# Hálózati architektúrák

## 1. Számítógép-hálózat fogalma

A számítógép-hálózat egy olyan speciális rendszer, amely számítógépes rendszereket kapcsol össze, valamilyen információátviteli cél érdekében. Ez az összekapcsolás szoftveresen és hardveren is létezik.

A cél:
- erőforrások megosztása
- kommunikáció
- teljesítménynövelés
- biztonságnövelés

## 2. Hálózati fogalmak

**Hálózati csomópont**:
Önálló kommunikációra képes. Minden csomópontnak saját, egyedi azonosítója van. Egy csomópont lehet adó és vevő is.

**Átviteli közeg** (médium):
Az információ fizikai továbbítását biztosítja.

**Átviteli sebesség** (sávszélesség):
Az egységnyi idő alatt átvitt információ mennyisége. Jelölések: bit/mp, bit/sec, bit/s, b/s, bps.

## 3. Mennyiségek a hálózatban

A számítógép memóriájából a legkisebb olvasható egység: bájt ( B ).
Információtovábbítás esetén a legkisebb olvasható egység: bit ( b ).

1 **octet** = 8 bit = 1 bájt

| SI | Mennyiség | Jel | Átszámítva | IEC | Mennyiség | Jel |
|---|---|---|---|---|---|---|
| kilo | $10^3$ | k | $10^3$ B = 1 KB | kibi | $2^{10}$ | Ki |
| mega | $10^6$ | M | $10^6$ B = 1 MB | mebi | $2^{20}$ | Mi |
| giga | $10^9$ | G | $10^9$ B = 1 GB | gibi | $2^{30}$ | Gi |
| tera | $10^{12}$ | T | $10^{12}$ B = 1 TB | tebi | $2^{40}$ | Ti |

Tehát egy 500 GiB –os winchester = $5 * 2^{30}$ bájt, ami ≠ 500 GB –al, mert az $5 * 10^9$ bájtot jelent.

A kommunikációban 10 hatványokkal számolunk, memóriakezelésnél 2 hatványokról van szó.

## 4. Hálózati parancsok Knoppix alatt

**ifconfig**:

Hálózati interfészek konfigurálásra használatos. Általában rendszer-indításkor alkalmazzák, ha szükséges. Ezután, általában csak akkor, ha hibakeresésre, vagy a rendszer finomhangolására van szükség.

Ha nincs argumentum megadva, megjeleníti az aktív interfészek állapotát. Ha egyetlen interfész argumentumot kap, akkor csak annak az egy interfésznek jeleníti meg az állapotát. Ha megadjuk a **–a** (all) kapcsolót, akkor megjeleníti azon interfészeket is, amelyek DOWN állapotban vannak. Egyébként interfészt konfigurál.

| | |
|---|---|
| ifconfig eth0 | Az eth0 interfész adatainak megjelenítése. |
| ifconfig eth0 down | Az interfészt passzív állapotba tesszük, így a link is passzív lesz. |
| ifconfig eth0 up | Az aktívba tevés ír a gyűrűs pufferbe, de a passzívba tevés nem! |

**dmesg**:

A kernel gyűrűs puffer (egy végtelenített napló, amely a boot-olás óta végbement kernel üzeneteket tárolja a memóriában) vizsgálatára, illetve ellenőrzésére használatos. A puffert fájlba irányíthatjuk, ha le akarjuk másolni a tartalmát: dmesg > boot.messages. A **–c** kapcsolóval megadhatjuk, hogy a puffer kiíratása után töröljük a tartalmát.

A dmesg által generált sorok elején egy időbélyeg áll, amely megadja, hogy a boot-olás utáni hányadik mp-t és mircomp-t jelenti.

**mii-tool**:

A hálózati kapcsolatról ad információkat. Kiírja az összes aktív interfészt, az adatátvitel módját, a kapcsolat irányítottságát és a linkek állapotát is. A **–v** kapcsolót használva bővebb adatokat tudhatunk meg, például: product info, basic status, link partner, stb…

Hálózati kapcsolatok irányítottsága:

- Szimplex kapcsolat: 1 adó, 1 vevő, 1 irányú kommunikáció.
- Duplex kapcsolat: 2 irányú kommunikáció.
- Half-duplex: Egy időben egyszerre csak egy irányban folyhat a kommunikáció (wifi).
- Full-duplex: Egy időben egyszerre több irányban is folyhat a kommunikáció (kábel).

**Interfészek**:

A rendszerben szabadon létrehozhatunk interfészeket **eth**{n} néven, de alapértelmezetten az **eth0** interfész UP-olódik fel. Ugyanakkor van egy szoftveres interfész is, amelynek **lo** (loopback) a neve. Ez egy operációs rendszer szintű hálózatot teremt. Amit ezen az interfészen kiküldünk, az vissza is jön.

## 5. Csomópontok azonosítása

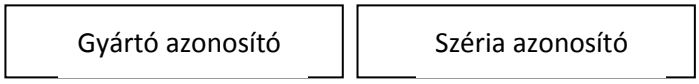
- **Fizikai cím** (hardveres vagy MAC cím)
  Egy globálisan egyedi 48 bites cím, amely egy adott hálózaton belül azonosítja a csomópontot. A hardver gyártója égeti bele a kártyába. Minden hálózati interfésznek saját fizikai címe van (például: háló kártya).

- **Logikai cím** (IP cím)
  Egy globálisan egyedi 32 bit hosszúságú cím. A hálózati kommunikáció során a hálózatok közötti azonosításra szolgál.

- **Domain név**
  Ez egy olvasható név, amelyet azért hoztak létre, hogy ne kelljen a felhasználónak a weboldalakat IP alapján azonosítani. A domain nevet a DNS (Domain Name System) fordítja IP címre.

A fizikai cím fix, amíg a logikai cím konfigurálható.

## 6. A fizikai cím felépítése




24 bit 24 bit

A gyártó azonosító (Vendor ID) és a szériа azonosító együtt tehát 48 bitet tárol. Ha egy gyártó igényel egy gyártó azonosítót és annyi szériát gyárt le, hogy már nem tud több azonosítót hozzájuk rendelni, akkor igényelhet egy újabb gyártó azonosítót.

Fizikai cím formátumai:

- Linux alatt: 00:13:D1:9B:6C:76 (pontozott hexadecimális formátum)
- Windows alatt: 00-13-D1-9B-6C-76
- CISCO: 0013.D19B.6C76

## 7. IP cím konfigurálás

| | |
|---|---|
| ifconfig eth0 0.0.0.0 | Megszűntetjük az eth0 interfészhez rendelt IP címet. |
| ifconfig eth0 10.1.1.12 | Statikusan rendelünk hozzá egy IP címet az eth0 interfészhez. |
| dhclient eth0 | Dinamikusan rendelünk hozzá egy IP címet az eth0 interfészhez. |

Ha a statikusan megadott IP cím egyezik egy másik csomópont IP címével, akkor hálózati címütközés áll fenn.

**DHCP: Dynamic Host Configuration Protokoll**

A DHCP kliens működése során küld egy DHCP kérést, majd mikor a hálózat DHCP szervere megkapja ezt a kérést (request), válaszol (answer) a kliensnek egy érvényes IP címmel. A DHCP nem adja örökbe az IP címet, erre szolgál a bérleti idő (lease time).

## 8. Rétegelt architektúra (OSI modell)

| | | | |
|---|---|---|---|
| 7. Alkalmazási réteg | 4. Szállítási réteg | 1. Fizikai réteg (kábel) | |
| 6. Prezentációs réteg | 3. Hálózati réteg (logikai cím) | | 1. → 7. : Adat fogadása (Rx) |
| 5. Viszony réteg | 2. Adatkapcsolati réteg (fizikai cím) | | 7. → 1. : Adat küldése (Tx) |

| | |
|---|---|
| 1. réteg feladata: | információ továbbítása. |
| 2. réteg feladata: | logikai szabályozás. |
| 3. réteg feladata: | hálózatok közötti kommunikáció. |
| 4. réteg feladata: | végponttól végpontig tartó megbízható adatfolyam áramlás biztosítása. |
| 5. réteg feladata: | a 2 kommunikáló alkalmazás közötti párbeszéd vezérlése. |
| 6. réteg feladata: | a különböző adat- és kódolási formátumok kódolása elérhető formába. |
| 7. réteg feladata: | közvetlenül az alkalmazásnak nyújt szolgáltatásokat. |

## 8. Ethernet

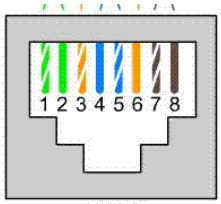
LAN környezetben a leggyakrabban alkalmazott technika.

| | | | |
|---|---|---|---|
| 10 Mbps (kezdeti kapacitás) | Vékony Ethernet, Vastag Ethernet | 100 – 500 m | Koax kábelezés, majd sodrott érpár |
| 100 Mbps (fejlesztés) | Fast Ethernet | 100 m | Optikai kábelezés |
| 1 Gbps | Gigabit Ethernet | 100 m | Sodrott érpár |
| 10 Gbps | | 100 m | Sodrott érpár |

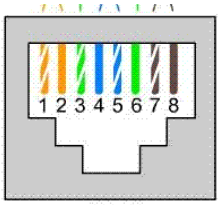
## 9. Sodrott érpárok kategóriái

| | | |
|---|---|---|
| - CAT3 (klasszikus telefonvonal) | 2 vagy 4 ér egy kábelben. | |
| - CAT5 | 8 érből álló kábel. | 100 MHz |
| - CAT5e (enhanced) | Csavart érpárok. | 112 MHz |
| - CAT6 | Zaj csökkentése műanyag elválasztóval. | 250 MHz |
| - CAT6a | | 500 MHz |
| - CAT7 | | 600 MHz |

## 10. EIA/TIA 568-as szabvány



T568A



T568B

| | | |
|---|---|---|
| **Egyenes kábel**: | A – A, B – B | Tx – Tx, Rx - Rx |
| **Fordított kábel**: | A – B | Tx - Rx |

Manapság a hálózati kártya logikailag tudja kezelni, ha egyenes kábellel kötünk össze 2 végpontot (Auto-MDI-X).

A Gigabit Ethernet mind a 8 színt használja, sőt küldésre és fogadásra egyaránt.

**Kétirányú kommunikáció**nál az információértékek összeadódnak. Ha a saját információértéket kivonjuk az érzékeltből, akkor megkapjuk a nekünk szánt információt. **Full-duplex hálózatokban már nincs ütközés**.

## 11. Alinterfészek

| | |
|---|---|
| ifconfig eth0:**2** 10.1.2.12 | Ahol **2** az alinterfész azonosító. |
| Ifconfig eth0:2 0.0.0.0 | Ekkor az alinterfész eltűnik. |

## 12. ARP protokoll

Az ARP (Address Resolution Protocol, azaz címfeloldási protokoll) arra használatos, hogy egy berendezés hozzájusson egy másik berendezés MAC-címéhez, ha annak csak az IP címe ismert.

Az IP cím meghatározása során az ARP egy úgynevezett ARP táblában keresgél. Ezt a táblát az **arp –a** paranccsal tudjuk manuálisan lekérdezni. A tábla minden sora tartalmazza az egyes IP címeket, a hozzájuk tartozó MAC címeket és azt, hogy melyik interfészt használják. Az ARP táblából egy idő után törlődnek a nem használt címek.

**Dinamikus ARP bejegyzések**

- Ha az ARP-nek találata van az ARP táblában, akkor megtalálta a címzett MAC címét és elő tudja állítani a keretet.
- Ha nincs találat, akkor az ARP üzenetszórási módszerrel szétküld egy kérést (request). Aki magára ismer, az küld egy ARP választ (reply), ez a válasz már unicast üzenet. Ekkor megvan a keresett MAC cím, a küldő beleírja az ARP táblába és ezután már előállítható a keret.

**Statikus ARP bejegyzések**

| | |
|---|---|
| arp –s IP_cim MAC_cim | Segítségével felvihetünk az ARP táblába egy IP címet és a MAC címét. |
| arp –d IP_cim | Törli a paraméterként kapott IP címet az ARP táblából. |

A statikus ARP bejegyzéseket egy PERM szócska különbözteti meg a dinamikus ARP bejegyzésektől.

**Az ARP tábla szüksége miatt belátható, hogy egy keret küldéséhez szükséges a címzett IP és MAC címe is.**

## 13. Ping parancs

A saját és egy másik csomópont közötti IP-kommunikáció ellenőrzése. Kötelező paramétere lehet IP cím vagy akár Domain cím is. Domain cím esetén a DNS oldja fel a domain címet IP címre.

A ping parancs működéséhez az IP protokollt és az ICMP –t (Internet Control Message Protocol) használja. Tehát a ping egy ICMP **echo request** –et generál. A csomópont pedig ugyancsak egy ICMP **echo reply** –al válaszol.

Kapcsolók:

| | |
|---|---|
| - s méret | Az echo request üzenet méretét adja meg. |
| - c darab | Megadja, hogy hány üzenetet küldjön egymás után a címzettnek. |
| - b | Speciális címekre is küldhetünk üzenetet, például broadcast. |
| - f | Floodolás, elárasztás (nagyon sok üzenetet küld ki egységnyi idő alatt). |

A ping parancs közbeavatkozás nélkül végtelen ideig fut, eközben folyamatosan újabb és újabb sorokat ír ki a képernyőre:

**PING** domain_cím **(**ip_cím**)** ICMP_üzenet_mérete [56 B] **(** ICMP_üzenet [56 B] + ICMP_fejléc [8 B] + IP_csomag [20 B] **) bytes of data**. **64** [8B + 56B] **bytes from** domain_cím **(**ip_cím**): icmp_seq=**sorszám **ttl=**ip_csomag_élettartama **time=**kiküldési_és_érkezési_idő **ms**

A ttl jelentése: Time to leave.
Ez az érték az IP csomagban van, mérete 1 B. A küldő 64-t ír bele, Windows alatt ez az érték 255. A csomagok útjuk során ha egy router-el találkoznak, akkor azok csökkentik a csomag ttl értékét egyel. Ha egy router látja, hogy az adott csomag ttl értéke 1, akkor lecsökkenti 0-ra, majd eldobja.

Példa a ping utolsó sorára:
rtt min/avg/max/mdev = 78.162/89.213/97.695/6.836 ms (RTT = Round Trip Time)

IPv6-os címet nem lehet megadni a ping parancsnak.

## 14. Traceroute parancs

Segítségével megállapítható, hogy a kezdő- és a célállomás között milyen útvonalon haladnak az ICMP csomagok. Működéséhez egy IP vagy egy Domain címet vár. Ezután sorban megszólítja (3x) a közte és a címzett között álló router-eket, majd végül a címzettet is.

A traceroute első sora:
**traceroute to** domain_cím **(**ip_cím**)**, n **hops max,** m **byte packets**

- **n** jelenti azt, hogy maximum hány router kötheti össze a két végpontot (ezt maximális hopp-távolságnak nevezzük),
- **m** jelzi a küldött csomagok méretét.

A traceroute többi sora (annyi sor, amennyi a hopp-távolság a küldő és a címzett között):
router_domain_címe **(**router_ip_címe**)** válaszidő_1 válaszidő_2 válaszidő_3

- ha egy adott router esetén a három megszólítás közül az egyikre nem jött időben válasz, akkor azt ***-al jelöli a program,
- ha mind a három megszóllításra ***-ot kapunk, akkor a hálózat tiltja az ICMP echo request utasítás fogadását.

**Ha ICMP protokollt használunk:** ( -I kapcsoló )
A traceroute ugyan B-nek küldi az echo request-et, de mindig csak az útvonalon soron következő sorszámú router küld választ. Ezt úgy éri el, hogy a TTL értékét kezdetileg 1-re állítja, majd addig növeli, amíg B nem válaszol. Amikor a csomagot megkapja egy router, csökkenti a TTL értékét.

Ha nem 0 a TTL új értéke, akkor továbbküldi B-nek. Ha 0 lesz az értéke, akkor eldobja a csomagot és a **time_exceeded** üzenetet küldi el nekünk, így közli velünk az IP címét. Amikor végül a megnövelt TTL érték olyan nagy, hogy elérjük B-t, akkor B már echo reply –al válaszol nekünk és az útvonalunk meg lett határozva.

**Ha UDP protokollt használunk:** ( alapértelmezett )
A traceroute egy UDP (User Datagram Protocol) üzenetet küld el, amelyhez egy külön portot használ.

Az ICMP-vel egyszerűen lehet támadni csomópontokat, az UDP alapú routerek azonban már nem válaszolnak az echo request-ekre, hanem helyette ***-ot küldenek. Tehát ha az ICMP protokollt használva az egyik router esetében mind a három megszólításra ***-ot kapunk, akkor (szinte) biztos, hogy az adott router nem válaszol az ICMP echo request-ekre.

Kapcsolók:

| | |
|---|---|
| traceroute –q darab | Hányszor szóllítson meg egy routert (alapértelmezetten 3). |
| traceroute –N darab | Egyszerre hány routert szólítson meg (alapértelmezetten 16). |
| traceroute –n | Numerikus kimenet (csak IP címeket ír ki, így gyorsabb). |
| traceroute –i interfész | Melyik hálózati interfész segítségével küldjük a kérést? |
| traceroute –m | Megadható vele, hogy mekkora legyen maximum a hopp-távolság. |

A traceroute nem garantálja, hogy létezik az útvonal.

## 15. IP címzés

_____ . _____ . _____ . _____
8 bit 8 bit 8 bit 8 bit

Ezt a 32 bitet **k** darab **hálózati azonosító** bit és (**32-k**) darab **csomópont**ot **azonosító** bit alkotja.

Hálózatosztályok:

| Név | Hálózati azonosító | Csomópont azonosító | Első 8 bit értéke | Hálózatok száma | Csomópontok száma |
|---|---|---|---|---|---|
| A: | 1-8 bit | 9-32 bit | 0 – 127 | 128 db. | $2^{24}$ db. |
| B: | 1-16 bit | 17-32 bit | 128 – 191 | 64 * 256 db. | $2^{16}$ db. |
| C: | 1-24 bit | 25-32 bit | 192 – 223 | 32 * 256 * 256 db. | $2^{8}$ db. |
| D: | Multicast címzésre használjuk. | | 224 – 239 | | |
| E: | Kutatási területre felhasználható. | | 240 – 255 | | |

A C típusú hálózatosztály esetében a hálózatok száma a következőképpen alakul:
- az első 8 bit értéke 192-223 közé eshet, amely összesen 32 felvehető különböző érték,
- mivel a hálózat azonosításhoz 24 bit kell és a másik 2*8 bit értéke nincs korlátozva, ők egyenként 256-féle értéket vehetnek fel,
- ezért a hálózatok száma: 32 * 256 * 256 darab.

A kisebb intézmények tipikusan C típusú hálózatosztályt alkalmaznak, de a nagyobb cégek A típusú osztályt használnak.

**Minden hálózatnak van 2 kitűntetett IP címe**: ( ezek a címek nem oszthatóak ki IP címnek )

| | | | |
|---|---|---|---|
| Hálózati cím | A hálózatban kiosztható legkisebb cím. | Csupa 0-s csomópont azonosító bit. | MINDIG PÁROS! |
| Üzenetszórási cím | A hálózatban kiosztható legnagyobb cím. | Csupa 1-es csomópont azonosító bit. | MINDIG PÁRATLAN! |

**Az IP címzés hierarchikus**: először azonosítjuk a hálózatot, majd azon belül a csomópontot.

## 16. Alhálózatok kialakítása

Alhálózatok esetén a definiált hálózatazonosító biteken kívül újabb hálózatazonosító biteket határozunk meg, természetesen az így létrejött alhálózatok esetében csökken a kiosztható csomópontok száma.

Legyen egy C típusú hálózatosztály IP címe: **192.1.1.0**
Bontsuk fel a hálózatot 4 alhálózatra:

A csomópont azonosításra felhasznált bitek közül az első kettőt (mindig előröl indulunk) válasszuk meg az alhálózat azonosítására (subnet, vagy alnet bitek). Ekkor a korábbi 8 bitből már csak 6 bitünk marad csomópont kiosztásra. Mivel a legkisebb cím a hálózati címet jelenti, a legnagyobb cím pedig az üzenetszórási címet, így a kiosztható IP címek száma az egyes alhálózatokban: $2^6$ – 2 darab.

A létrehozott 4 alhálózat hálózati és üzenetszórási címei, valamint a kiosztható IP címek:

| | | | |
|---|---|---|---|
| 00 bitek | 192.1.1.**0** | 192.1.1.**63** | 192.1.1.**1** – 192.1.1.**62** |
| 01 bitek | 192.1.1.**64** | 192.1.1.**127** | 192.1.1.**65** – 192.1.1.**126** |
| 10 bitek | 192.1.1.**128** | 192.1.1.**191** | 192.1.1.**129** – 192.1.1.**190** |
| 11 bitek | 192.1.1.**192** | 192.1.1.**255** | 192.1.1.**193** – 192.1.1.**254** |

**Milyen alhálózatot hozzunk létre, ha szeretnék egy hálózatot, ahol 55 IP címünk van?**

C típusú hálózatosztály esetében $2^6 > (55 + 2)$. // ahol 2 jelenti a hálózati és üzenetszórási címet
Így legalább 6 bit szükséges az 55 IP cím tárolására, a maradék 2 bit pedig az egyes alhálózatokat azonosítja.
A 4 alhálózatból, amelyek a 2 hálózatot azonosító bitből jön létre, nekünk csak 1re van szükségünk, például a 00-ásra.

**Milyen alhálózatot hozzunk létre, ha szeretnénk egy hálózatot, amelyben 4 alhálózatunk van és mindegyikben 12 IP címünk?**

C típusú hálózatosztály esetében $2^4 > (12 + 2)$.
Így legalább 4 bit szükséges a 12 IP cím tárolására, a maradék 4 bit pedig a hálózatéhoz.

Az **utolsó 2 bit**et meg kell hagyni csomópont azonosító biteknek, mert 1 bit esetén csak a hálózati és üzenetszórási címet tárolnánk.

## 17. Alhálózati maszk

Az alhálózati maszk hossza megegyezik az IP cím hosszúságával. A maszkot felépítő bitek jelzik, hogy az IP címünk egyes bitpozícióján hálózatot vagy csomópontot azonosító bit áll e. Az 1-es bit jelentése, hogy hálózatot azonosít, a 0-s bit pedig csomópontot azonosít.

A hálózati osztályok hálózati maszkjai:

| | | | |
|---|---|---|---|
| A | 255.0.0.0 | /8 | (az első 8 bit hálózatot definiál) |
| B | 255.255.0.0 | /16 | |
| C | 255.255.255.0 | /24 | |

**Legyen az alhálózati maszk: 255.255.224.0 és az IP cím: 150.16.32.0. Mi az üzenetszórási cím?**

Felírva az alhálózati maszkot: 11111111 . 11111111 . 11100000 . 00000000

Mivel az első 8 bit tárolt értéke 150, tudjuk, hogy ez egy B hálózatosztályú IP cím, ahol az első 16 bit azonosítja a hálózatot és további 3 bit az alhálózatot.

Az IP címet írjuk fel binárisan: **10010110** . **10000000** . **001**00000 . 00000000

Látható, hogy az IP címünk a 001 alhálózatba tartozik.

Ennek az alhálózatnak az üzenetszórási címe: 10010110 . 10000000 . 001**11111** . **11111111**

Azaz **150.16.63.255**.

**Legyen az IP címünk 193.6.153.64, és tudjuk, hogy az első 26 bit hálózatot definiál. Hálózati cím az IP címünk?**

Felírva az alhálózati maszkot: 11111111 . 11111111 . 11111111 . 11000000

Mivel az első 8 bit tárolt értéke 193, tudjuk, hogy ez egy C hálózatosztályú IP cím, ahol az első 24 bit azonosítja a hálózatot és további 2 bit az alhálózatot.

Az IP címet írjuk fel binárisan: **11000001** . **00000110** . **10011001** . **01**000000

A válasz: **igen**, mert látható, hogy az IP címünk a 01 alhálózatba tartozik és minden csomópontot azonosító bit 0.

**Legyen az IP címünk 193.6.153.65, és tudjuk, hogy az első 25 bit hálózatot definiál. Hálózati cím az IP címünk?**

Felírva az alhálózati maszkot: 11111111 . 11111111 . 11111111 . 10000000

Mivel az első 8 bit tárolt értéke 193, tudjuk, hogy ez egy C hálózatosztályú IP cím, ahol az első 25 bit azonosítja a hálózatot és további 1 bit az alhálózatot.

Az IP címet írjuk fel binárisan: **11000001** . **00000110** . **10011001** . **01**000001 ( láthatjuk azt is, hogy ez nem páros szám )

A válasz: **nem**, mert látható, hogy az IP címünk a 01 alhálózatba tartozik, azonban az utolsó csomópontot azonosító bit értéke 1.

**32 darab** /27-es hálózat kiad **1 darab** /22-es hálózatot, mert egy /22-es hálózatot $2^5$ darab alhálózatra akarunk felbontani és mivel ehhez 5 újabb hálózatot azonosító bitre van szükségünk, az alhálózatok /(22+5)-ös hálózatok lesznek, azaz /27-esek.

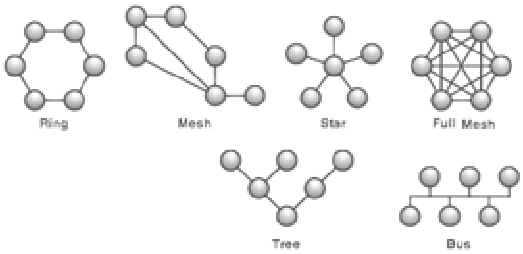
## 18. Alhálózat további alhálózatokra bontása

Legyen az IP címünk **193.6.140.64** és az első **26** bit hálózatot azonosító bit. Ezt a hálózatot szeretnénk 8 részre bontani.

Kiszámoljuk, hogy $2^3 = 8$.

Ezért lefoglalunk újabb 3 bitet a hálózat azonosítására. Tudjuk, hogy eredetileg (26-24) = 2 bit töltötte be az alhálózat azonosítási szerepét, de újabb 3 bit, azaz most már 5 bit azonosítja az alhálózatot. Így a korábban definiált 4 alhálózatnak mostantól alhálózatonként újabb **8 al-alhálózat**ja van és ezen al-alhálózatoknak egyenként ($2^3 – 2$) darab kiosztható IP címe van.

## 19. Hálózatok osztályozása topológia szerint




**Busz** (bus) Leggyakrabban koax kábelezést alkalmaznak.

**Gyűrű** (ring) Nagyobb a hibatűrése.

**Csillag** (star)

- repeater Jelismétlő, amely erősíti és kondicionálja a jelet.
- hub Több portos repeater.
- bridge Felhasználja a keret fejrészét és csak akkor kerül át a kimenetre, ha az eszköz azon az oldalon van.
- switch Több portos bridge, amely két tetszőleges interfész között tud kapcsolatot létesíteni.

**Fa** (tree)

**Háló** (mesh)

- részleges mesh
- full mesh

Csavart érpárok: UTP, FTP, STP, S/FTP (8 tűsek).