========== Bevezetés ==========

A szabványnak megfelelés - konformancia
Azonos részhalmaz - együttműködés

Szervezetek:
- IEEE (Institute of Electrical and Electronics Engineers)
  - 802.3 - Ethernet
  - 802.11 - WiFi
  - 802.15 - WPAN
  - 802.16 - BWA
- IETF (Internet Engineering Task Force)
  - RFC-k (Request For Comment)
- ISO (International Standardization Organization)
- ANSI (American National Standards Institute)
- ITU (International Telecommunacation Union)
- ETSI (European Telecommunication Standard Instututе)

Hálózatok igényei:
  - összekötöttség
  - gyors linkek, kis hibaarány, késleltetés stb.
  - forgalom elterelés, szabályzás kell!
  - best-effort nem elég

HTTP (HyperText Transfer Protocol)
  - független az átviteli rétegtől (de ált TCP/IP)
  - 1.1 újítások:
    - perzisztens kapcsolat
    - pipelineing (több kérés a válasz megvására nélkül)
    - byte serving (csak kijelölt rész lekérése)
  - 2.0 (jövő)
    - Google SPDY elemeit használja
    - multiplexált streamek egy TCP kapcs.-on belül
    - kérések priorizálása
    - fejrész tömörítés
    - statikus részeket nem kell újraküldeni
    - szerver 'push' és 'hint'

FTP (File Transfer Protocol)
  - out-of-band (adat 20-as protokoll, vezérlés 21-es protokoll)
  - aktív adatátvitel - szerver építi fel a kapcsolatot (tűzfal vagy NAT esetén nem megy)
  - passzív átvitel - kliens építi fel a kapcsolatot

Levelezés:
  - Három szereplő:
    - User Agent (pl: Outlook)
      - levél írás, olvasás, szerkesztés
    - Levél szerver
      - tárol
      - lekérés a szerverről:
        - POP (Post Office Protocol)
        - IMAP (Internet Mail Access Protocol)
          - Többen is módosíthatják (flagek)
        - HTTP (pl Gmail)
    - Levelző protokoll (SMTP)
      - store & forward

Peer to Peer:
  - Változatok:
    - Centralizált (Napster)
    - Elosztott (Gnutella)
      - robosztusabb, exp. forgalom növekedés
    - Párhuzamos kommunkáció (BitTorrent)
  - jobban skálázódik mint az FTP, de nagyobb sávszél igény

Eddigiek TCP alapon.

Multimédia elvárások:
  - ~150 ms delay
  - néhány 10 ms delay ingadozás
  - néhány százalék véletlen csomagvesztés belefér
  - UDP alapon.

Késleltetés okai:
  - Feldolgozás a csomópontban
  - Sorban állás
    - darabszám * L (csomaghossz) / R (adatátviteli sebesség)
  - Adási idő
    - L (csomaghossz) / R (adatátviteli sebesség)
  - Terjedési idő
    - d (hossz) / s (sebesség)
      - sebesség általában: 2*10^8 m/s

a = átlagos csomagérkezési ráta
Forgalom intenzitása: L * a / R

QoS (Quality of Service)

================= ProtokollArchitektúrák ====================

protokoll
  - statikus rész (alak, forma)
    - csomag szerkezete: header - adat - (trailer)
      - header és trailer: PDU (Protocol Data Unit)
  - dinamikus rész (mit kell csinálni vételkor, továbbításkor, kivételekkor...)

rétegek
  - kezelhető, független részek de feladatok duplikálódnak, overhead
  - két eszköz azonos rétegei között virtuális kapcsolatok
  - pl: ISO - OSI
    - csak a rétegekre bontás, a protokollok nem részei az OSI-nak.
    - Alkalmazási réteg
      - alkalmazások együttműködése
      - HTTP, FTP, SMTP
    - Megjelenítési réteg
      - felhasználói adatok szintaxisának ellenőrzése
      - konvertálás (pl. endiannes, CR LF / LF)
      - tömörítés
      - titkosítás
    - Viszony réteg
      - duplexitás kezelése
      - kapcsolat felépítése, lebontása
      - összefüggő adatfolyamok szinkronizálása
    - Szállítási réteg
      - megbízható kommunikáció biztosítása
      - forgalomszabályozás
    - Hálózati réteg
      - logikai címzés
      - útvonalkeresés
      - forgalomirányítás
      - router
    - Adatkapcsolati réteg
      - csomagok összeállítása
      - közeghozzáférés kezelése
      - fizikai címek kezelése: MAC (Media Access Control) címek
      - bridge, switch
    - Fizikai réteg
      - közeg specifikációja
      - vonali kódolás: szimbólumreprezentáció
      - moduláció: szinuszos vivő-be van belekódolva az infó

  - TCP-IP architektúra:
    - Alkalmazási réteg (Alkalmazási + Megjelenítési + Viszony)
    - Szállítási réteg (Szállítási)
    - Internet réteg (Hálózati)
    - Interfész réteg (Adatkapcsolati + Fizikai)

  - IEEE LAN-architektúra:
    - Adatkapcsolati:
      - LLC (Logical Link Control)
      - MAC (Medium Access Control)
    - Fizikai:
      - PHY (Physical)
      - PMD (Physical Medium Dependent)

A háromsíkú protokollarchitektúra:
  - felhasználói
    - felhasználói adat továbbítása
  - vezérlő
    - az adat továbbításához szükséges infó áramlása
  - menedzsment
    - a kapcsolat fenntartása, ellenőrzése

Cross-layer:
  - nem szomszédos rétegek kommunikációja

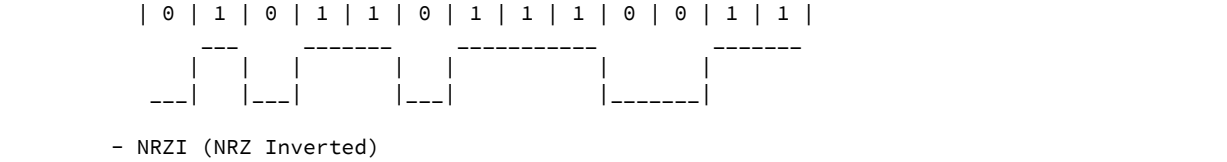
================ Fizikai szintű kommunikáció 1. ====================

bitsebesség - bit/s
szimbólum-sebesség, jelzési sebesség - baud, Bd
bitsebesség = szimbólum méret (bitben) * szimbólum-sebesség

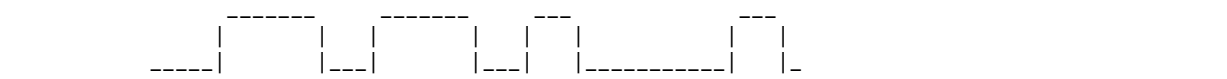
több szint (nagyobb szimbólum méret)
  - nagyobb sávszél
  - több hiba

Vonali kódolás:
  - adat bitsorozata -> jelsorozat (feszültség érték sorozat) leképzés
  - cél: ne legyen DC komponens, bitszinkron tartható legyen

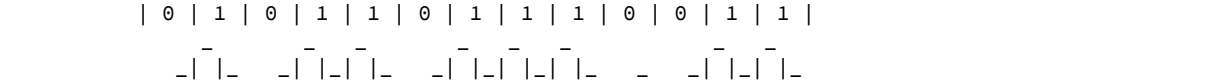
  - NRZ (Non-Return-to-Rero)
    - direkt leképzés: 0 -> 0, 1 -> 1
    - van DC, nincs szinkron



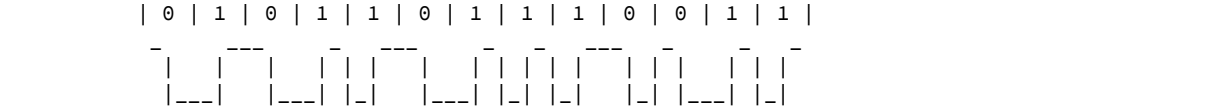
  - NRZI (NRZ Inverted)
    - 0 -> marad a jel, 1 -> bitidő közepén váltás
    - van DC, hosszú nulláknál nincs szinkron (bit shuffling megoldja)



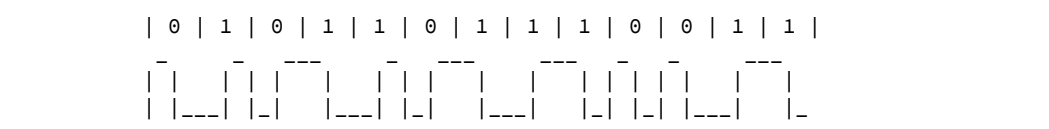
  - RZ (Return-to-Zero)
    - Két impluzus között mindig visszatér a nulla jelszintre
    - van DC, de a szinkron tartható



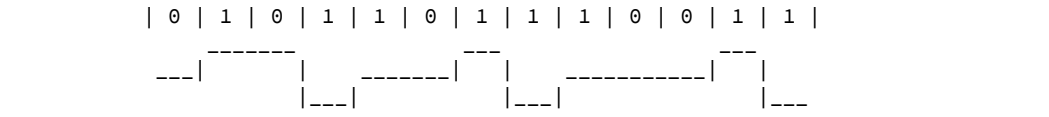
  - Manchester
    - 1 -> felfele váltás bitidő közepén, 0 -> lefele váltás
    - két egymásutáni azonos jel esetén bithatáron is váltás
    - szinkronizálható, nincs DC
    - Ethernet, RFID



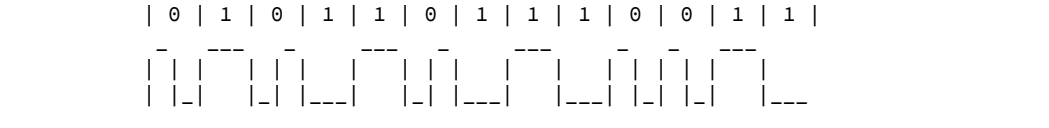
  - Differenciális Manchester
    - 1 -> középen váltás, 0 -> elején és középen is váltás
    - hiba-védettebb, mint a Manchester
    - token ring, optikai tárolás



  - MLT 3 (Multi-Level Transmit 3)
    - 3 fesz. szint, ha 1 -> ciklikusan mozog felfele, 0 -> marad



  - FM-0
    - minden bithatáron váltás, 0-nál középen is
    - Kb. ugyanaz, mint a Diff Manchester.



  - 4B/5B kódolás
    - minden bitnégyest 5 bitre cserél
    - 16 kombináció adatnak (ezekben legalább 2 váltás)
    - 8 jelzésre
    - 8 nem használt

Bitfolyamok strukturálása:
  - keret
    - szinkron miatt
    - hiba védelem (paritás bit / CRC (Cyclic Redundancy Check))
  - fejrész azonosítása:
    - SOH (Start of Header) bitminta
    - STX (Start of Text) bitminta

Moduláció:
  - infó belekódolása szinuszos vivőbe.

  - ASK (Amplitude Shift Keying)
    - sima: 0 -> nincs vivő, 1 -> van vivő
    - több szintű is lehet
    - egyszerű és olcsó implementálni
    - zajérzékeny
    - optikai átvitelben használják

  - FSK (Frequency Shift Keying)
    - 0 -> f1 frekvencia, 1 -> f2 frekvencia
    - egyszerű implementáció
    - kevésbé zajérzékeny
    - változatai:
      - AFSK (Audio FSK) - telefonmodemek
      - GMSK (Gaussian Minimum Shift Keying) - GSM

  - PSK (Phase Shift Keying)
    - BPSK (Bináris PSK) - 1 -> 180°-os fázistolás, 0 -> nop
      - RFID
    - QPSK (Quadrature PSK) - 4 szint
      - Bluetooth 2
      - komplexként ábrázolható (felső bit - valós, alsó - képzetes rész)
        - ábrázolva : konstellációs diagram
    - 8PSK - nagy a hibaarány, efelett QAM jobb
      - EDGE (Enhanced Data rates for GSM Evolution)

  - QAM (Quadrature Amplitude Modulation)
    - jel = komplex szám
    - valós résszel egy koszinuszos, képzetes résszel egy szinuszos vevőt modulálunk
    - a két jel összegét küldjük át
    - 16-QAM, 64-QAM, 128-QAM, 256-QAM
    - pl kábel tv

  - Adaptív moduláció:
    - link állapotától függően választ modulációt

Multiplexelés:
  - több csatorna jelét egy csatornán átküldeni
  - FDM (Frequency Division Multiplexing)
    - mindenki egyszerre, külön freq.-en
  - WDM (Wavelength Division Multiplexing)
    - FDM optikában
  - TDM (Time Division Multiplexing)
    - egymás utáni időszeletekben
  - CDM (Code Division Multiplexing)
    - csatornánként chip kód (orthogonálisak)
      - ezzel pszeudorandom zajt generálnak (srand(chip kód), n db rand() egymás után)
      - ebbe kódolják bele az infót (zaj xor adat jelet viszik át)
      - vevő oldalon a pszeudozajjal vissza xorolják
    - nehéz lehallgatni, blokkolni, nem zajérzékeny
    - bonyolult implementálni

================= Fizikai szintű kommunikáció 2. ====================

Hírközlő csatornák:
  - fémvezetők
    - TEM-hullámvezető (Transzverzális Elektro-Mágneses)
  - optika
  - vezeték-nélküli csatornák

Fémvezetők:
  - Szimmentrikus (sodrott) érpár
    - UTP: árnyékolatlan
      - típusai:
        - cat3: 16 MHz
        - cat5: 100 MHz (Fast Ethernet)
        - cat6: 250 MHz
        - cat7: 600 MHz
        - S: shielded (fémháló)
        - F: foiled (fémlemez)
    - vékony, rugalmas, olcsó
    - interferencia -> színtorzulás, szellemkép
  - Koax kábel
    - egyik vezető körülöleli a másikat, külső földelve
      - magas freq áramú vezeték antennaként viselkedik
      - koaxnál nem
    - végeken lezárás hullámimpedanciával
    - kábeltévénél tipikus

Strukturált kábelezés:
  - rendezőből (központi elosztó) minden végpontba külön kábel
  - UTP-t használ

Fényvezetők:
  - 3 népszerű hullámhossz:
    - 850 nm - 100 Mbit/s
    - 1310 nm - Gbit/s
    - 1550 nm - 10 Gbit/s
  - optikai szálak - mag és héj
    - step-index fiber (köpenyről visszaverődés)
    - graded-index fiber (nincs visszaverődés, csak törés)
  - multimódus: több független pálya van a vezeték belsejében (mag átmérője nagy)
  - monomódus: kis mag átmérő -> csak alapmódusú hullám fér, visszaverődés nélkül
    - kábel olcsóbb, adó/vevő drágább -> nagy távra éri meg
  - a hálózatba kell:
    - adó, vevő, erősítő, szűrő, szétosztó, összegző
    - ONT (Optical Network Terminator) - Internet - Optika összekötő

    - OLT (Optical Line Terminator) - optikai végpont, vevőegység
  - nagy sávszél, megbízható, de van diszperzió

Vezeték-nélküli csatornák:
  - 10 kHZ -> 1 THz
    - nincs teljesen kihasználva
  - antennanyereség - mekkora térszög alatt sugároz / teljes gömb
  - befolyásolja:
    - szabadtéri csillapítás (1 / s^2)
      - ez a legnagyobb korlát
    - visszaverődés (épületek, fák, ionoszféra)
    - törés
    - elhajlás
    - szóródás (troposzféra)
    - több úton terjedés (fading)
      - megoldás: többszörös vétel (diversity) frekvenciában, vagy térben
  - freq sávok engedélykötelesek
    - kivéve ISM (Industrial, Scientific, Medical) : ~2.4 GHz, ~5,6 GHz
  - Közvetlen rálátás - Fresnel-zónák
  - cellás mobil hálózat:
    - BTS (Base Transceiver Station): átjátszó
    - BSC (Base Station Controller): átjátszók jeleit összefogja, vezetékesen továbbítja

================= Többszörös hozzáférés ====================

Közös az átviteli közeg - többen akarnak hozzáférni egyszerre - hogyan?
 - Medium Access Control (adatkapcsolati rétegben)

Hasonló megoldások mint multiplexelésnél
  - FDMA (Frequency Division Multiple Access)
    - ortogonális, bonyolult
    - interferencia -> védősávok használata
    - real-time átvitelnél
  - TDMA (Time DMA)
    - ortogonális, rugalmas, egyszerű
    - interferencia, szinkronizációs problémák
    - lehet dinamikus (igény szerinti időrés kiosztás): Bluetooth, ViMAX
    - ez terjedt el
  - CDMA (Code DMA)
    - nem ortogonális, de információelméletileg a legjobb
  - SDMA (Space DMA)
    - térben választjuk el a felhasználókat (irányított antennák)

A hozzáférési módszerek jellemzése:
  - kihasználtság (throughput): az idő hány %-ában történik hasznos kommunikáció
  - késleltetés
  - igazságosság
  - stabilitás

ALOHAnet
  - csomagkapcsolt
  - központi hub, kliensek (csillag topológia)
  - hub->kilensek: broadcast, csomagban cím
  - kliens->hub: véletlenszerű, siker == nyugta
    - ha nem kap nyugtát -> véletlen késleltetés
  - max 18.4%-os kihasználtság
    - Kiszámolása:
      - Tételezzük fel, hogy egyenlő méretű csomagok vannak (pl byte-ok), amiket 'T' idő adni
      - Ha 'T' idő alatt átlagosan G db adási próbálkozás van:
        - Akkor Poisson eloszlást feltételezve annak az esélye, hogy egy 'T' hosszú időszeletben pontosan k-an próbálnak adni:
          - Pr[k] = G^k * e^(-G) / k!
        - Ahhoz, hogy adni tudjunk 't' és 't+T' között, az kell, hogy 't-T' és 't+T' között senki se próbálkozzon az adással.
          - így mire elkezdünk adni a 't' időpillanatban, addigra már mindenki befejezte az adást, amit legfeljebb 't-T'-kor kezdhetett el.
          - ennek a valószínűsége: e ^ (-2G)
        - De az nem elég, hogy tudunk adni, tényleg adnunk is kell, amit G valószínűséggel teszünk meg.
          - A throughput tehát: Spure = G * e^(-2G)
        - Ennek a maximuma 0.5 / e (amit G = 0.5-nél vesz fel), ami 0.184 kerekítve.
          - Tehát az Aloha az időnek max 18.4%-át használja ki adatátvitelre.

  - instabil, késleltetése nem korlátos, de hosszútávon fair

Réselt Aloha:
  - azonos hosszú csomagok, időréshatáron
  - nem kell '2T' hosszú szünet, hogy 'T' időt adni lehessen.
  - dupla akkora kihasználtság (36.8%)
  - késleltetés, stabilitás, fairness marad változatlan
  - RFID, műholdas katonai rendszerek

Helyfoglaló Aloha
  - az idő egy része igénybejelentésekkel megy el.
  - a felhasználók visszahallják az igényeket
  - sokkal kisebb átlagos késleltetés mint a Réselt Aloha esetében
  - GSM: réselt-helyfoglaló Aloha

Vivőérzékeléses többszörös hozzáférés (Carrier Sense Multiple Access)-
  - a csatorna foglaltságát is figyelembe veszik az adók
  - ha foglalt a csatorna:
    - nem perzisztens - később megpróbálja, nem figyeli
    - 1-perzisztens - addig vár, amíg fel nem szabadul, utána egyből ad
      - ha üközik ekkor: random backoff
    - p-perzisztens - p valószínűséggel ad, ha szabadnak érzi a csatornát
  - egyszerű implementálni
  - közel 100% throughput-ot eredményezhet
  - terjedési késleltetési érzékeny - max 2 * RTT (Round Trip Time) után érzékelünk az ütközést
  - túlterhelés hatására instabil
  - igazságos

  CSMA/CD
    - visszacsatolás adás közben, hogy volt-e ütközés
      - csak vezetékes csatornán lehetséges
    - ütközés -> foglaltsági jelzést ad (Jam signal) -> mindenki leáll
      - exponenciális backoff
    - fél-duplex esetén van csak értelme

  CSMA/CA
    - ütközés elkerülés
    - ha szabad a csatorna:
      - küld egy jelzést a többieknek, hogy adni akar, vagy vár, és ha utána is szabad, akkor ad
    - ha foglalt
      - exponenciális backoff
    - WLAN-ok
    - rejtett terminál: két terminál nem hallja egymást, és ugyanannak a harmadik terminálnak adnak egyszerre
    - exponált terminál: aki adni készül olyan kommunikációt is hall, amit a címzettje nem, adhatna, de nem mer
    - megoldás rejtett és exponált terminálra:
      - RTS (Ready To Send) - CTS (Clear To Send) jelzések (az átvitel hosszát is tartalmazzák)

Versenymentes protokollok:

  Bit-térkép módszer:
    - N időrésnyi foglalás, i. időrésbe i. terminál szól, ha adni akar.
    - utána annyi adat keret, ahányan adni akartak
    - alacsony forgalom esetén nem hatékony

  Bináris visszaszámlálás:
    - mindenki kap egy sorszámot
    - a legnagyobb sorszámú, adni akaró használhatja a csatornát
    - nem fair

Korlátozott verseny:
  - terminál csoportok, csak csoporton belül verseny

Adaptív fa protokoll:
  - minden ütközésnél két csoportra osztják a terminálokat
    - rekurzívan, amíg 1 vagy 0 eleme nem lesz a csoportoknak

Központilag vezérelt többszörös hozzáférés:
  - polling - akit kérdez a vezérlő, az adhat
  - probing - egy csoportot kérdez, azon belül verseny
  - lehet foglalás (azonos vagy külön csatornán, versennyel / nélkül)

  - stabil és fair

A polling megoldható elosztottan is: token passing
  - az adhat, akinél a token van

================= Lokális hálózatok ====================

IEEE-ANSI rétegek:
```
  LLC  |                  802.2
  ---------------------------------------------
  MAC  |
  PHY  |   802.3, 802.4, 802.5, 802.11, X3T9.5
  PMD  |
```

Mindegyik számára közös funkciók: 802.1
  - együttműködés, biztonság

802.2 - Logical Link Control
  - Hálózati réteg számára megbízható átvitel
    - forgalomszabályozás, hibaérzékelés, -javítás
  - lehet nyugtázott / nyugtázatlan datagram, vagy nyugtázott összeköttetés-alapú
  - nem mindig használják - IP-nek nincs szükségre rá
  - a fejrész miatt overhead

Az Ethernet
- IEEE 802.3 - a "kasszikus" Ethernet
- IEEE 802.3u - Fast Ethernet
- IEEE 802.3z - Gbit/s Ethernet
- IEEE 802.3ae - 10 Gbit/s Ethernet

DIX-nek is van Ethernet szabványa (Ethernet version 2), a MAC kerete más

logikailag busz topológia

Ethernet vs. Token bus vs.Token ring versenyt Ethernet nyerte
  - mert CSMA/CD-t használ

Jelölésrendszer:

A	B	C
10	Base	5
1000	Base	T

'A' rész:
  - adatsebesség Mbit/s-ben
'B' rész:
  - Base = alapsávi átvitel (Manchester kódolás)
  - Broad = szélessávú átvitel
'C' rész:
  - Átviteli közeg:
    - T = twisted pair
    - FX/LX/SX = fiber optics
    - CX = shielded balanced copper
    - T4 = 4 pair twisted pair
    - T2 = 2 pair twisted pair
  - Szegmenshossz:
    - 2 = 185m
    - 5 = 500m

10 Base 5 (Vastag Ethernet)
  - koax, vámpír csatlakozó,
  - max 100 adó, 2,5 méterenként

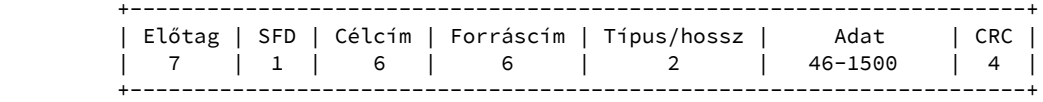
10 Base 2 (Vékony Ethernet)
  - koax, BNC csatlakozó, T elosztó
  - max 30 adó per szegmens

Ethernet ismétlő:
  - több szegmens összekötése
  - 5-4-3 suabály (10 Mbit/s):

    - 5 szegmenst 4 ismétlővel lehet összekötni, és max 3 szegmensen lehetnek terminálok
  - kábel meghibásodás: csak az adott szegmensen
    - rosszul csatlakozott terminál blokkolása

Ethernet hub (10 base T):
  - sokkapus ismétlő
  - ütközés a hubban jön létre
  - minden összekötött szegmens ugyanazon a sebességen és u.a. a keretezéssel működhet
  - használat:
    - régi 10 Base {2, 5} mai hálózathoz
    - protokoll analizátor
    - ha switch-en nincs STP (Spanning Tree Protocol)
      - 2 port összekötése megbénítja a hálózatot, hubnál csak ez a rész esik ki

Ethernet keret:



  A méretek byte-ban

  Előtag: fix bitminta - 10101010......
  SFD (Start Frame Delimiter) - keret kezdete
  Típus/hossz: az adatmező hossza vagy típusa (1500 alatt vagy 1536 felett)

Résidő: T = 2 * RTT = 2 * L / C
  - L: Szegmens hossz, C: jelterjedési idő
  - legfeljebb ennyi idő egy ütközést detektálni

Ethernet max résideje: 51.2 µs (2 * (2.5 km + 4 ismétlő késleltetése)
  - Ez 64 byte 10 Mbit/s esetén -> minimális kerethossz: 64 byte

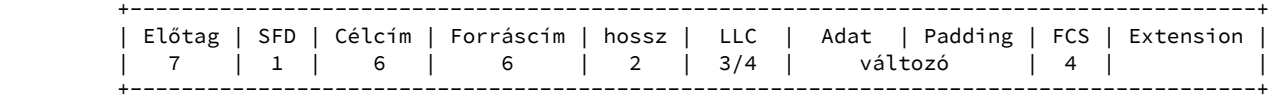
Az ethernet MAC protokollja (CSMA/CD):
  - várj, amíg szabad lesz a csatorna
  - várj 9.6 µs-et (keretek közti idő)
  - kezdj el adni
    - ha ütközés:
      - jam signal
      - Ha 'N'-szer ütköztél eddig:
        - Ha N == 15, akkor abort();
        - különben várj 51.2 * rand() % (1 << min(N, 10)) µs-et

Kapcsolt ethernet
  - nagyobb sebességű ethernet -> bridge, switch bevezetése
  - több szegmens összekapcsolása, eltérő szegmenssebességek
  - MAC címek alapján szűrik/irányítják a szegmenseken belüli/közötti forgalmat

Fast Ethernet (100 Base X)
  - Cat5 UTP (árnyékolatlan), vagy Cat5 STP (árnyékolt), vagy optika
  - 4B/5B kódolás

GigaBit Ethernet
  - 512 byteos résidő (eddig 64 volt)
    -> a keretet megnyújtották (csak a félduplex üzemmód esetén)

  Az új keret:



  FCS: Frame Check Sequence (CRC-32)

10 Gbit/s Ethernet:
  - full duplex (nincs CSMA/CD)

40 Gb és 100 Gb Ethernet: szabvány 2010-ben, 802.3ba

----- LAN hálózatok ------

Egy LAN szegmens korlátai:
  - távolság, állomások száma / típusa

átjátszók segítségével megoldható -> LAN hálózatok

Különböző átjátszók:
  - fizikai réteg: repeater (egy port) vagy hub (több port)
    - jelerősítés, továbbítás (minden porton)
  - adatkapcsolati réteg: bridge (közegek között, újrakeretezéssel), switch (azonos közegben, keretezés nélkül)
    - továbbítás csak a szükséges porton (MAC cím alapján, szelektíven)
    - többféle sebességet is képesek kezelni
    - store & forward
    - bridge különböző LAN-okat is összeköthet
  - hálózati réteg: router
    - útválasztás csak a szükséges porton
  - felette: gateway
    - protokollkonverzió, -együttműködés

Switch
  - full-duplex (vagy CSMA/CD ha nem full-duplex hálózathoz kapcsolódik)
  - Plug-and-play, self-learning
    - kapcsolótábla alapján
      - bejegyzések: MAC Address, Interface, Time Stamp
      - régi bejegyzéseket eldobja
      - működése, amikor egy csomagot kap:
        - ha van bejegyzés az adott MAC cél-címre
          - ha a cél azon a szegmensen van, ahonnan jött a keret akkor eldobja a keretet
          - különben továbbítja a megadott interfészre
        - különben elárasztás, kivéve az az interfész ahonnan jött.
          - ehhez a MAC címhez a következő csomagból (a válaszból) tudja meg az interfészt -> ekkor frissíti a kapcsolótáblát

================= Vezeték nélküli LAN-ok ====================

jellemzői:
  - pár száz méter
  - 1-2 Mbit/s -> 100 Mbit/s
  - ISM sávban

Szabványok:
  - 802.11 - 1-2 Mbit/s, 2,4 GHz, FHSS / (DSSS / infra)
  - 802.11a - 54 Mbit/s, 5 GHz, OFDM
  - 802.11b - 11 Mbit/s, 2,4 GHz, DSSS, 11-13 csatorna
  - 802.11g - 54 Mbit/s, 2,4 GHz, OFDM / DSSS, 13 csatorna
  - 802.11n - akár 600 Mbit/s, 2,3 / 5 GHz, OFDM MIMO (4 stream), 64QAM több csat mód
  - 802.11ac - 1+ Gb/s, 5 GHz, 8 stream, 256 QAM

További:
  - 802.11e - QoS
  - 802.11h - auto teljesítményszabályozás (ATPC, csatorna váltás (DFS)
  - 802.11i - titkosítás
  - 802.11j - HiperLAN2 együttélés
  - 802.11s - mesh-üzemmód

Az 'n'-nek (ált. 5MHz) együtt kell élnie a 'b/g'-vel (2,4 GHz) -> duál rádiós AP

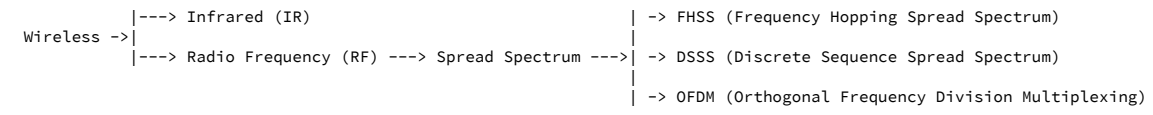
Csatornák: 2.4000 - 2.4835 GHz sáv
  - 13 db 22 MHz-es, egymástól 5 MHz középfrekvenciára

Spektrális maszk minden csatornára (átlapolódás miatt)
  - spektrális maszk: az egyes frekvenciasávokban kisugározható max. teljesítmények előírása
  - frekvencia közepétől 30dB-es csillapítás +/-11Mhz-re

3 csatornát lehet használni átlapolás miatt
 - de közel-távol probléma, interferencia

5 GHZ sávban 23 nem átlapoló csatorna

Vezeték-nélküli kommunikáció fizikai formái:



DSSS
  - az adathoz képest magas frekvenciás pszeudó zajjal modulált jelet továbbítanak (mint CDMA)
  - a saját pszeudo zajjal nagy a korrelációja az ilyen jelek összegének, de az összes többi jelre, és a külső zajokra kicsi.
  - minél nagyobb a chip-frekvencia annál jobb a zavarvédettség ("processing gain"), de kisebb a sávszél
  - IEEE 802.11 : 11 bites szórás

FHSS
  - több frekvenciát használ
  - adott időközönként pszeudorandom másik frekvenciára ugrik
  - A 2.4 GHz ISM sávban:
    - Min. 75 frekvencia használata
    - Max. 400 ms egy frekvencián: 2,5 ugrás/s;
    - Lehet adaptív is: csak „jó” frekvenciákat használja

OFDM
  - a sávok nem diszjunktak, a spektrumok "össze vannak tolva", egymásba átlógnak
  - a jeleknek azonos távolságra vannak zérushelyeik, de úgy vannak összetolva, hogy a zérushelyeknél pontosan egy jel értéke nem nulla.
  - ezekben a pontokban szétválaszthatóak (Nyquist-elv)
  - védettség a keskenysávú interferencia és frekvencia szelektív fading ellen
  - magas spektrális hatékonyság
  - de érzékeny a Doppler-effektusra és frekvencia szinkronizációra

Működési módok:
  - Ad-hoc mód: peer-to-peer
  - Infrastruktúra mód: AP (Access Point) - kliens

WLAN-ok topológiája:
  - BSS (Basic Service Set): egy cella, egy AP, és a kliensei
  - ESS (Extended Service Set): több összekapcsolt cella
  - DS (Distribution System): elosztóhálózat

WAP (Wireless Access Point)
  - Egy bridge, ami a vezetékes LAN-t (802.3) a WLAN-al (802.11) köti össze

802.11 keretformátum:

Frame Control	Duration ID	Addr 1	Addr 2	Addr 3	Seq.	Addr 4	Data	FCS
2	2	6	6	6	2	6	0 - 2312	4

  Vezérlési mező (Frame Control):
    - protokoll verzió (2 bit)
    - típus (2 bit): vezérlési adat vagy menedzsment keret
    - rész típus (4 bit):
      - pl: vezérlésen belül: ACK, RTS, CTS, menedzsment: Beacon, Authentication, Probe Request
    - ToDS - FromDS (1-1 bit): elosztóhálózatba / -ból

  802.11 címek:

To DS	From DS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	N/A
0	1	DA	Send AP	SA	N/A
1	0	Rec. AP	SA	DA	N/A
1	1	Rec. AP	Send AP	DA	SA

  DA: célcím (Destination Address)
  SA: forráscím (Source Address)

  BSSID: cella azonosító (Basic Service Set ID)
  Rec. AP: fogadó Access Point
  Send AP: küldő AP

Hozzáérési módszerek:

  CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)
    - vezeték-nélküli kapcsolatnál nem lehet ütközést detektálni

  DCF (Distributed Coordination Function):
    Különböző IFS-ek (Inter Frame Space):
      - SIFS (Short IFS): 10 v. 16 µs
        - vezérlő üzenetek számára
      - PIFS (PCF IFS): SIFS + Résidő
        - PCF képes AP várakozási ideje
      - DIFS (DCF IFS): SIFS + 2*Résidő
        - adatkereteknek
    Működése:
      - ha szabad, vár (D)IFS időt, ha addig is szabad marad, akkor ad.
      - ha foglalt, akkor vár, amíg szabad lesz, utána még (D)IFS + véletlen késleltetés.
        - a késleltetés értéke: Backoff_Time = INT(CW * RND()) * Slot_Time
          - Slot_Time: 20 µs (DSSS), 50 µs (FHSS)
          - CW (Contention Window)
            - max várakozás értékét határozza meg
            - n-edik foglaltságnál 2^(4+n) - 1 -> (31, 63, 127...)
          - RND: random float 0 és 1 között
          - (Megj: ha a CW pl 31, akkor az INT(CW * RND()) 0 valószínűséggel lesz 31. A CW értékei valószínűleg kettőhatványok, csak az előadó elírta).

  PCF (Point Coordination Function):
    - opcionális, DCF felett (vele együtt)
    - egyetlen AP vezérli a hozzáférést
      - AP -> beacon -> állomások abbahagyják a DCF-et
    - végigkérdezi, az állomásokat (ekkor adhatnak, ha akarnak)
      - nagyon nagy késleltetés
    - prioritásokat is rendelhet az állomásokhoz
    - nem terjedt el

  DCF (PCF-hez képest):
    - kevés kliens esetén gyorsabb, kisebb késleltetés
    - nincsenek prioritások, nincs QoS
    - ha egyszer megszerzi egy terminál a csatornát, sokáig tarthatja

  HCF (Hybrid Coordination Function)
    - "DCF prioritásokkal"

    EDCA (Enhanced Distributed Channel Access)
      - nagyobb prioritású terminál kevesebb ideig vár
      - versenymentes idő pl. hang és video számára

    HCCA (HCF Controlled Channel Access)
      - AP bármikor elrendelhet versenymentes periódust, egyébként EDCA verseny
      - forgalomosztály ütemezés is (előre veszi a magasabb prioritású osztályt: per-session service)
      - bonyolult, ritkán valósítják meg

RTS/CTS:
  - WLAN-nál opcionális
  - rejtett terminál problémát megoldja.
    - RTS foglalási időt is tartalmaz: NAV (Network Allocation Vector)
    - ezt mások is hallják, elindítanak egy számlálót, és addig nem adnak
  - exponált terminált nem oldja meg
  - kisebb keretek ütköznek, ha van RTS/CTS
  - kevés felhasználónál rosszabb kihasználtság
  - növeli a késleltetést

Mesh hálózatok:
  - ad-hoc vezeték-nélküli kapcsolatokból hálózat
  - nagyon rosszul skálázódik

WLAN biztonság:
  - WEP (Wired Equivalent Privacy): könnyen törhető

  - WEP2, WPA (Wi-Fi Protected Access): továbbfejlesztés, régi hardware-en
  - WPA2: erős titkosítás új hardware-en (802.11i)

================= BWA - Broadband Wireless Access ====================

802.15 WPAN (Wireless Personal Area Network):

Bluetooth - IEEE 802.15.1
  - cél: olcsó, rövid távú (10 m), max 1 MB/s
  - 2,4 GHz ISM
  - FHSS - f = 1600 1/s (79 db 1 MHz-es sáv)
  - TDD (Time Division Duplexing): a csatorna irányának vátlása 625 µs-enként.
  BR (Base Data Rate):
    - GFSK (Gauss szűrős FSK) - 1 Mbit/s
  EDR (Enhanced Data Rate):
    - Bluetooth 2.0
    - 2 Mbit/s - DQPSK (Differential QPSK): Az előző jelhez képesti fáziskülönbségbe van kódolva a bit, nem abszolút fázis értékekhez, mint a QPSK-nál.
    - 3 Mbit/s - 8DPSK

Hálózatszervezés:
  - piconetekből
    - max 8 aktív állomás, abból pontosan egy master
  - egy slave egy másik piconetbe lehet master (összes kombináció lehetséges)
  - piconetek hálózata: scatternet (dinamikus, ad-hoc)

  A kommunikáció menete:
    - szinkronizálás a master órájához (FHSS miat kell)
    - master mondja meg, hogy melyik időrés melyik slave-é

  Kapcsolat felépítése:
    - Inquiry: további Bluetooth-eszközök felderítése a hatótávolságon belül
    - Page: kapcsolat felépítése (pairing)

  Kétféle összeköttetés:
    - SCO (Synchronous Connection Oriented) - hangátvitel, előre lefoglalt időrések
    - ACL (Asynchronous Connectionless) - adat

Bluetooth verziók:
  - 1.0
  - 1.2: adaptív FHSS
  - 2.0: EDR (3 Mbit/s), kisebb fogyasztás
  - 3.0: 24 Mbit/s-ig, "High Speed": vezérlés Bluetoothon, adat WLAN-on
  - 4.0: alacsony fogyasztású (Bluetooth Smart)

Nagysebességű PAN-ok -> IEEE 802.15.3
- célkitűzés: médiakommunikáció
- 480 Mbit/s-ig, kb. 10 m

SNR (Signal-to-Noise Ratio)

Shannon egyenlet a csatorna-kapacitásra (C - Capacity, B - Bandwidth): C = B*log(1 + SNR)
  -> „jobban megéri” a sávszélességet növelni, mint az SNR-t

UWB (Ultra-Wideband)
  - B > 500 Mhz
  - nincs vivő, rövid, kis teljesítményű, széles spektrumú impulzusok
  - alapból egyenlő időközönként
    - Time-based moduláció: 0-t picit korábban, 1-et picit később küldjük
    - Shape-based moduláció: A 0 jel ellentétes fázisú az 1-hez képest
  - ellenáll a többutas terjedésnek
  - 3,1...10,6 GHz sávban
  - még nem terjedt el
  - 100 - 500 Mbs
    - 110 Mbit/s - 10m-es táv
    - 200 Mbit/s - 4m-es táv
    - 480 Mbit/s - < 4m
  - alacsony spektrális-, de magas területi hatékonyság

  Alkalmazások:
    - Digitális Otthon (Wireless USB)

    - Road Side Markers
    - Through-the-wall Intrusion Sensor (katonai célok)

EMC (ElectroMagnetic Compatibility)
 - Legnagyobb nehézség: az impulzusok sávközép-frekvenciájának és sávszélességének pontos kézben tartása

---

802.16 WMAN (Wireless Metropolitan Area Network)

WiMAX - Worldwide Interoperability for Microwave Access
  Felhasználási terület:
    - Wi-Fi hotspotokhoz történő csatlakozás
    - Szélessávú vezeték-nélküli alternatíva a kábel és a DSL részére.
    - Nagy sebességű mobil adatátvitel a telekommunikációs szolgáltatásokra (4G)
  Mikrohullám (visszaverődik a hétköznapi méretű tárgyakról)
    - LOS (line-of-sight) -> Fresnel zóna
      A fesnel zóna sugara (r - sugár (m), D - távolság (km), f - frekvencia (GHz))
      - r = 17.32 sqrt(D / (4*f))

  Szabványok:
    - "Fix" WiMax: 802.16d vagy 802.16-2004
      - OFDM
    - "Mobil" ViMax: 802.16e
      - MIMO

  Adaptív moduláció: BPSK-QPSK-16QAM-64QAM

  MAC:
    - A WLAN-al ellentétben: ütemező algoritmus
    - csak egyszer kell versenyezni, utána fix időrés
    - QoS, prioritások
    - Beengedés-szabályozás: biztosítja, hogy az új adatfolyam ne rontsa a meglevők minőségét

  Frekvenciasávok:
    - Engedélyköteles (QoS garantálható):
      - 2,500-2,690 GHz (MMDS) - csak USA
      - 3,410-3,600 GHz (ETSI) - csak EU
    - Engedélymentes (QoS nem garantálható):
      - 5,725-5,850 GHz

================= Kapcsolás, jelzés, címzés ====================

Kapcsolás: két nem szomszédos csomópont között "kapcsolat"
  - E.S. (End System) - végpont, felhasználó

  Fajtái:
    - áramkörkapcsolás
      - hullámhossz-kapcsolás

    - üzenetkapcsolás
    - csomagkapcsolás
      - virtuális áramkörkapcsolás

          | -> Broadcast (szóró) szghálók
          |
  hálók -> |
          | | -> Áramkör kapcsolt
          | -> Kapcsolt hálók -> | | -> Összeköttetés-mentes
                                  | -> Üzenet és csomagkapcsolt -> |
                                                                    | -> Összeköttetés-alapú

Összeköttetés-alapú: adatátvitel előtt felépül egy összeköttetés a végpontok között
Összeköttetés-mentes: előzetes összeköttetés létrehozása nélkül

Áramkörkapcsolás
  - dedikált fizikai kapcsolat, minden adat ezen megy
  - jellemzően nem állandó, fel kell építeni és le kell bontani
  - realtime adatátvitelre jó

  - torlódás csak az összeköttetés felépítése során jöhet létre
  - kapcsolatok általános felépítése:
    - kapcsolóelem (a tényleges kapcsolást végzi)
    - kapcsolóvezérlő (kiválasztja, hogy mely kapcsolóelemek legyenek aktívak)
  - blokkolás:
    - belső blokkolás: a kért kimenet szabad, de nincs útvonal a bemenettől a kimenetig
    - kimeneti blokkolás: két bemenet ugyanazt a kimenetet akarja használni
    - az egyszerű crossbar (keresztpontos) kapcsolóban nincs belső blokkolás
    - blokkolás megengedésével hatékonyabb kapcsológépet kaphatunk
  - időosztású kapcsológépek:
    - TDM jel memóriába írása, és más sorrendben kiolvasása
    - korlátos számú csatornát támogat csak, térosztású elvvel kombinálva használják
  - hullámhossz kapcsolás: WDM
    - különböző hullámhosszú fény egy üvegszálon -> külön logikai csatornák

Üzenetkapcsolás
  - az egész üzenet egyben megy át (store & forward)
  - üzenetben címrész kell
  - jobb csatorna kihasználtság
  - nincs ütközés, cserébe nagy késleltetés
  - lehet prioritásos, broadcast
  - adatátvitelre jó, médiára nem

Csomagkapcsolás
  - az üzenet csomagokra tördelődik
  - a csomag tartalmazza:
    - a küldő csomópont azonosítóját/címét
    - a címzett csomópont azonosítóját/címét
    - a csomag "helyét" az üzenetben
  - a csomagok egymástól független útvonalon, tetszőleges sorrendben érkezhetnek meg
  - csomagkapcsolt hálózat fontosabb elemei:
    - switch: helyi hálózatban továbbítás
    - router: hálózatok közötti továbbítás
  - csomagkapcsolás megvalósítható:
    - összeköttetés-mentes módon: datagram kapcsolás
    - összeköttetés-alapú módon: virtuális áramkörkapcsolás

Datagram-kapcsolás
  - minden csomag önálló egység
  - minden csomag tartalmazza a végpont globálisan egyedi címét
  - a két végpont közötti csomópontok
    - megvizsgálják a csomag fejrészét
    - kiválasztják az útvonal következő szakaszát
      - a választás során két tényező:
        - melyik az a csomópont, amely a csomagot a lehető legrövidebb úton juttatja rendeltetési helyére
        - hol található szabad csomópont, amely képes a csomag fogadására

Blokkolás csomagkapcsolókban:
  - belső és kimeneti is lehetséges
  - torlódás átmeneti, előre nem látható
  - elkerülések módjai:
    - túlbiztosítás: a belső kapcsolatok legyenek gyorsabbak, mint a bementek
    - párhuzamos kapcsolás: több útvonal kialakítása egy bemenet - kimenet pár között
    - pufferelés: csomagok késleltetése
    - visszaduzzasztás: kapcsoló szól a küldőnek, hogy ideiglenesen függessze fel az adást (csak korlátozott ideig hatásos)

Virtuális áramkörkapcsolás:
  - szakaszokból álló dedikált összeköttetés
  - minden csomag ezt az útvonalat használja
  - csomópontokban VCI (Virtual Circuit Identifier)

Csomag- vs áramkörkapcsolás:
  - Áramkörkapcsolás:
    - ha sok adatot sorrendhelyesen, állandó sebességgel, kis késleltetéssel kell továbbítani (pl média)
  - Csomagkapcsolás:
    - nem dedikált, burstös, késleltetés tűrő kommunikáció (pl levelek, weboldalak)

Jelzések:
  - összeköttetések létrehozása, lebontása

    - hívásvezérlő protokollok
  - 2 féle:
    - in-band
    - out-of-band
      - ha a jelző csatorna közös: CCS (Common Channel Signaling)
      - rugalmasság, jobb sávszélesség-kihasználás

Elnevezés és címzés:
  - név: google.com
  - cím: 173.194.39.136
  - kapcsolat: címfeloldás

Hierarchikus nevek:
  - prefix + domain, elválasztás pontokkal
  - DNS (Domain Name System)
  - root = üres string a záró pont után (hit.bme.hu.)
  - top level domains:
    - IANA (Internet Assigned Numbers Authority)
    - ország (.hu)
    - általános (.com)
    - infrastruktúra (.apra)

Címzés:
  - globális egyediek, hierarchikusak
    - hierarchikus címekkel egyszerűbb az útvonalválasztás
  - van nem prefix globális hierarchikus címzés is (Ethernet)
    - 3 byte gyártóé, 3 byte adapteré

Névfeloldás:
  - DNS végzi (névszerverek)
  - elvileg minden kérés először a rootnak megy
    - de szerver duplikáció (és anycast címzés)
    - cachelés

Több szintű címzés: MAC & IP együttélése

================= Routing ====================

Hálózati rétegben, IP alapján

Módszerek:
  - távolságvektor
  - linkállapot

Útvonalválasztás / csomagtovábbítás
  - a hálózat nem állandó
  - túl nagy, hogy pontos, aktuális információnk lehessen róla

Összeköttetés-alapú és -mentes hálózatokban is szükség van rá.
  - útvonal-kijelölés (egyszer) / -választás (minden csomagra)

Ismerni kell a csomópontokat, linkeket, lehetséges útvonalakat

Útvonaltáblák: cél - következő csomópont összerendelések
  - kitöltése: manuálisan / automatikusan (centralizáltan / elosztottan)

Centralizált:
  - egy pontban minden info a hálózatról
  - minden csomópontra meghatározza és elküldi az ő útvonaltábláját
  - konzisztens, de sérülékeny, nem up-to-date

Elosztott eset:
  - mindenki maga gyűjt infót és épít routingtáblát
  - mindenki mást hisz a hálózatról

Statikus routing:
  - telefonhálózatokra működött, internet túl dinamikus
  -> dinamikus, adaptív routing

Gyűjthető infók:
  - linkek sebessége, forgalma, kiszolgálási díja

Célok:
  kis routing tábla
    - olcsóbb csomópont, gyorsabb keresés
  robosztusság
    - lehető legkevesebb hibás tábla (pl "fekete lyukat" okozhat)
  optimális út
    - legrövidebb
    - legmegbízhatóbb
    - legolcsóbb

A lehetséges megoldások:
  - centralizált / elosztott
  - forgalomfüggő / -független
  - egy- vagy többutas
  - lepésenkénti / forrás általi

Elosztott módszerrel lehet a legjobb összekötöttséget elérni.

Módszerek:
  - Távolságvektor (Bellman-Ford): A csomópontok elmondják a hálózatról alkotott elképzeléseiket
  - Linkállapot (Dijkstra): A csomópontok elmondják a szomszédaikról nyert tapasztalataikat

A távolságvektor módszer:
  - az elképzelések: melyik csomópont milyen távol van (hopszám)
    - esetleg súlyozható átviteli sebességgel, sorbanállási hosszal vagy költséggel
  - ha egy csomópont a beérkező üzenetből kideríti, hogy rövidebb útra van lehetőség, akkor végrehajt egy cserét a vektorban
  - lehetséges hiba:
    - végtelenig számolás:
      - ha egy link megszűnik, de a szomszédoknak van múltbeli bejegyzése róla, akkor a múltbeli - már nem létező link - rövidebb utat jelent
        - ráadásul a nem létező út minden iterációban egyre hosszabb lesz, az ide küldött csomagok addig ping-pongoznak két csomópont között, amíg le nem jár a TTL (ha van)
  - javítás:
    - Split horizon (nem hirdetek útvonalat annak, akitől tanultam azt)
    - Route poisoning (rossz linkek hirdetése, ki kell törölni a táblából)
    - Holddown timer (elérhetetlen útvonalat csak adott idő után szabad frissíteni)
      - lassabb lesz tőle a protokoll
  - nem jól skálázódik
    - max egy ISP hálózatán belülre
  - lassan követi a változásokat
  - van jobb

Linkállapot módszer:
  - szomszédokhoz vezető linkek aktuális állapota
  - T (temporary)-be ismert szomszédok és távolságaik
    - ha egy szomszéd ismertnél rövidebb úton elérhető az egyik szomszéd tapasztalati alapján, akkor frissítsük
  - minden ütemben a legkisebb T-beli rakját át P-be (permanent)
  - ha T kiürül készen vagyunk

A hierarchikus routing
  - AS (Autonom System)-ek kialakítása
  - Egy AS kívülről egy "csomópont"
  - AS-ek globálisan egyedi azonosítót kapnak ASN (Autonomous System Number), a IANA-tól
  - Az AS-en belül a határ-router feladat az útvonal választás
  Típusok:
    - Multihomed: több mint egy másik AS-el van összekötve
    - Stub: csak egy másik AS-el van összekötve
    - Transit: csak átmenő forgalmat szolgál ki (ISP)

AS-ek közti útvonalválasztás:
  - Hot-potato elv:
    - minél rövidebb legyen az út a saját AS-én belül
    - annak ellenére, hogy lehet összességében így hosszabb lesz az út
  - Cold-potato:
    - minél hosszabb ideig a saját hálózatán belül akarja tartani a forgalmat, szinte a felhasználóig
    - szolgáltatás minőségi garanciákat nyújthat

Mobil végpontok:

  - Jó lenne, ha csak az eredeti helyét kéne ismerni (nem kell routing táblát frissíteni)
  - A Home agent (router) továbbítja a csomagot a Foreign agent felé

Multicast routing:
  - Nem egy végpont a címzett, hanem egy csoport
  Elnevezés:
    - broadcast: (műsor)szórás mintájára, mindenkinek a hálózaton
    - multicast: egy csoportnak
    - anycast: egy valakinek a csoportból (pl. aki a legközelebb van a forráshoz)
    - unicast: egy adott végpontnak

Csoportok címzése:
  - ne kelljen tudni minden elem címét
  - csoport cím
  - sűrű elhelyezkedés: elárasztás + lemondás (flood-and-prune)
  - ritka elhelyezkedés: explicit csatlakozás
  - elhelyezkedés-független: a kettő ötvözete

================= Internet Protocol ====================

Hálózati rétegben, feladata: címzés, útvonalválasztás, tördelés

Jellemzői:
  - csomagkapcsolt
  - összeköttetés-mentes
  - "best effort"

Címzés (IPv4):
  - 32 bites cím (kimerült)
  - hálózatazonosító + egyedi cím (hálózaton belül)

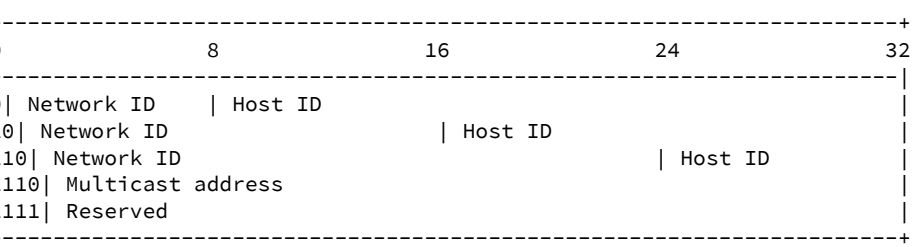
  Címosztályok (egyedi címzésre):

  +---+
osztály	hálózatazonosító	hálózatok száma	címek száma
A	8 bites	126	16 777 214
B	16 bites	16382	65534
C	24 bites	2097150	254
  +---+

  - IANA (Internet Assigned Number Authority) osztja ki őket

  D osztály:
    - multicast címzésre
    - 224.0.0.0-től 239.255.255.255-ig

  E osztály:
    - Fenntartott osztály és címtartomány
    - 240.0.0.0-től 255.255.255.255-ig



    - Osztályazonosító prefix (A: 0-127, B: 128-192, C: 192-223, D: 224-239, E:240-255)

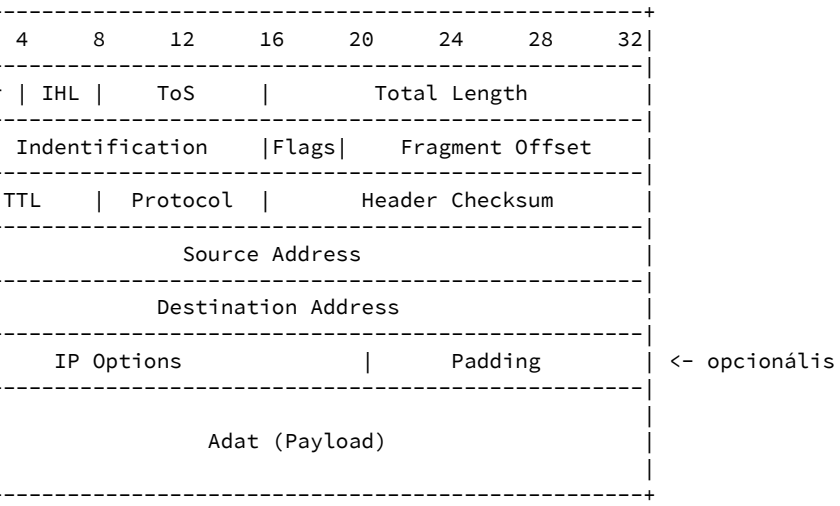
  Speciális címek:
    - Host ID = csupa nulla: hálózat címe
    - Host ID = csupa egyes: broadcast cím
    - 127.0.0.0 - 127.255.255.255: localhost, helyi gépet azonosítja

  Privát címtartományok:
    - első megoldás a címtartomány kimerülésére
    - pl cégen belül nem kell globálisan egyedi cím

    - nem kell hozzá RIR (Regional Internet Registry) jóváhagyás
    - kifele is kommunikálhat NAT-tal (Network Address Translation)
    - privát IP-címtartományok:
      - 10.0.0.0 - 10.255.255.255 (1 db A osztály)
      - 172.16.0.0 - 172.31.255.255 (16 db B osztály)
      - 192.168.0.0 - 192.168.255.255 (256 db C osztály)
    - késlelteti az IPv6 bevezetését
    - nem minden protokollra működik
    - két privát hálózat egyesítésekor mindenkit újra kell címezni
    - kiszivárogtak az internetre is -> felesleges terhelés a névszereknek
      - rosszul configolt routerek miatt

  CIDR (Classless Inter-Domain Routing)
    - VLSM (Variable Length Subnet Mask)
    - alhálózati maszk mondja meg, hogy hol van a hálózati és végpont azonosító határa
    - alhálózati maszkot az egyesek száma egyértelműen meghatározza
      -> pl: 152.66.0.0 /16 (16 db egyes = 255.255.0.0)

IPv4 fejrész:



  1. sor:
    Ver (Version) (4 bit) - IPv4: 4, IPv6: 6
    IHL (Internet Header Length) (4 bit) - IP fejrész mérete 4 byteos szavakban (min 5 (-> 20 byte), max 15 (-> 60 byte))
    ToS (Type of Service) (8 bit) - QoS osztályok, prioritások
      - 0-2 bit: precedencia
      - 3. bit: késleltetés (normal/low)
      - 4. bit: throughput(normal/light)
      - 5. bit: reliablity (normal/high)
      - használata: DSCP (Differentiated Services Code Point) - Diffserv, ECN (Explicit Congestion Notification)
    Total Length (16 bit) - A teljes csomag mérete (fejrésszel együtt) byteban (min 576, max 65535)

  2. sor (tördelés):
    Identification (16 bit) - Az IP-töredékek egyedi azonosítása
    Flags (3 bit)
      - 0. bit: fentartott
      - 1. bit: DF (Don't Fragment): ha tördelni kéne, akkor el kell dobni
        - MTU (Maximum Transmission Unit) Path Discovery ezt használja
      - 2. bit: MF - More Fragment
        - vannak-e még hátra további töredékek ebből a csomagból
    Fragment Offset (13 bit)
     - Az eredeti csomagban lévő kezdőpozícióját adja meg e töredékben lévő adatnak 8 byteos egységekben

  3. sor:
    TTL (Time To Live) (8 bit): csomag élettartalma hopszámban
    Protocol (8 bit): Az adatrészben lévő protokoll azonosítója
      - pl 1: ICMP, 2: IGMP, 6: TCP, 8: EGP, 17: UDP, 89: OSPF, 132: SCTP
    Header Checksum (16 bit) : A fejrész(!) minden 16 bitjére számolt egyes komplemens
      - minden csomag továbbításnál újra kell számolni (TTL változik)

  4-5 sor: A csomag feladójának és címzettjének az IP-címe (NAT-nál nem tényleges cím)

  6. sor:
    IP options: ritkán használt, pl útkijelölésre jó
      - SSRR (Strict Source Record Route) - pontos út kijelölése
      - LSSR (Loose Source Record Route) - kötelező útba ejtendő csomópontok kijelölése

    Padding: Az IP Options részt egészíti ki 4 bájt többszörösére

Útvonalválasztás
  "Hot potato"-elv
    - minél gyorsabban továbbítsuk: csak a következő csomópontot kell ismernünk
    - olcsó csomópontok, gyors
  Routing tábla:

Hálózat címe	Alhálózati maszk	Interfész	Közvetlenül kapcsolódik?	Következő csomópont
IP cím	/N	azonosító	igen/nem	IP cím

    - A táblázatban lévő hálózati címekkel (maszkolva) kell összehasonlítani az éppen beérkezett csomag hálózatát
      - Ha többel is egyezik, akkor az egyik leghosszabb egyezést válasszuk (leghosszabb hálózati maszk)
    - Bináris fával implementálják (gyorsabb keresés)
    - Saját címet érdemes külön megnézni, a táblázatban keresés előtt
    - Ha nincs találat -> alapértelmezett útvonal (vagy eldobja a csomagot, ha nincs alapértelmezett útvonal)

  Alapértelmezett útvonal: DG (Default Gateway)
    - erre megy a csomag, ha nem ismeri a célhálózatot
    - ehhez a hálózati cím: 0.0.0.0/0
    - végpontok gyakran az összes külső hálózati címre szánt csomagot az alapértelmezett útvonalra küldik

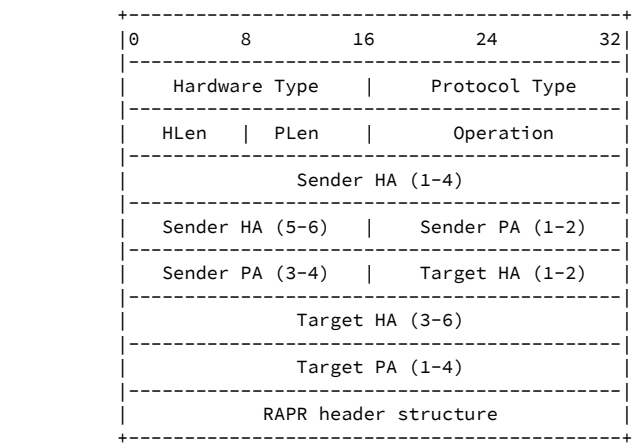
  A jobb metrikával rendelkező kapcsolaton küldjük ki a csomagot
    - pl elérhetőség, terheltség, késleltetés
    - más stratégia közvetlen és közvetett csomópontnál

  Közvetlenül kapcsolódó csomópont (LAN):
    - célszerű közvetlenül a címzettnek küldeni a csomagot, de ehhez kell a MAC címe

  Nem közvetlenül kapcsolódó csomópont:
    - küljük el a routernek, majd ő megoldja (ehhez kell a router MAC címe)

ARP (Address Resolution Protocol), RARP (Reverse ARP):
  - IP cím -> MAC cím "fordítás"
  - broadcast kérés: „Kinek az IP-címe a ...?”
  - az ARP üzeneteket közvetlenül az adatkapcsolati réteg protokolljának küldjük (Nem IP-csomag!)
  - sokféle protokollt támogat (nem csak Ethernet, és nem csak IP)

  ARP fejrész:



    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
    HA (Hardware Address) - MAC cím
    PA (Protocol Address) - IP cím

    MAC cím lekérdezésnél a Target HA csupa nulla (ismeretlen)
      - Az ethernet keretben a MAC cím csupa egyes(!) - broadcast

  ARP tábla:
    - MAC - IP párok
    - cachelés
    - lehetnek statikus elemek (kézzel felvitt), meg dinamikusak

  További alkalmazásai az ARP-nek:
    ARP probe: lekérdezi az IPv4 címet, hogy más használja-e azt
      - Broadcast ARP kérés, küldő IP címe csupa nulla, saját címét kérdezi le
    ARP hirdetmény: Ha változik valakinek a MAC vagy IP címe, üzenet, hogy mások tudják frissíteni
az ARP táblájukat (cache)
      - Request: Target PA = Sender PA, Target HA = 0
      - Reply: Target PA = Sender PA, Target HA = Sender HA

  RARP: adatkapcsolati rétegbeli címből IP-címet
    - hálózatmenedzsment
    - permanens tár nélküli eszközök
      - Semmi ismeretük nincs a hálózatról
      - Hálózatról töltődik be az operációs rendszer
      - IP-cím nélküli kezdeti kommunikáció
    - már nem használják, helyette:
     - BOOTP (Bootstrap Protocol): hálózatról történő betöltésre
     - DHCP (Dynamic Host Configuration Protocol): IP-cím kérésére

IP - Útvonalválasztó protokollok
  - útvonal irányítás
  - ki-/becsatlakozások kezelése
  Protokoll típusok:
    Ad hoc protokollok
      - kis és gyorsan változó hálózatokra
      - lehet proaktív, folyamatosan karbantartott táblákkal
      - reaktív, igény szerinti célfelderítés
      - hibrid (proaktív és reaktív együtt)
    IGPs (Interior Gateway Protocols)
      - kisebb hálózatokon pl egy AS-en belül
    EGPs (Exterior Gateway Protocols)
      - AS-ek között

  IGPs:
    Távolság-vektor módszer:
      - RIP (Routing Information Protocol)
        - egyszerű de lassú, hopszám alapján választ, max 15-ös hopszám (kis hálózatok)
      - IGRP (Interior Gateway Routing Protocol)
        - max 255 hop, többféle metrika
      - EIGRP (Enhanced Interior Gateway Routing Protocol)
        - VLSM-et használ
        - gyorsabb, megbízhatóbb mint IGRP és RIR (ez hurokmentes)
        - 6 metrika kombinációja

    Link-állapot módszer:
      - OSPF (Open Shortest Path First): nagy céges hálózatokban
        - LSDB (Link State DataBase)
        - gyors
        - többféle metrika
        - hierarchikus routing
        - IPv4 specifikus (IPv6-ra újra kellet specifikálni)
      - IS-IS (Intermediate System to Intermediate System): nagy kiterjedésű ISP hálózatokban
        - nem IPv4 specifikus

  EGPs:
    - EGP (Exterior Gateway Protocol): Sokáig az Internet EGP-je, ma már nem használt
    - BGP (Border Gateway Protocol): 1995-től használt EGP (BGPv4)
      - path vector protocol
      - egy routing tábla bejegyzés: | célhálózat | következő router | útvonal a célhálózatig |
      - a teljes útvonal dinamikus nyilvántartása
        - AS határ routerek végzik: útvonal-vektor üzenetekkel frissítés, hirdetve a többi AS elérhetőségét
        - útvonal-vektor üzenet: AS-ek sorozata a célhálózat felé

IP tördelés
  - a hálózatok alsóbb rétegei meghatározzák a keret maximális méretét
    - az adatkapcsolati réteg fej- és farokrészét leszámítva ez az MTU (Maximum Transmission Unit)
      - Ethernetnél 1500 byte, de LAN függő
  - ha tördelés szükséges, a TTL nem nulla és a DF nincs beállítva:
    - adat tördelése 8 byte-os egységekre
    - total length mezőt a töredék méretére kell állítani
    - Indentification-t be kell állítani (ha nem volt, generálni kell)
    - be kell állítani a fragment offsetet
    - utolsó kivételével az MF bit mindenhol 1-es
    - új ellenőrző összeg (a tördelés a TTL csökkentés után történik)

A töredékek összeállítása:
  - a végpont végzi (általában a címzett, de pl. NATnál nem)
  - ha egy darab nem, vagy hibásan érkezik meg, az összeset eldobja

A Router feladatai:
  - Hibás-e a csomag (fejrésze)?
  - Nekem címezték-e?
  - Ismerem-e a címzett hálózatát?
  - A TTL érték csökkentés után >0?
  - Kell-e tördelni? Lehet-e tördelni?
  - Kell-e visszajelzést küldeni?

ICMP és IGMP: az IP jelzés- és menedzsmentüzenetei

ICMP (Internet Control Message Protocol)
  - hibaüzenetek, kérések (pl echo), válaszok
  Pl:
    - Ping (RTT meghatározása): Echo request - Echo reply
    - Traceroute: Echo request - Echo reply + TTL (time exceeded hibaüzenet)
    - Source Quench: felszólítás a küldőnek a forgalma visszafogására
    - Redirect: felszólítás a küldőnek másik útvonal használatára
    - Address Mask Request: host lekéri a router-től az alhálózati maszkot
  - IP csomagba ágyazva, IPv4 fejrész után következik a fejrésze (8 byte)
    - best-effort, de különleges elbánással
  - fejrész: típus, típuson belül kód (pl miért történt a hiba), CRC, adat (pl hibás üzenete fejrészének eleje)

IGMP (Internet Group Management Protocol)
  - főként multicast csoportok kezelése
    - csoportok lekérdezése
    - csoporthoz csatlakozás
  - a TTL értéke általában 1 (csak a helyi hálózaton érvényes)
  - csak IPv4-hez, IPv6 máshogy oldja meg

================ IPv6 ====================

Az IPv4 hibái:
  - kevés cím
  - tördelés drága
  - nem biztonságos
  - mobilitás támogatás csak külön protokollal

A címkimerülést késleltették:
  - Privát IP-címtartományok
    - NAT (Network Address Translation)
  - CIDR (Classless Inter-Domain Routing)
    - VLSM
    - címek újraosztása
  - RIR-ek (Regional Internet Registry), ISP-k IP-címgazdálkodása

- Virtuális hosting (Név, IP illetve port alapú)

IPv6
- 128 bites címek
- 8x4 hexa szám pl: FEDC:BA94:7654:3210:FEDC:BA98:7654:3210
- vezető nullák elhagyhatók egy 4-es blokkon belül
  - FEDC:0094:0004:0000:000C:BA98:7654:3210
  - FEDC:94:4:0:C:BA98:7654:3210
- :0:-s blokkok elhagyhatóak, helyére :: (max egy darab, egybefüggő rész hagyható el)
- localhost: 0:0:0:0:0:0:0:1 helyet ::1
- Ha IPv4-es címből származik, akkor lehet "dotted decimal"
  - 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

  Prefixek:
  - 001: Unicast Address
  - 1111 1110 10 Link Local Use Address
  - 1111 1110 11 Site Local Use Address
  - 1111 1111 Multicast Address

  Címtipusok:
  Unicast: egyedi cím - pontosan egy csomóponthoz tartozik
  - Az IPv4-hez hasonlóan
  - Minden IPv6 csomópontnak legalább egy ilyen címe van

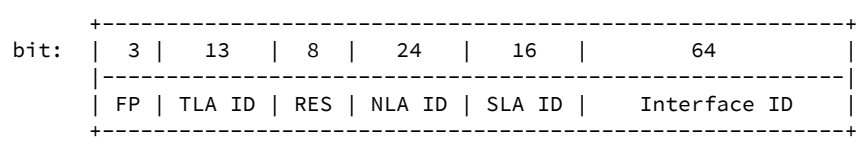
  Multicast:
  - csoportot azonosít
  - minden csoporton belüli csomópont megkapja az erre a címre küldött adatot
  - broadcast helyett is

  Anycast:
  - csoportot azonosít
  - biztosított, hogy a csoport egy csomópontja megkapja az erre a címre küldött üzenetet
  - pl. a küldőhöz legközelebbi

  Unicast címek
  Típusai:
  - Globális: Aggregálható vagy IPv4 kompatibilis
  - Link local: Prefix után a hálózati cím végig nulla, nem route-olható
  - Site local: (valószínűleg a gyakorlatban ezt nem fogják használni)
  - Beágyazott IPv4 címet tartalmazó (!= IPv4 kompatibilis)

  Globális unicast címek:
  - | global routing prefix | Subnet ID | Interface ID |
  - Interface ID megegyezés alapján 64 bit
    - Ethernet MAC címből származik: 48 bit -> 64 bit
    - Vagy DHCP-től random

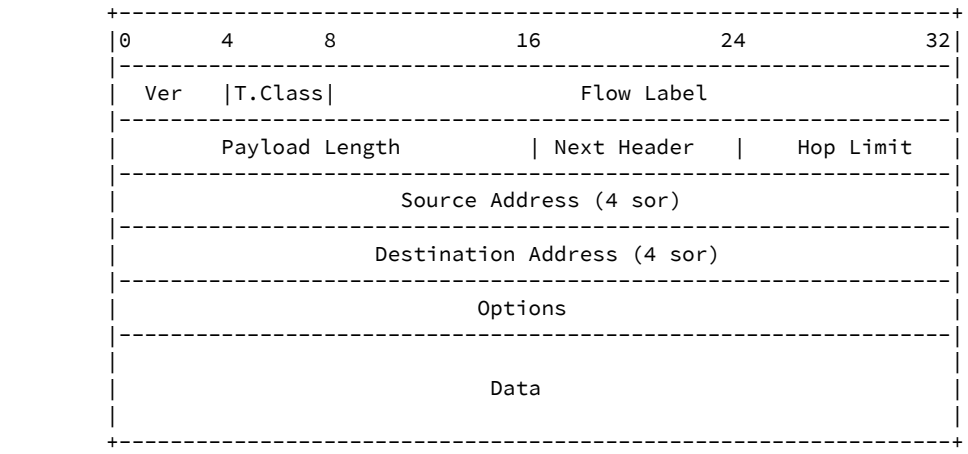
  Aggregálható globális unicast cím:



  FP: Format Prefix (001)
  TLA ID: Top-Level Aggregation Identifier (régió)
  RES: Reserved
  NLA ID: Next-Level Aggregation Identifier (szolgáltató)
  SLA ID: Site-Level Aggregation Identifier (előfizető és alhálózat)

  IPv4-es IPv6 címek:
  - 000-val kezdődik
  - IPv4 kompatibilis IPv6 cím pl. ::10.0.0.1
  - IPv6-ra képzett IPv4-es cím pl. ::FFFF:10.0.0.1

IPv6 fejrész:



- fix méret: 40 byte (IPv4 20-60)
- nincs CRC (gyorsabb feldolgozás)
- nincs tördelés (helyette Path MTU Discovery)

Verzió: IPv6 esetén 6 (IPv4:4)
Traffic Class: QoS, Prioritás
Flow Label: folyam azonosító címke
  - adott kapcsolatot azonosító generált mező: nem független datagrammok!
  - QoS és igazságos adatsebesség megosztást tesz lehetővé
  - Egy forrás-célállomás pár esetén több folyam is lehet
  - Adatfolyamot a cím és a flow label együtt azonosítja

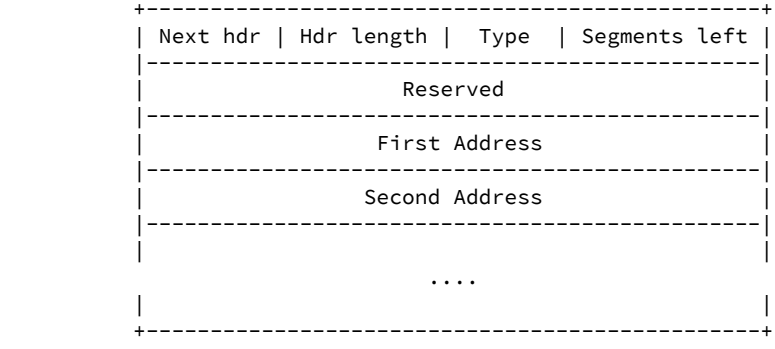
Payload Length: adathossz -> max 64 kB-os csomag
  - de jumbogram opció (4 GB)

Hop Limit = TTL

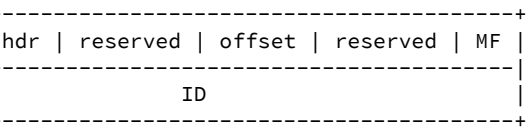
Next Header - következő fejrész
  2 lehetőség:
    - A beágyazott PDU típusát adja meg (hasonlóan az IPv4 Protocol mezőjéhez)
    - Az IPv6 fejrész kiterjesztését jelentő "Extension header" típusát adja

IPv6 opciók (header extension-ön keresztül):
  - Hop-by-hop Options Header (0)
    - QoS cselekvések csomópontonként
  - Routing Header (43)
    - csomópontok felsorolása, amelyeket útba kell ejteni
    - Az IPv4 opciókhoz hasonlóan szigorú / laza forrás forgalomirányítás
  - Fragmentation Header (44)
    - Mint az IPv4-ben, de csak a forrás darabolhat
  - Authentication Header (AH) (IPSec-ből) (51)
  - Encapsulation Security Payload Header (ESP) (IPSec-ből) (50)

IPv6 fejrész kiterjesztések
  Routing Extension
    - az érintendő csomópontok IPv6 címe (max 24 db)
      - következő csomópont megtalálása anycast címzéssel



Fragmentation Extension
  - Mint IPv4
    - 32 bit Fragment ID, 13 bit Fragmentation offset, 1 bit MF (nincs DF)
  - Tördelni csak a feladó fog
  - Összeállítani csak a fogadó fog



IPsec az IPv6-ban
  - Két Extension Header a biztonság növelésére
  - IPSec-re minden IPv6 csomópontnak képesnek kell lennie

  - Authentication Header
    - Valóban a látszólagos feladó küldte-e?
    - Lett-e módosítva a csomag?

  - Encapsulating Security Payload Header
    - Titkosított a csomag tartalma

NDP (Neighbor Discovery Protocol)
  - ARP, ICMP Router Discovery, Redirect helyett
  - ICMPv6 üzenetekkel
  - Automatikus címkonfiguráció (Stateless Autoconfiguration)
    - Automatikusan minden beállítást megkap, nem kell DHCP!
    - Duplikált cím észlelése

  ICMPv6 üzenetek:
    - Router hirdetés:
      - Hoszt automatikusan kiépíti a default router listáját
      - Link prefix, link MTU terjesztése
    - Router kérelmezés
    - Redirect: routerek megmondják a jobb útvonalakat a hostoknak
    - Szomszéd kérelmezés (Neighbor Solicitation)
      - Szomszéd adatkapcsolati címének lekérése (ARP helyett)
      - Szomszédok elérhetősége/elérhetetlensége
      - Cím duplikáció felderítés
    - Szomszéd hirdetés (Neighbor Advertisement)
      - pl. új adatkapcsolati cím hirdetése

    NDP adatbázisok (cache)
      - szomszédok, célcímek, prefixek, default routerek

Állapotmentes automatikus címkonfiguráció
  - duplikáció ellenőrzés (multicast szomszéd kérelmezés)
  - ha egyedi a linken: link-local multicast router kérelmezés
  - routerek hirdetéssel felelnek konfigurációs paraméterekkel
  - cím: routerek által közölt prefix + link local cím
  - Ha alkalmazásnak nem felel meg: lehet állapotfüggő konfiguráció (DHCPv6)

Szükséges eszközök:
  - "Okos" routerek: DHCP-funkcionalitás
  - Anycast és Multicast címzés

Routing protokollok:
  - RIPng
  - OSPFv3
  - IS-IS Extensions for IPv6
  - EIGRP for IPv6
  - MP-BGP

IPv6-ra áttérés:
  - alkalmazások támogatják

  Áttérés Dual Stackkel:
   - A csomópontok egyszerre IPv4 és IPv6 csomópontok
   - Külön hálózati és szállítási réteg a két protokollnak

   - Protokollválasztás: DNS szerver IPv6 vagy IPv4 címmel válaszol

  Áttérés Dual Layerrel
    - Mint a Dual Stack, de a szállítási réteg közös (több alkalmazás működhetne, ha nem használnak IP címeket)

  Áttérés alagutazással (Tunneling)
    - Beágyazása az egyik protokoll változatú csomagnak a másikba

================ IPv6 Transition ====================

IPv4 -> IPv6
  - 1983. 01. 01. (Flag day) NCP (Network Control Program) -> TCP/IP sikerült, ma már nem megy
  - hosszú átmenet, együtt kell működniük

Problémák:
  IPv6 kliens, IPv4 szerver: DNS64 + NAT64
  IPv4 kliens, IPv6 szerver: DNS46 + NAT46
  Végpontok IPv6-osak, de környezet IPv4: 6in4 tunnel
  Végpontok IPv4-esek, de környezet IPv6: 4in6 tunnel
  IPv6 képes kliens IPv4 környezetben (csak IPv4 címet kapott), IPv6 szerver: 6to4

NAT (Network Address Translation)
  - IP eredetileg végpont - végpont kommunikáció
  - Az alkalmazások arra számítanak, hogy a címek és portszámok a hálózati átvitel során változatlanok
  - De IPv4 címek kifogytak:
    - belül privát IP címek, kifele címfordítás

  Alapötlet:
    - A kimenő router kicseréli a forrás IP-címét a sajátjára
    - A router - valahogy - továbbítja a választ a forrásnak
      - Minden privát IP-hez hozzárendel egy kimenő portot
        - A port alapján tudja, hogy kinek jött a beérkező csomag
        - Azt a portszámot is meg kell jegyeznie, amin a belső hálózaton kommunikálnia kell a csomóponttal
        - belső IP - belső port - külső port összerendeléseket tárol

  Ez a megoldást:
    - Source NAT-nak (SNAT) hívjuk, ha a router publikus IP-címe fix,
    - Masquerade-nek, ha DHCP-vel kapta azt

  Hívják még NAPT-nak (Network Address and Port Translation) is vagy many-to-one NAT-nak
    - a basic NAt vagy one-to-one NAT pedig csak az IP címcsere

A másik irányú feladat az, hogy pusztán privát IP-címmel rendelkező gépeket elérhetővé tegyünk az Internet felől.
  - NAT-nál kívülről csak válaszolni lehet, kezdeményezni nem.

DNAT (Destination NAT) = port forwarding
  - a router az adott portjára érkező datagramokat egy meghatározott privát IP című gépnek továbbítja úgy, hogy a célcímet kicseréli a csomagban.
  - ICMP-nél nem működik (nincs portszám)
    - kivéve ha az egy ICMP hibaüzenet, amibe benne van a TCP vagy UDP fejrész első 8 byteja, ami tartalmazza a portot.
    - de az ICMP fejrész valamelyik mezője felhasználható erre a célra.

Alkalmazások gyakran azt hiszik, hogy a hálózatban az IP és a port változatlan marad
  - Pl. FTP aktív módban megmondja a szervernek, hogy az melyik porton építse fel a kapcsolatot
    - Ez privát IP-címmel nem megy, de protocol helper meg tudja oldani
  - Ez a probléma NAT64-nél is előjön

IPv6-only kliens és IPv4-only szerver
  DNS64 szerver:
    - Ha nincs IPv6 cím, csak IPv4, vagy egy IPv4 csomópontot kell azonosítani IPv6 hálózatban akkor az IPv4 címből generál egy speciális IPv6 címet
      - 64:ff9b::/96 well known prefix + az IPv4 cím
      - vagy egy NSP (Network-Specific Prefix) + az IPv4 cím

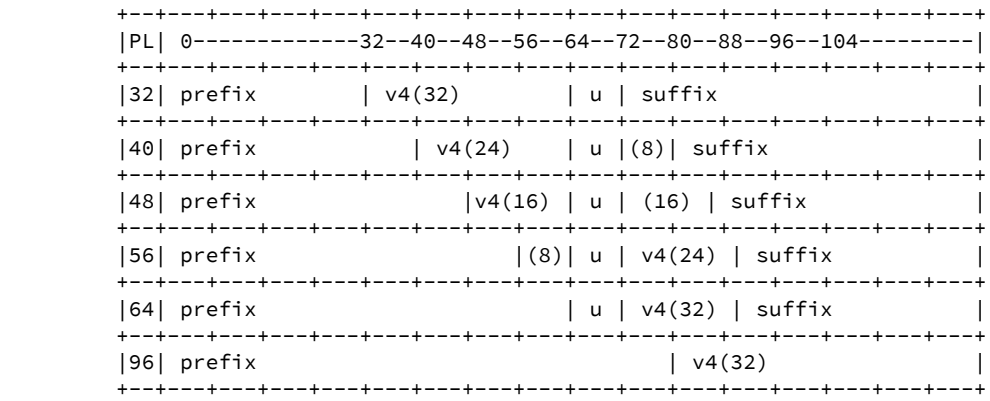
  A megoldáshoz szükséges:
    - A kliensben névkiszolgálóként a DNS64 szerver van beállítva

    - Az útválasztási táblázatok szerint a well-known prefix felé az út egy NAT64 átjárón keresztül vezet (Anycast címzés használható)

  Példa:
    - Az IPv6-only kliens csatlakozni szeretne az IPv4-only szerverhez
    - Lekéri a szerver IPv6 címét a szimbolikus neve alapján
    - Megkapja a szerver IPv4 címét tartalmazó speciális IPv6 címet
    - IPv6 csomagot küld a kapott címre
    - A NAT64 átjáró kapja meg, IPv4 csomagot készít belőle, és továbbküldi
      - A célcím az IPv6 cím utolsó 32 bitje
      - A forráscím a NAT64 átjáró IPv4 címe
    - Az IPv4-only szerver megkapja, és válaszol a NAT64 átjárónak (tőle kapta)
    - A NAT64 továbbítja az IPv6-only kliensnek
      - Forráscímként az a speciális IPv6 cím kerül, amit a DNS64 szerver generált
      - A NAT64 a portból ki tudja találni, hogy ki a kliens -> az ő IPv6 címét írja be célcímnek

IPv4-Embedded IPv6 Address
  - IPv4 IPv6 címbe ágyazva
  - Két fajtája van:
    - IPv4-Converted IPv6 Address (IPv4 állomást képvisel IPv6 hálózatban) <- ez az eset áll fenn NAT64 + DNS64-nél
    - IPv4-Translatable IPv6 Address (IPv4 eszközök által is elérhető IPv6 állomás)
  - A prefix mérete szigorúan csak 32, 40, 48, 56, 64 vagy 96 lehet
  - Az IPv6 címben 64-71 biteknek 0-nak lenniük
  - Az IPv6 cím 32 bitjét általában a prefix után írjuk, de a 64-71 bitek helyét „átugorjuk”

A címek lehetséges formátuma
  - PL: prefix length, v4: IPv4 cím bitjei, u és suffix: 0 értékű bitek



6in4 tunnel
  - IPv6 szigetek csak IPv4 hálózaton keresztül tudnak egymással kommunikálni
  - Az IPv6 csomagokat IPv4 csomagokba csomagolja be
  - Az IPv4-ben az 41-es protokollt használja az IPv6 csomagok azonosítására

6to4
  - IPv6 képes eszköz IPv4-only környezetben van
  - IPv6 protokollal egy másik IPv6-os eszközt szeretne elérni (akár az is lehet IPv4-only környezetben)
  - A 6to4 megoldás egy "automatikus" tunnel, ami az IPv6 csomagokat IPv4 csomagokba csomagolja be (41-es protokol)

  Megvalósítás szempontjából kétféle konfiguráció lehetséges:
    - Az IPv6 kliens maga végzi a becsomagolás (6to4 host)
    - Egy IPv6 hálózat, aminek a routere végzi a becsomagolást (6to4 border)

  A kicsomagolás a 6to4 relay feladata
    - Ilyen több is lehet, legközelebbi a 192.88.99.1 anycast címen

  A megoldás működésének lépései, ha a cél natív IPv6
    - A kliens a szerver felé IPv6 csomagot küld
    - A becsomagolás elvégzője (host vagy router) az IPv6 csomagot IPv4 csomagba ágyazza, és az IPv4 segítségével elküldi egy 6to4 relaynek
    - A 6to4 relay megkapja az IPv4-be ágyazott IPv6 csomagot, kicsomagolja és továbbítja a natív IPv6 hálózatba
    - A natív IPv6 hálózatban a csomag megérkezik a címzetthez
    - A címzett válaszol

    - A válasz visszaér vagy az előbbi vagy egy másik 6to4 relayhez
    - Visszafele a 6to4 relay az IPv6 csomagokat IPv4-be csomagolva küldi a korábban a kliens IPv6 csomagját IPv4-be csomagoló eszköznek (host vagy router). - Ennek publikus IPv4 címe van!
    - A host vagy router kicsomagolja az IPv4 csomagból az IPv6 csomagot és eljuttatja a kliensnek

  Működéséhez szükséges feltétel: a kliensnek legyen publikus IPv4-címe: Ha nincs neki, akkor Teredot kell helyette használni
    - A kliensnek lehet privát IP címe, a hoszt pedig használhat NAT-ot, 6to4 + NAT = Teredo

  A 6to4 továbbfejlesztett változata a 6rd

  A 6to4 segítségével IPv6 szigeteket is összeköthetünk IPv4 fölött.
    - A host a cél IPv6 címéből (6to4 cím) tudja meg, hogy nem egy IPv6 relaynek, hanem egy IPv4 eszköznek küld csomagot.

  A 6to4 címzéshez 2002::/16 prefixet foglalták le. A címek képzése:
    Hálózatcím:
    - 2002::/16 prefix + publikus IPv4 cím 32 bitje + 16 bit subnet ID
    - Host esetén a subnet ID egy generált véletlen szám
    - Ha a 6to4 mechanizmust router használja, akkor akár több IPv6 hálózat is lehet mögötte, ekkor hasznos a subnet ID.
    Gépcím:
      - A szabványos módosított EUI-64 (Ethernet 48 bitje -> 64 bites azonosító) azonosító

  Ilyen módon a 6to4 minden publikus IPv4 címhez egy 2002::/16 kezdetű, /48 méretű IPv6 címtartományt rendel
    - Mindegyik "mögött" elérhető lehet egy ilyen méretű IPv6 hálózat

================= Forgalomszabályozás ====================

Forgalomszabályozás (flow control):
  - az adatforrás az aktuális átviteli sebességét illeszti a vevő és a hálózat kiszolgálási sebességéhez

Torlódásvezérlés (congestion control):
  - csomópontok, linkek időszakos túlterheltségét próbálja megszüntetni, vagy megelőzni
  - a flow control a congestion control egyik eszköze

Célok:
  - legyen egyszerű, stabil, igazságos
  - minimális hálózati erőforrást vegyen igénybe
  - ne függjön a források számától (skálázható legyen)

Visszacsatolás:
  - Nyílthurkú: nincs visszacsatolás
  - Zárthurkú: van visszacsatolás
  - Hibrid

Nyílthurkú:
  - felhasználó elmondja a forgalmi igényeit, a hálózat dönt, hogy beengedi-e
    - ha igen, akkor erőforrások dedikál neki, és ellenőrzi, hogy ennél több ne használjon

  Forgalomleírók ("forgalmi igények"):
    - csúcssebesség
    - átlagsebesség

Szereplők:
 - forgalomszabályozó (regulator): késlelteti a túlzott forgalmat
 - a felügyelő (policer): eltávolítja a bottlenecket a hálózatból

Zárthurkú szabályozás:
  Feltétlenül szükséges, ha:
    - nincs erőforrás-foglalás
    - túlfoglalást (overbooking) alkalmazunk statisztikus nyereség elérése érdekében

Típusai:

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. sebesség	lépések

on-off: a nyelő engedélyezi az adást

stop-and-wait: a küldő minden csomag után megvárja a nyugtát
- kihasználtság (L: csomaghossz, R: adatátviteli sebesség, RTT: körbefordulási idő): U = L/R / (RTT + L/R)

statikus ablak: a küldő az ablak méretével megadott számú csomag elküldése után vár csak nyugtára
- sorszámozásra és tárolásra van szükség
- ha az ablak mérete 'k', akkor U = min(k * L/R / (RTT + L/R), 1)
- ACK-ra várás - timeout kell (RTT függvénye)
- szabály kell arra, ha nem jött ACK

Szállítási protokollok

Hálózati réteg: végpontok („host”-ok) közötti logikai kapcsolatok

Szállítási réteg: alkalmazások (process) közötti logikai kapcsolatok
- forgalom szabályozás
- multiplexelés
- hibadetektálás, javítás (pl. Automatic Repeat reQuest - ARQ)
- sorrendhelyes átvitel
- csomagképzés (szegmensek) és csomagok visszaállítása a felsőbb rétegek számára

UDP (User Datagram Protocol)
TCP (Transmission Control Protocol)

Az UDP és TCP közös képességei:
- Portok kezelése
- Multiplexelési képesség

Alapvető különbség az UDP és a TCP között:
- UDP összeköttetésmentes (connectionless)
- TCP összeköttetés-alapú (connection-oriented) transzport-szolgáltatást nyújt

Portok kezelése:
- Az IP réteg egy csomópontot címez meg ("egy gépet")
- A végpontokon belül az alkalmazások a portok segítségével különböztethetőek meg
- Foglalt portok (0...1023)
  - 80: HTTP, 21: FTP, 69: TFTP

Multiplexelés: egy alkalmazáshoz port rendelése, csomagba a portszám beírás
Demultiplexelés: csomagok eljuttatása a megfelelő alkalmazáshoz a port alapján
- ezek alapja a socket
- mux/demux-ot általában az oprendszer végzi

Socket: interfész az alkalmazás és a hálózat között
A socket-et az alábbiak jellemzik:
- transzportprotokoll (UDP v. TCP)
- saját IP cím
- saját portszám

→ helyi socket cím

- (opcionális) távoli IP
- (opcionális) távoli port

→ távoli socket cím

→ socket pár

Leegyszerűsítve: socket = IP cím + portszám

  Típusok:
    Datagramm socket: UDP
    Stream socket: TCP
    Raw (IP) socket: ICMP, IGMP, OSPF
      - itt nem kell portokat kezelni

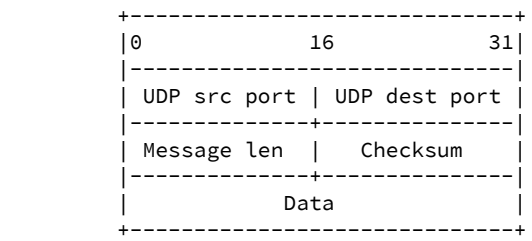
TCP kliens-szerver socket kezelés:
  - szerver létrehoz "hallgató" módban lévő socketeket: várja a kliensek kapcsolatfelvételét
  - kapcs. felvétel után dedikált socket minden kapcsolathoz
    - virtuális áramkörkapcsolás (full duplex)
  - szerver különböző TCP socketeket hozhat létre ugyanazzal a helyi IP címmel és port számmal
    - mivel a távoli host IP címével, portjával más socket párt alkot
    - szerver gyerek processz összerendelése a kliens processzével

Alsóbb rétegekben (Router, Switch):
  - tűzfalak, NAT-ok és proxy szerverek figyelik az aktív socket párokat és fenntartanak socket interfészeket
  - ütemezéshez, QoS támogatáshoz routerekben a csomagfolyamokat a socket párokkal lehet azonosítani

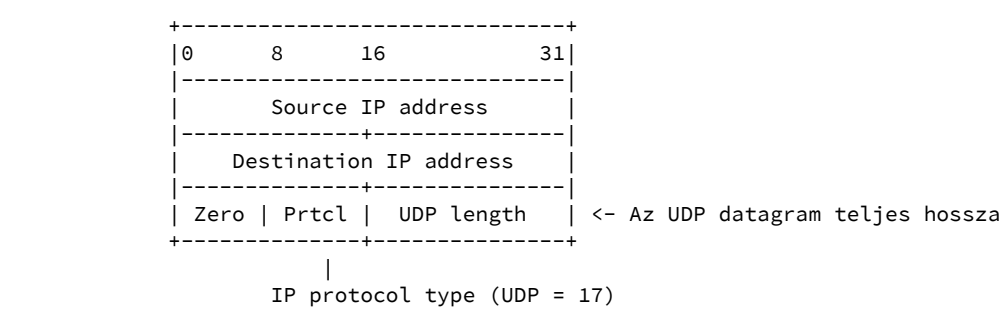
Egyszerű (raw) socket:
  - Közvetlenül az alkalmazásnak továbbítja a csomagot a fejléccel
  - Nincs TCP/IP feldolgozás, alkalmazás látja el fejléccel/veszi le a fejlécet
  - Nem biztonságos

UDP (User Datagram Protocol)
 - Multiplexálást (port kezelést) és opcionálisan adatintegritás ellenőrzést nyújt

UDP fejrész:



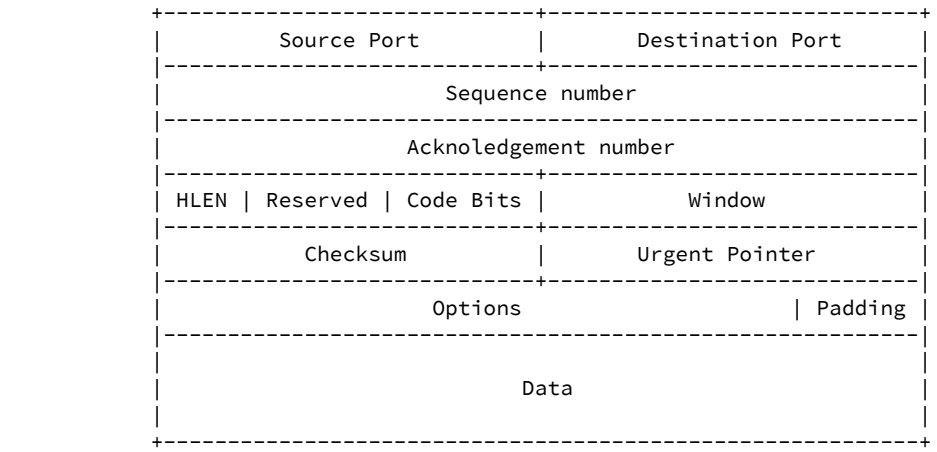
  Source port: opcionális
  Length: az adatrész hossza byteban, max (65 535 - 8 - 20)
  Checksum: opcionális (nincs: 0, IPv6-nál már nem opcionális)
    - Egy "pszeudo-fejléc" alapján 1-es komplemens 16 bites szavakra
    - A pszeudo-fejléc az IP címeket is tartalmazza az IPv4 headerből



UDP alkalmazása:
  - Broadcast, multicast (TCP nem képes rá)
  - Média: streaming, valós idejű játékok, VoIP, IPTV
  - Gyors és rövid lekérdezések: DNS, DHCP, RIP
  - Forgalomszabályozás: DCCP-vel (Datagram Congestion Control Protocol)

TCP (Transmission Control Protocol)
  Jellemzői:
    - Virtuális összeköttetések: összeköttetés épül fel és marad fenn a kommunikáció tartamára
    - Stream-típusú szolgáltatás: bytestreamek sorrendhelyes átvitele
    - Strukturálatlan stream: nincsenek határolók a streamen belül
    - Pufferelt átvitel: a streamből a datagramm megtöltéséhez szükséges mennyiséget várja össze
    - Duplex kapcsolatok: két független stream
    - Vezérlő információk küldése: az ellenkező irányban folyó streambe ágyazva (piggybacking)

Fejléc:



Sequence no.: a szegmensben levő adat első byte-jának pozíciója a küldő byte stream-jében
Ack no.: annak a byte-nak a sorszáma, amelyet a forrás legközelebb vár
HLEN : fejléc mérete, minimum 20 byte, max 60 byte
Code bits (flags): URG, PSH, ACK, RST, SYN, FIN bitek a kapcsolat kezeléséhez használt jelzőbitek
Window : a küldő ismertté teszi a vételi pufferének méretét
Checksum : mint az UDP-ben (pszeudorandom)
Urgent pointer : ha URG=1, a szegmens „urgent” részt tartalmaz, ilyenkor a végére mutat (pl. jelszóküldés)

Hívásfelépítés a TCP-ben:
- "3-way handshake"
  Az elküldött csomagok:
  - A: SYN: seq=x
  - B: SYN: seq=y, ACK x+1
  - A: ACK y+1 (lehet benne adat)

Híváslebontás a TCP-ben:
- "Modified 3-way handshake"
  - A: FIN seq=x
  - B: ACK x+1, FIN seq=1
  - A: ACK y+1

Összevont nyugta:
- ha X nyugta elveszik de X+Y-ra kap nyugtát (vagyis X+Y-ig minden csomag megjött)
- akkor a végpont ebből tudja, hogy X-et nem kell újraadnia
- nem tökéletes, ha 1000 csomagból az első veszik el, a fogadó ezt nem tudja jeleni
  - megoldás: szelektív nyugtázás: SACK (Selective ACK)
    - opcionális fejrészmezőben
    - népszerű

Várakozás ACK-ra: time-out (RTT függvénye)
- elveszett csomagoknál visszafogja magát

Gond a sorrend keveredéssel:
- a küldő elveszettnek hiszi a csomagot-> forrás visszafogás, amikor nem kéne
- D-SACK (Duplicate SACK): Fogadó szól, hogy az újraküldött csomag duplikátum -> visszagyorsul a forrás

Fast retransmit
- A time-out idő gyakran túl hosszú -> nagy késleltetés, ha valami elveszik
- Ha a vevő hézagot vesz észre a vett szegmensek sorozatában (elveszett csomag) akkor újból lenyugtázza a megelőző helyesen vett szegmenst.
- Fast retransmit szabály: ha az adó 3 egymást követő ACK-t kap (plusz az eredeti ACK) ugyanarra a szegmensre, feltételezi, hogy az
  azt követő szegmens elveszett és újraküldi azt még mielőtt lejárna a timeout.

MSS (Maximum Segment Size)
- A legnagyobb adatméret byteban, amit a TCP hajlandó küldeni egy szegmensben
- Össze kell egyeztetni az adatkapcsolati réteg MTU-jával
- TCP kapcsolatfelépítésnél kell egyeztetni, MSS opció a fejlécben
- TCP adó használhat Path MTU discovery-t is: dinamikus MSS változtatás

  Forgalomszabályozás: a csúszóablak
    - az ablak mérete megadja a "kintlevő", nyugtázatlan csomagok max. számát
    - A vevő közli a szabad helyének a méretét (RcvWindow) a küldött szegmensben
    - Az Adó legfeljebb RcvWindow mennyiségű nyugtázatlan adatot küld

    Problémák:
      Ha a vevő nullás csúszóablak méretet hirdet: adó leáll a küldéssel
        - Ha elveszik a vevő csomagja az új csúszóablak méretről, az adó vár hiába
        - Megoldás: adó egy timert indít, lejárta után felkéri a vevőt, hogy küldjön ACK-ot az új ablakméretről
      Buta ablak jelenség
        - Ha a vevő oldalon kicsi (akár 1 byte) szabadul fel, lehet 1 byte az ablak
        - A küldő 1 byte-ot küld, a vevő megint 1 byte-tal nyit
        - Erőforrás pocsékolás: kisebb az adat mint a fejléc!
        - Megoldás: A vevő nem nyitja az ablakot, csak akkor, ha MSS nagyságrendűt nyithat
          - A küldő nem küld, hacsak nem MSS-nyit küldhet vagy mindent küldhet, amit az alkalmazás kért

  Torlódásvezérlés a TCP-ben:
    Két fő módszer:
      "Hálózat által segített" (network assisted) torlódásvezérlés:
        - A hálózati elemek szolgáltatnak információt a túlterhelésről az adónak
        - Pl. TCP/IP ECN, ATM: külön jelzéscsomagok ehhez
      Végpontok közötti (end-to-end) torlódásvezérlés:
        - Nincs visszacsatolás a hálózatból, a végpont következtet arra, hogy torlódás léphetett fel
        - TCP ezt használja

    TCP implementációja: növeljük az adatsebességet amíg nincs torlódás, csökkentsük ha van
      - congestion window, CongWin (max adatsebességet befolyásolja)
      - növeljük a CongWin-t minden RTT alatt MSS-sel, amíg vesztést nem érzékelünk
      - csökkentsük a CongWin-t felére minden vesztéskor
        - többszörös nyugta ugyanarra a szegmensre
      - AIMD (Additive Increase - Multiplicative Decrease)

    Az AIMD kiegészítései:
      "Slow Start": az összeköttetés kezdetén minden ACK hatására CongWin kétszerezés az első vesztésig, utána AIMD
      Eltérő viselkedés timeout és többszörös (3-szoros) nyugták esetén
        - 3 duplikált ACK -> CongWin felezés
        - timout lejér -> CongWin = 1 MSS (ez sokkal rosszabb torlódási helyzetre utal)
          -> vissza "Slow Start" állapotba

    TCP-nél a CongWin max értéke 8 MSS

    Kétféle implementáció:
      - Tahoe: 3 D-SACK hatására is slow start 1 MSS-ről
      - Renoe: 3 D-SACK -> CongWin felezés

    Vegas TCP:
      - megpróbálja előre jelezni a torlódást, mielőtt elveszne csomag
      - RTT alapján jósolja meg, előre csökkenti az ablakot ha kritikus az RTT
      - Ha más csomópontok Renoe-t használnak, akkor túl sok értelme nincs

    Átlagsebesség:
      - Csomagvesztés előtt az átbocsátás: W/RTT
      - Vesztés után az ablak W/2 lesz, átbocsátás: W/2RTT
      - Közte lináris -> átlagos átbocsátás: 0.75 W/RTT

    TCP fairness
      - két kapcsolat, egyforma MSS és RTT -> fair
      - de kisebb RTT-vel rendelkező kapcsolatok gyorsabban növelik a CongWin-t -> nagyobb átviteli sebesség
      - UDP nem fair, nem fogja vissza magát torlódáskor, kiszorítja a TCP-t
      - párhuzamos TCP kapcsolatok
        - Böngészők több párhuzamos TCP kapcsolatot építenek fel a webszerverhez -> minél több kapcsolat, annál nagyobb átviteli sebesség

================= Multimédia IP felett ====================

Beszédjel feldolgozása:
  Analóg-digitális átalakítás:

- A 64 kbit/s-os beszéd: 8 bit 125 µs-nként (8 kHz)
- Inaktív szakaszok kivonása: VAD (Voice Activity Detection)
- Tömörítés
- Csomagokká alakítás

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: 5 - 20 ms

Videó-tömörítési formátumok
MPEG-1: Kb. VHS videó, CD audió minőség, 1,5 Mbit/s
MPEG-2: DVD, digitális tv-adások, 3...10 Mbit/s (SD minőség), 10...20 Mbit/s (HD minőség)
MPEG-4: Nagyobb tömörítés, jobb minőség az MPEG-2-hez képest, tipikusan 1,5 Mbit/s (SDTV), 8 Mbit/s (HDTV)

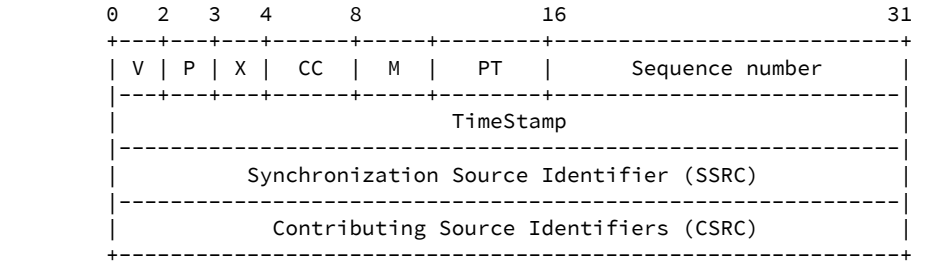
UDP & IP mellé kellenek még + protokollok (szállítási rétegben):
Médiakezelés: RTP, RTCP, RTSP
Hívásvezérlés: (H.323) és SIP
QoS: DiffServ

RTP (Real-time Transport Protocol):
- Payload-típusok, sorszámozás, időbélyeg

RTCP (Real-Time Transport Control Protocol):
- végpontok közötti QoS-monitorozás
- médiastreamek közötti szinkronizálás

UDP portokat használnak párban - RTP: 2*k, RTCP: 2*k + 1, ahol k egész

RTP fejrész:



V: Version (2 bit)
P: Padding (1 bit)
X: Extension (1 bit) - van-e kiterjesztés (változó hosszú, a header után)
CC: CSRC count (4 bit) - a CSRC azonosítók száma = a multiplexált források száma
M: Marker (1 bit) - valamilyen esemény történt
PT: Payload Type (7 bit) - az adat típusa ("profile"), kódolása
- pl G.711, G.722, GSM, JPEG, MPEG2

Sequence number (16 bit) - egyedi sorszám (kezdőértéke véletlen, utána egyesével növekszik)
- csomagvesztés detektálás, csomagsorrend helyreállítás

Timestamp (32 bit) - az első bytenak megfelelő pozíció valódi ideje a médiafolyamban
SSRC (32 bit) - Az RTP csomagfolyam forrását azonosítja, az RTCP rendeli hozzá, véletlenszerűen
- pl egy mikrofont vagy kamerat azonosít. Egy forrástól akár több stream lehet, egyedi SSRC-vel.

CSRC (0...15-ször 32 bit)
- contributing source: az RTP mixer által létrehozott kombinált csomagfolyam komponensét azonosítja

RTP mixer:
Közbülső rendszer, amely fogadja az RTP-csomagokat egy vagy több forrásból
- megváltoztathatja az adatformátumot
- kombinálja a csomagokat (az új csomag SSRC-je a mixer, de CSRC-be felsorolja az eredeti forrásokat)
- kombinált streamre új időzítés

RTCP (Real-time Transport Control Protocol)
  Végpontok között információt szolgáltat a minőségről a kapcsolat résztvevőinek:
    - késleltetés
    - jitter
    - vett csomagok, elveszett csomagok stb.
    - RTT
  Hatására az alkalmazás kodeket válthat, vagy folyamot korlátozhat
  Folyam max 5%-át használhatja
    - sok felhasználó esetén ritkán jön RTCP jelentés
      - elavult jelentés miatt rossz döntés QoS szempontból
    - megoldás: hierarchikus aggregáció
      - több jelentés összevonása egy összefoglaló jelentésbe
      - összevonás: node-ok hierarchiája, fastruktúra

RTSP (Real Time Streaming Protocol)
  - Kapcsolat felépítése és ellenőrzése a session végpontok között
  - VCR-jellegű funkciók
  - Általában RTP-t használ
  - Állapotfüggő, sok kérés típus

Fejrésztömörítés
  IP + UDP + RTP fejrész helyett egy fejrész
    - Redundáns információ (pl. megvan az adatkapcsolati fejlécben)
    - Sok mezőben nincs változás a kapcsolat alatt
    - Vannak mezők, amelyek változnak, de jósolható módon, pl. sequence number
  Először tömörítetlen és teljes fejrész elküldése
  Utána a kompresszor és a dekompresszor megállapodnak egy Context Session ID-ben (CID) és a tömörítés formátumában
    - CID: IP címek, UDP portok, RTP SSCR halmazát azonosítja (8/16 bit hosszú)

  Amik változnak:
    IP-nél:
      - Teljes hossz: ezt az adatkapcsolati fejrész is tartalmazza, nem kell átvinni
      - Header checksum: elhagyható, adatkapcsolatira bízva
      - Identification: inkrementálódik, IPv6-nál már nincs
    UDP:
      - Ugyanaz az első kettő, mint IP-nél
    RTP:
      - Seq. number: inkrementálódik
      - Időbélyeg: mintavételi időkkel növekszik
      - M bit
      - CSRC lista ritkán, csak akkor kell átvinni

  Compressed RTP (cRTP) protokoll
    4 formátum:
      FULL_HEADER: teljes tömörítetlen fejrészek plusz CID és Sequence No
        - Seq. No: csomagvesztés ellen kom. és dekom. között
      COMPRESSED_UDP: IP + UDP tömörített (2 v 6 byte), RTP tömörítetlen
        - Ha változik az RTP payload típus
      COMPRESSED_RTP: normál eset, minden fejrész tömörítve
        - Delta kódolás
      COMPRESSED_NON_TCP: IPv4 ID mező tömörítetlen
        - Védekezés nagy csomagvesztés esetén
    - 20 + 8 + 12 byte helyett akár 5 byte összesen
    - de ront a hibaarányon

================= QoS ====================

Garanciák pl:
  - rendelkezésreállás
  - átviteli sebesség
  - késleltetés, késleltetésingadozás,
  - csomagvesztés

Eszközök:
  - Forgalmi méretezés
  - Protokollválasztás
  - Hálózati architektúra megválasztása, hálózati biztonság
  - Tárak menedzselése

Módszerek:
  "Nyers erő" (over-provisioning)

  Folyamonkénti (per-flow) QoS-biztosítás
    - IETF's Integrated Services (IntServ) módszer
    - QoS az egyedi csomagfolyamokra: „finom felbontású" módszer

  Forgalomosztály-alapú (class-based) QoS-biztosítás
    - IETF's Differentiated Services (DiffServ) módszer
    - QoS folyamosztályokra: „durva felbontású" módszer

IntServ (Integrated Services)
  - szolgáltatás-osztályok specifikálása
  - az RSVP (Resource reSerVation Protocol) használata
  - eszköze: ütemezés
  - best effort

  Alkalmazások és szolgáltatásosztályok:
    Guaranteed Quality:
      - Az ún. "Real-time intolerant (RTT)" alkalmazások számára
      - Garantált korlátokat nyújt bármely csomag késleltetésére és a sávszélességre.
    Controlled-Load:
      - Az ún. "Real-time tolerant (RTT)" alkalmazások számára
      - Törekszik kb. ugyanolyan szolgáltatást nyújtására, mint amilyet a folyam kapna, ha terheletlenek lennének a hálózati csomópontok.
      - Beengedés-szabályozást használ

  Működése:
    - beengedés-szabályozás (admission control)
    - erőforrás-foglalás RSVP-vel
    - forgalom ellenőrzés és formálás (traffic policing)
    - csomópontokban ütemezés

  Forgalomleírás és ellenőrzés
    - eszköze: a token bucket
      - a vödörbe r sebességgel töltődnek a tokenek
      - max b token lehet benne, az efelettiek elvesznek
      - Ha egy n hosszú csomagot akkor továbbít, ha van n token (és akkor ezeket kiveszi), amúgy vár.

DiffServ (Differentiated Services)
  - forgalomosztályokhoz rendel erőforrások
  - kevés osztály, pl Premium, Regular
    - a fejlécben néhány bit elég ennek a jelzésére, nem kell külön RSVP

  Működése:
    Edge router (lokális hálózat -> internet):
      - folyamonkénti forgalom-menedzselést végez
      - megjelöli a csomagokat in-profile ill. out-profileként
    Core router (internet -> internet):
      - elsőbbség adása az in-profile csomagoknak

  Csomagok megjelölése az edge routerben:
    - profil: egyeztetett r sebesség és B vödörméret (folyamonként)
    - a profilnak megfelelő részek megjelölése in-profileként többi out-profile
    - osztályok külön elbánási móddal rendelkeznek

  Policing: ha szükséges, a nem konform csomagokat formáljuk / eldobjuk
    - pl ha a legmagasabb osztály in-profile elemeinek nincs elég sávszél, akkor az out-profile-ból dobunk el
    - utána foglalkozunk az utána következő legmagasabb osztállyal

  PHB (Per-Hop-Behavior): a forgalomosztályhoz tartozó csomagtovábbítási elveket definálja
    - minden csomópontnál egyedi döntés, nincs együttműködés, mint IntServ-nél

  PHB-k megadására: ToS (IPv4), Traffic Class (IPv6)
    - 6 bit: DiffServ Code Points (DSCP)
    - 2 bit: Explicit Congestion Notification

  Két alapvető PHB-típus:
    - "expedited forwarding" (EF)
    - "assured forwarding" (AF)
  Default PHB: best effort

  EF: a csomagok továbbítása minimális késleltetéssel, kis csomagvesztéssel
    - olyan sorhoz rendeljük hozzá a csomagot, amelynek a kiszolgálási üteme legalább a csomagok beérkezési ütemével egyezik meg
    - beszéd, videó
    - túl nagy EF forgalom sorbanállási késleltetést okoz
      -> beengedés-szabályozás erre az osztályra
      - A szolgáltatók korlátozzák, pl. max. 30%-a lehet a link kapacitásának

  AF:
    - 12 alosztály (AFxy)
      - x: prioritási osztály (4 különböző érték)
      - y: "eldobási stílus" (3 különböző érték)
      - torlódásnál 4*x + y dönt (akié nagyobb, az élvez előnyt)

================ Hálózati alkalmazások ====================

Alkalmazás-rétegbeli protokollok
  - legtöbbször az alkalmazásban kerül implementálásra
  - mégis szükséges szabványosítani (az alkalmazásoknak együtt kell működnie)
  - alsóbb rétegeket - mint szolgáltatásokat - az operációs rendszer biztosítja
    - csak egy interfészt (API-t) biztosít: SAP (Service Access Point)
    - az alkalmazás által használható végződés: socket
      - kell hozzá tudni az IP-t, a szállítási protokollt és a portszámot

  Port-hozzárendelés:
    Szerveren
      - szolgáltatást azonosítja (pl 80 = HTTP)
      - egy port maximum egy szolgáltatáshoz lehet hozzárendelve
      - statikus
      - well-known ports ("jól ismert" portok): 1-1023
    Kliensen
      - Dinamikusan kerül kiosztásra a még nem használtak közül
      - Regisztrált (1024-49151) illetve dinamikus portok (49152-65535)

  Nyers IP alkalmazások és portok:
    - Vezérlés: 1: ICMP, 2: IGMP
    - Routing: 8: EGP, 89: OSPF
    - Multimédia: 132: SCTP (Stream Control Transmission Protocol)
    - 6: TCP, 17: UDP

  UDP felett:
    53: DNS (Domain Name System)
    67: BOOTP, DHCP (szerver oldal), 68: BOOTP, DHCP (kliens oldal)
    69: TFTP (Trivial File Transfer Protocol)
    123: NTP (Network Time Protocol)
    161: SNMP (Simple Network Management Protocol)
    520: RIP (Routing Information Protocol)

  TCP felett:
    20 és 21: FTP (File Transfer Protocol)
    22: SSH (Secure Shell)
    23: Telnet
    25: SMTP (Simple Mail Transfer Protocol)
    53: DNS (Domain Name System) (Ugyanaz a protokoll UDP-n és TCP-n is)
    80: HTTP (HyperText Transfer Protocol)
    110: POP3 (Post Office Protocol version 3)
    143: IMAP4 (Internet Message Protocol version 4)
    443: HTTPS (HTTP Secure)
    465: SMTPS (SMTP Secure)
    993: IMAP4S (IMAP4 Secure)
    995: POP3S (POP3 Secure)

Névfeloldás (DNS)
  - Követelmények:
    - Jó skálázhatóság
    - Hibatűrés
    - Aktuális (friss) információk
  - Hierarchikus - FQDN (Fully Qualified Domain Name)

  Root: '.'
    - 13 szerver valójában (anycast címzés)

Top level domains:
- egy szinttel a Root alatt
  - .hu, .de
  - .com, .org, .net
  - .arpa

DNS zóna: címtartomány, pl kisnyuszi.hu.
DNS szerver: Egy vagy több zónát tárol, szolgál ki
- elsődleges DNS szerver: r/w (pontosan egy darab)
- másodlagos DNS: r (legalább egy)
  - terhelés elosztás
  - szinkronizálás monoton növekvő verziószám alapján (pl: YYYYMMDDnn)

A zóna elemei: erőforrásrekordok - RR (Resource Records)

SOA (Start of Authority)
- adminisztratív adatok
- az elsődleges DNS szerver neve
- zóna verziószáma (ez alapján a szinkronizálás)
- kapcsolattartó e-mail címe

A (Address)
- név - IP-cím
- a legtipikusabb felhasználás

CNAME (Canonical Name)
- más néven „alias”
- név - név összerendelés

PTR (Pointer)
- IP-cím - név
- ún. reverse zónában

NS (Name Server)
- az adott zónát kiszolgáló DNS szerverek
- legalább kettő kell

MX (Mail Exchange)
- SMTP kiszolgálót azonosít
- Több is megadható preferenciával (prioritással)

SRV (Service Locator)
- MX általánosítása
- tetszőleges szolgáltatásra (pl. SIP)

Névfeloldás menete:
Rekurzív kérés: a konkrét (vagy a negatív) választ várja, a címzettnek a feladata a névfeloldás
Iteratív kérés: a következő csomópontot várja

A DNS-kéréseket a helyi gépen az operációs rendszer oldja fel egységesen
DNS gyorsítótár (cache) a helyi gépen és a DNS szerveren

Minden rekordnak TTL-je (Time To Live) valódi másodpercben megadva -> elévülés

Autoritatív válasz: ha a rekordért felelős szerverek valamelyikétől származik a válaszol
Nem autoritatív: ha gyorsítótárból származik

Névfeloldás menete
- (0. Böngésző gyorsítótára)
- 1. Helyi gép gyorsítótára
- 2. Helyi gépen "hosts" fájl
- 3. Lekérdezés DNS szerverektől
  Ha van DNS szerver megadva:
  - Lekérdezés az elsődleges DNS szervertől, ha elérhető (rekurzív)
    - Az a cache-ből kiszolgál vagy névfeloldást végez (iteratív)
  - Lekérdezés a másodlagos DNS szervertől, ha meg van adva és az elsődleges nem érhető el (rekurzív)

Ha nincs DNS szerver megadva vagy nem elérhetők, akkor lekérdezés valamely root NS-től, majd a hivatkozott NS-ektől (iteratív)

DNS protokoll:

```
    Lekérdezés: DNS kliens <-> DNS szerver
      - UDP 53 <- rövid, gyors üzenetváltás (512 bájt felett TCP)
    Zónaletöltés: Elsődleges DNS szerver <-> Másodlagos DNS szerver
      - TCP 53 <- hosszabb, megbízhatóbb

    Kérés elemei:
      - Kért rekord típusa(i)
      - Feloldandó név vagy IP-cím
      - Rekurzív kérés esetén RD (Recursion Desired) bit beállítva
    Válasz:
      - Pozitív válasz: egy vagy több elemű lista
        - Ebből "véletlenszerűen" (round-robin) választ
      - Authoritatív válasz esetén az AA (Authoritative Answer) bit beállítva
      - Referral válasz (egy vagy több elemű lista)
        - Kiegészítő hivatkozásokat tartalmaz, mely közelebb visz a feloldáshoz
        - pl.: illetékesebb NS; A rekord helyett azonos nevű CNAME rekord
      - Negatív válasz: nem található bejegyzés, nem oldható fel
        - Nincs válasz != negatív válasz

DHCP (Dynamic Host Configuration Protocol):
  - IP-beállításokat oszthatunk ki vele dinamikusan
    - Kliensek egyszerű beállítása
    - Módosítások központilag
    - Mobilitás hálózatok között (eltérő beállítások)

  IP cím igénylése:
    kliens -> DHCP szerver
      -> Discover - Bérlet kérése
      <- Offer    - Bérlet ajánlat
      -> Request  - Bérlet kiválasztása
      <- Ack      - Bérlet nyugtázása

  Kérés:
  0.0.0.0-tól                       255.255.255.255-nek
  (még nincs IP-címe)               (bárkinek)

  Ajánlat:
  255.255.255.255-nek               pl. 192.168.1.1-től
  (bárkinek; nem címezhető)         (egy DHCP szerver címe)

  Ajánlat tartalma
    - IP-cím
    - Alhálózati maszk
    - Bérleti idő
    - DHCP szerver IP címe
    - Igénylő MAC címe!

  Kiválasztás
    - az első ajánlatot (pl. ha több DHCP szerver)
    - Kiválasztási üzenet: 0.0.0.0-tól 255.255.255.255-nek
    - Az üzenet tartalma
      -  kért IP-cím
      -  DHCP szerver IP-címe

  Nyugta
    - 192.168.1.1-től, 255.255.255.255-nek
    - Nyugtázó üzenet tartalma
    - kiosztott IP-cím
    - alhálózati maszk
    - bérleti idő

  Bérleti idő (TTL)
    Kezelése:
      - Félidőben hosszabbítási kérés
      - 7/8 TTL-nél új igénylése

    Rövid TTL mellett
      - Ha a kliens szabálytalanul távozik a hálózatból
        -  a bérletét nem adja vissza
      - Ha a kliens szabálytalanul újraindul
        -  nem adja vissza a bérletét, és még újat is igényel
      - A beállításváltozások gyorsan életbe lépjenek
```

    Hosszú mellett
      - Ne legyen nagy hálózati forgalom

  DHCP opciók (csak a lényegesek):
    - Subnet Mask (alhálózati max)
    - Domain Name (FQDN suffix)
    - Router (alapértelmezett átjáró(k))
    - DNS (DNS szerver(ek))
    - Host Name (gép neve is kioszthtató)
    - Requested Address (igényelt IP-cím)
    - Lease Time (TTL)
    - DHCP Server (DHCP szerver IP-címe)
    - Parameter Request List (igényelt paraméterek listája)
    - Renewal Time (megújítási idő)

  DHCP hibatűrés: több DHCP használata egy hálózatban, de diszjunkt IP-címtartományok osztása

  DHCP kiterjesztése több hálózati szegmensre
    - A routerek nem engedik át a DHCP üzeneteket
    - A routerekre ún. "DHCP Relay Agent"-et telepítve az továbbítja a DCHP forgalmat a DHCP szerverek és kliensek között

  IPv6-ban minden router egyben DHCP szerver is

Szöveg és fájlátvitel
  Telnet
    - távoli parancssor
    - nem biztonságos, SSH helyette

  FTP (File Transfer Protocol):
    - TCP 20, 21-es port

Levelező Rendszerek
  - címzett meghatározásához DNS kell
  - SMTP: levél továbbításra
  - POP3, IMAP4: levelek lekérdezése

  POP3 (Post Office Protocol)
    - Parancs orientált
    - TCP 110-es port
    - Levelek lekérdezésére
    - POP3S: POP3 TLS tikosítással

  IMAP4 (Internet Message Protocol)
    - Parancs orientált
    - TCP 143-as port
    - Levelek lekérdezésére
    - IMAP4S: IMAP4 TLS tikosítással
    - Intelligensebb a POP3-nál:
      - Könyvtárstruktúra támogatása
      - Keresés támogatása
      - Nem törli automatikusan a szerveren tárolt leveleket

  SMTP (Simple Mail Transfer Protocol)
    - Levelek továbbítására
      - TCP 25-ös port
      - Nem közvetlen továbbítás is támogat: SMTP szerverek (SMTP relay) közbeiktatásával
      - SMTPS (SMTP Secure): TCP 465

    A leggyakoribb SMTP parancsok:
      HELO: Üdvözlés
        - ESMTP esetén EHLO
      MAIL FROM:<feladó e-mail címe>
      RCPT TO:<címzett e-mail címe>
      DATA: Adat következik
      <CR><LF>.<CR><LF>: Adat vége
      QUIT: SMTP kapcsolat bontása
      VRFY <e-mail cím>: Létezik-e az adott e-mail cím
      HELP
      NOOP

Webes Rendszerek
  - HTTP (HyperText Transfer Protocol)
  - Parancsorientált állapotkódokkal
  - Speciális fejlécek
  - Állapotmentes
  - TCP 80
  - Proxy: kliens nevében jár el (főként a hatékony gyorsítótárazás miatt)

  Nem-perzisztens kapcsolat: minden kéréshez fel kell építeni / le kell bontani a kapcsolatot
    - 2 RTT/objektum (+ TCP overhead)
      - ezért böngészők párhuzamos TCP kapcsolatokat nyitottak
  Perzisztens: egy kapcsolaton belül több kérés/válasz (HTTP 1.1 óta)
    - 1 RTT/objektum
      - a kérések átlapolhatóak

  Pl: GET /somedir/page.html HTTP/1.1
      Host: www.someschool.edu
      User-agent: Mozilla/4.0
      Connection: close
      Accept-language:hu
      <CR><LF>

  Gyakori kérések:
    GET <URL>: adott URL tartalmának lekérése
    HEAD: mint a GET, de csak a metaadatokat adja vissza
    POST: a kliens ezzel tud adatokat küldeni a szervernek
    PUT: a POST-hoz hasonló, fájlfeltöltésre alkalmas
    DELETE: adott URL tartalmának törlése

  Gyakori HTTP állapotkódok:

    Kód   Jelentés

    200   OK
    201   Created
    202   Accepted
    204   No content
    400   Bad request
    401   Unauthorized
    403   Forbidden
    404   Not found
    500   Internal Server Error
    503   Service Unavailable

  HTTP 2.0
    - Google SPDY elemeit használja
    - csökkenti a web oldalak betöltési idejét:
      - multiplexált stream-ek egy TCP kapcsolaton belül
      - kérések priorizálása
      - kérés és válasz fejrészek tömörítése
      - a statikusok nincsenek újraküldve
      - szerver push és hint funkció

================= Hálózati alkalmazások II. ====================

SMTP képes ellátni: (MTA = Mail Transport Agent, MUA = Mail User Agent)
  - MUA - MTA
  - MTA - MTA

Spamek miatt MUA-MTA-ra "beengedés" korlátozás:
  Message Submission Protocol
    - Hasonló, mint SMTP, de
    - Eltérő portszám: 587
    - Autentikáció (pl. username, password)
    - További funkciók (pl. FQDN ellenőrzése, szintaxis ellenőrzése, hibák naplózása)
  MTA helyett: MSA - Message Submission Agent

Két cím is van STMP-nél (mint postai levélen és borítékon)
  - A levelező kliensben látható mezők (From, To, Cc, esetleg Bcc) tartalma a levél törzsében utazik
  - A kézbesítés pedig az ún. envelope recipient mező alapján történik, illetve
    hiba esetén az envelope sender alapján talál vissza a feladóhoz a hibaüzenet
  Ezeket a küldő MUA állítja be

    - a feladót a saját beállítása alapján ("Saját e-mail cím"): From mező, és envelope sender
    - a címzettet(ek)et a felhasználó rendelkezése alapján: To, Cc, Bcc mezők, illetve envelope recipient
  Az SMTP DATA parancs esetén először a fejrész mezőit kell megadni (From, To, Cc, Bcc, Date, Subject), majd egy üres sor után a levél szövege
    - lehet mást írni a levél fejlécébe, mint envelope sender és envelope recipient értéke (a MUA-k konzisztensen töltik ki)
    - a szerver, akinek a level küldjük ezt ellenőrizheti (de pl a telnet nem teszi)

Levelek letöltése POP3-mal
  POP3: Post Office Protocol
    - Postafiók távoli elérésére használható
    - Szabványos portszáma: 110
  A POP3 protokoll legfontosabb parancsai
    - USER username - felhasználó nevének megadása
    - PASS password - jelszó megadása (nyílt szövegként!)
    - STAT - lekérdezi a levelek számát és összesített méretét
    - LIST - levelek lekérdezése (sorszám + méret)
    - RETR n - az n. levél letöltése
    - DELE n - az n. levél kijelölése törlésre
    - RSET - törlésre való kijelölés(ek) megszüntetése
    - QUIT - postafiók aktualizálása (törlések véglegesítése) és kilépés

FTP (File Transfer Protocol)
  Az FTP szerver a 21-es porton várja a kliens csatlakozását
    - Csatlakozáskor létrejön a vezérlő kapcsolat (TCP): A kliens parancsokat ad ki rajta, a szerver válaszol
  Fájlok, könyvtárlisták átviteléhez: adat kapcsolat (TCP)
    - Szerver oldalon a 20-as portot használja
    - Szükség esetén létrejön, majd lebomlik
    - Létrehozásának iránya kétféle lehet
      - Aktív mód esetén: szervertől a kliens felé
      - Passzív mód esetén: klienstől a szerver felé
    Kliens oldali tűzfal vagy privát IP-cím+NAT esetén passzív mód
      - A tűzfal nem engedné meg a kapcsolódást befele
      - Privát IP-cím alapján nem jönne meg a SYN (De létezik protocol helper)

  FTP parancsok
    - USER username - felhasználói név megadása
    - PASS password - jelszó megadása
    - PORT h1,h2,h3,h4,p1,p2 - IP-cím és portszám megadása a szerver által kezdeményezett adatkapcsolathoz (aktív mód)
    - PASV - IP-cím és porszám kérése a szervertől, ennek alapján építi fel az adatkapcsolatot a kliens (passzív mód)
    - LIST [directory] - könyvtár tartalmának listázása
    - RETR filename - fájl letöltése
    - STOR filename - fájlt feltöltése
    - HELP [command] - argumentum nélkül parancslistát ad
    - SYST - információk a szerverről
    - QUIT - kilépés

  FTP átvitel közben:
    - megjelenítései réteg: LF <-> CR/LF csere szöveges fájlok eseték

  Anonymous FTP:
    - a felhasználói név: Anonymous
    - a jelszó az e-mail címünk, de legalább egy @ karakter

    Tipikus könyvtárak
      - pub - itt találhatók a nyilvános anyagok
      - incoming - ide lehet feltölteni

    Böngésző ezt használja, ha ftp kapcsolatot nyitunk meg

Virtuális webszerverek
  - Webtárhely-szolgáltatók szeretnék egy szerveren több ügyfél weblapját kiszolgálni
  Nem igazán jó megoldások:
    - Külön könyvtárba tenni, könyvtár neve az URL-ben:
      - www.webhostingkft.hu/bogyoesbaboca
      - www.webhostingkft.hu/micimacko
      - nem jó, mert a felhasználók saját domaint szeretnének, ilyeneket:
        - www.bogyoesbaboca.hu

      - www.micimacko.hu
  - Minden weblapnak külön IP-cím, ugyanahhoz az interfészhez mindet hozzárendelni
    - nem jó, mert sok IP-cím kell, ami IPv4-ben szűkös erőforrás!

Probléma: Szimbolikus név → IP-cím leképzésnél a hostnév elveszik
  - Megoldás: HTTP/1.0 helyett HTTP/1.1: a hostnév elküldhető
Összes ügyfél esetén CNAME rekord, például:
  - www.bogyoesbaboca.hu CNAME www.webhostingkft.hu
  - www.micimacko.hu CNAME www.webhostingkft.hu
A GET parancs után a Host: adja meg a hostnevet, pl.:
  GET / HTTP/1.1
  Host: www.bogyoesbaboca.hu

Be kell állítani a szerveren, hogy melyik virtuális webszerverhez milyen tartalmat rendelünk.