

HAÜ Vizsga

Tartalomjegyzék

DNS (Domain Name System).....	4
Jellemzők.....	4
Hiteles (authoritative) szerver	4
Helyi (local) szerver:	4
Gyorsítótár (cache):.....	4
Névfeloldás	4
Bejegyzések	5
Szállítási réteg funkciói	6
Socket	6
TCP-alapú socket.....	6
UDP-alapú socket.....	6
Multiplexálás	7
Demultiplexálás	7
UDP (User Datagram Protocol)	8
Jellemzők.....	8
UDP-szegmens fontos mezői.....	8
Ellenőrzőösszeg.....	8
TCP (Transmission Control Protocol)	9
Jellemzők.....	9
TCP-szegmens fontos mezői.....	9
Nyugtázás.....	9
Megbízható adatszállítás	10
Forgalomszabályozás (flow control).....	10
Kapcsolatkezelés	10
Torlódáskezelés.....	11
Szelektív nyugtázás (SACK)	11
QUIC (Quick UDP Internet Connections).....	12
Jellemzők.....	12

Kapcsolatfelépítés.....	12
QUIC csomagformátum	12
Küldés és nyugtázás	12
IPv4.....	13
Jellemzők.....	13
Fejléc fontos mezői.....	13
Darabolás és összerakás.....	13
IPv6.....	14
Jellemzők.....	14
Fejléc fontos mezői.....	14
Címzés.....	14
Automatikus címkiosztás.....	15
DHCP (Dynamic Host Configuration Protocol).....	15
SLAAC (Stateless Address Autoconfiguration).....	15
Címfordítás	16
Alapok.....	16
NAT (Network Address Translation)	16
IPv4-IPv6 átjárás.....	16
Datagram továbbítása és útvonalválasztás.....	17
Továbbítás routerben.....	17
Routing tábla.....	17
Statikus irányítás	17
Dinamikus irányítás.....	17
Dinamikus routing protokollok.....	18
RIP (Routing Information Protocol)	18
OSPF (Open Shortest Path First)	18
BGP (Border Gateway Protocol)	19
Adatkapcsolati réteg.....	20
Szolgáltatások	20
ARP (Address Resolution Protocol).....	20
VLAN	21

Többszörös hozzáférés.....	21
Véletlen hozzáférés.....	21
WLAN.....	22
Mobilhálózatok.....	23
Cellás rendszerek.....	23
Rádióhálózat-típusok.....	23
SDN (Software-defined Networking).....	24
Jellemzők.....	24
OpenFlow.....	24
SNMP (Simple Network Management Protocol).....	25
Jellemzők.....	25
MIB (Management Information Base).....	25

DNS (Domain Name System)

Jellemzők

- Az interneten minden hostot IP-címek azonosítanak, ami az emberek számára nehezen értelmezhető
- Az IP-címekhez emberi használatra is alkalmas neveket rendelünk
- Hierarchiába szervezett, elosztott adatbázis, sok szerverrel
- Kiválóan skálázható és robusztus
- UDP felett az 53-as porton kommunikál
- **Szolgáltatások:**
 - Hostnév IP-címre fordítása
 - Host álnév (alias) valódi (kanonikus) névre fordítása
 - Levelezőszerverek neve
 - Terheléelosztás: több IP-cím tartozik egy kanonikus névhez
- **Szintek:** Root → TLD → SLD → Subdomain → Host

Hiteles (authoritative) szerver

- Internetes hostokat üzemeltető szervezetek szervere
- Gyakorlatilag minden szervezethez tartozik egy
- A szervezet hostjainak és szervereinek névfeloldására szolgál
- Nem csak a szervezet üzemeltetheti

Helyi (local) szerver:

- Nem tartozik a hierarchiába
- Minden ISP működtet egyet
- A hostok ettől kérik egy ismeretlen doménnév vagy hostnév feloldását

Gyorsítótár (cache):

- Egy megismert összerendelést a szerver bizonyos ideig tárol
- Minden szerver alkalmazhatja

Névfeloldás

- **Iteratív:** a megszólított névszerver megadja, hogy melyik névszerverhez kell fordulni helyette a kéréssel
- **Rekurzív:** a névfeloldás feladatát a megszólított névszerverre bízta

Bejegyzések

- **Resource Record (RR):**
 - Típus (type)
 - Név (name)
 - Érték (value)
 - Élettartam (TTL)
- **A (Address):**
 - **Név:** hostnév
 - **Érték:** a névhez tartozó IPv4-cím
- **AAAA (IPv6 Address):**
 - **Név:** hostnév
 - **Érték:** a névhez tartozó IPv6-cím
- **NS (Name Server):**
 - **Név:** egy domain neve
 - **Érték:** a domain autoritativ szerverének a hostneve
- **CNAME (Canonical Name):**
 - **Név:** egy kanonikus névhez tartozó álnév
 - **Érték:** a kanonikus név
- **MX (Mail Exchange):**
 - **Név:** a levelezőszerver álneve
 - **Érték:** levelezőszerver kanonikus neve

Szállítási réteg funkciói

Socket

- Kapcsolat az alkalmazást és a hálózati működést segítő rendszerfunkciók között
- A hálózati alkalmazás folyamata nyitja meg
- A host operációs rendszere kezeli
- Feladata üzenetek **küldése** és **fogadása**
- Szállítási réteg szinten teremt kapcsolatot a folyamatok között
- Alapvető típusok:
 - **Megbízható:** kapcsolat-orientált (TCP – `SOCK_STREAM`)
 - **Megbízhatatlan:** adatcsomag-orientált (UDP – `SOCK_DGRAM`)

TCP-alapú socket

- **Szerver:**
 - Állandóan futó folyamat
 - Várakozik a kliensre
 - Hallgatózik egy adott porton: nyitott **szerver socket**
- **Kliens:**
 - Kezdeményezi a csatlakozást a szerverhez
 - Egy **kliens socketet** nyit
 - A szerver socket adatait használja: IP-cím és portszám
 - Általában az operációs rendszer rendel portszámot a kliens sockethez
- **Adatátvitel:**
 - A szerver fogadja a klienst
 - Minden egyes klienshez egy **kapcsolat-socketet** nyit meg
 - A szerver kliens-oldali IP-cím és portszám alapján azonosítja a klienseket
 - Így képes több kliens is kommunikálni a szerverrel ugyanazon a porton

UDP-alapú socket

- Nincs felépített kapcsolat a szerver és a kliens között
- Nincs nyugtázás és sorrendezés
- **Datagramok küldése (kliens):** a socket a kapott datagramhoz hozzáteszi az IP-címet és portszámot
- **Datagramok fogadása (szerver):** a kliens azonosítása csak közvetlenül a küldési IP-cím és portszám alapján történhet
- Egy folyamat egy socketen keresztül több kliens-socket felől kaphat adatokat
- A különböző kliensek datagramjai keverve érkeznek

Multiplexálás

- A küldő hostnál történik
- Összegyűjti a különböző socketek adatait
- Ellátja fejléccel a demultiplexáláshoz

Demultiplexálás

- A fogadó hostnál történik
- Továbbítja a vett szegmenseket a megfelelő socketeknek
- **Működése:**
 - A host megkapja az IP datagramot
 - Minden datagramnak van forrás és cél IP-címe
 - Minden datagram pontosan 1 db szállítási rétegbeli szegmenst hordoz
 - Minden szegmensnek van forrás és cél portszáma
 - A host a szegmenst az IP-cím és portszám alapján irányítja a megfelelő sockethez

UDP (User Datagram Protocol)

Jellemzők

- Leegyszerűsített szállítási protokoll
- „Best effort” szolgáltatás: a szegmensek elveszhetnek és összekeveredhetnek
- **Összeköttetés nélküli:** nincs kézfogás és a szegmensek kezelése független
- Mivel nem szükséges összeköttetés-felépítés, ezért **nincs kezdeti késleltetés**
- Nincs kapcsolatállapot-kezelés sem a küldő, sem a fogadó oldalon
- **Veszteségtűrő,** így nincs szükség torlódáskezelésre
- Kis méretű fejléc
- A lehető **leggyorsabb** adatátvitelt biztosítja
- Leggyakoribb alkalmazása a multimédia adatfolyam-szállítás (streaming)
- További alkalmazásai: DNS, SNMP
- UDP feletti megbízható adatátvitelről az alkalmazási rétegben kell gondoskodni

UDP-szegmens fontos mezői

- Forrás portszám
- Cél portszám
- Datagram hossza (bájtokban)
- Ellenőrzőösszeg (checksum)

Ellenőrzőösszeg

- Cél a hibák érzékelése
- **Küldő oldalon:**
 - A szegmenst 16-bites egész számok szorzataként kezeli
 - Az ellenőrzőösszeg ezen számok összege, invertálva
 - Ezt beleteszi a fejlécbe
- **Fogadó oldalon:**
 - Azonos módszerrel számítja az összeget
 - Hozzáadja az ellenőrzőösszeg mezőt
 - Az eredmény csak 1-eseket tartalmazhat
- Az összeadásból adódó legmagasabb helyiértékű 1-et hozzáadjuk az összeghez

TCP (Transmission Control Protocol)

Jellemzők

- Több különböző RFC írja le a működését
- **Pont-pont** elrendezésű: egy küldő és egy fogadó
- **Megbízható**, sorrendtartó adatfolyam
- **Csővezetékezett** (pipelined) protokoll: nem vár az előző nyugtára
- **Full-duplex** adatátvitel: kétirányú adatáramlás ugyanazon a vonalon
- **Összeköttetés-alapú**: kézfogással indul, lezárással ér véget
- Forgalomszabályozás a fogadó túlterhelése ellen
- Torlódáskezelés: igazodás a hálózat aktuális állapotához

TCP-szegmens fontos mezői

- Forrás portszám
- Cél portszám
- Sorszám (Sequence number)
- Nyugtaszám
- Fejléc hossz
- Flag bitek:
 - **URG**: sürgős üzenet (általában nem használt)
 - **ACK**: nyugtamező érvényes
 - **PSH**: azonnal továbbítandó (általában nem használt)
 - **RST**: végzetes hiba miatti azonnali megszakítás
 - **SYN**: szinkronizációs bit az első csomagokban
 - **FIN**: kapcsolat lezárási kérelem
- Vételi ablak
- Ellenőrzőösszeg (checksum)

Nyugtázás

- **Sorszám (SEQ)**: a teljes adat hányadik bájtával kezdődik a szegmens része
- **Nyugta (ACK)**:
 - Vételi információ
 - Melyik bájtal kezdődjön a következő küldése
 - Nyugta arról, hogy az eddigiek megjöttek
 - Összegző nyugta minden eddig megjött sorszámról

- **Timeout idő:** nem jó, ha túl kicsi vagy túl nagy, az RTT alapján határozzuk meg
- **RTT becslés:** folyamatosan ingadozó érték, ezért becsülni kell, mérés alapján
 - A mérés-alapú minta **SampleRTT**: értéke egy szegmens elküldése és az arra kapott nyugta érkezési ideje közötti különbség
 - A minták értéke nem állandó, ezért simítani kell
- Exponenciálisan súlyozott mozgó átlagot számítunk:
 - **EstimatedRTT** = $(1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$
 - Az „ α ” paraméter javasolt értéke 0.125
- Kell némi ráhagyás, amely a minták dinamikájával arányos:
 - **DevRTT** = $(1 - \beta) * DevRTT + \beta * |SampleRTT - EstimatedRTT|$
 - A „ β ” paraméter javasolt értéke 0.25
- Így a **Timeout** értéke: **TimeoutInterval** = $EstimatedRTT + 4 * DevRTT$

Megbízható adatszállítás

- Újraküldést kiváltó okok: **időtúllépés** és **kettőzött nyugta** (duplicate ACK)
- Időtúllépés során az ezt okozó szegmens újraküldése és az időzítő újraindítása
- Ha még nem nyugtázott szegmensre érkezik nyugta:
 - A már nyugtázott szegmensek halmazának frissítése
 - Időzítő újraindítása, ha van még nyugtázatlan szegmens
- Késleltetett nyugta (delayed ACK): legfeljebb 500 ms-ot vár a következő szegmensre és utána nyugtáz
- Gyors újraküldés (fast retransmit):
 - A harmadik duplikált nyugta után **csomagvesztést** feltételezünk
 - A kimaradt szegments újraküldjük, időtúllépés nélkül

Forgalomszabályozás (flow control)

- A vevő oldalán az adatok egy véges méretű bufferbe érkeznek
- Megakadályozza a buffer túltöltődését, ha a küldő túl gyorsan vagy túl sok üzenetet küld
- A feldolgozás üteméhez illeszti a küldési sebességet
- Buffer szabad helye: **RcvWindow** = $RcvBuffer - (LastRcvd - LastRead)$
- A fogadó jelzi a szabad hely mennyiségét a nyugta **RcvWindow** mezőjében
- A küldőnek legfeljebb ennyi nyugtázatlan adata lehet

Kapcsolatkezelés

- A kliens kezdeményez
- **Kezdeményezés – 3-utas kézfogás (three-way handshake):**
 1. A kliens **SYN** szegmenst küld a szervernek: kezdő sorszám, adat nélkül
 2. A szerver **SYN + ACK** szegmensekkel válaszol: inicializáló lépések itt is
 3. A kliens **ACK** szegmensessel válaszol, ami már tartalmazhat adatot is

- **Lezárás – 4-utas kézfogás (four-way handshake):**
 1. A kliens **FIN** szegmenst küld a szervernek
 2. A szerver **ACK** szegmessel válaszol
 3. A szerver ezután **FIN** szegmenst küld, amivel elindítja a lezárást
 4. A kliens megkapja a szegmenst és **időzített várakozás** állapotba lép:
 - A kliens válaszol egy **ACK** szegmessel
 - A szerver megkapja az **ACK**-ot és lezártnak tekinti a kapcsolatot

Torlódáskezelés

- Minden küldő a lehető legnagyobb sávszélességgel küldi az adatokat
- A küldők összes forgalma meghaladja egy link átbocsátó képességét
- Ekkor a csomagok feltorlódnak (congestion), a küldő tud ellene tenni
- **Congestion window:**
 - A következő nyugtáig átküldhető adatok mennyiségét korlátozza
 - $LastByteSent - LastByteAcked \leq CongWin$
 - Az átlagos sávszélesség közelítőleg $CongWin / RTT$
 - Ha nincs probléma, akkor **növeljük** az ablakot, egyébként **csökkentjük**
 - Ha **nincs duplikált nyugta**, akkor minden rendben van
- **AIMD (additive increase, multiplicative decrease):**
 - **Lineáris növelés:** minden RTT idő alatt 1 MSS-el növeljük az ablakot
 - **Exponenciális csökkentés:** csomagvesztés esetén a felére csökkentjük
- **Lassú indítás (Slow Start):**
 - A kapcsolat kezdetén $CongWin = 1 MSS$
 - Minden nyugta érkezésekor 1 MSS-el növeljük az ablakot
 - **Threshold** változó: csomagvesztés után az ablak felére állítjuk be
- Ha $CongWin < Threshold$: **Slow Start**, exponenciális ablaknövelés
- Ha $CongWin \geq Threshold$: **Congestion Avoidance**, lineáris ablaknövelés
- 3 duplikált nyugta után: $Threshold = CongWin = CongWin / 2$
- Időtúllépés esetén: $Threshold = CongWin / 2$ és $CongWin = 1 MSS$

Szelektív nyugtázás (SACK)

- A fogadó nyugtában küldi vissza a sorrenden kívüli bájtok tartományait
- Korlátozott mennyiségben
- A fejléc opciót felhasználva
- A kapcsolat felépítésekor a felek egyeztetnek a használatáról

QUIC (Quick UDP Internet Connections)

Jellemzők

- Eredetileg Google fejlesztés, jelenleg Internet Draft
- TCP változatok előnyeit ötvözi
- Fő feladata megbízható adatátvitelt biztosítani UDP felett
- A szállítási és az alkalmazási réteg között fél réteggel feljebb van
- Egyelőre nem a TCP helyett, csak mellett használják
- Megoldja a titkosítást és a szállítási rétegbeli funkciókat is
- **Gyors** kapcsolatfelépítés
- **Hatékony** újraküldés és torlódáskezelés
- Adatfolyamok **multiplexálása**
- Kapcsolati végpontok **migrálása**
- Beépített **titkosítás (TLS)**

Kapcsolatfelépítés

- **Ismeretlen szerver:**
 - A felépítés **1 RTT** idejű
 - CHLO (client hello), REJ (reject), SHLO (server hello)
 - Része a titkosítás kulcskiosztása is
- **Ismert szerver:**
 - A felépítés **0 RTT** idejű
 - CHLO/SHLO után egyből küldhető adat

QUIC csomagformátum

- A csomag az **UDP szegmens adata**
- **Egyedi csomagszám:** monoton nő, újraküldés esetén már más lesz

Küldés és nyugtázás

- A küldésnél torlódáskezelést használ (TCP CUBIC)
- A nyugta egy azonosított csomagra vonatkozik:
 - SACK: a TCP-nél jóval több intervallumot kezel
 - Időbélyeges: könnyebb az RTT számítás (csomagonként lehet számláló)
- TLP-t (Traffic Light Protocol) is használ

IPv4

Jellemzők

- 32-bites címtér
- Pontozott decimális címformátum, 4 oktett (bájt)
- Összesen 2^{32} kiosztható cím (~4 milliárd)
- Bizonyos tartományok le vannak foglalva speciális célokra

Fejléc fontos mezői

- Internet protokoll verziószám
- Fejléc hossz
- ToS (Type of Service)
- Datagram teljes hossza
- Fragment azonosító
- Flagek (Don't Fragment, More Fragments)
- Fragment offset
- Time to live (TTL)
- Átviteli protokoll
- Ellenőrző összeg
- Forráscím
- Célcím

Darabolás és összerakás

- Az IP alatti rétegekben korlátozott a keretek mérete
- **MTU (Maximum Transmission Unit):** adott linken küldhető maximális méret
- A routerek végzik a darabolást
- Az összeillesztés csak a fogadónál történik
- Darabolási információk a csomagok fejlécében
- Terheli a routert: feldolgozást igényel, késlelteti a csomagot
- Ha egy darab elvész, az egész csomagot újra kell küldeni (TCP)
- Célszerű elkerülni – megoldás:
 - **Path MTU Discovery:** az útvonal legkisebb MTU-ját kell használni
 - **Don't Fragment** flag beállítása a fejlécben
 - Kisebb MTU-val újrapróbáljuk, ameddig megfelelő MTU-t nem találunk

IPv6

Jellemzők

- 128-bites címtér
- Hexadecimális címformátum, 8 hextett
- Összesen 2^{128} kiosztható cím ($\sim 3.4 \cdot 10^{38}$)
- Egyszerűbb fejléc, gyorsabb feldolgozás
- QoS (Quality of Service) korrektebb támogatása
- Nincs darabolás és ellenőrzőösszeg
- A bevezető nullák elhagyhatók

Fejléc fontos mezői

- Internet protokoll verziószám
- Prioritás (QoS)
- Flow-label
- Adathossz bájtokban
- Next header (hivatkozás további fejléc mezőkre)
- Hop limit (olyan, mint a TTL)
- Forráscím
- Célcím

Címzés

- **Részei:** hálózati cím (48-bit), alhálózati cím (16-bit), interfész azonosító (64-bit)
- A hálózati maszk hossza alapértelmezetten 64
- Egy interface-nek több globális IPv6-címe is lehet
- A hálózaton belüli forgalomirányításhoz van lokális címe is
- **Link-local cím:**
 - Csak az adott hálózatban kell egyedinek lennie
 - **FE80::/8** tartomány

Automatikus címkiosztás

DHCP (Dynamic Host Configuration Protocol)

- **IPv4**
- Dinamikusan, egy pool-ból oszt címeket meghatározott időre
- **Lease:** IP-cím + kliens + időintervallum
- A szerver az összes lease-ről nyilvántartást vezet
- A címosztás lehet permanens is, lejárató idő nélkül
- A DHCP nem képes DNS-bejegyzések közzétételére
- **UDP-alapú**, a szerver a 67-es porton, a kliens a 68-as porton kommunikál
- **4-step handshake:**
 1. DHCP**DISCOVER** (broadcast kérés – 255.255.255.255:67)
 2. DHCP**OFFER**
 3. DHCP**REQUEST**
 4. DHCP**ACK**

SLAAC (Stateless Address Autoconfiguration)

- **IPv6**
- Az IPv6 host-ok automatikusan konfigurálják önmagukat
- Induláskor a hálózati eszköz automatikusan allokal link-local címeket minden interface-nek, amelyen az IPv6 engedélyezett
- Ezek a címek **FE80::/64** prefixummal állnak elő
- NDP-vel (Neighbour Discovery Protocol) állíthat elő globálisan egyedi címeket

Címfordítás

Alapok

- Az IPv4 címtér relatív kicsi mérete és kimerülése miatt van rá szükség
- IPv6 esetében szükségtelen, mivel bőséges a címtér
- A hálózatból kifelé ugyanaz a cím látszódjon, belső változásoktól függetlenül

NAT (Network Address Translation)

- **Egy-egy hozzárendelés**
- A hálózati határrouter **ki- és visszacseréli** a belső címet egy globálisan egyértelmű címre
- A router a fordításokat (hozzárendeléseket) nyilvántartja egy táblában

IPv4-IPv6 átjárás

- A NAT csak ideiglenes megoldás az IPv4 kimerülésére, végső megoldás az IPv6
- Két teljesen külön címtér, de működhet párhuzamosan (dual stack)
- **NAT64:** IPv6 és IPv4 közötti címfordítás
- **Tunneling:** az IPv6 datagramot IPv4 datagramba teszi és átküldi az alagúton

Datagram továbbítása és útvonalválasztás

Továbbítás routerben

- A host a cím és a maszk alapján eldönti, hogy a cél vele egy hálózatban van-e
- Ha igen, akkor a szomszédos hostnak küldi
- Ha nem, akkor egy **átjárón** az internet felé:
 - **Default gateway**
 - A hostnál meg kell adni, hogy lehessen indítani a datagramot ebbe az irányba
 - A host felé eső interface az első routeren
 - Egy hálózatban van a hosttal
- A router egy datagramot célcím alapján továbbít, ekkor 2 eset lehetséges:
 - Közvetlenül továbbítható egy hozzákapcsolt hálózatba
 - Távoli hálózat esetén az irányítási tábla bejegyzései alapján

Routing tábla

- Irányítási és továbbítási szabályok listája
- Lokálisan érvényes, de célszerű globális ismeretek alapján is feltölteni
- A szabályok bejegyzések:
 - Részhálózatokra vonatkoznak
 - CIDR elveket figyelembe véve maszkot is tartalmaznak
 - A következő ugrást mutatják: kimenő interface vagy hálózat (rekurzív)
 - Időről-időre változhatnak
 - Lehet köztük átfedés (ezt fel kell oldani címösszevonással)
- **Gateway of last resort:** mindenre illeszkedő cím – `0.0.0.0/0`
- Ha nincs illeszkedő bejegyzés, akkor a router eldobja az üzenetet és jelzi a hibát

Statikus irányítás

- A hálózat állandóságát feltételezi
- A routerekben ritkán frissítik a bejegyzéseket
- Elég egyszer kiszámolni
- Kézi konfiguráció
- Bejegyzéstípusok:
 - Statikus út
 - Tartalék út
 - Alapértelmezett út

Dinamikus irányítás

- Követi a hálózat változásait
- A routerek rendszeresen frissítik a bejegyzéseiket
- Rendszeresen újra kell futtatni az algoritmusokat
- Automatizált konfiguráció

Dinamikus routing protokollok

RIP (Routing Information Protocol)

- **Távolságvektor-alapú** útválasztás
- **Intra-AS** protokoll (IGP)
- Két fő változata a **RIPv1** és **RIPv2**, lefelé kompatibilis
- IPv6-támogatás: **RIPng**
- **Metrika:**
 - **Ugrásszám (hop count)**
 - A közvetlenül kapcsolt hálózat metrikája 1
 - Maximális értéke 15, a végtelen távolság 16
- **RIP hirdetések:**
 - Távolságvektorokat küld a szomszédoknak, UDP felett
 - 30 másodpercenként automatikusan vagy közvetlen kérésre küldi
- **Szomszéd elérhetősége:**
 - 180 másodperc után feltételezi, hogy az eszköz elérhetetlen
 - **Poisoned reverse** megoldással elterjeszti ezt a hírt
- **RIPv1:**
 - Osztályalapú konfiguráció
 - Maszkokat nem tárol
 - **Auto-summary:** alapértelmezetten összevontan hirdet
- **RIPv2:**
 - Osztályfüggetlen konfiguráció (CIDR)
 - Maszkokat is tárol, nem jelentenek gondot az eltérő osztályok
 - Nincs összevonás

OSPF (Open Shortest Path First)

- **Kapcsolatállapot-alapú** útválasztás
- **Intra-AS** protokoll (IGP)
- Ma használt változata az **OSPFv2**
- IPv6-támogatás: **OSPFv3**
- **Hierarchikus működésű**
- Beépítetten támogatja az osztályfüggetlen címzést (CIDR)
- Minden OSPF router a hozzákapcsolt linkekről gyűjt információkat
- Információk hirdetése minden OSPF routernek:
 - **LSA (Link State Advertisement):** IP datagramba csomagolt, multicast
 - Periodikusan és változás esetén
 - **Hálózati térképet** épít fel a szerzett információk alapján
- Körzetek (area):
 - Pontosan 1 gerinchálózati (backbone) körzet: **area 0**
 - A többi a gerincből ágazó lokális körzet

BGP (Border Gateway Protocol)

- **Útvektor-alapú** útválasztás
- **Inter-AS** protokoll (EGP), de facto szabvány
- Ma használt változata a **BGP-4**
- **Feladatai:**
 - Megismerjük a szomszédos AS-ekben lévő hálózatok elérhetőségeit
 - Ezeket az információkat keresztüljuttassuk a saját AS-en
 - Útvonalakat határozzunk meg egy távoli hálózat felé (policy alapján)
 - Saját hálózataink elérhetőségeit más AS-ek felé hirdessük
- **Működése:**
 - A BGP-t futtató routerekből párokat képzünk (BGP peers)
 - A párok egymást között információt cserélnek
 - TCP feletti kommunikáció megvalósított viszonylatokkal (BGP session)
 - Azonos AS-ben lévő routerek esetén belső BGP viszony (iBGP)
 - Eltérő AS-ekben lévő routerek esetén külső BGP viszony (eBGP)
- **Amiket hirdet:**
 - Prefix mellett útvonal-attribútumokat is hirdet
 - **AS-PATH:** azon AS-ek listája, amiken a prefix hirdetése keresztülment
 - **NEXT-HOP:** többértelműség miatt meg kell határozni a kilépési irányt
 - A kapott hirdetések kezelése a beállított **policy**-tól függ
- **Policy:**
 - **Inter-AS:** egy szervezet irányítani szeretné, hogy milyen forgalom és hogyan mehet át az AS-en
 - **Intra-AS:** saját forgalom és adminisztráció, nincs szükség döntésekre
- **Nagyságrend:**
 - A hierarchikus útválasztás csökkenti a táblabejegyzéseket és a routing hirdetési forgalmat
 - **Inter-AS:** kritikus
 - **Intra-AS:** nem annyira kritikus
- **Teljesítmény:**
 - **Inter-AS:** a politika fontosabb a teljesítménynél
 - **Intra-AS:** koncentrálhat a teljesítményre

Adatkapcsolati réteg

Szolgáltatások

- **Cél:** datagramok (keretek) átvitele két szomszédos, hálózati funkciókat megvalósító eszköz között
- A szomszédos elemek között egy **link** van:
 - Vezetékes
 - Vezetéknélküli
 - LAN
- **Keretezés: header** és **trailer** illesztése a datagramhoz
- Címzés **fizikai cím** (MAC-cím) alapján
- Keretek megbízható átvitele:
 - Bufferelés a végpontokon
 - Bithiba-észlelés (esetleges újraküldetés)
 - Bithiba-korrekció
- Küldési irányok kezelése egy linken:
 - Simplex
 - Half-duplex
 - Full-duplex
- **LLC (Logical Link Control) alréteg:**
 - Az adatok átvitelét szervezi
 - Multiplexáló funkció (a felsőbb rétegből érkező PDU-kra)
 - Opcionális funkciók (nyugtázás, folyamszabályozás)
- **MAC (Media Access Control) alréteg:**
 - Hibaészlelés és javítás
 - Közeghozzáférés
 - Címzés
 - Keretezés

ARP (Address Resolution Protocol)

- A hálózati elemek bizonyos ideig tárolják a feloldott címeket (ARP cache)
- Automatikusan indul, nincs szüksége kézi konfigurációra
- **Működése:**
 - Az eszköz küldene egy datagramot a cél hostnak, de a címe nincs benne az ARP-táblában, így küld egy **ARP query** keretet, benne a célcímmel
 - A cél host felismeri a saját címét és válaszol egy **ARP reply** kerettel, benne a saját MAC-címével
 - Az eszköz felveszi a cél IP-címét feloldó bejegyzést az ARP-táblájába

VLAN

- Fizikai Ethernet logikai szintű felosztása (broadcast domainek)
- Azonosítani kell, hogy melyik forgalom melyik VLAN-ba tartozik
- A keret fejléc speciális mezőjében **VLAN tag** van:
 - IEEE 802.1Q (Dot1Q)
 - 12 bájtt
 - VLAN 1 (default)
 - Normál VLAN-tartomány: 1-1005
 - Kiterjesztett VLAN-tartomány: 1006-4096
- Management VLAN kijelölése: csak ezen keresztül megengedett a konfiguráció
- A switch funkciókat VLAN-onként kell ellátni: külön MAC-címtábla és feszítőfa
- A VLAN-okat portokhoz rendeljük:
 - **Elérési (access) port:**
 - Tipikusan 1 host felé
 - Egy porthoz egy VLAN
 - Nem kell jelölni a kereteket
 - **Trönk (trunk) port:**
 - Switchek között
 - Egy porton több VLAN forgalma is megy
 - A kereteket jelölni kell, a jelöletlen keretek natív VLAN-ban vannak
 - **Router-on-stick:** trónkolt hozzáférés alinterfészek segítségével

Többszörös hozzáférés

- **TDMA** (Time Division Multiple Access): időosztás
- **FDMA** (Frequency Division Multiple Access): frekvenciaosztás
- **SDMA** (Space Division Multiple Access): térosztás
- **CDMA** (Code Division Multiple Access): kódosztás
- **WDMA** (Wavelength Division Multiple Access): hullámhosszosztás

Véletlen hozzáférés

- Egyszerű ALOHA: nincs szinkronizáció, legfeljebb 0.18 valószínűséggel ütközik
- Réselet ALOHA: csak időrések elején lehet küldeni, kétszeres esély a sikerre
- **CSMA (Carrier Sense Multiple Access):**
 - Küldés előtt ellenőrzi, hogy szabad-e a csatorna
 - A jel terjedési késleltetése miatt lehetséges ütközés
 - **CSMA/CD (ütközés érzékelés):**
 - Csak akkor küld, ha üres a csatorna
 - Ütközés esetén abbahagyja a küldést
 - Véletlen ideig vár, majd újrapróbálkozik
 - **Exponential backoff:** ha megint ütközés van, egyre tovább vár

WLAN

- Vezetéknélküli, rádióhullám-alapú kommunikáció
- Jellemző összefüggés: a vivőfrekvencia nagysága egyenesen arányos a sávzélességgel, de fordítottan arányos a hatótávolsággal
- Elérhető hálózat felépítési módok:
 - **Infrastruktúra mód:** a végpont elérési ponton keresztül kapcsolódik
 - **Ad-hoc mód:** egymáshoz kapcsolódó végpontok hálózata
- **Bázisállomás:**
 - Access Point (AP)
 - Azonosított cellák: Basic Service Set (BSS)
 - Azonosítás: Service Set ID (SSID)
- A switch/router funkció gyakran egybe van építve az AP-val
- Általában DHCP-vel konfigurál magának címeket
- Ütközések esélyének csökkentése: **CSMA/CA (ütközés elkerülés)**

Mobilhálózatok

Cellás rendszerek

- A mobil eszköz csak az idő egy részében kapcsolódik a hálózathoz
- Mobilhálózatokban komoly gond lehet a felhasználók követése
- Mobilitás-kezelés:
 - **Helyzet-nyilvántartás (Location Management):**
 - **Helyzetfrissítés (Location Update):** mobil terminálok követése
 - **Paging:** mobil terminál megtalálása broadcast üzenettel
 - **Hívásátadás-kezelés (Handover Management):**
 - **Cellán belüli átadás:** a felhasználó nem hagyja el egy adott cella lefedettségi területét, de megváltoztatja az eddig rádiós használt csatornát, csökkentve ezzel a csatornák közti interferenciát
 - **Cellák közötti átadás:** mobil terminál cellák között vándorol, felsőbb réteg támogatását is igényli
- **Elemek:**
 - **Mobil terminál (MS)**
 - **Bázisállomás (BS):** a mobil terminálok közvetlenül ezzel kommunikálnak
 - **Cella:** a bázisállomás egy adó-vevője által lefedett terület
 - **Kapcsolóközpont**
 - **Átjárók:** a mobilhálózat kapcsolódási pontja külső hálózatok felé

Rádióhálózat-típusok

- **Makrocella:**
 - Nagy területi lefedettség (1-35 km)
 - Ritkán lakott területek, gyorsan mozgó felhasználók
 - Nagy adóteljesítmények (1-20W), nagy G
- **Mikrocella:**
 - Kis területi lefedettség (200-1000 m)
 - Sok felhasználó, lassabb mobilok
 - Kis teljesítmény (10-5000 mW), nagy kapacitás
- **Pikocella:**
 - Főként beltéri és nagyon nagy forgalmú területi lefedettség
 - Kis teljesítmény (<100 mW), beltéri antennák

SDN (Software-defined Networking)

Jellemzők

- Cél: a vezérlési sík és az adatsík szétválasztása
- Az erőforrások szoftver-alapú, centralizált vezérlése
- SDN-vezérlő:
 - Interface-ek, absztrakciók a vezérléshez
 - A hálózatban elosztott adatbázis (állapot-nyilvántartás)
 - Kapcsolat a vezérelt eszközökkel: OpenFlow, SNMP
- **Adatsík:**
 - Forgalomtovábbító eszközök
 - A kapcsolótábla tartalmát a vezérlő biztosítja
 - Néhány protokollt is képesek futtatni (pl. ARP, LLDP)
- **Vezérlési sík:**
 - Logikai értelemben központosított
 - Topológiai és hálózati állapotokra vonatkozó információk kezelése
 - Eszközök felderítése
 - Forgalomtovábbítási út meghatározása
 - Biztonsági funkciók
 - Interface az alkalmazási sík felé
- **Alkalmazási sík:**
 - Meghatározzák a hálózattól kért erőforrásokat és a működést üzleti, illetve policy szempontoknak megfelelően
 - Vezérlési alkalmazások alapvető funkciói:
 - Routing/forwarding (irányítás/továbbítás)
 - Közeghozzáférés-vezérlés (access control)
 - Terhelésmegosztás (load balancing)

OpenFlow

- Az SDN megvalósításának egyik meghatározó technológiája
- Nyílt interface a vezérlési sík és az adatsík között
- A switchek TCP felett kommunikálnak a vezérlővel (6653-as port)
- **Szabályok (flow-k):**
 - L2, L3 és L4 fejléc adatok alapján integrált továbbítási szabályok
 - A switchek flow bejegyzések tábláit tárolják
 - **Információk:**
 - Illeszkedési szabály
 - Végrehajtandó akció
 - Hálózati maszk

SNMP (Simple Network Management Protocol)

Jellemzők

- De facto hálózatmenedzsment szabvány
- Gyors alkalmazhatóság
- Alkalmazkodás az eszközök változásaihoz
- Lekérdezés és változtatás
- **Kulcselemek:**
 - **Menedzser:** külön hoston futó szoftver
 - **Ügynök (agent):** a menedzselt eszközön fut
 - **MIB adatok:** ügynök szolgáltatja, menedzser nyilvántartja
 - **Parancsok**
- UDP felett a 161-es porton kommunikál (trap: 162)
- Parancsvezérelt

MIB (Management Information Base)

- A hálózatmenedzselési adatok elosztott adatbázisa
- **Faszerkezetben** hivatkozott adatelemek (változók):
 - **Azonosító: OID (Object Identifier)**
 - Számokkal jelölik az elágazásokat
 - Pl. 1.3.6.1.4.1.9.2.2 (Cisco eszközök lokális interfészei)
- Gyártófüggetlen és gyártófüggő részek

Structure of Management Information (SMI)

- Adatleíró nyelv az MIB objektumokhoz
- Abstract Syntax Notation 1 (ASN.1)
- Az ajánlás adattípusokat definiál