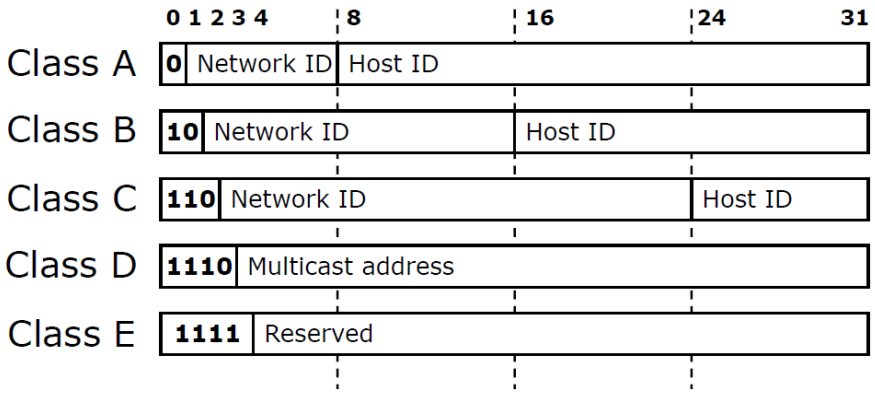
- 4 bájt (32 bit)
    - $2^{32} \approx 4*10^9$ ➔ KIMERÜLT
- jelölések:
    - bináris: 1011000 10010011 00111110 11100001
    - „dotted decimal”: 176.147.62.255
- cím felépítése: hálóazonosító + egyedi azonosító

12.2.2 Címosztályok

- eleinte az első 8 bit hálózati, majd bevezették a címosztályokat
- különböző méretű hálózatok → különböző méretű címosztályok
- IP címeket az IANA adja ki
    - A és B típusú címeket kaphatnak tipikusan országok, ISP-k, egyetemek

Osztály	Hálózatazonosító bitjeinek száma*	Hálózatok száma*	Hálózaton belüli címek száma*
A	8	126	16777214
B	16	16382	65534
C	24	2097150	254

- A, B, C osztályú címek **unicast** (egyedi) címzésre
- D osztály: **multicast** (többes címzés)
    - 224.0.0.0 – 239.255.255.255
- E osztály: fenntartott címosztály
    - 240.0.0.0 – 255.0.0.0
- osztályazonosítók kiosztása:



- osztályazonosító prefix, diszjunkt címtartományok

- Host ID:
  - csupa 0 – hálózat címe, eszközök kezelhetik
  - csupa 1 – broadcast cím, mindenkinek szól
  - emiatt csak $2^{N}_{host} - 2$ számú terminál címezhető
- 127.0.0.0 – 127.255.255.255
- gond a routing aggregációnál
- nem skálázható osztályokon alapuló rendszer
- egyes nagy cégek és egyetemek A osztályú címet kaptak, amit nem használtak ki (túl sok cím), B viszont túl kicsi lett volna. Ugyanez kisebb cégeknél: B túl nagy, C túl kicsi

12.2.3 Privát címtartományok

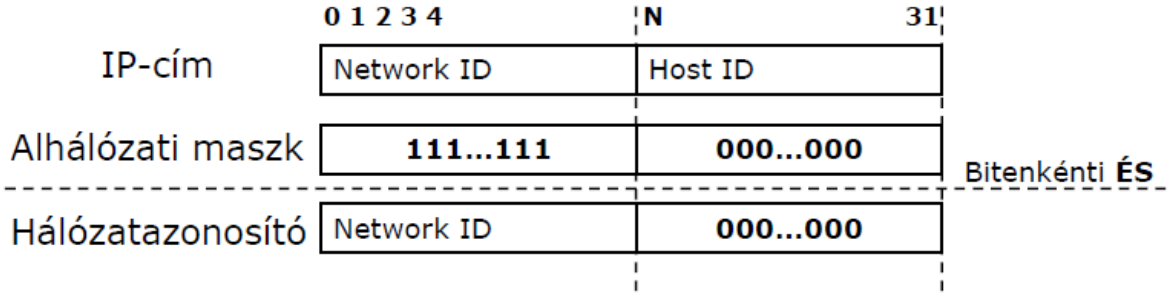
- első ötlet az IPv4 címek kimerülésének kezelésére
- ötlet: ha pl. a cég nem kommunikál kifelé, nem kell neki globálisan egyedi IP cím
- ha kifelé is szeretne kommunikálni: NAT-olás (Network Address Translation)
- helyi hálózatban:
  - 1 db A osztályú cím – 10.0.0.0 – 10.255.255.255
  - 16 db B osztályú cím – 172.16.0.0 – 172.31.255.255
  - 256 db C osztályú cím – 192.168.0.0 – 192.168.255.255
- gondok:
  - késleltette az IPv6 bevezetését
  - megsérti a végpont-végpont konnektivitás elvét (néhol nem is működik)
  - privát hálózatok egyesítésével címduplikáció
    - köztük újracímzés, NAT-olás
  - privát címek kiszivárogtak az internetre
    - névszerver-terhelés ezáltal
    - határ routerek szűrik

- Milyen IP-címek az alábbiak?
  - 152.66.115.35 (www.bme.hu)
    - Első bitjei: 10… ⇨ B osztályú cím
    - Utolsó 16 bitje nem csupa 0 vagy 1 ⇨ egyedi cím
  - 195.199.23.255
    - Első bitjei: 110… ⇨ C osztályú cím
    - Utolsó 8 bitje csupa 1 ⇨ broadcast cím
  - 10.97.15.255
    - Első bitje: 0… ⇨ A osztályú cím
    - Utolsó 24 bitje nem csupa 0 vagy 1 ⇨ egyedi cím
    - Privát címtartománybeli ⇨ globálisan nem használható
- Mi legkisebb és legnagyobb C osztályú cím?
  - 11000000 00000000 00000000 00000000 ⇨ 192.0.0.0
  - 11011111 11111111 11111111 11111111 ⇨ 223.255.255.255

12.2.4 Címosztályok általánosítása

- osztály alapon nem skálázható + címtartomány kimerülése

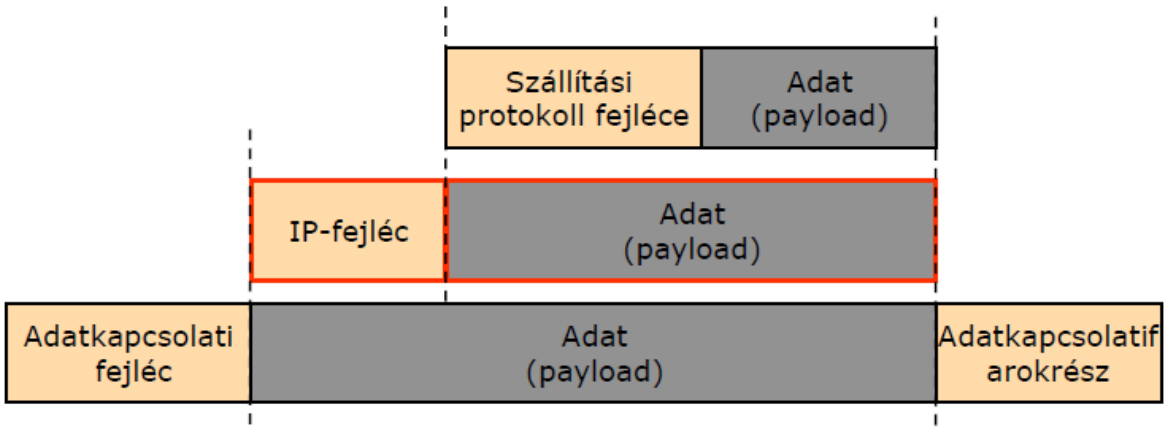
- **CIDR** – Classless Inter-Domain Routing
  - **VLSM** – Variable Length Subnet Mask
    - osztályok eltörlése
    - a cím 32 bitje tetszőleges helyen lehet szétvágva
  - alhálózati maszk alkalmazása
- subnetting: alhálózatokra való osztály
  - 1 db A cím → 256 B osztályú cím
  - 1 db B cím → 256 C osztályú
  - nem lehet prefix alapján megállapítani



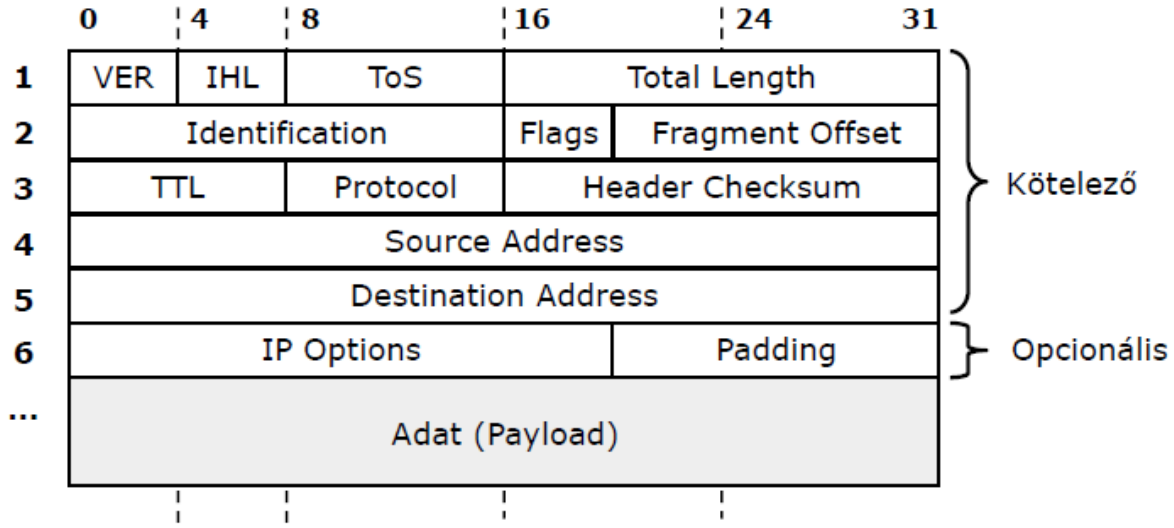
- Jelölés a hálózat egyértelmű azonosítására
  - <hálózat IP-címe> / <alhálózati maszk egyeseinek a száma>
  - Így pl. a BME hálózata: 152.66.0.0 /16 (alhálózati maszkja 255.255.0.0)
  - Hasonlóan:
    - A osztály: /8
    - B osztály: /16
    - C osztály: /24

12.3 IP csomag

- IP-fejléc/fejrész (IP header) + IP adat (payload)
- alsóbb rétegbeli protokoll adatrészébe ágyazódik be
- magasabb rétegbeli protokollok ágyazódnak be az ő adatrészébe



12.3.1 Fejléc



25. ábra IP fejléc szerkezete

- **VER** – version, 4 bit
  - értéke **4** (IPv4) vagy **6** (IPv6)
- **IHL** – Internet Header Length
  - IP fejléc mérete **32 bites szavakban**
  - minimum **5**, maximum **15**
- **ToS** – Type of Service (8 bit)
  - QoS osztályok, legtöbb router még nem támogatja

Bitek	Jelentés
0-2	**Precedencia** **Példák:** **„network control"** **„priority"** **„routine"**
3	**Késleltetés (normal/low)**
4	**Throughput (normal/light)**
5	**Reliability (normal/high)**
6-7	**Fenntartott**

- **Total Length** – Maximális hossz (16 bit)
  - teljes IP csomag mérete **byte**-okban
  - minimum **576** (IP-fejléc 20 + 512 adat + 44 IP opciók + alsóbb fejlécek)
  - maximum **65536** (max. 65515 adat)
- **Identification** (16 bit): az IP töredékek egyedi azonosítása
- **Flags** (3 bit)
  - 0: fenntartott – 0-nak kell lennie

    - 1: DF – Don't Fragment
        - 1: ha tördelni kéne, el kell dobni
        - ezt használja az MTU (Maximum Transmission Unit) és az útválasztás néhány TCP verzióban és az IPv6-ban
    - 2: MF – More Fragment
        - 1: ha nem utolsó töredék
        - 0: utolsó töredék, vagy tördeletlen
- **Fragment Offset** (13 bit)
    - eredeti csomagban lévő kezdőpozícióját adja meg e töredékben lévő adatnak 8 bájtos formában
- **TTL** – Time to Live (8 bit)
    - csomag élettartama
    - eredetileg másodpercben
    - gyakorlatban **hopszámban** mérve
        - mindig csökkentjük, ha nem nagyobb 0-nál, eldobjuk
- **Protocol** (8 bit)
    - adatrészben lévő protokoll azonosítója
    - IANA felügyeli ezeket
    - pl. 6 – TCP, 17 – UDP
- **Header Checksum** (16 bit)
    - az IP fejléc minden 16 bites szavának összegére számolt egyes komplemens
    - csomag érkezésekor ellenőrizni kell és újraszámolni a helyességét továbbítás esetén
- **Source/Destination Address** (2*32 bit)
    - feladó és címzett IP címe
- **IP Options:** csomag útját lehet kijelölni, legtöbb router biztonsági okokból eldobja
- **Padding:** IP Options kiegészítése

12.4 IP IP – útvonalválasztás

- csomagkapcsolt, connectionless, best-effort (nincs garancia)
- „hot potato"-elv
    - minél gyorsabb továbbítás
    - csak következő csomópontot kell ismernünk
    - előnye, hogy kis erőforrásigényű, gyorsan továbbítható, gyors, kis memóriaigényű
    - kevés ismeret a hálózatról
    - datagramra tökéletes (nincs garancia, nyugtázás kezelése → gyors marad)
- továbbításhoz kell tudnunk:
    - célcím → csomag tartalmazza
    - saját routing tábla ismerete
    - mekkora egységek érkeznek → csomag tartalmazza
    - mekkora továbbítható → következő hálózat MTU-ja tartalmazza
    - QoS → nincs garancia
        - opcionálisan ToS mező felhasználásával egyéb protokollok
- routing tábla felépítése

Merre megy?		Merre küldjük?		
Hálózat címe	Alhálózati maszk	Interfész	Közvetlenül kapcsolódó	Következő csomópont
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>
<IP-cím>	</N>	<azonosító>	<igen/nem>	<IP-cím>

- a cél IP cím akkor tartozik egy hálózatba, ha a hálózatazonosító része megegyezik
- ha több hálózatra is illeszkedik, a leghosszabb egyezőséget kell keresni a táblában
- rosszul skálázódik, minden bejegyzést végig kell nézni, általában bifával implementálják
- keresés csak akkor, ha az nem saját cím

12.4.1 Alapértelmezett útvonal (Default Route)

- alapértelmezett cím
  - akkor megy erre, ha nem ismeri a hálózatot: 0.0.0.0/0
- alapértelmezett átjáró
  - fenti bejegyzéshez tartozó következő csomópont címe
  - inkább útvonalválasztónak szokás hívni
- nem feltétlenül van ilyen → ha nincs, a csomag eldobásra kerül
- végponton gyakran két cím szerepel csak
  - helyi hálózat (helyi végpont közvetlenül elérhető)
  - alapértelmezett útvonal (minden más távoli hálózat)

12.4.2 Metrikák alkalmazása

- hálózati utak közti preferenciát tudunk megadni (elérhetőség, terheltség, költség stb.)
- lehet statikus és dinamikus is
- jobb metrikával rendelkező kapcsolaton küldünk csomagot

12.4.3 Megfelelő bejegyzés kiválasztása közvetlen és nem közvetlen esetben

- közvetlenül kapcsolódó (helyi) hálózat
  - címzettnek közvetlenül küldeni
  - ehhez kell az adatkapcsolati réteg beli címe
- nem közvetlenül kapcsolódó (távoli) hálózat
  - útválasztónak küldjük – IP címet tilos módosítani
  - csak adatkapcsolati rétegben kell útválasztónak címezni

12.5 IP – ARP/RARP

12.5.1 ARP – Address Resolution Protocol

- ha ismerjük az IP címet és a hozzá tartozó adatkapcsolati címre van szükség
- szinte minden adatkapcsolati és hálózati réteget támogat
- broadcast mindenkinek → releváns IP cím válaszol
- közvetlenül adatkapcsolati réteg protokollja kapja meg

0 - 15 bits		16 - 31 bits
Hardware Type		Protocol Type
HLen (1 byte)	PLen (1 byte)	Operation
Sender HA (bytes 1 - 4)		
Sender HA (byte 5- 6)		Sender PA (byte 1 - 2)
Sender PA (byte 3 - 4)		Target HA (byte 1 -2)
Target HA (bytes 3 - 6)		
Target PA (bytes 1 - 4)		
RARP header structure		

- Hardware Type
  - Ethernet: 0x0001
- Protocol Type
  - IP: 0x8000
- HLen
  - Ethernet: 6 byte
- PLen
  - IPv4: 4 byte
- Operation
  - ARP request = 1
  - ARP reply = 2
  - RARP request =3
  - RARP reply =4
- Sender Hardware Address
  - <MAC-cím>
- Sender Protocol Address
  - <IP-cím>
- Target Hardware Address
- Target Protocol Address

- ARP tábla: adatkapcsolati rétegbeli és IP címpárok
- statikus (kézzel felvitt) vagy dinamikus (ARP feloldás eredménye) – cache funkció, egy idő után elévül és törlődik
- **ARP probe:**
  - IP cím használata előtt felderíti, hogy nem használja-e már más
  - broadcast kérés, küldő IP címe csupa nulla – saját címét kérdezi le
  - IPv4 Address Conflict Detection
- **ARP hirdetmény**
  - ha változik a MAC vagy IP cím, küld egy üzenetet, hogy tudják frissíteni a többiek az ARP táblájukat
  - oprendszer boot (hálózati kártya változott) vagy load balancing (több kártya)

12.5.2 RARP – Reverse ARP

- adatkapcsolati rétegből IP címet kérdez le
- hálózatmenedzsmentnél, permanens tár nélküli eszközöknél használják
  - pl. hálózatról töltődik be a rendszer
- már nem használják, helyette BOOTP, DHCP

12.6 IP - Útválasztó protokollok

- útvonal-irányítási információk begyűjtése
- hurokmentes routing
- újabb csomópontok csatlakozásának biztosítása
- csomópontok és hálózatok leválasztásának kezelése

12.6.1 Útvonalválasztó protokollok osztályozása

- Ad hoc – kis és gyorsan változó hálózatokra

- Interior Gateway Protocols – kisebb AS-eken belül
- Exterior Gateway Protocols – AS-ek között (internet routing protokollja)

12.6.1.1 Ad hoc routing protokollok

- proaktív – folyamatosan karbantartott táblák
- reaktív – igény szerinti célfelderítés
- hibrid – a fenti kettő együttesen
- hierarchikus, multicast, földrajzi elhelyezkedés alapján, energiatakarékos

12.6.1.2 IGPs

- lehet távolság-vektor alapú és linkállapot alapú (lásd 11.3)
    - távolság-vektor alapúak: RIP, IGRP, EIGRP
        - **RIP** – nem preferált
            - nem preferált
            - hopszám metrika
            - rosszul skálázódik
            - könnyen konfigurálható
            - maximum 15 hop hálózaton belül
            - update: 30 másodperc
        - **IGRP** – Cisco specifikus
            - többféle metrika
            - max. 255 hop
            - update: 90 másodperc
            - már nem használják
        - **EIGRP** – IGRP javítása
            - gyorsabb konvergencia
            - hurokmentes működés
            - 6 metrika kombinációja
    - linkállapot alapúak: OSPF, IS-IS
        - **OSPF** – nagy céges hálózatokban
            - link state database elárasztással
            - gyorsan konvergál
            - többféle metrika
            - hierarchikus
            - v2: IPv4 – népszerűbb
            - v3: IPv6-ra átspecifikálva
        - **IS-IS** – nagy kiterjedésű ISP-knél
            - OSI fejlesztés, nem IP csomagokban
            - könnyebb átállás IPv6-ra

12.6.1.3 EGPs

- **EGP** – sokáig Internet EGP-je, ma már nem használt
- **BGP** – 1995-től használt EGP (BGPv4)
    - routing táblában:
        - célhálózat
        - következő router
        - **útvonal célhálózatig**
    - teljes útvonal dinamikusan nyilvántartva
    - AS határ routerek végzik → hirdetve a többi AS elérhetőségét

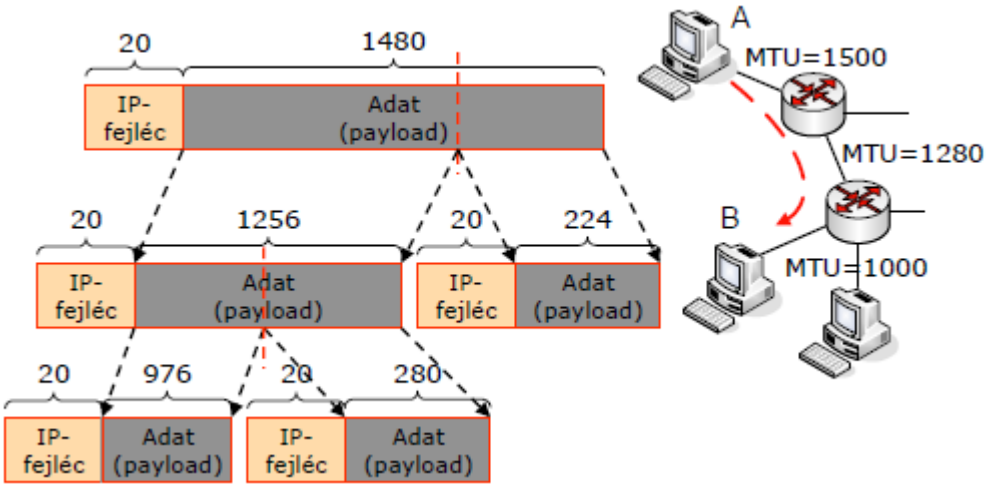
    - üzenet: AS-ek sorozata a célhálózat felé
    - minden AS határ router fogadáskor megnézi, megfelel-e saját szabályainak
- címaggregáció → kisebb routing táblák, gyorsabb keresés, gyorsabb terjesztés

12.7 IP – tördelés

- hálózatok alsóbb rétegei meghatározzák a keret maximális méretét
- adatkapcsolati réteg fej- és farokrészét leszámítva ez az MTU (Maximum Transmission Unit)
    - Ethernetnél 1500 byte
- eltérő technológiák, eltérő kapcsolatok → tördelni kell

12.7.1 Tördelés menete

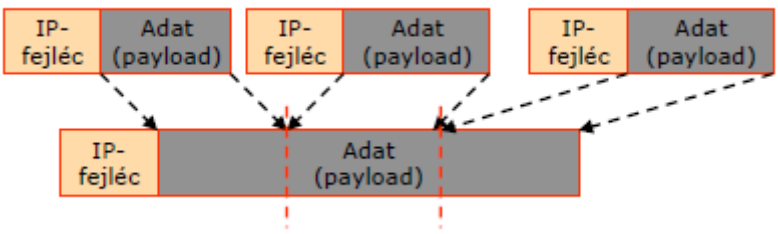
- minden csomagnak kell fejléc



- IP fejlécben tördeléshez mezők: Identification, Flags, Fragment Offset
- IP fejléc változása tördelés közben:
    - Total Length mezőt a töredék méretére állítani
    - Identification mezőt generálni, ha nem volt
    - Adat tördelése 8 byte-os egységekre
    - tördelésnek megfelelő Fragmentation Offset beállítása
    - minden csomagban MF = 1, kivéve az utolsóban
    - checksum változása

12.7.2 Töredékek összeállítása

- fragment offsetekből a töredék helye meghatározható
- csak a címzett végezheti el
- ha hiányzik egy darab → a többit is eldobja
- csak teljes csomagot továbbít felsőbb szint felé



- router megvizsgálja, hogy

- neki címezték-e
- hibás-e a fejléc
- ismeri-e a címzett hálózatát
- TTL értéke >0
- kell/lehet-e tördelni
- kell-e visszajelzést küldeni

12.8 ICMP és IGMP

12.8.1 ICMP – Internet Control Message Protocol

- jelzés- és menedzsmentüzenetek → visszajelzés a „best effort" hálózatról
- IP felett
- hibaüzenetek, kérdések, válaszok
- gyakran használt: Ping (RTT), Traceroute
- továbbiak: Source Quench (forgalom visszafogása), Redirect (forgalom más irányba terelése), Address Mask Request (alhálózati maszk lekérése)
- IP csomagba ágyazva, IPv4 fejléce után következik a fejléce (8 byte)
- **különleges elbánásban** részesül (de best-effort szolgáltatás)
- fejléc:
  - típuson belül kód pl. címzett elérhetetlen, ezen belül kód az okról pl. letiltott hálózat
  - ellenőrzőösszeg az egész ICMP üzenetre
  - hibaüzenetnél: hibás üzenet fejléce + adatmezőjéből az első 8 byte

12.8.1.1 Traceroute

- először TTL = 1, TTL = 2 …
- routerek sorban eldobják, Time Exceeded üzenetet küldenek róla
- megadja a csomópontok listáját, amit a csomag érint a célig
- alkalmazzák hibaelhárításra, tűzfal felderítésre, letöltésnél, pentestingnél
- további

12.8.2 IGMP – Internet Group Management Protocol

- IP-t futtató csomópont-csoportok kezelésére
  - online streaming, játékok
  - IP csomagokban
- multicast csoportok kezelése
- csoportok lekérdezése és csoportokhoz csatlakozás
- TTL általában 1 → csak a helyi hálózaton érvényes
- csak IPv4-hez

13 2014.04.03 – IPv6

13.1 Motivációk – az IPv4 hibái

- 32 bites címtartomány kimerülése
- sok felesleges mező a fejlécben, tördelés a köztes csomópontokban → **erőforrás-igényes**
- nem támogatja a hitelesítést és a titkosítás → **nem biztonságos**
- csak külön protokollal oldható meg a mobilitás
- címtartomány kimerülése:
    - egyre nagyobb az internet elterjedtsége → egyre több eszköz
        - mobiltelefonokon **4G-n** már **IPv6** van!
        - szenzorok, intelligens otthon, stb.
    - pazarló címfelhasználás, dial-up helyett always-on elterjedtsége
    - virtualizáció
- 2011. februárjában utolsó öt blokk kiosztásra került (16 millió cím/blokk)
- a kimerülést késleltette
    - **NAT**
    - **CIDR** (**VLSM**, újrakiosztás)
    - ISP-k IP-**címgazdálkodása**
    - **virtuális hosting**

13.2 IPv6 címzés

- 64 bites címek ideálisak, nem túl nagy fejléc-növekedéssel járnak, viszont tovább elegendőek
- megállapodás szerint 128 biten ábrázoljuk
    - **3,4 * 10^{38} cím** kezelésére elég
    - **8*4 db hexadecimális szám**ként
        - pl. FEDC:BA94:7654:3210:FEDC:BA98:7654:3210
    - a **vezető nullák elhagyhatók**
        - FEDC:**0094**:**0004**:**0000**:**000C**:BA98:7654:3210 → FEDC:94:4:0:C:BA98:7654:3210
    - 16 bites nullákat tartalmazó részek kihagyhatók (max. 1 blokk)
        - FEDC:**0000**:**0000**:**0000**:**000C**:BA98:0000:3210 → FEDC::C:BA98:0:3210
        - localhost – 0:0:0:0:0:0:0:1 → ::1
- vannak címek, amelyek IPv4-ből származnak, ekkor megengedett:
    - 0:0:0:0:0:0:A00:1
    - ::10.0.0.1
- hálózati címek (**prefixek**) jelölése:
    - FEDB:ABCD:ABCD::/48
    - FEDB:ABCD:AB00::/40
- hivatkozásként: http://[FEDC::C:BA98:0000:3210]/index.html

0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP (non-IP addresses used by ISO)
0000 010	Reserved for IPX (non-IP addresses used by IPX)
0000 011	Unassigned
0000 1	Unassigned
0001	Unassigned
001	Unicast Address Space
010	Unassigned
011	Unassigned
100	Unassigned
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link Local Use addresses
1111 1110 11	Site Local Use addresses
1111 1111	Multicast addresses

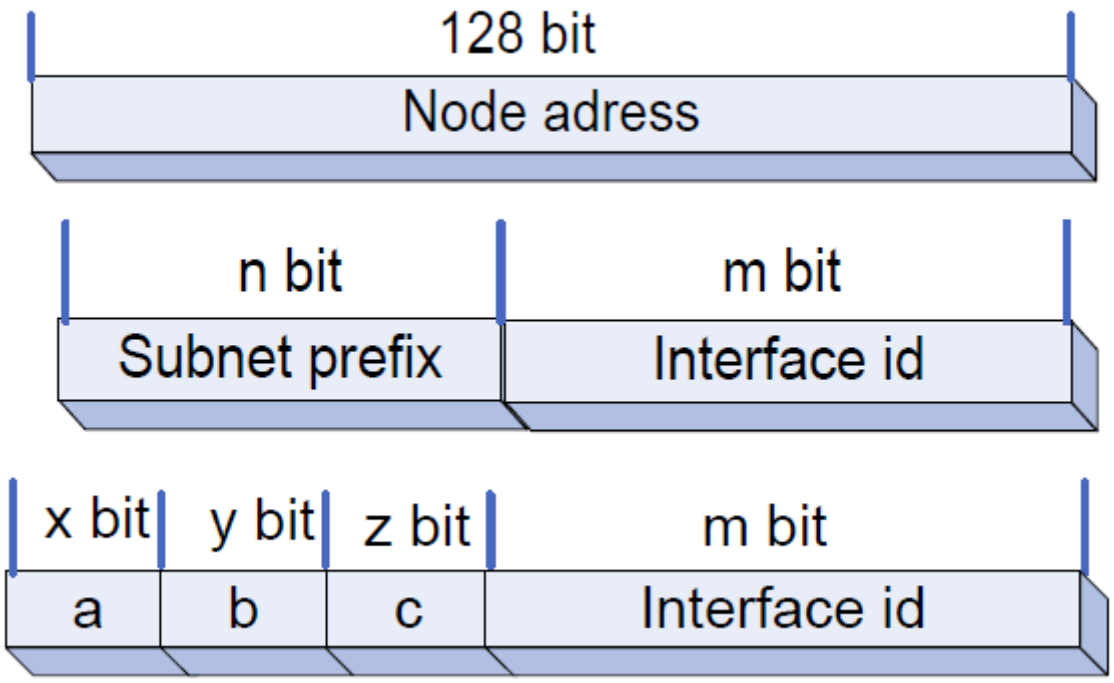
26. ábra Cím prefixek

13.2.1 Címtípusok IPv6-ban

- **unicast:**
  - IPv4-hez hasonlóan
  - pontosan egy csomóponthoz tartozik (egyedi)
  - minden IPv6 csomópontnak **legalább egy** ilyen címe van
- **multicast:**
  - csoportot azonosít
  - minden, csoporton belüli csomópont megkapja az erre küldött adatot
  - broadcast helyett is!
- **anycast:**
  - csoportot azonosít
  - biztosított, hogy a csoport egy tagja megkapja az erre küldött üzenetet (pl. küldőhöz legközelebbi)

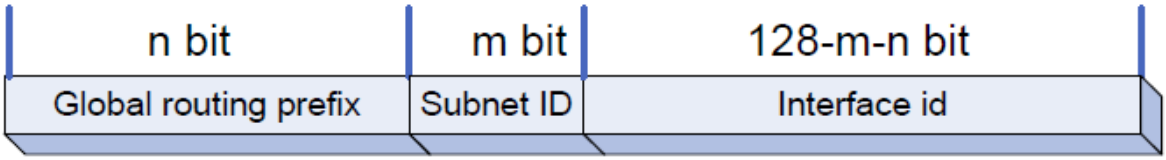
13.2.2 Unicast címek

- típusai
  - globális (aggregálható vagy IPv4 kompatibilis)
  - link local (prefix után a cím végig 0, nem routolható)
  - site local
  - beágyazott IPv4 címet tartalmazó



27. ábra Strukturált unicast cím felépítése

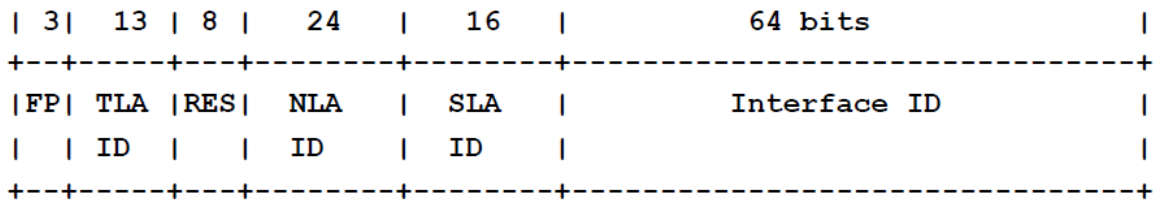
13.2.2.1 Globális unicast cím



28. ábra Globális unicast cím felépítése

- csomópont és kapcsolódósának azonosítása („ki vagy és hová kapcsolodódsz?”):
  - routing prefix → routing topológia
  - subnet ID → hálózati adminisztrátor
  - interface ID → csomóponti azonosítás
    - megegyezés szerint 64 bit
    - adatkapcsolati rétegbeli címből képződik
    - Ethernet MAC címből: 48 bit → 64 bit (EUI-64)
    - kaphatja DHCP szervertől is véletlenszerűen

13.2.2.2 Aggregálható globális unicast cím

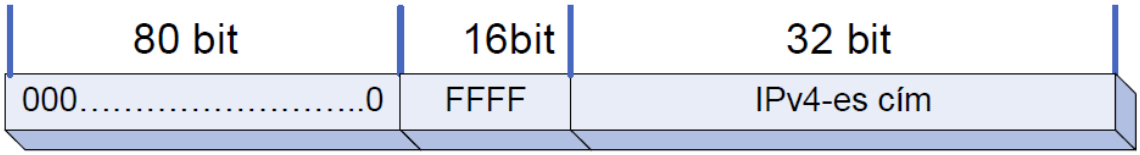


- FP — Format Prefix (001)
- TLA ID — Top-Level Aggregation Identifier (régió)
- RES — Reserved for future use
- NLA ID — Next-Level Aggregation Identifier (szolgáltató)
- SLA ID — Site-Level Aggregation Identifier (előfizető és alhálózat)
- INTERF. ID — Interface Identifier

29. ábra Aggregálható globális unicast cím felépítése (RFC 2347)

13.2.2.3 IPv4 kompatibilis címek

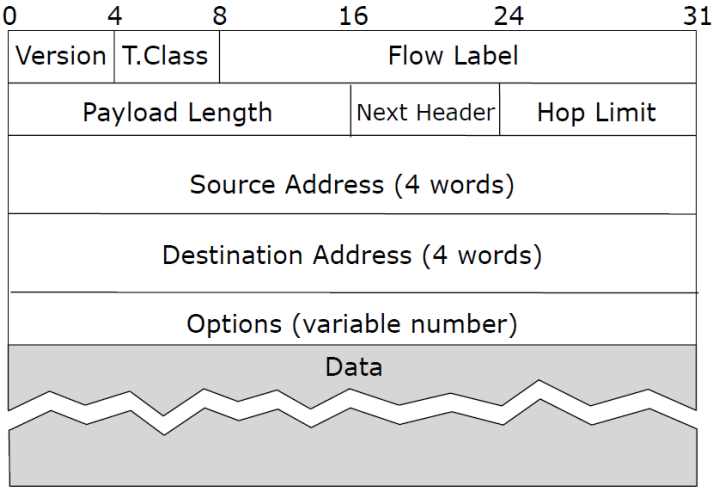
- 000-val kezdődnek
- IPv4 kompatibilis IPv6 címek → pl. ::10.0.0.1
- IPv6-ra képezett IPv4 címek → pl. ::FFFF:10.0.0.1



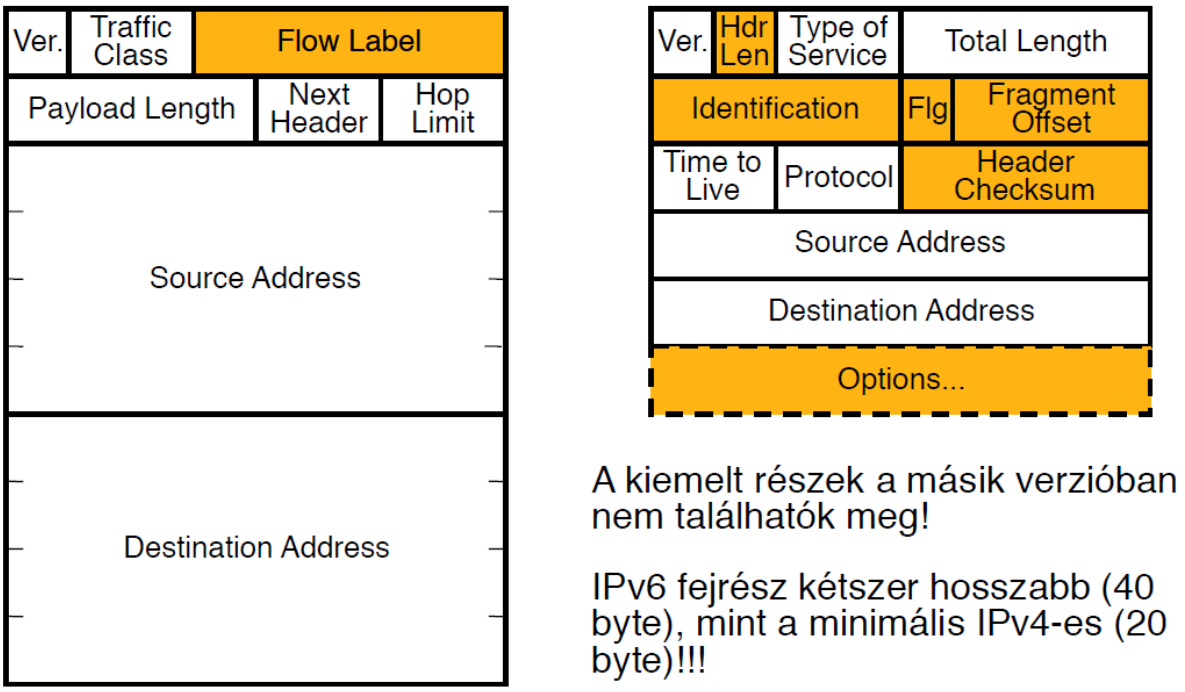
30. ábra Kompatibilis címek felépítése

13.3 IPv6 csomag felépítése

- IPv4 csomag felépítését lásd a 12.3.1 fejezetben



31. ábra IPv6 csomag felépítése



32. ábra IPv4 és IPv6 fejlécek összehasonlítása

- ami az IPv6 fejlécéből kimaradt:
  - nincs ellenőrzőösszeg (checksum)
    - nem kell minden routernek ellenőriznie → nő a feldolgozási sebesség
    - kisebb hibaarány, jobb kapcsolatok
  - nem változik a fejléc mérete
    - fix méret miatt nő a feldolgozási sebesség
    - nem kell IHL mező
  - nincs ugrásonkénti tördelés
    - nő a feldolgozási sebesség
    - nem kell ehhez mező → kisebb fejrész
    - kell viszont MTU felderítés (Path MTU Discovery)

13.3.1 IPv6 fejléc mezői

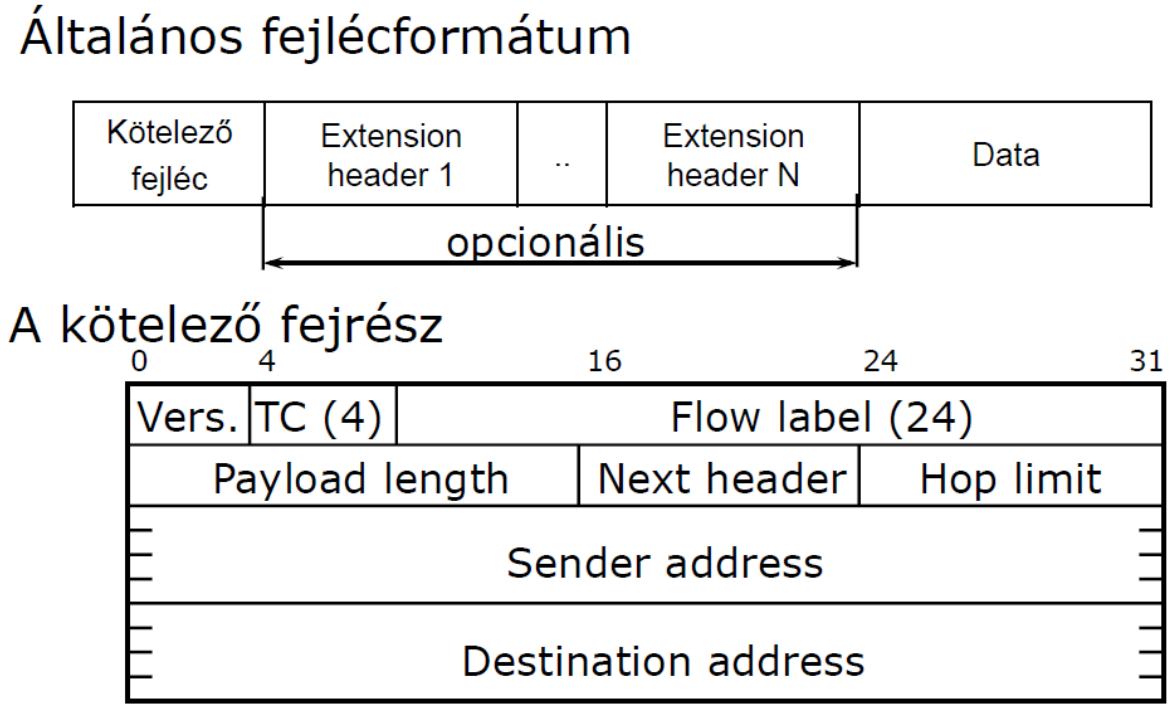
- **Version** (4/6 bit)
- **Traffic Class** – forgalmi osztály
  - IPv4-ben lévő ToS mezővel megegyező módon működik
  - QoS lehetőség biztosítása
- **Flow Label** – folyam azonosító címke (24 bit)
  - generált, adott kapcsolatot azonosító mező ← datagrammok nem függetlenek!
  - QoS és igazságos adatsebesség megosztás segítője
  - egy forrás-célállomás pár esetén több folyam is lehet – cím és flow label együtt azonosít
- **Payload Length** – adathossz
  - fejrész nélkül!
  - maximum 64 KB (jumbogram opció – 4 Gb)
- **Hop Limit** – ugrás korlát (8 bit)
  - IPv4 TTL mezőjével azonos
- **Címmezők** – 2*128 bit

- **Next header** – következő fejléc (8 bit)
    - beágyazott PDU típusát adja meg **VAGY** IPv6 Extension header (kiterjesztés) típusát adja meg

13.3.2 IPv6 opciók – Header Extension

- célja: egyszerű fejléc kiegészítése opcionális lehetőségekkel
- **Hop-by-hop Options Header** (0)
    - QoS támogatás
    - óriás datagrammok (jumbogrammok)
    - minden közbülső csomópont meg kell, hogy vizsgálja
- **Routing Header** (43)
    - útba ejtendő routerek felsorolása
    - IPv4 opciókhoz hasonlóan – laza/szigorú forgalomirányítás
- **Fragmentation Header** (44)
    - mint IPv4-ben, de **csak a forrás** darabolhat
- **Authentication Header** (51) – **IPSec**-ből
- **Encapsulation Security Payload Header** (ESP) – IPSecből

13.3.3 IPv6 fejrész kiterjesztése



13.3.3.1 Routing Extension

- IPv4-hez hasonlóan
- **Header length** – fejrész hossza **64 bites szavakban**
- **Type** – típus
- **Érintendő címek listája**
    - max. 24 cím
    - következő csomópont megtalálása anycast címzéssel

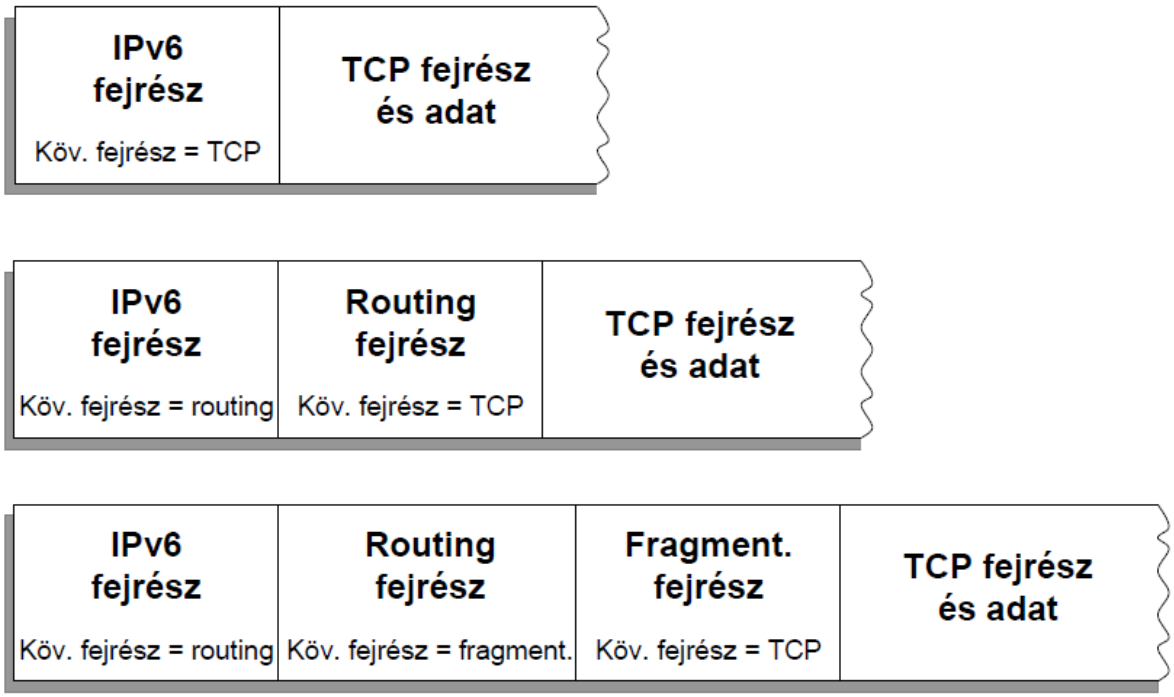
13.3.3.2 Fragmentation Extension

- 13 bites **fragmentation offset**
- **MF bit** itt is jelen van

- 16 bites helyett 32 bites **Fragmentation ID** mező
- tördelni és összeállítani **csak a fogadó** fog

13.3.4 IPSec az IPv6-ban

- két Extension Header a biztonság növelésére
- IPSec-re minden IPv6 csomópontnak képesnek kell lennie
- megvalósítás módja:
    - **Authentication Header** – Valóban a látszólagos feladótól jött? Lett-e módosítva?
    - **ESP Header –** Titkosított csomag tartalma



33. ábra IPv6 fejléc példák

13.4 IPv6 koncepció összefoglalása

- megnövelt címtartomány
- egyszerűbb, rugalmasan bővíthető fejléc formátum
- gyorsabb feldolgozás csomópontokban
- erőforrás-allokáció támogatása
- biztonságos kommunikáció támogatása
- mobilitás megoldott

13.5 NDP – Neighbor Discovery Protocol

- IPv4-nél megismert **ARP**, **ICMP Router Discovery**, **Redirect** egy protokollban
- szomszédok és routerek felderítése ICMPv6 üzenetekkel
- **automatikus címkonfiguráció** (stateless autoconfiguration)
    - nem kell DHCP (stateless), ha mégis, akkor **stateful**
- duplikált címek észlelése

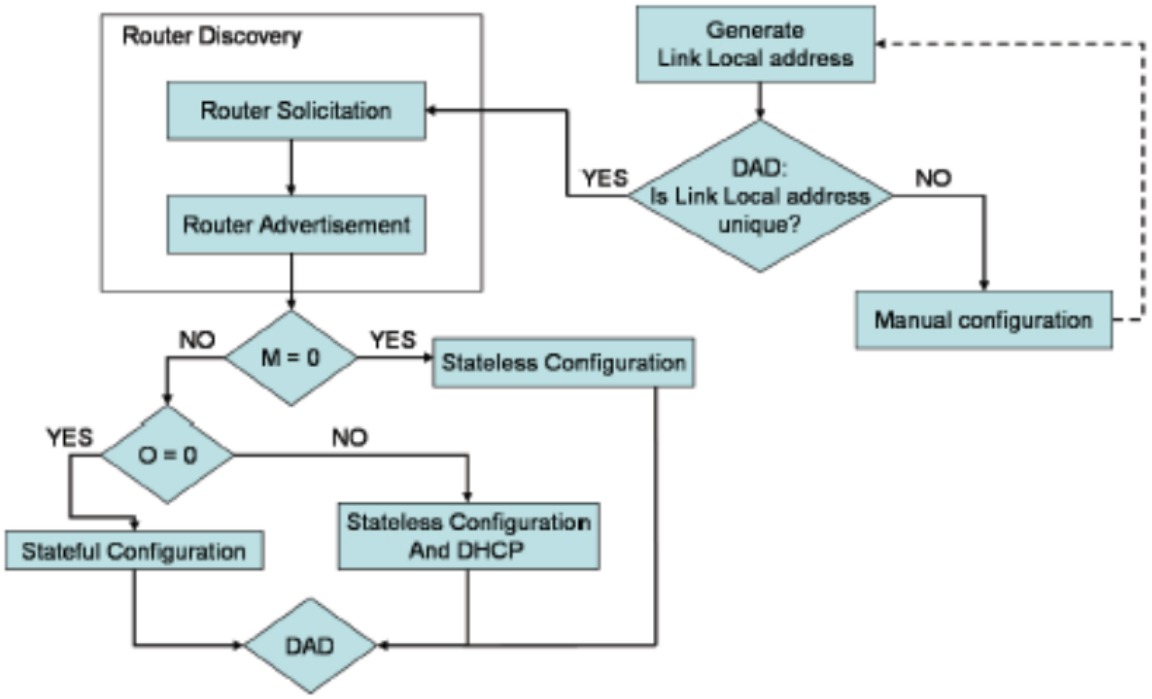
13.5.1 ICMPv6 üzenetek

- **Router Advertisement** (router hirdetés)
    - routerek terjesztik a környezetükben

    - host automatikusan kiépíti, nem manuálisan, mint IPv4-ben
    - link prefix, MTU terjesztése
- **Router Solicitation** (router kérelmezés)
- **Redirect** – routerek megmondják a jobb útvonalakat a hostoknak
- **Neighbor Solicitation** (szomszéd kérelmezés)
    - **ARP** helyett szomszédok adatkapcsolati címének lekérésére
    - elérhetőség/elérhetetlenség megállapítása
    - cím duplikáció felderítése
- **Neighbor Advertisement** (szomszéd hirdetés)
    - pl. új adatkapcsolati cím hirdetése

13.5.2 Állapotmentes automatikus címkonfiguráció

- **duplikáció** ellenőrzése
- **link-local multicast** típusú **router kérelmezés**
- **paraméterek hirdetése** routerektől
- **cím**: router prefix + link local cím
- (ha nem jó az alkalmazásnak, lehet stateful → DHCPv6)



34. ábra Stateless autoconfiguration folyamata

13.5.3 Szolgáltatásokat lehetővé tevő eszközök

- **okos routerek**
    - DHCP funkcionalitás: IP-osztás koordinálása, hálózati beállítások nyilvántartása
- **anycast címzés**
    - szolgáltatás elérésére
- **multicast címzés**
    - azonos szolgáltatásokat nyújtó egységek egymás közötti kommunikációjára

13.6 IPv6 routing

- többféle routing protokoll:
    - RIPng
    - OSPFv3
    - IS-IS Extensions for IPv6
    - EIGRP for IPv6
    - MP-BGP

13.7 Migráció és együttélés

- **pillanatnyi helyzet**
    - 2008-ban volt 10 éves az IPv6
    - 2012. novemberében internetes forgalom 1%-át tette ki
    - 2013. végére a hálózatok több, mint 16%-a támogatja az IPv6-ot
    - lassú terjedés okai:
        - kevés IPv6 szolgáltató
        - kevés kompatibilis szolgáltatás és alkalmazás
        - (nincs igazi kényszer)
    - nagy lökések az áttéréshez:
        - Pekingi olimpia (2008)
        - 4G mobil cellás rendszerekben hang **VoIPv6**

13.7.1 IPv6 alkalmazások napjainkban

- hálózatban: routing protokollok, DNS, DHCPv6…
- operációs rendszerekben (ill. ennél újabbaknál): XP SP1, Mac OS X 10.2, Linux, Redhat, Solaris, FreeBSD…
- tűzfalakban: ip6fw (FreeBSD), E-Border…
- alkalamzásokban: Java IPv6 (JDK1.4 óta), 3GPP…

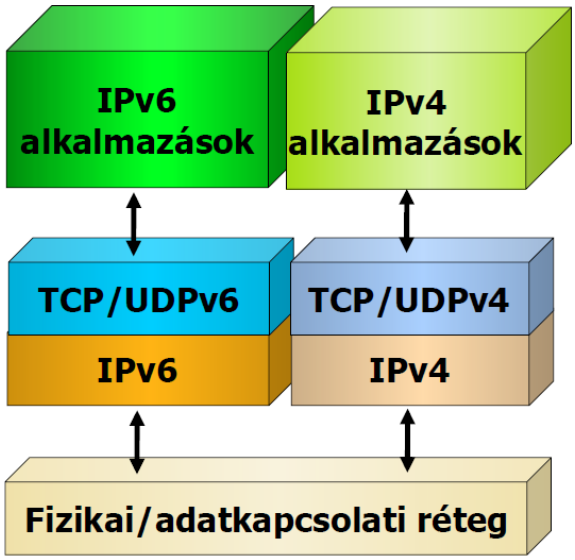
13.7.2 IPv6 kompatibilis IPv4 címzés

- **compatible**: 96 db 0 bit után IPv4 cím
- **mapped**: 80 db 0 bit után **16 db 1-es** majd IPv4 cím ← ezt használjuk
- menet közbeni címfordítás

13.8 IPv6 áttérés lehetőségei

13.8.1 Dual Stack

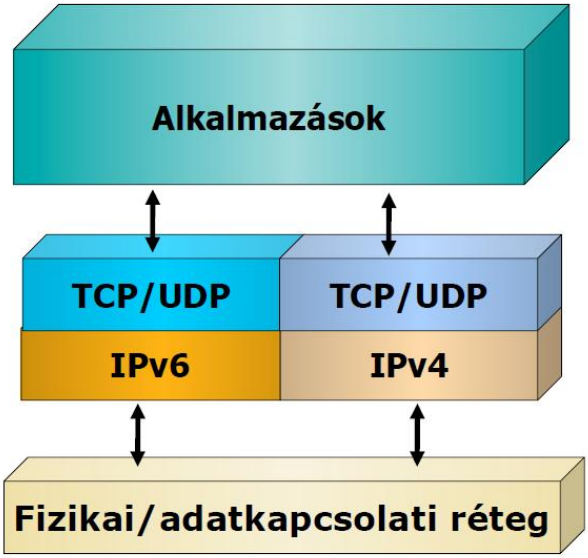
- csomópontok egyszerre IPv4 és IPv6 csomópontok
- IPv6 olyan kommunikáció esetében, ahol lehetséges, de ha szükséges, vissza tud lépni IPv4-be régebbi rendszerekhez
- DNS szerver IPv4 vagy IPv6 címmel válaszol – protokollválasztás módja
- legtöbb IPv6 implementáció ezt használja



35. ábra Dual Stack felépítése

13.8.2 Dual Layer

- csomópontok **egyszerre** IPv4 és IPv6 csomópontok
- a szállítási réteg megmarad
- így több alkalmazás működne, ha nem használ IP címeket



36. ábra Dual Layer felépítése

13.8.3 Tunelling

- egyik protokoll változatú csomag beágyazása a másikba
- **IPv4 hálózaton IPv6 csomagokat** szállítunk
  - végpontok már IPv6 képesek, de a hálózat még nem
  - 6to4 – IPv4-es végpont címe az IPv6-os cím hálózati prefixében → elterjedt
  - 6in4 – statikus alagút-végpont konfiguráció
- **IPv6 hálózaton IPv4 csomagokat** szállítunk
  - hálózat már IPv6 képes, de a végpontok még nem

14 2014.04.09 – IPv6 transition

14.1 IPv4 – IPv6 áttérés jellemzői és eszközei

- NCP→TCP/IP áttérés még egyszerű volt az akkori fejlettség mellett
- ma lehetetlen „óraütésre” áttérni IPv6-ra
    - több milliárd csomópont
    - sok alkalmatlan alkalmazás
    - pénzügyi, gazdasági káosz stb.
- **hosszú idejű átmenet** a két protokoll párhuzamos együttélésével
    - bizonyos vendorok nem is fogják megoldani az IPv6 kompatibilitást
    - régi eszközökhöz a felhasználók ragaszkodhatnak
- meg kell oldani az **együttműködést**
    - sok alkalmazásunk szerver-kliens üzemmódban működik
        - milyen protokollokat beszél a kliens és a szoftver, milyet a hálózat?
        - ha az alkalmazások képesek Dual Stack működésre, a kommunikáció megoldott – feltéve, hogy a hálózat alkalmas rá!
        - (kevés szabad IPv4 cím miatt Dual Stack Lite kerül előtérbe)
    - IPv6 kliens IPv4 környezetben, IPv6 képes szerverrel
        - erre jó megoldás a **6to4 használata**
    - IPv6 kliens IPv4 képes szerverrel
        - **DNS64 + NAT64** átjáró használata
    - IPv4 kliens és IPv6 szerver
        - megoldást jelenthet a **NAT46 + DNS46** használata
    - IPv6 kliens és IPv6 szerver, de útközben csak IPv4
        - IPv6 szigetek összekötésére **6in4 tunnelling**
    - IPv4 kliens és IPv4 szerver, de útközben csak IPv6
        - IPv4 datagramok szállítása IPv6 fölött (**4in6 tunelling**)

14.2 Kitérő - hálózati címfordítás

- **alapprobléma:** egy szervezet belső hálózatában privát IP-címek vannak, de el kell érni bentről az internetet és az internetről is szükség lehet elérni a belső hálózatban lévő klienseket → a külső kommunikációhoz **címfordítást** használunk
    - rendelkezésünkre áll egy router, aminek van publikus IP-címe

14.2.1 SNAT – Source NAT

- a publikus routernek nem elég csak a privát IP-címet visszakövetnie, a TCP/UDP esetén a forrás portszámokat is nyilván kell tartania
    - ezek a portok **gépenként egyediek**
- a router egyedi portszámokra cseréli ezeket, majd kapcsolatonként nyilvántartja **mit mire** cserélt

14.2.2 DNAT – Destination NAT

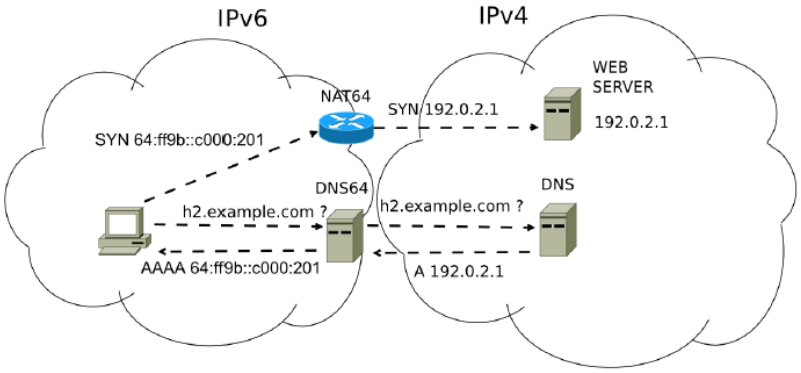
- más néven port forwarding
- a router adott portjára érkező üzeneteket a privát IP-című gépnek továbbítja úgy, hogy a célcímet kicseréli a csomagban

14.2.3 ICMP üzenetek kezelése

- hibaüzenetben megtalálható az azt kiváltó TCP/UDP adategység első 64 bitje portszámokkal
- esetleg az ICMP valamilyen azonosító mezőjét használják

14.3 DNS64 + NAT64 és egyéb technikák

14.3.1 IPv4-only kliens + IPv4-only szerver



37. ábra NAT64 megoldás

14.3.2 DNS64 szerver

- recursive query-kre válaszol
  - ha van IPv6 cím → átadja
  - ha nincs IPv6 cím → IPv4 alapján generál egyet és azt adja át
- speciális IPv6 cím – 64::ff9b::/96 + utolsó 32 biten kapott IPv4 cím

14.3.3 NAT64 használata

- 64::ff9b::/96 – IETF iránymutatása szerinti **well-known prefix** ← problémás lenne világszerte ezt a prefixet használni
- kliens oldalon névkiszolgálóként DNS64 szerver szerepel
- a well-known prefix felé az út egy NAT64 átjárónkeresztül vezet
- folyamat:
  - IPv6 kliens csatlakozni szeretne az IPv4-only szerverhez
  - lekéri a szerver IPv6 címét a szimbolikus neve alapján
  - megkapja az IPv4 címet tartalmazó speciális IPv6 címét
  - TCP SYN szegmenst tartalmazó IPv6 csomagot küld a megadott címre
  - ami áthalad a NAT64 átjárón
  - ami IPv4 csomagot készít belőle úgy, hogy
    - a célcím az IPv6 cím utolsó 32 bitje
    - a forráscím a NAT64 átjáró IPv4 címe
  - az átjáró a csomagot elküldi a címzettnek
  - a szerver megkapja az IPv4 csomagot, amire szabványosan válaszol (SYN+ACK)
    - a válasz címe a NAT64 átjáró IPv4 címe, ami eddig forráscím volt
  - a NAT64 átjáró megkapja a választ, amiből IPv6 csomagot készít:
    - forráscím a DNS64 által generált speciális IPv6 cím lesz
    - célcím az IPv6-only kliens címe lesz
    - a kapcsolattábla alapján tölti ki ezeket
  - a NAT64 átjáró elküldi a csomagot a címzettnek, aki veszi azt
  - és így tovább…

14.3.4 IPv4-Embedded IPv6 addresses

- IPv4-Converted IPv6 Address (IPv4 cím képviselete IPv6 hálózatban)
- prefix szigorúan 32, 40, 48, 56, 64 vagy 96 lehet
- az IPv6 cím 64-71 biteknek 0-nak kel lenniük

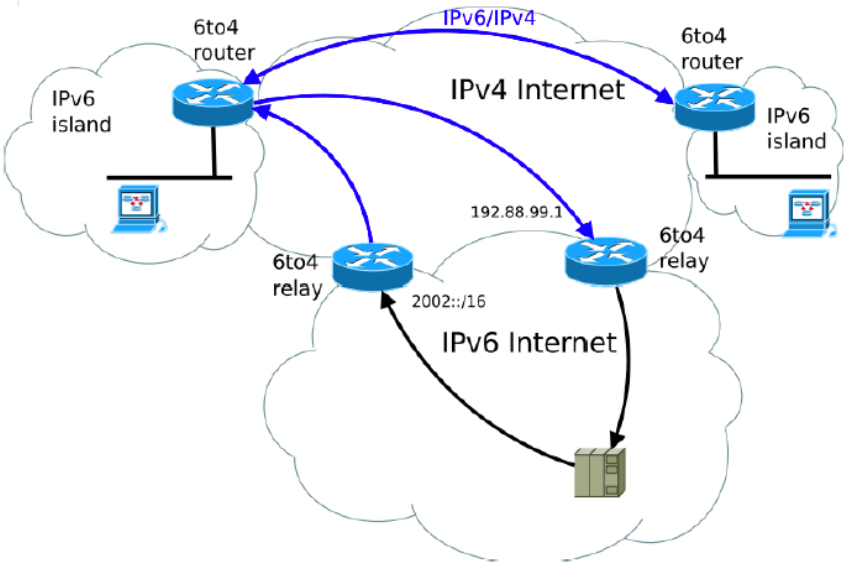
- IPv6 cím 32 bitjét a prefix után szoktuk írni
- cím végén ugyancsak 0 értékű bitek

14.3.5 6in4 tunnelling

- IPv6 szigetek csak IPv4 hálózaton keresztül tudnak kommunikálni egymással
- IPv6 csomagokat IPv4 csomagokba csomagoljuk, amiknek be- és kicsomagolását az IPv6 szigetek határán lévő átjárók végzik

14.3.6 6to4 tunelling

- IPv6 képes eszköz IPv4-only környezetben van és egy másik IPv6 eszközzel kíván kommunikálni IPv6 protokollal
- automatikus tunelling megoldás
- becsomagolást végezheti a **host** (6to4 host) vagy a **router** (6to4 border)
- kicsomagolás a **6to4 relay** feladata – 192.88.99.1 anycast címen
- folyamata:
    - kliens IPv6 csomagot küld a szerver felé
    - a küldött csomagot becsomagoljuk IPv4 csomagba (host/router elvégzi) és elküldi IPv4 használatával az általa választott 6to4 relaynek
    - a relay kicsomagolja, továbbítja a natív IPv6 hálózatba
    - a címzett válaszol
    - relay továbbítja az IPv4-be csomagolt IPv6 csomagot a küldő eszköznek
    - az eszköz kicsomagolja és IPv6 csomagot ad át a címzettnek
- működéséhez kell, hogy kliensnek legyen publikus IPv4-címe
- 6to4 használatával IPv6 szigeteket is összeköthetünk IPv4 fölött
- 6to4 címzéshez a speciális **2002::/16 prefixet** foglalták le
    - hálózatcím: 2002::/16 prefix + publikus 32 bites IPv4 cím + 16 bites Subnet ID (host esetén random szám)
    - gépcím: szabványos, módosított EUI-64 azonosító



38. ábra 6to4 működése

15 2014.04.15 – Forgalomszabályozás

- hívásvezérlés, routing, címzés, ütemezés újabb funkcióval bővül – forgalomszabályozás
- két rokon **feladat megkülönböztetése**:
    - **forgalomszabályozás** (flow control)
        - módszerek, melyek lehetővé teszik, hogy az **adatforrás aktuális sebességét** a vevőnél illetve a hálózatban rendelkezésre álló kiszolgálási sebességhez illesszük
    - **torlódásvezérlés** (congestion control)
        - linkek, csomópontok időszakos túlterheltségének megszüntetésére szolgáló módszerek
        - kialakuló torlódások **megelőzésére** szolgáló módszerek
        - a flow control tekinthető a congestion control eszközének
- átvitel során (forrás → nyelő) kialakulhatnak **szűk keresztmetszetek** (bottleneck)
- alapvető **cél**, hogy:
    - **egyszerű**en megvalósítható **legyen**
    - **lehető legkevesebb** hálózati erőforrás felhasználását igényelje
    - **ne függjön** a hatékonysága a **források számától**
    - biztosítson **igazságos részesedést** az igénybe vett erőforrásokon a forrásoknak
    - **stabil** legyen a működése

15.1 A forgalomszabályozás fajtái

- nyílthurkú – nincs visszacsatolás
- zárthurkú – van visszacsatolás
- hibrid – a kettő kombinációja

15.1.1 Nyílthurkú szabályozás

- kommunikáció előtt felhasználó és hálózat **forgalmi paramétereket egyeztet**
- hálózat dönt az új összeköttetés elfogadásáról (**admission control**)
- ennek megfelelően **erőforrásokat dedikál**
- működése során **ellenőrzi a paramétereket**
- **forgalomleírók:** egy paraméterkészlet, ami jellemzi az adatforrás viselkedését továbbá alapját képezi a szolgáltatási szerződés forgalmi részének (pl. csúcs-, átlagsebesség)
    - **bemenő** adatai a szabályozónak (regulator), aki késlelteti valamint az felügyelőnek (policer), aki eltávolítja a túlzott forgalmat

15.1.2 Zárthurkú szabályozás

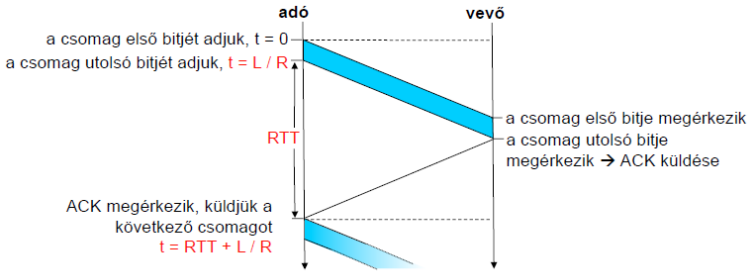
- feltétlenül szükséges, ha nincs erőforrás-foglalás
- túlfoglalást (overbooking) alkalmazunk a nyereség elérése céljából

1. generáció (csak a nyelő képessége)	ki – bekapcsolás (on-off)		
	stop-and-wait		
	statikus ablak (static window)		
2. generáció (a nyelő és a hálózat képessége)	állapot vizsgálat	vezérlés módja	vezérlés helye
	explicit	din. ablak	végpont
	implicit	din. seb.	lépések

39. ábra Zárthurkú szabályozás típusai

- **on-off:** a vevő engedélyezi az adást
- **stop-and-wait:** a küldő egy csomag után a nyugtára vár
- **statikus ablak:** a küldő az ablak méretével megadott számú csomag elküldése után vár csak nyugtára

15.1.3 Stop-and-wait működése

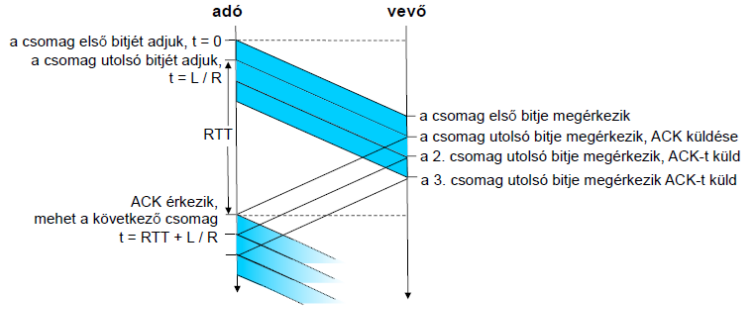


A kihasználtság: $U = \frac{L / R}{RTT + L / R}$

RTT: round-trip-time (teljes körbefordulási idő)

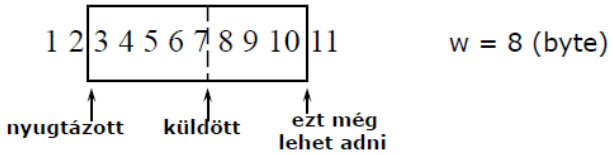
15.1.4 Statikus ablak működése

- az adónak lehet több, adásban lévő, még nem nyugtázott csomagja
- meg kell oldani a **sorszámozást** és a **tárolást az adóban**
- nagyobb kihasználtság stop-and-wait-hez képest



$U_{sender} = \frac{3 * L / R}{RTT + L / R}$

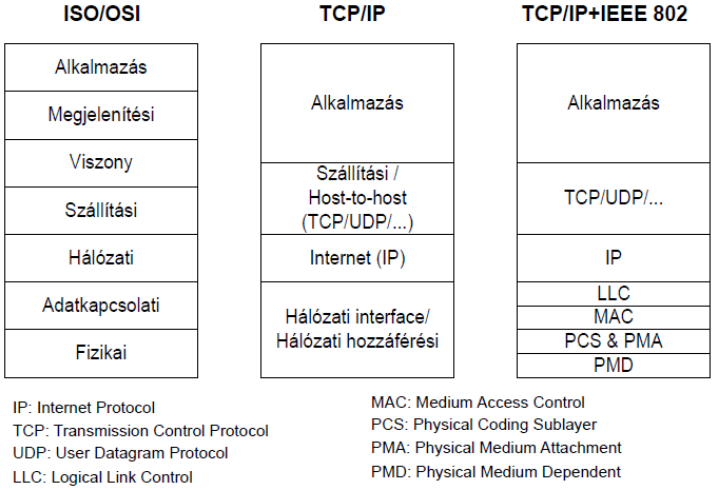
15.1.5 Csúszóablakos (sliding window) szabályozás



Várakozás ACK-ra: **time–out**

- Mekkorára válasszuk a time–out-ot?
- Probléma a túl kicsivel és a túl naggyal
- Megoldás: a teljes körbefordulási időhöz (RTT) igazítani
  - **adaptívvá** tenni
- Szabályok arra, hogy mit tegyünk, ha nem jön ACK a time-out alatt

16 2014.04.15 – Szállítási (Transport, Host-to-Host) protokollok



40. ábra OSI és TCP/IP összehasonlítása

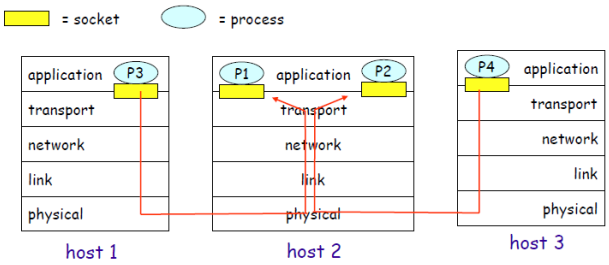
- **hálózati réteg:** végpontok közti logikai kapcsolatok (host-to-host)
- **szállítási réteg:** alkalmazások közti logikai kapcsolatok (process-to-process)
    - **hálózati réteg** szolgáltatásaira építve megbízható átvitel a **transzport entitások** között
    - **feladatai:**
        - flow control
        - multiplexelés, demultiplexelés
        - hibadetektálás és –javítás
        - sorrendhelyes átvitel
        - csomagképzés és csomagok visszaállítása más rétegek számára
- szállítási protokollok a **végpontokban** futnak, nem a hálózat köztes csomópontjaiban
- alkalmazások adategységeit protokoll-adategységekbe (PDU) tördeljük és a kapottakból összerakjuk őket

16.1 UDP (User Datagram Protocol) és TPC (Transmission Control Protocol) általánosságban

- **közös képességük:** portkezelés, multiplexelés
- **eltérő képességük:** UDP **összeköttetés-mentes**, míg TCP **összeköttetés alapú**

16.1.1 Portok kezelése

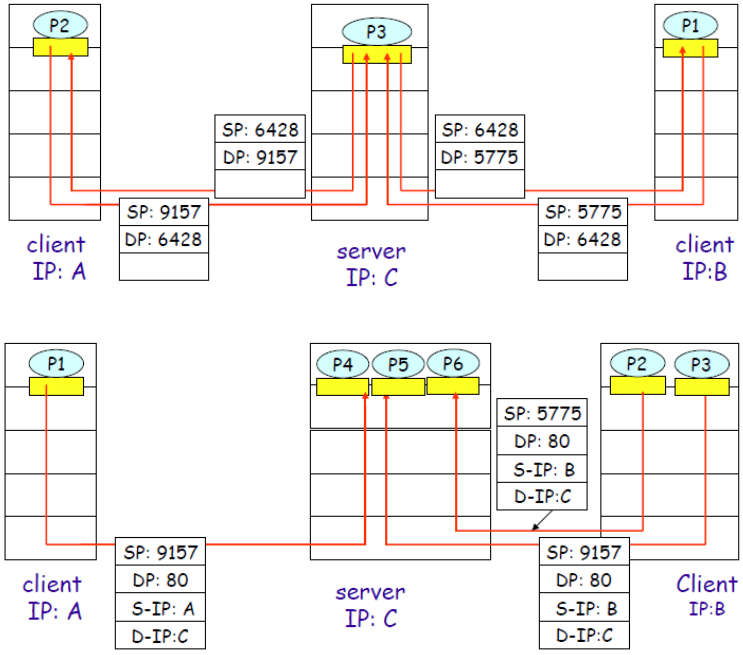
- IP rétegben a csomagok „hostoknak” vannak címezve, amin belül több alkalmazás is lehet
    - megkülönböztetésük portok szerint történik → **well-known** (0..1023) és **rendelkezésre álló** portok szerint, pl. 21: FTP, 80: http…
- UDP-ben megállapításra kerülnek az alkalmazandó portszámok → multiplexelés/demultiplexelés alapja



41. ábra Multiplexelés/demultiplexelés

16.1.2 Socketek

- a socket egyfajta „ajtó” az alkalmazás és a hálózat közt
- leegyszerűsítve socket = IP-cím + portszám
- jellemzői – ún. socket-pár
  - transztportprotokoll (TCP/UDP)
  - saját IP-cím
  - saját portszám
  - (opcionális) távoli host IP-címe
  - (opcionális) távoli host portszáma
- az operációs rendszer a fentiek alapján továbbítja az alkalmazások felé a csomagokat PDU alapján
- **socket típusai:**
  - datagram socket – UDP használja, connectionless
  - stream socket – TCP használja, connection-oriented
  - raw (IP) socket – routerekben és hálózati eszközökben, pl. ICMP, IGMP



42. ábra Mux/DeMux UDP (fent) és TCP esetben

16.1.3 TCP kliens-szerver socket kezelés

- szerver **hallgató** módban létrehoz socketeket → várja a kapcsolatot
- kapcsolatfelvétel után **dedikált** socketet alakít ki
  - socket-socket közt **virtuális áramkörkapcsolás**, TCP, duplex bytefolyam

- különböző socketek hozhatók létre egyazon IP-címmel és portszámmal, mivel más-más párt fog alkotni a távoli socket IP+port párosával
    - szerver oldalon gyerek process összerendelése kliens processzével
- **UDP-ben nincs dedikált process** → nincs gyerek process minden távoli processzhez, egy processz kommunikál velük
- ahol nincs implementálva a transzport réteg (router – hálózati réteg, switch – adatkapcsolati réteg), ott a tűzfalak, NAT-ok stb. figyelik az aktív socket párokat és fenntartanak nekik interfészeket
- ütemezéshez és QoS támogatáshoz a csomagokat socket-párokkal azonosíthatjuk

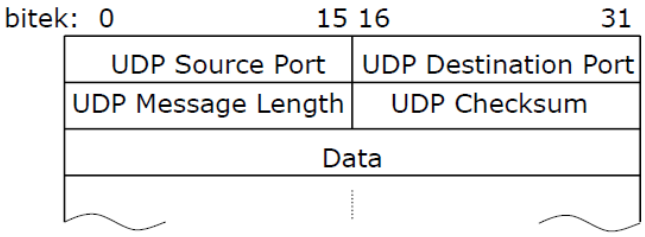
16.1.4 Raw socket

- közvetlenül az **alkalmazások** továbbítják a csomagot és látják el fejléccel
- nincs TCP/IP feldolgozás
- biztonsági aggályok – TCP reset támadástól való félelem
    - harmadik fél beékelődve megszakíthatja a kapcsolatot, peer-to-peer forgalom szűrésének következménye stb.

16.2 UDP

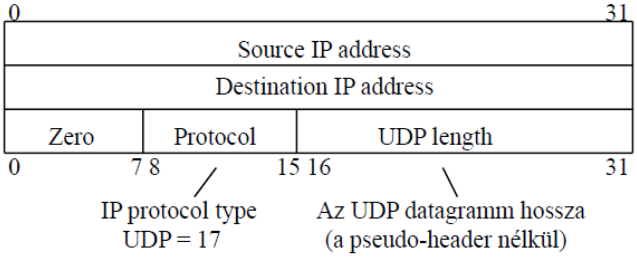
- nincs kapcsolatfelépítés és handshake
- **nincs garancia:** sorrendiség, adatintegritás, elvesztének detektálására
- megbízható átvitel garantálást az alkalmazásra bízza
- multiplexitást, opcionális integritás-ellenőrzést nyújt

16.2.1 UDP felépítése



- **Source port** opcionális
- **Length**: oktettben, max (65 535-8-20)
- **Checksum**: opcionális (nincs: 0, IPv6-nál már nem opcionális)
- Az UDP checksum az egyetlen lehetőség annak ellenőrzésére, hogy a datagram helyesen érkezett-e meg
    - **adatra és fejlécre!**
    - az IP-csomag ellenőrzése csak a fejrészt fogja át

- A checksum számítási elve
    - 1-es komplemens 16 bites szavakra
- Tartalmazza az IP-címeket is
    - annak ellenőrzésére, hogy a datagramm elérte a helyes címzettet, nemcsak a helyes portot
- „Pseudo-header" hozzáadásával (IPv4-re):



- portkezelés: különböző alkalmazások megkülönböztethetőek

- több alkalmazás egyidejű kezeléséhez **port-hozzárendelés** és **mux/demux**
- hibajelzés az UDP datagram tartalmára és az IP csomag további részeire
- **ehhez mind kell minimum 8 oktettnyi overhead**

16.2.2 UDP alkalmazása

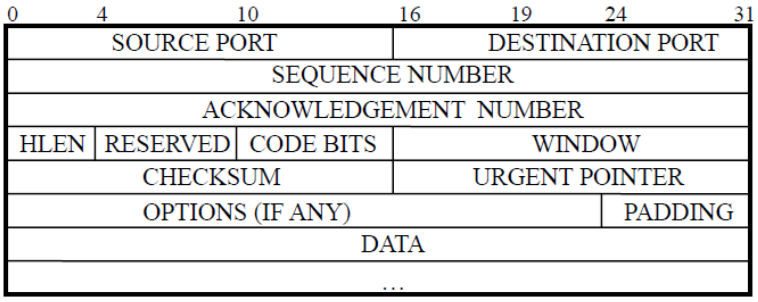
- broadcast és multicast (TCP nem tudja)
- streaming, valós idejű játékok, VoIP, IPTV
- gyors lekérdezésekhez: DNS, DHCP, RIP
- hálózati forgalom pár százalékát teszi ki, de növekszik → aggályos, mert nincs congestion control (megoldás: DCCP – Datagram Congestion Control Protocol)

16.3 TCP

- **megbízható szállítás** nyújtása az IP nem megbízható datagram-szolgáltatásán
- **jellemzői:**
  - **virtuális összeköttetés** alapú – felépül és fent marad a kapcsolat során
  - **stream-típusú szolgáltatás** – byte- stream-ek helyes átvitele
  - **strukturálatlan stream** – nincsenek határolók a streamen belül
  - **pufferelt átvitel** – a datagram megtöltéséhez szükséges mennyiséget összevárja
  - **duplex kapcsolatok** – két független stream
  - **vezérlő információk beágyazása** – ellenkező irányban folyó stream-be ágyazva

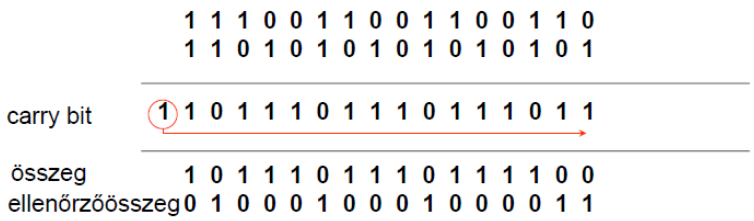
16.3.1 PDU felépítése

- TCP-ben szegmensnek hívjuk a PDU-t



43. ábra Szegmens felépítése

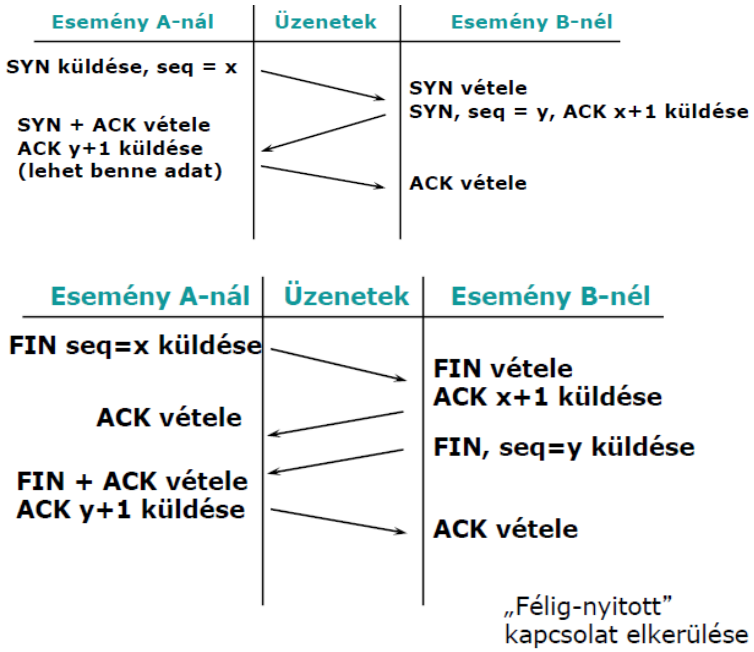
- **Sequence no.:** a szegmensben lévő adat első btye-jának helye a külső stream-jében
- **Ack. no.:** a forrás által legközelebb várt byte sorszáma
- **HLEN:** fejléc mérete; **minimum 20** byte, **maximum 60 byte**
- **Code bits:** a kapcsolat kezeléséhez használt jelzőbitek (SYN, ACK, RST, FIN, URG, PSH)
- **Window:** küldő vételi pufferének mérete
- **Checksum:** ugyanaz, mint az UDP-ben
- **Urgent pointer:** URG=1 esetén „urgent” részt tartalmaz a szegmens (pl. jelszóküldés), ilyenkor a végére mutat



44. ábra Checksum összetétele

16.3.2 Hívásfelépítés és –lebontás TCP-ben

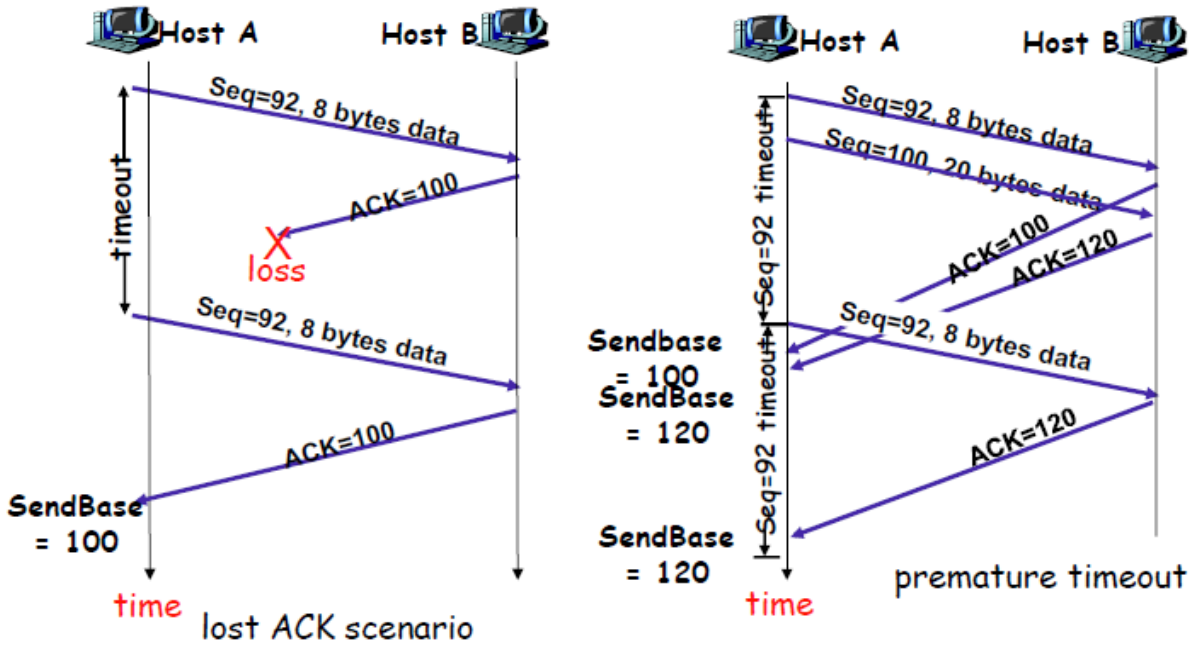
- felépítésnél **three-way handshake** (elején véletlenszerű sorszámmal a támadás elkerülésére), lebontásnál **modified three-way handshake**



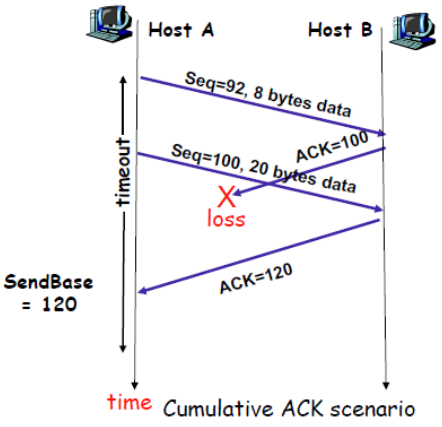
- ACK-ra várakozás (time-out) **adaptív** kell legyen és a **RTT**-hez (round-trip time) kell igazodjon

16.3.3 ACK-val kapcsolatos problémák

16.3.3.1 Elveszett ACK és korai timeout



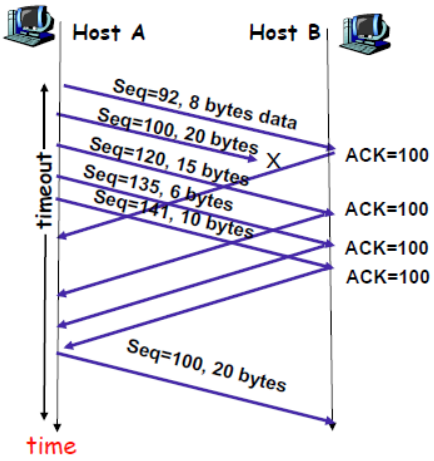
16.3.3.2 Összevont nyugta



- **gond az összevont nyugtával:**
  - pl. 1000 byte küldése 10 szegmensben → ha az első szegmens elvész, nem tudja a fogadó jelezni, hogy csak a 0-99 hiányzik ➔ újra kell küldeni az 1000 byte-ot
  - megoldás: **szelektív nyugtázás** (SACK)
    - népszerű, minden TCP implementációban
    - opcionális fejrészmezőben
    - meg tudja mondani, hogy mely szegmens érkezett meg
- **gond a sorrend keverésével:**
  - megoldás: **D-SACK** (duplikált SACK) – fogadó szól, hogy az újraküldött csomag duplikátum

16.3.4 Fast retransmit

- a **timeout** idő gyakran **túl hosszú** → minél később lehet újraadni az esetleg elveszett csomagot
- az elveszett szegmensekre utalhatnak a duplikált ACK-ok
- ha a vevő hézagot vesz észre a vett szegmensek sorozatában, akkor megint lenyugtázza az előzőleg helyesen vett szegmenst
- **fast retransmit szabály:** ha az adó 3 egymást követő ACK-ot kap (+ eredeti ACK), akkor feltételezhető a szegmens elvesztése, így még timeout előtt újra tudja adni

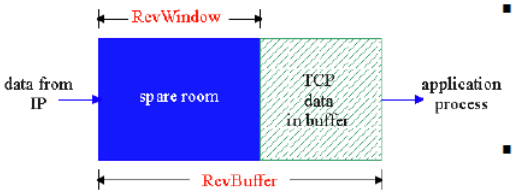


45. ábra Fast retransmit működése

16.3.5 Forgalomszabályozás – „sliding window” elve

- TCP-n byte-ok helyett oktettekkel működik
- működéshez lásd 15.1.5 fejezetet

16.3.5.1 TCP flow control – hogy működik?



- Spare room:

= `RcvWindow`

= `RcvBuffer-[LastByteRcvd - LastByteRead]`

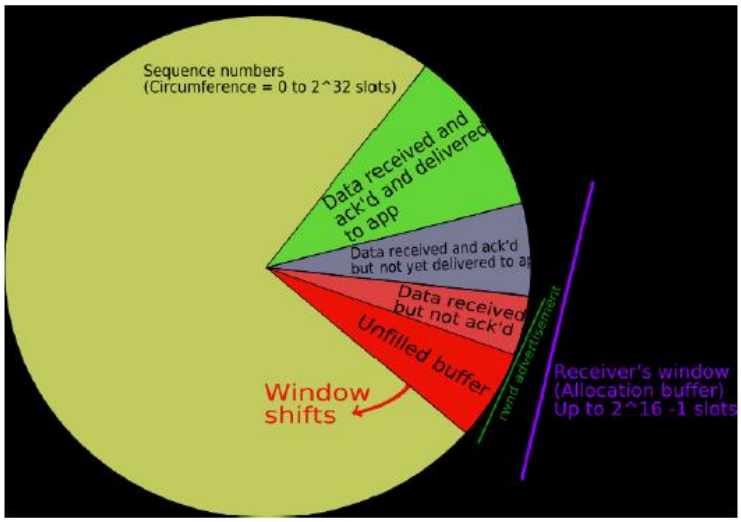
- A vevő közli a szabad helyének a méretét (`RcvWindow`) a küldött szegmensben
- Az Adó legfeljebb `RcvWindow` mennyiségű nyugtázatlan adatot küld
- Nem szabad összekeverni a torlódási ablakkal (**CongWin**)

16.3.5.2 MSS (Maximum Segment Size)

- a maximálisan **egy szegmensben küldhető** adatméret **byte-ban**
- össze kell egyeztetni az adatkapcsolati réteg MTU-jával → TCP kapcsolat felépítésénél MSS opció a fejlécben
- TCP adó használhat Path MTU Discovery-t is → dinamikus MSS változás lehetősége

16.3.5.3 Csúszóablakos forgalomszabályozás problémái

- nullás csúszóablaknál leáll az adás → egy számláló lejárta után a vevőtől megkérdezzük újra a czúszóablak méretét
- ún. **buta ablak jelenség**: a vevő oldalán kicsi (akár 1 byte) felszabadulás esetén az ablak mérete nagyon kicsi lesz (kisebb, mint a fejléc)
    - megoldás: a vevő nem nyitja meg az ablakot, ha az kisebb, mint az MSS → erőforrás pocsékolás elkerülése
        - küldő nem küld, hacsak nem MSS-nyit küld vagy mindent küldhet, amit az alkalmazás kért



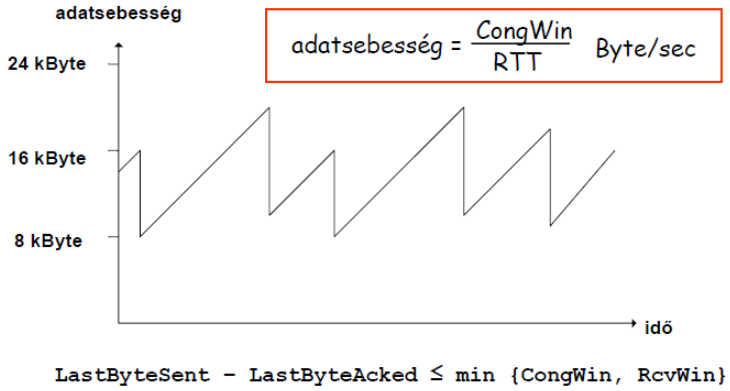
46. ábra TCP "óra"

16.3.6 Torlódásvezérlés a TCP-ben

- torlódásvezérlés fogalmához lásd 15. fejezetet
- két fő módszer:
  - **hálózat által segített** – network assisted: a hálózat elemei szolgáltatnak információt a torlódásról (pl. TCP/IP ECN, ATM – külön jelzéscsomagokkal)
  - **végpontok közti** – end-to-end: nincs visszacsatolás a hálózatból, a végpont következtet a torlódás lehetőségére ← **ezt csinálja a TCP is!**

16.3.6.1 TCP módszere

- **növeljük az adatsebességet**, amíg van elég throughput
- ha **torlódásra utaló jeleket észlelünk**, **csökkentsük** a sebességet
  - **congestion window** (CongWin) alkalmazása: minden RTT alatt MSS-sel növeljük az ablak méretét, amíg problémát nem észlelünk; ekkor felére csökkentjük a CongWin-t
    - **additive increase, multiplicative decrease (AIMD)**
- észleléshez segítség: **nem jött nyugta**, de letelt a time-out, esetleg **több nyugta** érkezett **ugyanazon szegmensre**



47. ábra Adatsebesség alakulása AIMD esetén

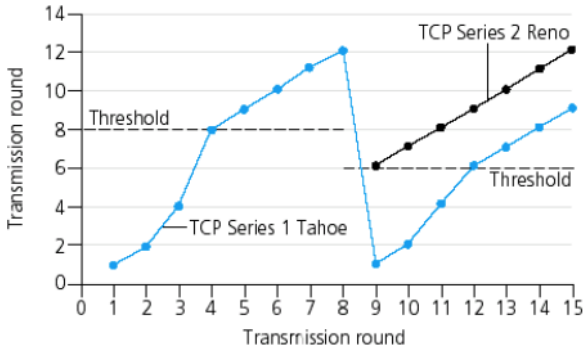
- „slow start” módszer: kapcsolat felépítése után exponenciális sebességnövelés az első ütközésig, aztán AIMD
  - minden ACK után kétszerezzük a sebességet
  - eltérő viselkedés timeout és többszörös nyugták esetén
  - elején pl. CW = 1 MSS (500 byte) és RTT = 200 msec, kezdeti sebesség = 20 kbps → mivel az átbocsátó képesség nagyságrendekkel nagyobb lehet a kezdeti sebességnél, ezért növeljük exponenciálisan az első ütközésig
- **3 D-ACK** esetén: CW felére csökken, majd lineárisan növeljük
- **timer lejárta** esetén: CW 1 MSS lesz, majd exponenciálisan nő egy korlátig, ahonnan lineárisan nő

- Ha a `CongWin` egy korlát alatt van, adó slow-start fázisban, exponenciális ablak növelés
- Ha `CongWin` a korlát felett, az adó torlódás elkerülési fázisban, lineáris ablak növelés
- Ha 3 duplikált ACK, a korlát `CongWin/2` lesz és a `CongWin` pedig a korlát
- Ha timer lejárt, a korlát `CongWin/2` lesz, míg a `CongWin` 1 MSS lesz

48. ábra A különböző esetek összefoglalása

16.3.7 TCP implementációk

- korlát: 8 MSS
- 3 D-ACK a 9. átvitelnél:
    - **Tahoe:** nincs fast recovery, 1 MSS-ről indul újra, exponenciális korlátig, majd lineárisan
    - **Renoe:** CW/2-ről lineárisan



- **Vegas TCP:** előre próbálja jelezni a torlódást RTT alapján (ha kritikus), de Renoe elnyomhatja

16.3.8 TCP átbocsátóképessége

- CW és RTT függvényében átlagos átbocsátó képesség? – nem foglalkozunk a slow starttal
- ablak mérete: W
    - csomagvesztés előtt – W/RTT
    - csomagvesztés után ablak feleződik – W/2*RTT
    - átlagos átbocsátóképesség – 0,75*W/RTT

16.3.9 TCP fairness

- idealizált esetben mindkét kapcsolatnál egyforma MSS és RTT
- nincs más TCP vagy UDP kapcsolat
- AIMD eset
- az átviteli sebesség jelenti a szűk keresztmetszetet

- kisebb RTT → gyorsabb CongWin növelés → nagyobb átviteli sebesség!
- UDP nem fair, kiszorítja a TCP-t
- böngészők több, párhuzamos TCP kapcsolatot építenek fel a webszerverrel → nagyobb sebesség!
    - R sebességgel, 9 kapcsolattal
        - 1 új TCP kérés – a sebesség R/10
        - 11 új TCP kérés – a sebesség több, mint R/2!

Alkalmazás	*Alkalmazási rétegbeli protokoll*	*Használt transzport-protokoll*
E-mail	SMTP	TCP
Távoli elérés	Telnet	TCP
Web-elérés	HTTP	TCP
File-átvitel	FTP	TCP
Routing	RIP	UDP
Hálózatmenedzsment	SNMP	UDP, TCP
VoIP, média-streaming	Többnyire nem szabványos	UDP

49. ábra Pár alkalamzás és transzportprotokolljaik

17 2014.04.17 – Multimédia továbbítása IP felett

- multimédia alkalmazások fő csoportjai:
  - tárolt audio- és video-streaming
  - élő audio- és video-streaming
  - interaktív, valósidejű audio és video
- mind késleltetés-érzékenyek
- viszont az adatvesztésre nem
- vagyis elvárásaik ellentétjei az adatkommunikációban elvártaknak

17.1 Média streaming

- **azonnali adatfolyam** – általában tömörített multimédiás információ, mely az **azonnaliságra** összepontosít, nem a teljes hűségű visszaállítása
- hagyományos médiafájlokkal ellentétben (egyetlen nagy csomagban) itt kisebb csomagokban, folyamatosan
- az elterjedéshez kellett a csomópontok fejlődése, a kapcsolatok sebességének növekedése és a hatékonyabb tömörítési algoritmusok elterjedése

17.1.1 Tárolt mediastream

- a tartalom a **forrásnál** tárolódik, a kliensnél megkezdődik a lejátszás, mielőtt a teljes tartalom letöltődne; lehetőség van előre-visszatekerésre, megállításra stb. → **interaktivitás**

17.1.2 Élőmédia streaming

- rádió-, TV-állomások az interneten
- élő előadások („webinar”)
- élő sportesemények
- lejátszás késleltetéssel, playback-tár
- néhány, tárolt médiánál használt fukció (de pl. fast forward nincs)

17.1.3 Interaktív real-time audió és videó

- VoIP (IP feletti telefonálás)
- videokonferencia
- **igényei:**
  - kis késleltetés
  - kis késleltetés**ingadozás**
  - beszédnél 150-300 ms késleltetés, pár 10 ms ingadozás
- megjelenik a hívásvezérlés

17.2 Beszédjel feldolgozás

- 64 kbit/s-os beszédet 125 µs-onként 8 bites „csomagokra” bontjuk
- detektálnunk kell tudni a szüneteket és inaktivitást/aktivitást, redundanciát, tömöríteni kell
- csomagokká alakítás → mérete függ a beszédszakasz méretétől és a forrássebességtől is

Szabvány*	Kódolási módszer	Bitsebesség [kbit/s]	Bonyolultság	Késleltetés [ms]
G.711	PCM	64	1	0,125
G.726	ADPCM	32	10	0,125
G.728	Low Delay-CELP	16	50	0,625
G.729	CS-ACELP	8	30	15
G.723.1	ACELP	6,3/5,3	25	37,5

- tipikus beszédcsomag méretek (payload méretek) → cél, hogy csomagvesztésnél minél kisebb essen ki
    - 10-20 ms-os beszédszakaszok kerülnek egy csomagba
        - 10 ms, tömörítés nélkül – 80 byte
        - 20 ms, kétszeres tömörítéssel – 80 byte
        - 6 ms, tömörítés nélkül – 48 byte

17.3 Videó tömörítési eljárások

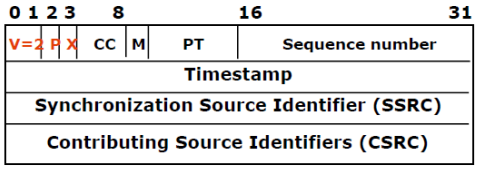
- MPEG-1: VHS minőség, CD minőségű hanggal, 1,5 Mbit/s
- MPEG-2: DVD, digitális TV-adások, 3-10 (SD) vagy 10-20 Mbit/s (HD)
- MPEG-4: tipikusan 1,5 Mbit/s (SDTV) vagy 8 Mbit/s (HDTV), Blu-ray-ekben is

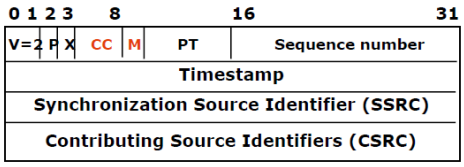
17.4 Médiakezelő protokollok

17.4.1 RTP és RTCP

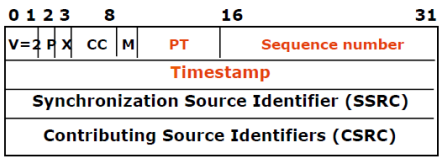
- **RTP:**
    - payloadjában médiainformációt hordoz
    - transzferprotokoll felett működik
    - támogatja a streamek szállítását
    - de nem nyújt QoS-t
- **RTCP:** végpontok közti QoS monitorozás eszköze, streamek közti szinkronizálás
- különböző UDP portokat használnak: RTP párosat, RTCP a következő páratlant
- **RTP** helye Alkalmazási/Szállítási rétegben van
- **RTP** szolgáltatásai: payload **típusok**, **sorszámozás**, **időbélyeg**
- RTP-t használ pl. QuickTime, RealAudio, RealVideo, NetMeeting, Cisco IP/TV

17.4.2 RTP csomagfejrész formátuma



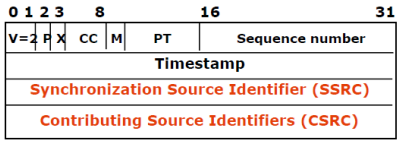


- Version (V, 2 bits)
  Az RTP verziószáma; az RFC 1889-ben definiált módon
- Padding (P, 1 bit)
  Ha 1-es: van padding az RTP csomag végén
  Padding - utolsó byte: hányat kell figyelmen kívül hagyni
- Extension (X, 1 bit)
  Ha 1-es: a fejrész után változó hosszúságú fejrész kiterjesztés.
  Ha van kiterjesztés: első 2 byte a hosszát adja meg
  A kiterjesztés a fix fejrész utolsó érvényes mezője után következik



- Payload type (PT, 7 bit)
  - "profile", amely legtöbbször a médiakódolási típusokat kezeli, azoknak payload-formátumokat feleltet meg
- Sequence number (16 bit)
  - lehetővé teszi az elveszett csomagok detektálását és a csomagsorrend helyreállítását
  - kezdőértéke véletlen szám (l. később); minden elküldött RTP csomag után eggyel növelődik
- Timestamp (32 bit)
  - Az RTP csomag első oktettjének megfelelő pozíció valódi ideje a médiafolyamban

- CSRC count (CC, 4 bits)
  - a CSRC azonosítók száma = a multiplexált források száma (a források megadása: a CSRC mezőben)
  - csak egy forrás: CC = 0
- Marker (M, 1 bit)
  - a csomagfolyam szignifikáns eseményeinek megjelölése
  - példák:
    - kerethatárok a különféle kódolási módszereknél
    - a beszéd aktív időszakainak kezdete/vége
  - az aktuális interpretációt a "profile" adja meg

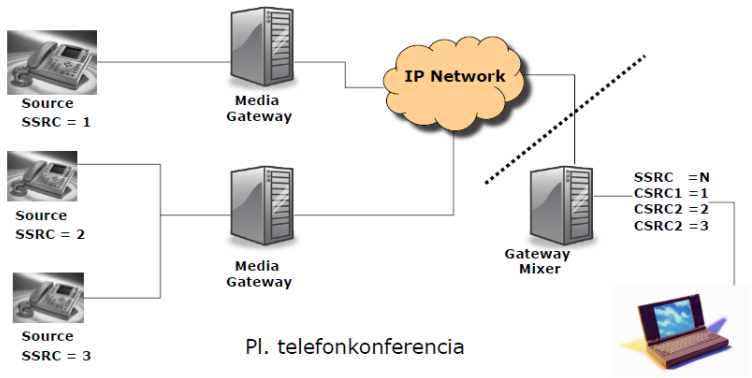


- SSRC (32 bit)
  - Az RTP csomagfolyam forrását azonosítja, az RTCP rendeli hozzá, véletlenszerűen
- CSRC (0…15-ször 32 bit)
  - contributing source: az „RTP mixer" által létrehozott kombinált csomagfolyam komponensét azonosítja

- különböző **payload profile**-ok pl. G.711 – 0, G.722 – 9, MPEG-2 – 33 stb.
- **SSRC:**
    - stream forrását azonosítja, **függetlenül a hálózati címtől**
    - véletlen szám, egyedi kell legyen a session-ön belül
    - több forrás esetén több SSRC generálódik, mindegyik más
    - vevő ez alapján válogatja egy csoportba a csomagokat
    - adatformátumot változtathat időben

17.4.3 RTP Mixer

- közbülső rendszer, mely fogadja az RTP csomagokat egy vagy több forrásból és új RTP csomagot készít belőlük
- megváltoztatja az adatformátumot, kombinálja a csomagokat
- új időzítés a kombinált streamre
- SSRC = Mixer, CSRC = források felsorolva



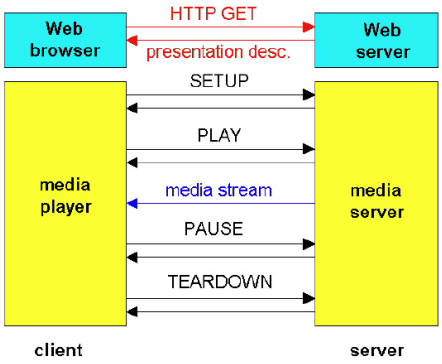
50. ábra RTP mixer működése

17.4.4 RTCP – Real-time Transport Control Protocol

- végpontok közti QoS monitorozása (akár harmadik fél által is)
- **nem** jelzésátviteli protokoll!
- hatására az alkalmazás kodeket válthat, folyamot korlátozhat

17.5 RTSP – Realt-time Streaming Protocol

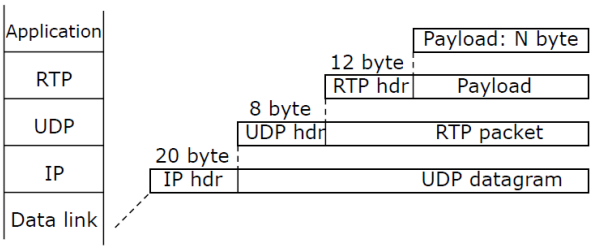
- kapcsolat felépítése és ellenőrzése a végpontok közt
- VCR jellegű funkciók (lejátszásvezérlés)
- RTSP szerverek nagy része RTP-t használ



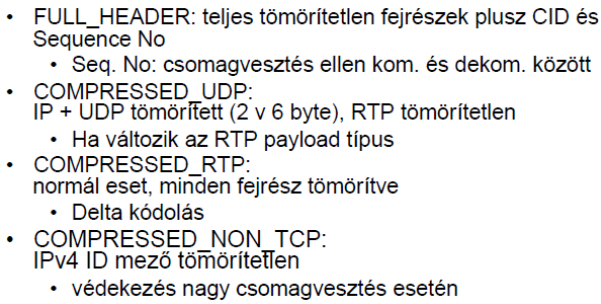
51. ábra RTSP működése

- HTTP-hez hasonló, de **több kéréstípussal** rendelkező, **állapotfüggő** protokoll
    - SETUP: átvitel specifikáció, RTP-RTCP portok meghatározása stb.
    - PLAY: lejátszás (egy vagy több stream)
- **nem** nyújt QoS-t
- **nem** kezeli a médiaformátumokat

17.6 Fejrésztömörítés



- egy médiacsomag hossza 40 + N + link-layer header
- cél az overhead csökkentése → kezeljük együtt mindhárom protokollt! (IP + UDP + RTP)
- arra alapozzuk, hogy a kapcsolat során minimális változás van a fejlécekben
- először tömörítetlenül küldjük az egész fejrészt, majd a tömörítő/kitömörítő megállapodnak a részletekben (formátum, Context Session ID – CID)
    - a CID azonosítja az IP címeket, UDP portokat, RTP SSCR-jét
- csak ezek változnak:
    - IP
        - teljes hossz - tartalmazza a link-layer is, nem kell átvinni
        - header checksum – elhagyható, link-layerre bízzuk
        - identification – inkrementálódik, IPv6-nál nincs
    - UDP: ugyanaz a első kettő, mint az IP-nél
    - RTP:
        - seq. number: inkrementálódik
        - időbélyeg: mintavételi idővel növekszik
        - M bit
        - CSRC lista ritkán kell
- Compressed RTP (cRTP) protokoll



52. ábra cRTP 4 különböző formátuma

    - kis sebességű linkeken érdemes használni, ahol alacsony a hibaarány
    - videokonferencia és IP-telefonoknál

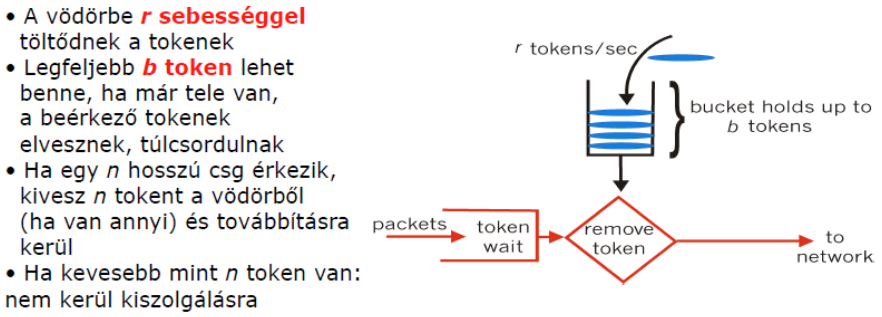
18 2014.04.22 – QoS IP-hálózatokban: túl a Best Effort-on

18.1 Mi a QoS?

- Quality of Service (szolgáltatásminőség) = végpontok közti garanciák
  - rendelkezésreállás
  - „sávszélesség”
  - késleltetés, -ingadozás
  - packet loss stb.
- QoS biztosításának többféle eszköze lehet
  - **csomagkapcsolt hálózatokban**
    - forgalmi méretezés
    - protokollválasztás
    - hálózati architektúra megválasztása, hálózati biztonság
    - tárak menedzselése
  - **IP hálózatokban**
    - nyers erővel (overprovisioning)
    - folyamatonként (per-flow) QoS – **IntServ módszer** (létezik finom felbontású változata is)
    - **forgalomosztály alapú** (class-based) – **DiffServ módszer** (durva felbontású változat QoS folyamosztályokra)

18.2 Integrated Services – IntServ

- szolgáltatásosztályok vannak definiálva
- **Quaranteed Quality**: real-time **intolerant** alkalmazások számára nyújt garantált korlátokat késleltetésre és sávszélességre
- **Controlled Lead**: real-time **tolerant** alkalmazások számára beengedés-szabályozással igyekszik kb. azonos szolgáltatást nyújtani, mintha nem lennének túlterhelve a csomópontok
- **best effort** módszer, ütemezéssel
- ha a hálózat **beengedi** (admission control) a folyamot, akkor **jelzésátviteli protokoll** (RSVP – ReSource reserVation Protocol) segítségével **erőforrás foglalás** történik
- a forgalmat **ellenőrizzük** és **formáljuk**
- csomópontokban **ütemezünk**
- policing és forgalomleírás eszköze a **token bucket** → küldhetünk b méretű burst-öt, de az átlagsebesség csak r lehet

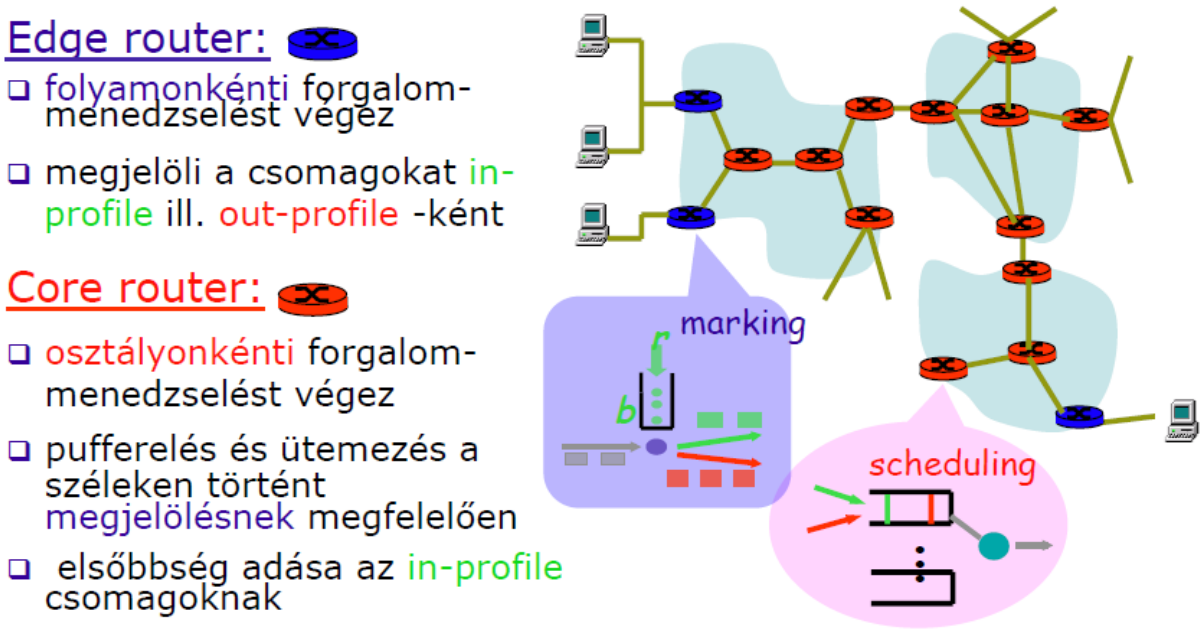


53. ábra Token bucket működése

18.3 Differentiated Services – DiffServ

- alapötlet: a gerinchálózatra kell megoldást találni, ahol nagy és aggregált forgalom megy át

- nem egyedi folyamhoz, hanem **kisszámú forgalomosztályhoz** rendel erőforrásokat (pl. Regular, Premium) → ezt a fejrészben egyetlen bittel is tudjuk jelezni
    - a betállítás történjen **adminisztratív** határokon
    - a router ennek megfelelően kétféle módon reagálhat, kezelhet csomagot



54. ábra A DiffServ működése

- csomagok megjelölése edge routerekben: egyeztetett **r** sebesség és **B** vödörméret
    - a különböző osztályokhoz tartozó csomagokat eltérően jelöljük
    - az osztályokon belül a profilnak megfelelő (**konform**) és nem megfelelő (**nem konform**) eltérő jelölést alkalmazunk
- megfigyeljük a forgalmat és a nem konform csomagokat **alakítjuk** (shaping) vagy **eldobjuk** (dropping)

18.3.1 DiffServ – csomópontok viselkedése

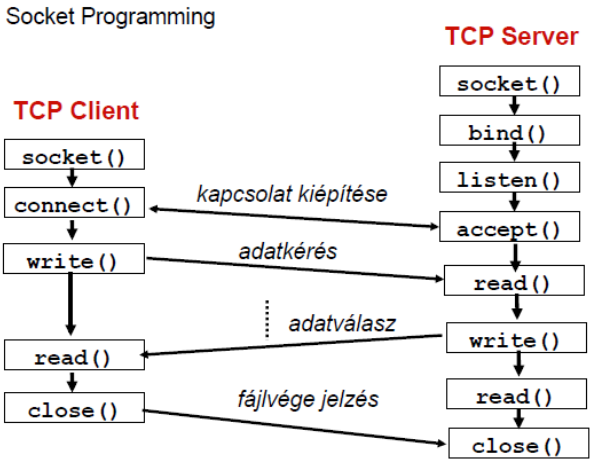
- Per-Hop-Behaviour (**PHB**) – forgalomosztályokhoz tartozó csomagtovábbítási elveket definiál
- nincs együttműködés a csomópontok közt
- IPv4-nél ToS mezőjében 6 bit, IPv6-nál Traffic Class néven
    - DiffServ Code Points (DSCP) – ebből 2 bit torlódásjelzésre
- PHB két típusa:
    - **EF** – Expedited Forwarding – DSCP: 101 110
        - **minimális késleltetés**, **kis** packet **loss**
        - célszerű, ha csak a link sebessége korlátozza az érkezési ütemet
        - szigorú prioritás garantálható („expressz” továbbítás)
        - túl nagy sorbanállás késleltetés okoz
        - szigorú admission controlt igényel
        - késleltetés és jitter leromlásával járhat osztályon belül
        - szolgáltató korlátozza – pl. max. 30%-a lehet a linkkapacitásnak
        - alkalmazása: audió, videó
    - **AF** – Assured Forwarding – DSCP: eltérő módon
        - legnépesebb osztály 12 elemmel

        - négy csoport, bennük 3-3 eldobási stílus
- default PHB pedig **best effort** (DSCP: 000000) jellegű
- prioritásos sorok és eldobási jellemzők alapján valósul meg a megfelelő kiszolgálás
    - osztályok **közti** torlódás – prioritás dönti el
    - osztályon **belüli** torlódás – legkisebb eldobási valószínűségű élvez előnyt
    - a szerződött ráta átlépése után a szolgáltatók az eldobási valószínűséget növelik az eldobási valószínűséget, kezdetben alacsony

19 2014.05.06 – Hálózati alkalmazások

19.1 Alkalmazásprotokollok

- legtöbbször az alkalmazásban kerül implementálásra
- szükséges szabványosítani még akkor is, ha egy alkalmazás-rétegbeli protokollt tipikusan kevés alkalmazás használ
- alsóbb rétegeket (mint szolgáltatást) az **operációs rendszer** biztosítja, elfedi a tényleges rétegeket
- API-t biztosít (ami gyakorlat SAP – Service Access Point), ennek **socket** végződését használhatja az alkalmazás, hogy megfelelő kommunikációt alakítson ki
- kliens-szerver architektúra (lásd 16.1.3 fejezet)
- a portok hozzárendelése a **szerveren** statikus, a **kliens** oldalán dinamikusan kerül kiosztásra
    - szerveren tipikusan 1-1023, kliens oldalon 1024-65535 (ebből regisztrált: 1024-49151, dinamikus: 49152-65536)



55. ábra Socket-hívás folyamata

	Natív IP	UDP	TCP
Kapcsolatorientált	×	×	✓
Megbízható	×	×	✓
Üzenetméret (tipikus)	rövid	rövid	hosszú
Adattovábbítás jellege	datagram	datagram	bitstream pipe
Portkezelés	×	✓	✓
Overhead	minimális	kicsi	nagy
Alkalmazások	vezérlési és menedzsment •ICMP, IGMP •Routing	multimédia -átvitel, névfeloldás	fájlátvitel, web, levelezés

56. ábra Alkalmazások és igényelt transzferprotokolljaik

19.1.1 Natív IP feletti protokollok és portszámaik

- 1: ICMP, 2: IGMP → **Vezérlés**
- 8: EGP, 89: OSPF → **Routing**
- 132: SCTP → **Egyéb**
- 6: TCP, 17: UDP → **Valódi szállítási protokollok**

19.1.2 UDP feletti protokollok és portszámaik

- 53: DNS (névfeloldás)

- 67-68: DHCP (konfiguráció)

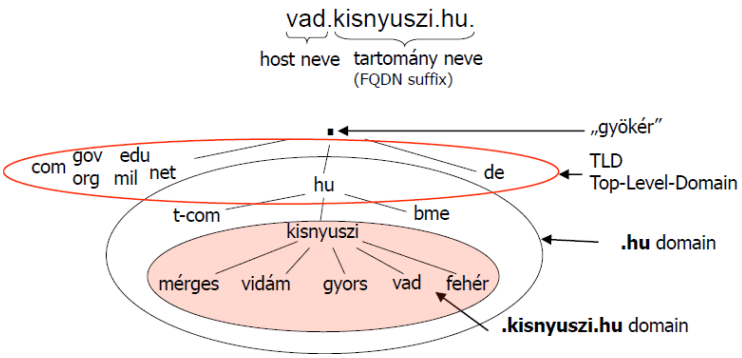
19.1.3 TCP feletti protokollok és portszámaik

- 20-21: FTP
- 22: SSH
- 23: Telnet
- 25/465: SMTP/SMTPS
- 53: DNS
- 80/443: HTTP/HTTPS
- 110/995: POP3/POP3S (Secure)
- 143/993: IMAP4/IMAP4S (Secure)

19.2 Infrastrukturális szolgáltatások

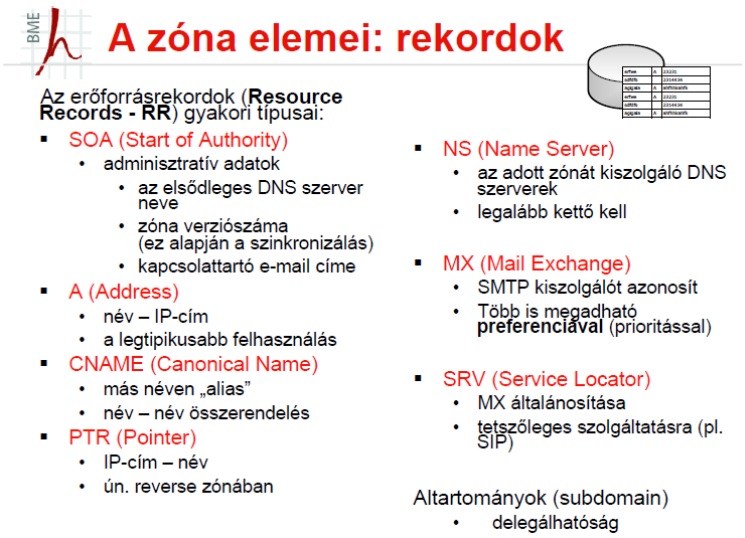
19.2.1 DNS – Névfeloldás

- könnyen megjegyezhető nevek megfeleltetése IP-címeknek
- elvárások: skálázhatóság, hibatűrés, friss információk
- hierarchikusan szervezett
- FQDN – Fully Qualified Domain Name



57. ábra Klasszikus kisnyuszis példa

- root (gyökér) „.”-al jelöljük
- domainek adminisztrálása – IANA adminisztrálja lsd. korábban
- elsődleges (írható/olvasható) és másodlagos (csak olvasható) DNS szerver
    - másodlagosból **legalább** egy kell, elsődlegesből **pontosan** egy



58. ábra A kép magáért beszél

- **rekurzív** vagy **iteratív** kérés
  - **rekurzív:** konkrét vagy negatív választ vár, a címzett végzi a névfeloldást
  - **iteratív:** minél közelebbi felelős megtalálása, referral (hivatkozás) adása
- helyi gépen és szerveren cache a feloldás gyorsítására
  - minden rekord TTL értéke másodpercben → elévülés
  - autoritatív válasz (rekordért felelős szerverek valamelyikétől származik) és nem autoritatív válasz (gyorsítótárból)
- böngésző gyorsítótára → helyi gép gyorsítótára → helyi gép hosts fájlja → DNS szerver (elsődleges/másodlagos, rekurzív/iteratív)
- lekérdezéshez - DNS szerver ←→ DNS szerver: UDP 53 – rövid, gyors üzenetváltás
- zónaletöltéshez – elsődleges → másodlagos DNS: TPC 54 – hosszabb, megbízhatóbb
- **DNS kérés elemei:**
  - kért rekod típusa(i)
  - feloldandó név vagy IP-cím
  - rekurzív kérés esetén vonatkozó bit (Recursion Desired)
- **DNS válasz elemei:**
  - pozitív válasz esetén egy/több elemű lista, round-robin választás belőle
    - ha autoritatív a válasz, azt egy jelzőbit mutatja
    - referral válasz kiegészítő információkkal
  - negatív válasz → nem található a bejegyzés
  - nincs válasz != negatív válasz

19.2.2 Címkonfiguráció – DHCP

- IP beállításokat oszthatunk ki vele dinamikusan
- központi módosításokkal, egyszerűen állíthatunk be klienseket, mobilitást adva

59. ábra DHCP működése

- **DISCOVER** és **OFFER**
  - kérés 0.0.0.0- tól 255.255.255.255 felé (bárkinek)
  - ajánlat 192.168.1.1-től (DHCP szerver) 255.255.255.255 felé (bárkinek)
  - ajánlat tartalmazza: IP-cím, Subnet mask, Bérleti idő, DHCP szerver IP címe, **igénylő MAC** címe
- **REQUEST** és **ACK**
  - kiválasztáskor 0.0.0.0-tól 255.255.255.255-nek a **kiválasztott IP-címmel, DHCP szerver IP-címével üzenet**
  - 192.168.1.1-től 255.255.255.255-nek **nyugta**, benne: kiosztott IP-cím, Subnet mask, bérleti idő
- **bérleti idő** kezelése: félidőben hosszabbítás kérés, TTL 7/8-ánál új igénylése
  - **rövid** bérleti idő mellett szól: kliens **szabálytalanul kiléphet**, szabálytalaul **újraindulhat**, **gyorsan** jussanak érvényre a **beállításváltozások**
  - **hosszú** bérleti idő mellett szól: ne legyen nagy a hálózati forgalom
- **DHCP-vel beállítható dolgok**:
  - 0x01 – Subnet Mask
  - 0x0F – Domain Name
  - 0x03 – Alapértelmezett átjárók
  - 0x06 – DNS
  - 0x0C – Host Name
  - 0x33 – Lease Time (TTL)
  - 0x3A – Renewal Time
  - stb.
- IPv6-ban minden router egyben DHCP szerver is
- hibatűrés: több DHCP használata egy hálózatban, de **diszjunkt IP-címtartományok** osztása
- **DHCP Relay Agentekkel** megoldható, hogy a routerek a DHCP-nek címzett forgalmat (amit alapból nem engednének át) a DHCP szerver felé továbbítsák

19.3 Szöveg- és fájlátvitel

19.3.1 Telnet

- egyik legrégebbi alkalmazás
- **távoli parancssor** parancsok küldésére és visszajelzések megjelenítésére
- ma is alkalmazzák hálózati eszközök egyszerű, távoli adminisztrációjára
- nem biztonságos, ezért **SSH-t érdemes használni helyette**

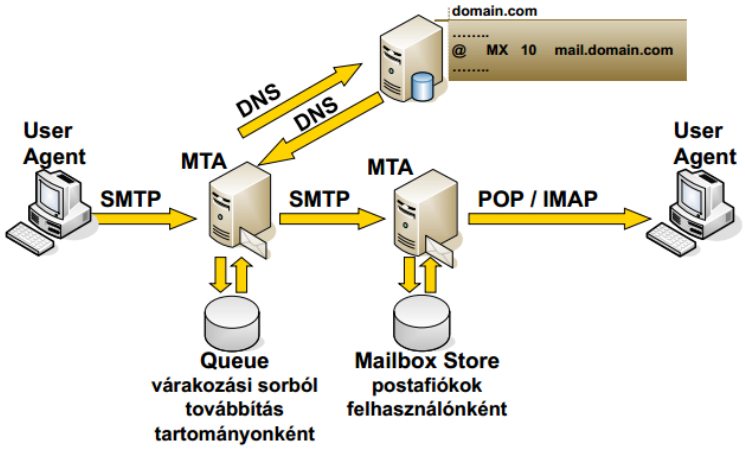
19.3.2 FTP

- egyik legkorábbi fájlátviteli protokoll

- TCP 20-21-es portját használja: 20 – adatátvitel, 21 – vezérlés
- különböző parancsok pl. open, ls (listázás), put (feltöltés), get (letöltés)

19.4 Levelezési rendszerek

- komponensei a MUA (Mail **User Agent**) és az MTA (Mail **Transfer Agent** – **SMTP**)
- használt protokolljai: SMTP (küldéshez), POP3 és IMAP4 (lekérdezéshez)
- címzettek meghatározása **DNS MX** rekordja segítségével



60. ábra Levélküldés folyamata

19.4.1 POP3, IMAP és SMTP

19.4.1.1 POP3 – Post Office Protocol version 3

- parancsorientált
- **TCP 110**
- **biztonságos** változata **TCP 995**-ös portján, TLS titkosítással

19.4.1.2 IMAP4 – Internet Message Protocol version 4

- parancsorientált
- **TCP 143**
- **biztonságos** változata **TCP 993**-as portján, TLS titkosítással
- könyvtárstruktúra támogatása
- keresés támogatása
- nem törli automatikusan a szerveren tárolt levelet

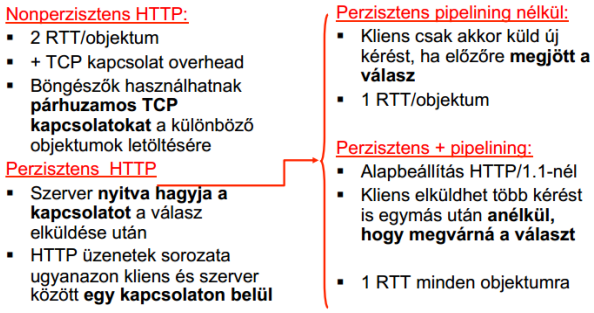
19.4.1.3 SMTP – Smart Mail Transfer Protocol

- parancsorientált, állapotkódokkal
- **TCP 25**
- SMTP relay – nem közvetlen küldés, relay szerverek közbeiktatásával
- biztonságos változata **TCP 465**-ös portján, TLS titkosítással
- gyakori parancsok: HELO, MAIL FROM:<feladó>, RCPT TO:<címzett>, <CR><LF>.<CR><LF>, QUIT, HELP, NOOP, VERIFY<cím>

19.5 Webes rendszerek

- Sir Tim Berners-Lee, CERN
- parancsorientált, állapotkódokkal
- speciális fejlécek
- **TCP 80**
- proxy használatának lehetősége: kliens nevében, főként a hatékony gyorsítótárazás miatt

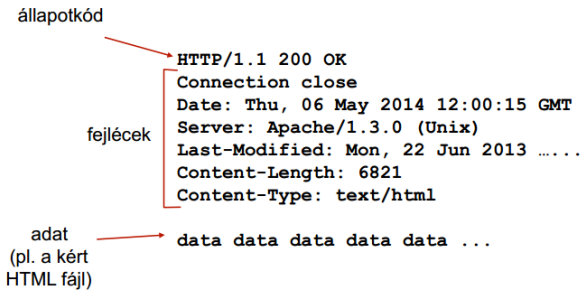
- HTTP1.0 – nem perzisztens kapcsolat (minden kérés/válasz külön TCP kapcsolaton keresztül)
    - **késleltetés: 2RTT + adási idő**
- HTTP1.1 – perziszten kapcsolat (egy TCP kapcsolaton belül több kérdés/válasz)



61. ábra Perzisztens és nonperzisztens HTTP működése

62. ábra Példa HTTP kérésre

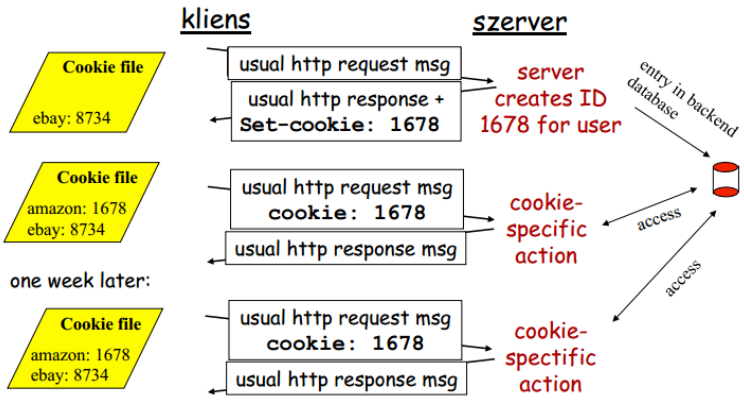
- gyakori **HTTP parancsok:**
    - GET <URL> - lekérdezés
    - HEAD – ugyanaz, mint a GET, csak fejléc metaadatokkal
    - POST – adatot tudunk küldeni vele a szervernek
    - PUT – feltöltésre alkalmas
    - DELETE – adott URL tartalmának törlése
- gyakori **HTTP fejlécek**
    - Accept – elfogadható **MIME** (Multipurpose Internet Mail Extensions) típus
    - Accept-Charset – elfogadható karakterkészlet
    - Content-Type – MIME típus
    - User-Agent – böngésző neve és verziója
    - Server – szerver neve és verziója



63. ábra Példa HTTP válaszra

Kód	Jelentés	Leírás
200	OK	
201	Created	POST sikeres
202	Accepted	Kérés elfogadva
204	No content	Nincs semmi a kliensnek
400	Bad request	Hibás kérés
401	Unauthorized	Hitelesítés szükséges
403	Forbidden	Hozzáférés megtagadva
404	Not found	Nem található
500	Internal Server Error	Belső szerver hiba
503	Service Unavailable	Pillanatnyilag nem szolgálható ki

64. ábra Gyakori HTTP kibakódok



65. ábra Cookie-k kezelése

- HTTP jövője – HTTP2.0, lásd 2.2 fejezet