Kiegészítés a *Számítógép-hálózatok* jegyzethez a 2. ZH témakörében

v0.8.5.1, 2012. 06. 04.

Internet Protocol

Az osztálymentes címzés

Miért van rá szükség?

Problémák:

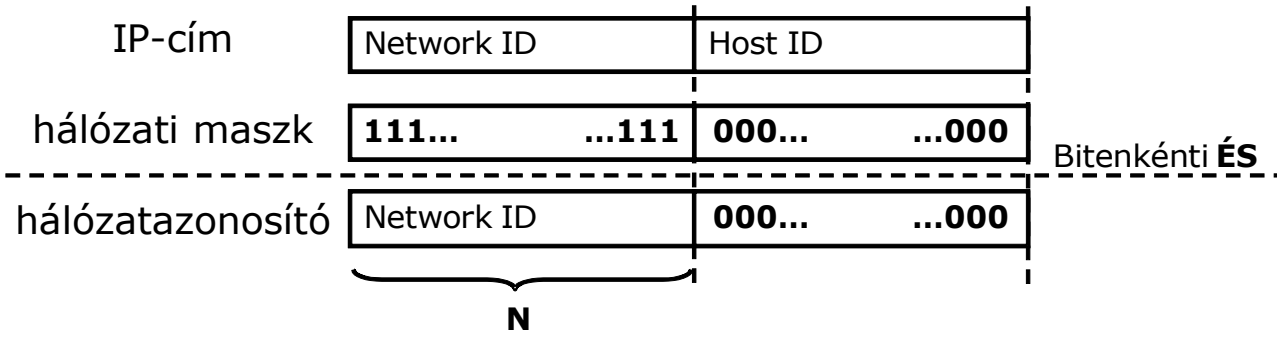
1. Az *osztály alapú címzés* (classful addressing) nem igazodik a fizikai hálózatstruktúrához
   - Egy nagy hálózat kisebb fizikai hálózatokból épül fel.
     → Az A, B osztályú hálózatokat kisebb hálózatokra KELL bontani.
2. A címzésben a két hierarchiaszint túl kevés
   - Az útválasztási (routing) táblázatok mérete drasztikusan nőtt
     → aggregációval megoldható a több hierarchia szint bevezetése.

Megoldás: *osztálymentes címzés* (classless addressing) és CIDR (Classless Inter-Domain Routing) 1517-20 RFC-k

- Nem az IP-cím első néhány bitjéből, hanem egy maszk alapján állapítjuk meg, hogy hol a határ az IP-cím két része közt.
- VLSM – Variable Length Subnet Mask
  A cím 32 bitje tetszőleges helyen lehet kettéosztva hálózat- és végponti azonosítóra.

Az osztálymentes címzés megvalósítása

Az osztálymentes címzést a *hálózati maszk* használatával lehet megvalósítani. Az IP-cím és a hálózati maszk közötti bitenkénti logikai ÉS művelet eredményeként előáll a hálózatcím:



Jelölés a hálózat egyértelmű azonosítására:

<hálózat IP-címe> / <hálózati maszk egyeseinek a száma>

Így pl. a Távközlés-informatika Labor egyik publikus címtartománya: 193.224.130.160/27 (hálózati maszkja 255.255.255.224).

Hasonlóan: A osztály: /8, B osztály: /16, C osztály: /24

Az IP címzés történeti fejlődése és következményei

Az osztály alapú címzéstől az osztálymentes címzésig a fejlődés több lépcsőben valósult meg. Előbb a nagyobb hálózatokat alhálózatokra bontották, ez a *subnetting* (lásd jegyzet 4.1.5. alfejezet), majd hasonló technikával megoldották a több kisebb hálózat nagyobb hálózattá való összevonását, ez a *supernetting*, de ettől kezdve már a kettőt nem volt érdemes megkülönböztetni, így *osztálymentes címzés*ről beszélünk. Az útválasztási algoritmus is előbb a *subnet rounting* volt, aztán jött a CIDR (Classless Inter-Domain Routing), de mi már csak az utóbbival foglalkozunk!

A fejlődés közbenső lépcsőjének „melléktermékei”:

- Visszamaradt kifejezések ma is élnek: pl. „alhálózati maszk” (subnet mask) kifejezés használata; helyesen ma már „hálózati maszk” (netmask), hiszen a CIDR-ben a supernetting is benne van!
- Visszamaradt szabályok élnek a köztudatban: pl. egy nagyobb címtartomány kisebb hálózatokra való bontásakor az első és utolsó subnet (ahol az alhálózatot megadó bitek értéke csupa 0 és csupa 1) nem osztható ki, legfeljebb további felosztásra használható (RFC 950), pedig ez már régen nincs így, lásd RFC 1878.

Speciális IPv4 címek

- **Host ID: csupa 0**
  - Hálózat címe
  - Az adott hálózat eszközei (elvileg) opcionálisan kezelhetik
- **Host ID: csupa 1**
  - Broadcast cím (directed broadcast; elavult kifejezéssel: subnet broadcast)
  - Az adott hálózaton mindenkinek szól
- **127.0.0.0/8**
  - Loopback cím
  - A helyi gépet azonosítja
  - Bármelyik használható, a 127.0.0.1 a szokásos
- **Privát IP-címtartományok (RFC 1918)**
  - Csak helyi hálózaton (Interneten nem) érvényes címek
  - 10.0.0.0/8 (1 db A osztály)
  - 172.16.0.0/12 (16 db B osztály)
  - 192.168.0.0/16 (256 db C osztály)
- **Link lokális IP-címtartomány (RFC 3927)**
  - 169.254.0.0/16
  - Csak helyi kommunikációra, a routerek nem továbbítják!
  - Ebből az automatikus címkonfigurációhoz használható címekbe az első és utolsó 256 db cím nem tartozik bele, ezeket további célokra tartják fenn!
  - további info: http://en.wikipedia.org/wiki/Link-local_address

További speciális célra lefoglalt tartományok: RFC 5735

SLAAC: Stateless Address Autoconfiguration (IPv4, RFC 3927)

Az állapotmentes automatikus címkonfigurációt IPv4 esetén az RFC 3927 *IPv4 link lokális címek dinamikus konfigurációja* néven (Dynamic Configuration of IPv4 Link-Local Addresses) említi, csak az IPv6 hasonló megoldása miatt hívjuk SLAAC-nek.

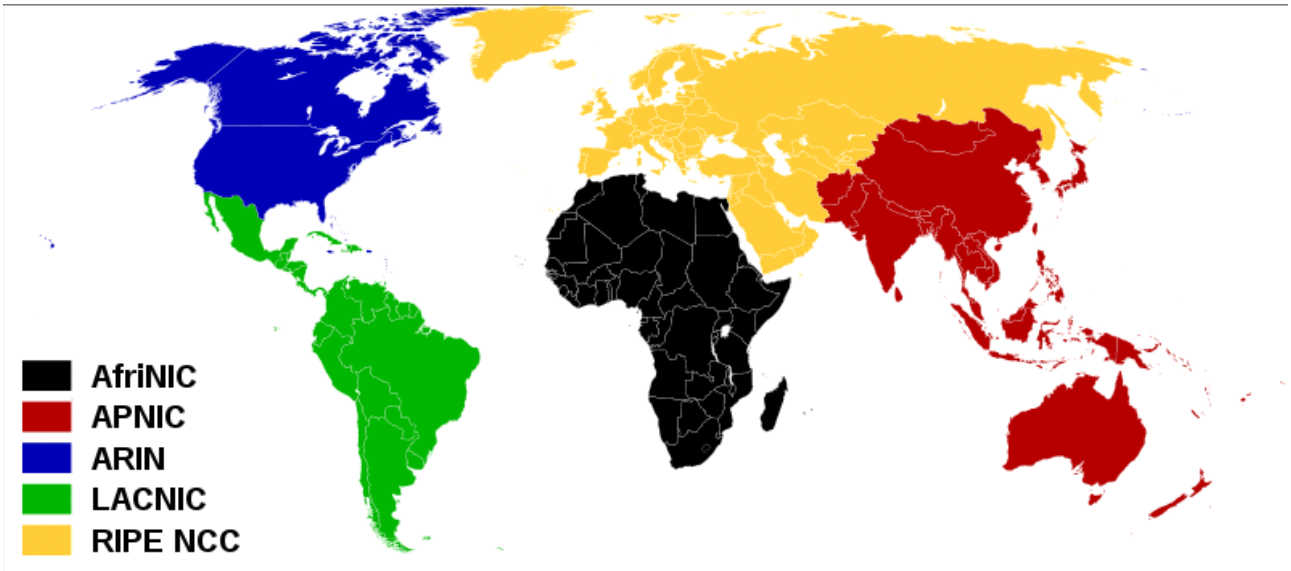
Ha nincs IP-cím statikusan beállítva, és az állomás nem kap választ egyetlen DHCP szervertől sem, akkor a 169.254.1.0 – 169.254.254. 255 tartományból véletlenszerűen választ magának egy IP-címet. Ellenőrzi (*ARP probe* segítségével), hogy már használatban van-e, és csak akkor használja, ha más nem használja azt (különben másik címmel próbálkozik). Ezzel csak az adott fizikai hálózaton tud kommunikálni.

APIPA: Automatic Private Internet Protocol Addressing (microsoftos kifejezés az IPv4 SLAAC megvalósítására), az RFC 3927 megsértésével 169.254.0.1-től 169.254.255.254-ig az összes címet használják: http://msdn.microsoft.com/en-us/library/aa505918.aspx

Az IP-címek kiosztása

Az IP-címeket az IANA (Internet Assigned Numbers Authority) osztja ki az 5 RIR (Regional Internet Registry) számára „/8” blokkonként.

A RIR-ek kisebb blokkonként osztják tovább internet szolgáltatóknak, oktatási intézményeknek, nagy cégeknek, stb.

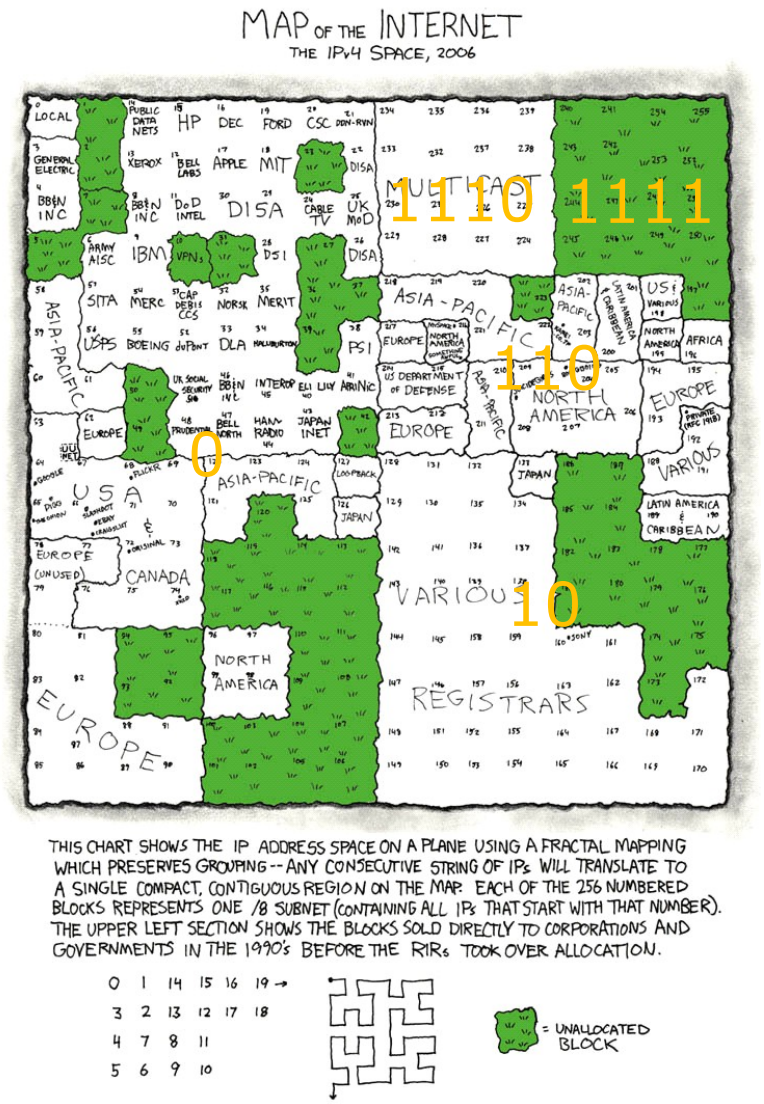


A kép forrása: http://en.wikipedia.org/wiki/File:Regional_Internet_Registries_world_map.svg

A kiosztott IPv4 címek - 2006 évi állás szerint

A 2006-os állapot fraktálos ábrázolásán figyeljük meg:

- kezdeti „/8” blokkok nagy cégeknek (főleg az első negyedben)
- A „C” osztálynál látható a regionális elv érvényesülése
- A „D” osztály a multicastnak van lefoglalva
- Az „E” osztály fenntartott



A kép forrása: http://xkcd.com/195/

Az IPv4 címek elfogyása

- A „B” osztályú IP-címek fogytak el először…
- Különböző módszerekkel sikerült jelentősen kitolni az IPv4 címtartományának kimerülését
  - osztálymentes címzés
  - privát IP-címek + NAT (RFC 1631) használata
  - a kezdetben kiosztott túlságosan nagy címtartományok visszaadása
- 2011. 01. 31-én az IANA kiosztotta az utolsó „/8” blokkokat a RIR-eknek
- Az „E” osztályt végül nem osztották ki, mert nem minden eszköz kezelné...
- Az APNIC címtartománya 2011. 04. 15-én kimerült
- A RIPE NCC címtartományának kimerülését 2012 közepére várják…
  → Az IPv6 bevezetése tovább Magyarországon sem halogatható!
  → Az IPv6, valamint a két rendszer együttélésének oktatása kulcsfontosságú!

Transmission Control Protocol

TCP ellenőrző összeg

A TCP protokoll az ellenőrző összeg számításakor a TCP szegmens elé helyez egy úgynevezett *pseudo header*t, ami tartalmazza az IP-címeket is. Így ki fog derülni, ha útválasztási hiba miatt a TCP szegmenst szállító datagramot tévesen kézbesítették.

TCP torlódásvezérlés

A *torlódásvezérlés* (congestion control) célja annak az elkerülése, hogy valamely közbenső hálózati eszköz (router, link) túlterhelése miatt a hálózat teljesítőképessége radikálisan csökkenjen (congestive collapse).

Miért is fordulhatna elő ilyen állapot?

- Ha a hálózat terhelése túl nagy (megközelíti a kapacitását), akkor a csomagvesztés megnő, és a TCP elkezdi újra küldeni a nyugtázatlan szegmenseket.
- Az újraküldés okozta terhelés növekedés további csomagvesztéseket okoz, és az önmagát gerjesztő folyamat odáig jut, hogy a hálózat kapacitásának csak a töredékét képes átvinni rendkívül rossz minőségi jellemzők mellett.

A TCP torlódásvezérlésre számos algoritmus létezik, az aktuális szabvány kereteit az RFC 5681 adja meg, a használt implementációkról érdeklődőknek bővebben:
http://en.wikipedia.org/wiki/TCP_congestion_avoidance_algorithm

A torlódásvezérlésre használt algoritmusok bevezetik a *torlódási ablak* (congestion window) fogalmát. (Figyelem! Ez nem azonos a TCP *ablakméret* (Window) paraméterével!) A torlódási ablak célja, hogy korlátozza két kommunikáló fél között a nyugtázatlan szegmensek (pontosabban a bennük levő adat oktettek) számát. De a TCP ablakméret paraméterével szemben ennek mérete nem a két kommunikáló fél képességeitől, hanem a hálózat aktuális áteresztő képességétől függ. Az algoritmusok ennek méretét változtatják attól függően, hogy tapasztalnak-e torlódásra utaló jeleket; ha ilyen jelek nincsenek, a méretét növelik, ha vannak, akkor csökkentik. Természetesen egy host adásakor a nyugtázatlanul elküldhető adatmennyiséget mindkét ablak mérete korlátozza:

küldhető oktettek száma = min(a vevő által megadott Window, congestion window)

Mik lehetnek a *torlódásra utaló jelek*?

- Egy szegmens küldése után a (hálózati viszonyok alapján adaptívan állított) timeout letelt és nem jött nyugta. Ennek az oka hálózati torlódás is lehet, de persze más is (például bithiba miatt egy IP router eldobta a szegmenst vagy annak nyugtáját szállító datagramot).
- Egy szegmensre többszörös nyugta érkezett. Ennek több oka lehet, például:
    - Torlódás miatt egy szegmens vagy annak nyugtája késett, emiatt a szegmenst újra elküldték. A nyugta aztán mindkét szegmensre megjött. – Ebben az esetben valóban torlódásra utal!
    - Csomagok sorrendje felcserélődött: egy később küldött csomag előbb érkezett meg, és a fogadó fél ezzel jelzi, hogy nem azt várta, hanem egy másik, korábban küldött

csomagot (kisebb sorszámmal). – Ebben az esetben nem biztos, hogy a sorrendcsere oka torlódás!

    - A nyugtát szállító IP datagram megduplázódott az IP alatti réteg hibája miatt. – Ez egyáltalán nem jelent torlódást.
- Az RFC 3168 szerinti *Explicit Congestion Notification* érkezett.

Még két felhasznált fogalom:

- RTT: Round-Trip Time: az az időtartam, ami TCP szegmens elküldésétől a szegmens nyugtázására alkalmas nyugta megérkezéséig eltelik. (Arra gondolunk, hogy a nyugtát a szegmens váltotta ki, de lehet, hogy valójában nincs ok-okozati összefüggés köztük, csak a kapott nyugta nem megkülönböztethető.)
- MSS: Maximum Segment Size: a TCP szegmens számára megengedett maximális méret. Ez az IP-t szállító hálózat MTU-jától függ, a TCP kapcsolat felépítésekor egymásnak TCP opcióval megadhatják ezen paraméterüket.

Az **AIMD – Additive Increase / Multiplicative Decrease algoritmus**

Az algoritmus lényege, hogy a növelés fix értékek hozzáadásával, a csökkentés viszont 1-nél kisebb számmal való szorzással történik. Az algoritmus általánosabb, mint nekünk kell, a pontos leírása megtalálható: http://en.wikipedia.org/wiki/Additive_increase/multiplicative_decrease

A TCP-nél használt változatát az RFC 5681-ben *congestion avoidance* névvel illetik és működése a következő:

Minden RTT alatt:

- Növeljük congestion window értékét MSS-sel, ha nincs torlódásra utaló jel.
- Csökkentsük congestion window értékét a felére, ha van torlódásra utaló jel.

Vegyük észre, hogy a növelés csak lineáris (azaz viszonylag lassú), a csökkentés viszont exponenciális; torlódás esetén nagyon gyorsan a felére, negyedére, stb. tud csökkenni.

Kezdőértékként használható egy fix érték, jobb híján akár MSS is. Azonban az algoritmust megelőzően az ún. *slow start* algoritmust szokták használni, aminek csak egyik jellemzője a lassú kezdés, a másik éppen a gyors (exponenciális) növekedés.

A slow start működése:

Kezdetben congestion window := MSS. (Pontosabban MSS méretétől függően kb. 2xMSS vagy 3xMSS, de ez részletkérdés.) Minden RTT alatt:

- Növeljük congestion window értékét a duplájára, ha nincs torlódásra utaló jel.
- Fejezzük be az algoritmust, ha van torlódásra utaló jel.

A slow start befejezésekor áttérünk a *congestion avoidance* algoritmusra.

A kettő kombinációjának az előnye, hogy a slow start segítségével kezdetben kis értékről gyorsan növekszik a congestion window, de amint torlódásra utaló jel van, rögtön átvált a stabil congestion avoidance algoritmusra.

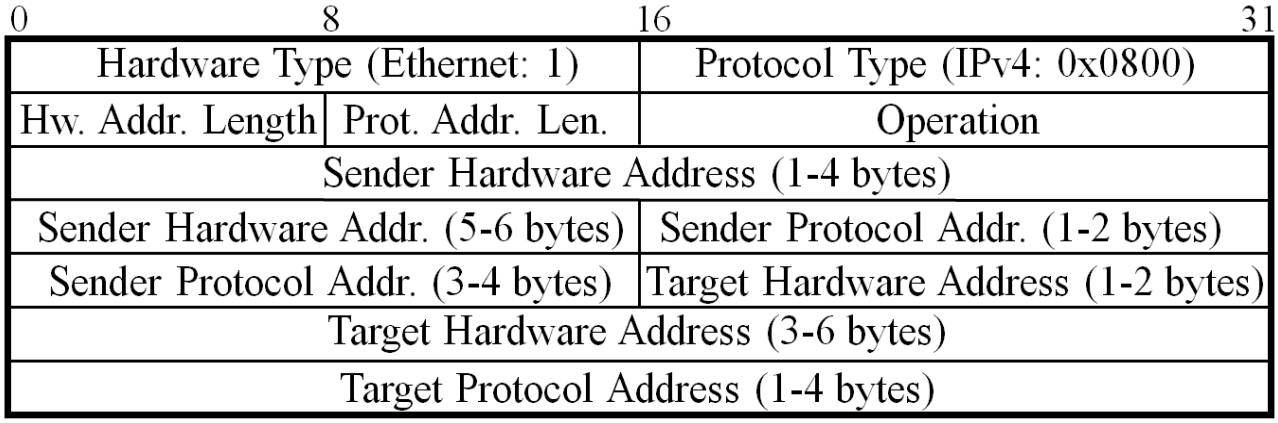
Megjegyzés: Az RFC 5681 leírja, hogy egy küszöbnél (ssthresh) mindenképpen váltani kell, illetve leír két további algoritmust is (fast retransmit, fast recovery). Ezekkel mélyebben nem foglalkozunk.

Address Resolution Protocol

ARP üzenetek felépítése

Az ARP üzenetek az Ethernet keret adat mezőjében utaznak. Ethernet szinten a célcím *ARP Request* esetén broadcast (FF:FF:FF:FF:FF:FF), *ARP Reply* esetén általában a kérés küldőjének unicast címe, de elvileg lehet broadcast is. Az *EtherType* mező értéke mindig 0x0806, erről ismerhető fel, hogy az Ethernet fölött ARP üzenet utazik.

Az ARP üzenet felépítése a következő:



Ahol:

- Hardware Address Length: 6 (Ethernet MAC-cím hossza)
- Protocol Address Length: 4 (IPv4 IP-cím hossza)
- Operation: *ARP Request* esetén: 1, *ARP Reply* esetén: 2.

A névfeloldás menete ARP-vel

Az „A” állomás szeretné megtudni a „B” állomás MAC-címét annak IP-címe alapján.

1. „A” Ethernet szinten az FF:FF:FF:FF:FF:FF (broadcast) célcímre küld egy *ARP Request*et, melyben a forráscím a sajátja, az EtherType mező értéke 0x0806. Az *ARP Request* (nem triviális) mezői:
   - Operation: 1 (Request)
   - Sender HA: <„A” MAC-címe> (Megegyezik a keret fejrészében találhatóval)
   - Sender PA: <„A” IP-címe>
   - Target HA: 00:00:00:00:00:00 (ismeretlen) (De a keret fejrészében a cél MAC-cím: FF:FF:FF:FF:FF:FF !!!)
   - Target PA: <„B” IP-címe>
2. Az *ARP Request* üzenetet a *broadcast domain* összes állomása veszi, és tárolja az „A” IP-cím – MAC-cím párosát az *ARP Cache* táblájában.
3. „B” felismeri a saját IP-címét az *ARP Request* üzenetben, ezért válaszként „B” Ethernet szinten az „A”-nak címezve küld egy *ARP Reply*t, melyben a forráscím a sajátja, az EtherType mező értéke most is 0x0806. Az *ARP Reply* (nem triviális) mezői:
   - Operation: 2 (Reply)

- Sender HA: <„B” MAC-címe> (Megegyezik a keret fejrészében találhatóval)
- Sender PA: <„B” IP-címe>
- Target HA: <„A” MAC-címe> (Megegyezik a keret fejrészében találhatóval)
- Target PA: <„A” IP-címe>

4. „A” veszi a választ, és eltárolja „B” IP-cím – MAC-cím párosát.

Az ARP Cache tábla kezelése

Az állomások bizonyos ideig tárolják az ARP-vel megszerzett névfeloldási információkat. Mivel az *ARP Request*et adatkapcsolati szinten broadcast címre kell küldeni, a küldő IP-cím – MAC-cím párosát minden állomás el tudja tárolni. Ha esetleg a választ is broadcast címre küldik, akkor azt is el lehet tárolni. A dinamikus bejegyzéseken kívül (az **`arp`** menedzsment szoftverrel) statikus bejegyzések is felvehetők. Az ARP Cache tartalma általában az **`arp -a`** paranccsal meg is jeleníthető, az **`arp -d`** paranccsal pedig bejegyzések törölhetők belőle.

Az ARP Cache tábla felépítése

Az ARP Cache táblában természetesen IP-cím – MAC-cím párok vannak. Két bejegyzéstípus létezik:

- **Statikus**: Manuálisan felvitt bejegyzés eredménye. Amíg nem törlik, változatlanul marad.
- **Dinamikus**: ARP címfeloldás eredménye. Gyorsítótár (cache) funkciót valósít meg; ne kelljen mindig lekérdezni. Egy idő után elévül és törlődik.

Az ARP Cache tábla elvi felépítése:

IP-cím	**HW-cím**	**típus**
<IP-cím1>	<MAC-cím1>	statikus
<IP-cím2>	<MAC-cím2>	dinamikus

A gyakorlatban még egyéb információt is tárolnak, például a hardver típusát (ami Etherneten kívül más is lehet) és az interfészt, amelyiken keresztül az a hálózat elérhető, amelyen az adott szomszéd található. Például Linux alatt az ARP Cache táblát megvizsgálva:

```
root@dev:~# arp -n
Address             HWtype   HWaddress           Flags Mask    Iface
193.224.130.161     ether    00:15:17:54:99:78   C             eth0
```

(A **`C`** flag jelzi a cache-elt (dinamikus), az **`M`** pedig a manuálisan beállított (statikus) értékeket.)

IPv4 Address Conflict Detection (RFC 5227)

Mielőtt egy host elkezd egy IP-címet használni, meg kell (SHOULD) vizsgálnia, hogy az IP-cím nincs-e már használatban. Erre való az *ARP Probe* üzenet: ez egy speciális *ARP Request*, amellyel a használni kívánt IP-címhez tartozó MAC-címre kérdez rá a TPA (Target Protocol Address)

mezőben, de az SPA (Sender Protocol Address) mezőben a 0.0.0.0 IP-cím található; ennek célja, hogy ne szennyezze mások ARP Cache-ét, azaz ha mégsem használhatja a címet, akkor ne tárolják el a hamis információt. Ha az *ARP Probe* üzenetre választ kap, akkor tudja, hogy más valaki már használja a kérdéses IP-címet.

Ha DHCP-vel kapott IP-címről derül ki, hogy más valaki már használja, akkor kötelező (MUST) a DHCP szerver felé DHCPDECLINE üzenetet küldeni.

Az RFC 5227 nem rendelkezik róla konkrétan, hogy az *ARP Probe* üzenetet hányszor kell elküldeni, de megemlíti, hogy a megfelelően alacsony hibavalószínűség érdekében többször.

Ha az IP-cím szabadnak bizonyult, akkor a fentiek szerint eljáró host köteles (MUST) *ARP Announcement* üzenettel jelezni mindenki számára, hogy az adott IP-címet ő fogja használni. Ez olyan *ARP Request* típusú üzenet, ahol a sender és a target IP mezőben egyaránt az adott IP-cím szerepel. Mivel broadcast címre küldik, mindenki megkapja, és az ARP Cache tábláját frissíteni tudja.

Sajnálatos módon *ARP Probe* helyett bizonyos implementációkban *Gratuitous ARP* (kéretlen ARP) üzeneteket használnak. Így hívják mind az *ARP Request* nélkül, broadcast címre küldött *ARP Reply* üzeneteket, mind az *ARP Probe* nélkül küldött *ARP Announcement* üzeneteket.

Ez a módszer azért nem jó, mert:

- Nem óvja meg a már működő gépek működőképességét.
- Nem teszi lehetővé a most induló gépnél sem azt, hogy automatikusan (emberi beavatkozás nélkül) más IP-címet használjon.

Dynamic Host Configuration Protocol

A DHCP általános jellemzői

A DHCP az alkalmazási rétegben működő protokoll. Segítségével a hostok automatikusan juthatnak hozzá a kommunikációjukhoz szükséges hálózati azonosítókhoz: IP-cím, hálózati maszk, alapértelmezett átjáró, stb. Eredetileg az RFC 1531 a BOOTP kiterjesztéseként definiálta. Újabb RFC-k: 1541, 2131 (aktuális).

A DHCP lehetőségei:

- IP-címek osztása MAC-cím alapján DHCP szerverrel

  Szükség esetén (a DHCP szerveren előre beállított módon) egyes kliensek számára azok MAC-címéhez fix IP-cím rendelhető.

- IP-címek osztása dinamikusan

  A DHCP szerveren beállított tartományból „érkezési sorrendben” kapják a kliensek az IP-címeket. Így elegendő annyi IP-cím, ahány gép egyidejűleg működik.

- Az IP-címeken kívül további szükséges hálózati paraméterek is kioszthatók:
  - Hálózati maszk
  - Alapértelmezett átjáró
  - Névkiszolgáló
  - Domain név

    - Hálózati rendszerbetöltéshez szerver és fájlnév

Az IP-címek bérlésének szabályai:

- A DHCP szerver a klienseknek az IP-címeket bizonyos *bérleti időtartam*ra (lease time) adja oda használatra.
    - Az időtartam hosszánál a szerver figyelembe veszi a kliens esetleges ilyen irányú kérését.
    - Az időtartam hosszát a szerver beállításai korlátozzák.
- A bérleti időtartam lejárta előtt a bérlet meghosszabbítható.
- Az IP-cím explicit módon vissza is adható.

A DHCP működése és üzenetei

A kliens és a szerver *DHCP üzenetek*kel kommunikálnak. A DHCP üzenetek BOOTP üzenetekben opcióként jelennek meg. A BOOTP üzenetek IP fölött, UDP-be ágyazva haladnak. Amíg a kliensnek nincs érvényes IP-címe, addig 0.0.0.0-t használ. Broadcast esetén IP szinten természetesen 255.255.255.255 címre küldi az üzenetet (Ethernet szinten pedig FF:FF:FF:FF:FF:FF címre). UDP-ben a kliens portszáma: 68, a szerveré: 67.

A továbbiakban a DHCP üzenetek neve mellett feltüntetjük, hogy ki küldi kinek: „küldő → címzett” formában. Jelölések:

- K: kliens
- S: szerver
- B: broadcast (IP és Ethernet szinten is)

Egy IP-cím megszerzéséhez a következő négy üzenetre van szükség:

DHCPDISCOVER K→B

Egy kliens küldi broadcast címre, hogy feltérképezze az elérhető DHCP szervereket és ajánlataikat.

A kliens opcionálisan (nem az IP fejrészben, hanem DHCP opcióként) megadhatja a legutoljára használt IP-címét, de ez NEM azonos a bérlet meghosszabbításával!

DHCPOFFER S→K

Egy DHCPDISCOVER üzenetre egy vagy több szerver válaszol, megadja, hogy milyen IP-címet és paramétereket tud kínálni.

Ekkor még ezek a kliens számára NEM használhatók!

DHCPREQUEST K→B

A kliens ezzel az üzenettel egyidejűleg elfogadja valamely szerver ajánlatát, és implicit módon elutasítja a többiekét (broadcast miatt minden szerverhez eljut).

A kliens megjelölheti benne a kért bérleti időtartamot is.

DHCPACK S→K

A szerver ekkor megerősíti a kliensnek az IP-cím bérletét, és megadja, hogy milyen időtartamra kapja meg a kliens.

A kliens utána a bérleti idő lejártáig használhatja az IP-címet, de a címütközés elkerülése érdekében

erősen ajánlott (SHOULD) *ARP Probe* segítségével ellenőriznie, hogy más nem használja-e.

- Ha más nem használja, a kliens *ARP Announcement*tel kihirdeti, hogy az övé.
- Ha más használja, a kliens DHCPDECLINE-nal jelzi a DHCP szervernek, és természetesen másik IP-címet kér.

Kedvezőtlen esetben további két üzenet fordulhat még elő:

DHCPNAK S→K

Ezzel az üzenettel jelzi a szerver, hogy a kliens kérése nem teljesíthető.

DHCPDECLINE K→S

A kliens jelzi a szervernek, hogy az adott IP-címet már más használja.

A bérleti idő lejártán belül a kliens hosszabbítást kérhet, ekkor nem kell az egész folyamatot lejátszania, elegendő:

DHCPREQUEST K→S

Az üzenet küldésekor a kliens még használja az érvényesen bérelt IP-címét és a kérést nem broadcast címre, hanem a szervernek küldi.

DHCPACK S→K

A szerver ekkor meghosszabbítja kliensnek az IP-cím bérletét és megadja, hogy milyen időtartamra kapja meg a kliens.

Természetesen ilyenkor a kliensnek nem kell további ellenőrzést végeznie, hiszen ezt a címet már eddig is használta. Így most a DHCPDECLINE üzenet sem jön szóba, de a szervertől most is kaphat DHCPNAK üzenetet.

A bérleti idő lejártán belül a kliens korábban is visszaadhatja (MAY) az IP-címet:

DHCPRELEASE K→S

Ezzel a kliens lemond a hátralevő bérleti időről, a szerver újra kioszthatja a címet.

Amennyiben egy kliensnek már van IP-címe (például statikusan be van állítva), akkor is kérhet más paramétereket:

DHCPINFORM K→S

Ezt a kliens a szervernek unicast üzenetként küldi.

Válaszul a szerver ugyanígy unicastként küldi egy DHCPACK üzenetben a további hálózati beállításokat. Ilyenkor a szerver nem ellenőrzi, hogy a kliens rendelkezik-e érvényes IP-cím bérlettel.

IP routing

Az IP csomagtovábbításának alapelvei

Az IP jellemzői:

- Csomagkapcsolt
- Összeköttetés-mentes (connectionless)
- „Best effort” – nincs garancia

„Hot potato”-elv

- Minél gyorsabban továbbítsuk
- Csak a következő csomópontot kell ismernünk (hop-by-hop)
- Előnyei:
    - Kis erőforrásigény
        - Egyszerű, ezért gyors működésű
        - Gyors, ezért nem kell (sokáig) tárolnunk a csomagokat (kis memóriaigény)
    - „Kevés” ismeret kell a hálózatról
    - Datagram szolgáltatásra tökéletesen alkalmas
      nem vállal garanciát, ezért nincs szükség nyugtázásra és egyéb hibakezelésre
      → gyorsasága megmarad

A továbbításhoz szükséges információk

- Kinek küldjük tovább? (routing)
    - Cél címe alapján → csomag tartalmazza
    - Saját ismeret alapján → útválasztó tábla (routing table) tartalmazza
- Hogyan küldjük tovább?
    - Mekkora egységekben? (tördelés)
        - Mekkora érkezik? → csomag határozza meg
        - Mekkora továbbítható? → a következő hálózat MTU-ja (Maximum Transmission Unit) határozza meg
    - Milyen QoS biztosításával?
        - „Best effort” – nincs garancia
        - Opcionális megoldás: ToS mező felhasználásával egyéb protokollok és mechanizmusok

Az útválasztó tábla (routing table)

Szükséges információk

- Hova tart?
- Merre küldjük tovább?
    - Ki a következő csomópont?
    - Melyik interfészen kell továbbítani?

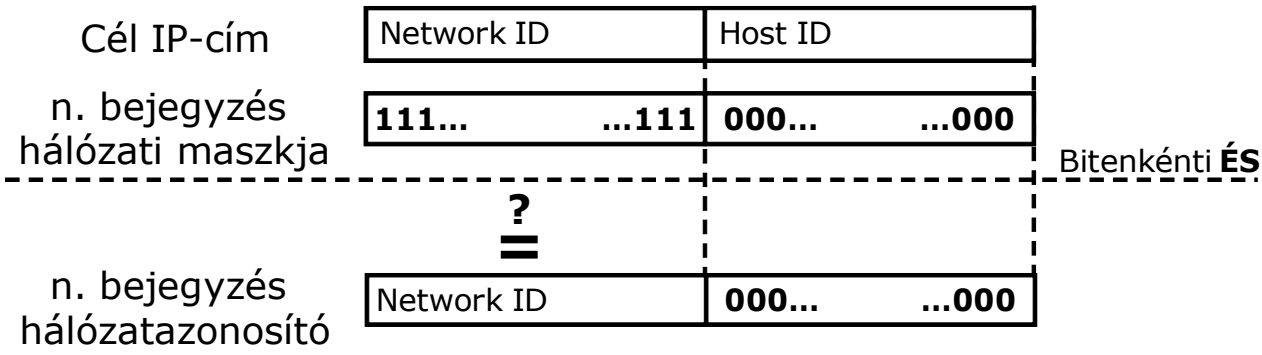
Az útválasztó tábla elvi felépítése:

Hova tart?		*Merre küldjük tovább?*		
Hálózat címe	Hálózati maszk	Következő csomópont	Interfész	Közvetlenül kapcsolódó
<IP-cím>	</N>	<IP-cím>	<azonosító>	<igen/nem>
<IP-cím>	</N>	<IP-cím>	<azonosító>	<igen/nem>
<IP-cím>	</N>	<IP-cím>	<azonosító>	<igen/nem>

Az útválasztó tábla gyakorlati felépítése (Linux alatt):

```
lencse@ns:~$ /sbin/route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
193.224.131.128 0.0.0.0         255.255.255.240 U     0      0        0 bond0
193.225.151.64  0.0.0.0         255.255.255.240 U     0      0        0 bond0
193.224.129.160 0.0.0.0         255.255.255.240 U     0      0        0 bond0
193.224.130.160 0.0.0.0         255.255.255.224 U     0      0        0 bond0
192.168.100.0   0.0.0.0         255.255.255.0   U     0      0        0 bond0.11
193.224.128.0   0.0.0.0         255.255.255.0   U     0      0        0 eth6
10.9.0.0        0.0.0.0         255.255.255.0   U     0      0        0 br0
0.0.0.0         193.224.128.9   0.0.0.0         UG    0      0        0 eth6
```

Hálózat kiválasztása



Az *illeszkedés* megállapítása:

A cél IP-cím akkor tartozik a táblázat n. sorában leírt hálózatba, ha a cél IP-cím és az n. sorban található hálózati maszk bitenkénti ÉS műveletének eredménye az n. sorban található hálózatazonosítóval megegyezik.

Keresés az útválasztó táblában

Ha a cél IP-cím több hálózatra is illeszkedik, akkor *legspecifikusabb illeszkedés* alapján kell továbbítani (ez az, ahol a leghosszabb a hálózati maszk).

(A problémának komoly irodalma van, akit mélyebben érdekel, Longest Prefix Match Algorithm néven érdemes keresni.)

Lineáris keresés esetén az útválasztó tábla összes bejegyzését végig kell nézni, ez akadályozza a „növekedést”.

Megoldás:

- általában bináris fával implementálják, így a bejegyzések számában lineáris keresés helyett a lépésszám csak a prefix hosszával arányos
- gerinchálózatban hardveres megoldás.

Alapértelmezett útvonal (default route)

Az *alapértelmezett útvonal* adja meg, hogy merre menjen a csomag, ha nem ismert a célhálózat. Az útválasztó táblában megadható egy megfelelő bejegyzéssel; ez a minden címet tartalmazó hálózat: 0.0.0.0/0

Az *alapértelmezett átjáró* (default gateway) a fenti bejegyzéshez tartozó következő csomópont. (Természetesen ez is egy útválasztót (router) jelent; az útválasztási RFC-kben viszont az átjáró kifejezést használják, ami egyébként máshol általában alkalmazás rétegbeli továbbítót jelent.)

Nem feltétlenül kell lennie alapértelmezett útvonalnak; ha a cél IP-cím egyik sorra sem illeszkedik, akkor a célhálózat ismeretlen, így a csomagot az útválasztó eldobja.

Végpontokon (host) gyakran csak két bejegyzés szerepel:

- Helyi hálózat (a helyi végpontok közvetlenül elérhetők)
- Alapértelmezett útvonal (minden más távoli hálózaton)

Metrika szerepe az útválasztásban

Metrika (mérték): számérték, amely a hálózati utak közötti preferenciát adja meg. A metrika alapja lehet például:

- Elérhetőség
- Terheltség
- Késleltetés

A metrika típusa lehet:

- Statikus (manuálisan megadott)
- Dinamikus (a link vagy a hálózat állapotától függően automatikusan változó)

Mindig a jobb értékkel rendelkező kapcsolaton küldjük ki a csomagot!

Teendők helyi és távoli hálózat esetén

Közvetlenül kapcsolódó (helyi) hálózat esetén: a címzettnek kell közvetlenül küldeni. (Ehhez a címzett adatkapcsolati rétegbeli címére van szükség.)

Nem közvetlenül kapcsolódó (távoli hálózat) esetén a megtalált „következő csomópont” útválasztónak kell küldeni, de a cél IP-címet tilos módosítani, csak az adatkapcsolati rétegben kell az útválasztónak címezni. (Ehhez szükség van az útválasztó adatkapcsolati rétegbeli címére.)

V*együk észre: mindig egy velünk szomszédos állomás (csomópont vagy végpont) adatkapcsolati rétegbeli címét kell kideríteni annak IP-címe alapján! Megoldás: ARP (Address Resolution Protocol).*

A router feladatai (összefoglalás)

- 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?
  visszajelzés küldéséhez: ICMP (Internet Control Message Protocol)

IPv6

Az IPv6 címek írásához

Prefix írásakor az utolsó olyan csoportot, ami nem csupa 0-t tartalmaz, mindig végig ki kell írni!

Példa: FEDB:ABCD:AB00::/40 a helyes, és helytelen: ~~FEDB:ABCD:AB::/40~~

Az RFC 3986 definiálja az IPv6 címek használatát URL-ekben. Példák:

http://[FEDC::C:BA98:0:3210]/index.html

http://[2001:738:2001:2012:221:70FF:FEC2:BA33]:8080/index.html

Az IPv6 címzési architektúrája

Az IPv6 címtartomány különféle célokra való felosztását *előtag*ok (prefix) segítségével lehet megadni.

A következő főbb kategóriákat definiálták (RFC 4291):

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

A következő kategória is létezett, de érvénytelenítették:

Site-Local unicast	1111111011	FEC0::/10

Az IPv6 multicast címzése

Az IPv4-gyel ellentétben *broadcast nincs*.

A multicast címek felépítése az alábbi:

bitek száma	8	4	4	112
mező neve	*prefix*	*flags*	*scope*	*group ID*

Az egyes mezők jelentése:

- prefix: a cím csoportcím voltát jelző előtag, értéke: FF
- flags: ebben a mezőben jelzőbiteket definiáltak: 0, R, P, T
- scope: a cím érvényességének a hatókörét fejezi ki (0-F)
- group ID: az egyes csoportok azonosítására használható bitek

A *flags* mezőben található jelzőbitek értelmezése:

- 0: fenntartott (reserved)
- R: A csoporthoz tartozó randevú pont (Rendezvous point) címe a csoportcímbe beágyazott-e (RFC 3956), 1: igen, 0: nem.
- P: A csoportcímet az azt létrehozó szervezet hálózatának előtagja (Prefix) alapján generálták-e (RFC 3306), 1: igen, 0: nem.
- T: A csoportcím dinamikus-e (Transient), 1: igen, 0: nem, azaz az IANA által kiosztott *well-known multicast address*. http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml

A csoportcímek érvényességi körét kifejező 4 bites *scope* mező lehetséges értékei:

- 0: Reserved – fenntartott
- 1: Interface-local – multicast loopback átvitelére használható
- 2: Link-Local – hatóköre a fizikai hálózat broadcast domain-je
- 4: Admin-Local – hatóköre a lehető legkisebb értelmes adminisztratív tartomány
- 5: Site-Local – hatóköre egy fizikai telephely
- 6-7: Unassigned – a hálózati adminisztrátorok definiálhatják
- 8: Organization-Local – egy szervezet összes telephelyére kiterjed
- 9-D: Unassigned – a hálózati adminisztrátorok definiálhatják
- E: Global – globális
- F: Reserved – fenntartott

Az IANA által kiosztott néhány általános célú csoportazonosító (group ID):

- FF02::1 – link local all nodes – az összes eszköz az adott fizikai hálózaton (broadcast domainben)
- FF02::2 – link local all routers – minden útválasztó a fenti körben
- FF05::2 – site local all routers – a telephely összes útválasztója

Vegyük észre, hogy az FF02::1 jelentése nem más, mint IPv4 esetén egy adott hálózatra vonatkozó broadcast cím!

Megjegyzés: IPv4-ben is van ilyen csoportcím: 224.0.0.1

Vannak minden scope esetén érvényes csoportazonosítók is, például:

- FF0x::C: SSDP – Az SSDP (Simple Service Discovery Protocol) által használt IPv6 csoportcím. (IPv4 alatt a 239.255.255.250 csoportcímre „szemetel” az SSDP.)

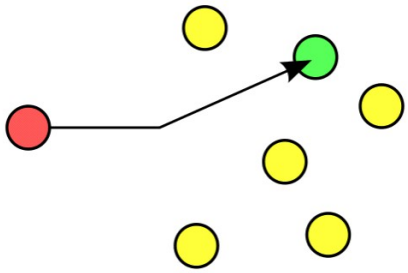
Az Anycast címzés

Ez NEM egy újabb IPv6 címtípus, mert egyrészt az IPv4-nél is alkalmazzák (tehát nem IPv6 specifikus), másrészt unicast címeket használ (tehát nem újabb típus).

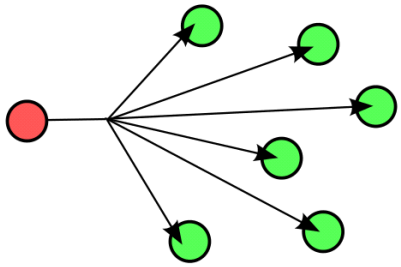
Tekintsük át a címzési módszereket!

- **Unicast**: A csomag pontosan egy általunk kiválasztott állomásnak szól.
- **Broadcast**: A csomag az adott hálózaton az összes állomásnak szól.
- **Multicast**: A csomag egy csoport összes tagjának szól.
- **Anycast**: A csomag egy csoport tagjai közül egynek szól, de nem a feladó, hanem a hálózat dönti el, hogy melyik legyen az. Tipikus választás, hogy legyen a küldőhöz legközelebb eső csoporttag.

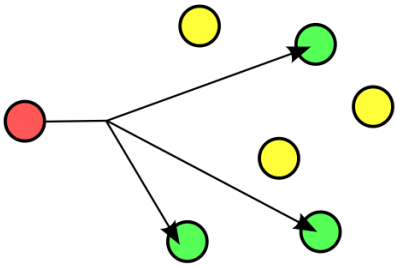
Unicast:



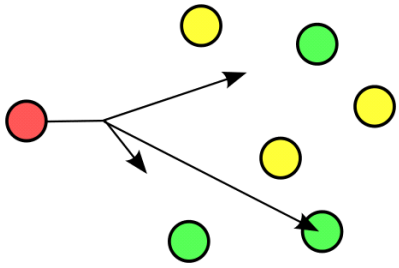
Broadcast:



Multicast:



Anycast:



Az Anycast címzés megvalósítása:

- Ugyanazt azt az IP-címtartományt (közönséges unicast címek!) több helyen is alkalmazzák és hirdetik az Interneten (BGP-vel).
- Az IP datagramok a legközelebbi célhoz fognak megérkezni.

Az Anycast címzés használata:

- DNS kiszolgálók esetén: ugyanolyan nevű (IP-című) legfelső szintű névkiszolgáló a világ több pontján is létezik. (A 13-ból 8 ilyen.)
  A kliensek a legközelebbit érik el.
- Az *IPv4-ről IPv6-ra való átállás* (IPv6 transition) során használt 6to4 megoldás is ezt használja a legközelebbi 6to4 átjáró elérésére.
- *Tartalomszolgáltató hálózat*ok (CDN: Content Delivery Network) is alkalmazzák abból a célból, hogy a kliens a legközelebbi szerverről tudja letölteni a médiatartalmat.

Globális Unicast címek

Bár jelenleg az IANA még csak a 2000::/3 tartományból delegált a RIR-eknek, ez elvileg semmiben sem különbözik a többi globális unicast IPv6 címtartománytól!

A jelenleg érvényes szabvány az RFC 3587: „*IPv6 Global Unicast Address Format*”.

Általános felépítésük :

```
|         n bits         |   m bits   |       128-n-m bits          |
+------------------------+------------+-----------------------------+
| global routing prefix | subnet ID |       interface ID           |
+------------------------+------------+-----------------------------+
```

- global routing prefix
    - hierarchikus felépítésű
    - az aggregációról a RIR-ek és az ISP-k gondoskodnak
    - a site-ok az ISP-ktől kapják
- subnet ID
    - hierarchikus felépítésű
    - az aggregációról a site-ok adminisztrátorai gondoskodnak
- Interface ID
    - a hálózati interfészt azonosítja

Az IPv6 címzési architektúráját definiáló RFC 4291 szerint a 0000/3 tartomány kivételével az Interface ID mérete 64 bit.

Így a felépítésük:

```
|         n bits         | 64-n bits |       64 bits               |
+------------------------+-----------+-----------------------------+
| global routing prefix | subnet ID |       interface ID           |
+------------------------+-----------+-----------------------------+
```

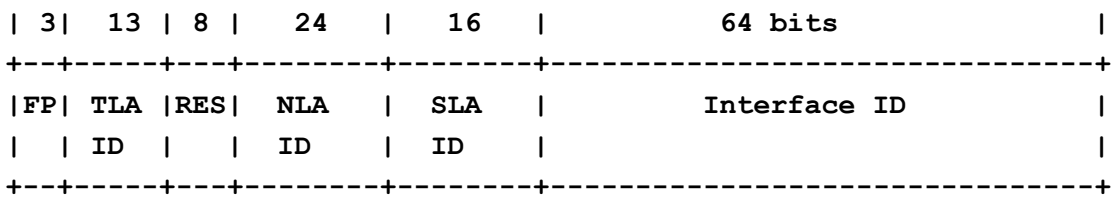
Ha a jelenleg használt 2000::/3 tartományban az egyes szervezetek a már *elavult RFC 3177* által ajánlott /48-as tartományokat kaptak, akkor a címeik felépítése a következő:

```
| 3 |     45 bits        |  16 bits  |       64 bits               |
+---+--------------------+-----------+-----------------------------+
|001|global routing pre.| subnet ID |       interface ID           |
+---+--------------------+-----------+-----------------------------+
```

Például a BME tartománya is ilyen méretű, a prefixe: 2001:738:2001::/48

A címek szerkezetét korábban RFC 2374 írta le „*An IPv6 Aggregatable Global Unicast Address Format*” címmel. Erről annyit kell tudni, hogy *elavult* (obsolete), NEM HASZNÁLJUK!

Az alábbi struktúrával támogatták volna az aggregálhatóságot:



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

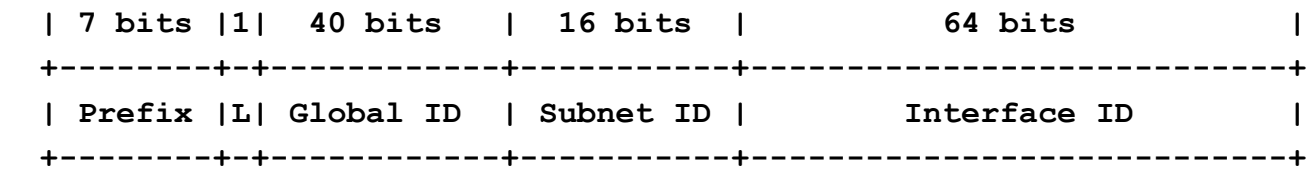
A fenti kötött formát tehát már nem használjuk!

Speciális Unicast címek

Unique Local IPv6 Unicast Adresses (RFC 4193)

Segítségükkel érvényes globális unicast IPv6 prefix nélkül is lehet IPv6-ot használni. Szerepükben hasonlítanak az RFC 1918 szerinti privát IP-címekhez és az érvénytelenített site-local unicast címekhez (de azok hibáit kiküszöbölve).

Felépítésük:



- Prefix: FC00::/7
- L: kötelezően 0 értékű
- Global ID: véletlenszerűen választott, igen jó eséllyel globálisan egyedi
- Subnet ID: a használó osztja ki a hálózatainak
- Interface ID: a hálózati interfész 64 bites azonosítója, lásd később

Az *IPv4-ről IPv6-ra való átállás* (IPv6 transition) támogatására szánták az alábbi IPv6 címtípusokat azzal a céllal, hogy IPv6 hálózatokban IPv4 címeket reprezentáljanak:

IPv4-Compatible IPv6 Addresses (érvénytelenített!)

IPv6 hálózatokban az *x.y.z.w* IPv4 címet reprezentálta volna a következő alakban: *::x.y.z.w* (az elején 96 db 0 értékű bit volt).

IPv4-Mapped IPv6 Addresses (ez használható!)

IPv6 hálózatokban az *x.y.z.w* IPv4 címet reprezentálja a következő alakban: *::FFFF:x.y.z.w* (az elején 80 db 0 értékű bit van).

Címallokáció megfontolásai

Az elavult (obsolete) RFC 3177 szerint a site-ok számára az alapértelmezett allokációs egység mérete a /48 volt.

Az új irányelvet az RFC 6177 „*IPv6 Address Assignment to End Sites*” fogalmazza meg.

- Kis felhasználóknál nem szükséges a pazarló /48 méret.
- A /64 viszont nem elég, mert idővel több fizikai hálózatra lehet szükség.
- Szempont a takarékosság is, de az is, hogy később sem legyen szükség NAT-ra (a site kapjon elegendő címet).
- A reverse DNS feloldás szempontjából megfontolandó a 4 bites határra való illeszkedés.
- Megemlíti a egyes RIR-ek gyakorlatát (/56-os méret), de alapelv, hogy nincs új alapértelmezett méret.
- Bár a /128 bizonyos esetekben elfogadott, site esetén semmiképpen sem javasolt.

Az IPv6 címtér felosztása

IPv6 Prefix	Allocation	Reference	Note
0000::/8	Reserved by IETF	[RFC4291]	[1][5][6]*
0100::/8	Reserved by IETF	[RFC4291]	
0200::/7	Reserved by IETF	[RFC4048]	[2] (spec. célú volt)
0400::/6	Reserved by IETF	[RFC4291]	
0800::/5	Reserved by IETF	[RFC4291]	
1000::/4	Reserved by IETF	[RFC4291]	
2000::/3	Global Unicast	[RFC4291]	[3] (IANA ebből oszt)
4000::/3	Reserved by IETF	[RFC4291]	
6000::/3	Reserved by IETF	[RFC4291]	
8000::/3	Reserved by IETF	[RFC4291]	
A000::/3	Reserved by IETF	[RFC4291]	
C000::/3	Reserved by IETF	[RFC4291]	
E000::/4	Reserved by IETF	[RFC4291]	
F000::/5	Reserved by IETF	[RFC4291]	
F800::/6	Reserved by IETF	[RFC4291]	
FC00::/7	Unique Local Unicast	[RFC4193]	
FE00::/9	Reserved by IETF	[RFC4291]	
FE80::/10	Link Local Unicast	[RFC4291]	
FEC0::/10	Reserved by IETF	[RFC3879]	[4] (Site local volt)
FF00::/8	Multicast	[RFC4291]	

* Számos speciális célút tartalmaz: ::/128, ::1/128, ::/96, ::ffff:/96, 64:ff9b::/96 (Well-Known Prefix, IPv4-Embedded IPv6)

Forrás: http://www.iana.org/assignments/ipv6-address-space/

A Neighbor Discovery Protocol (NDP)

Az NDP a TCP/IP referencia modell internet rétegében működik. Működéséhez ICMPv6 protokoll üzeneteket használ. Aktuális definíciója: RFC 4861.

Egy host működéséhez szükséges azonosítókat, paramétereket képes beszerezni, többek között képes:

- Állapotmentes automatikus címkonfigurációra (SLAAC)
- Routerek címének megállapítására
- DNS szerverek címének megállapítására
- Annak ellenőrzésére, hogy egy IPv6 címet már használnak-e (DAD)
- Címfeloldásra (IPv6 címből MAC cím)
- Az adott linken érvényes prefixek és MTU kiderítésére

Állapotmentes automatikus címkonfiguráció

A módszer az IPv6-cím utolsó 64 bitjében található Interface ID-t a hálózati interfész MAC-címéből állítja elő az ún. módosított EUI-64 algoritmussal.

Az állapotmentes automatikus címkonfiguráció (SLAAC: Stateless Address Autoconfiguration) lépései:

- Link-lokális cím generálása (FE80::/64 + módosított EUI-64 azonosító)
- Link-lokális cím ellenőrzése (ICMPv6 Neighbor Solicitation)
  Hasonlóan, mint IPv4-nél az ARP Probe üzenettel; ha nincs válasz, akkor OK. A használt Neighbor Solicitation ICMPv6 üzenetben a küldő IPv6 címe érvénytelen (::/128).
- Hálózati prefix kérése (ICMPv6 Router Solicitation)
- Hálózati prefix információ vétele (ICMPv6 Router Advertisement)
- Global Unicast cím előállítása (a kapott prefix + módosított EUI-64 azonosító)
- Global Unicast cím ellenőrzése (ICMPv6 Neighbor Solicitation)

Az SLAAC biztonsági kockázattal jár: hamis Router Advertisement üzenettel a kliens megtéveszthető:

- SLAAC Attack http://resources.infosecinstitute.com/slaac-attack/
- RFC 6104: „*Rogue IPv6 Router Advertisement Problem Statement*”

Demonstráció

Előre elkészített capture fájl alapján tanórán Wireshark segítségével vizsgáljuk az SLAAC folyamatát, otthoni elemzésre a capture fájl elérhető a tárgy anyagai között.

- A nem kívánatos egyéb forgalom kiszűrése érdekében a capture fájl készítése az **`ip6`** capture filterrel történt.
- A capture fájlba így is bekerültek a DNS szerver hiányában küldött *multicast DNS* üzenetek,ezeket az **`ipv6.nxt!=17`** display filterrel tudjuk megjelenítéskor kihagyni. (A 17 az UDP protokollt azonosítja.)

Figyeljük meg:

- Az üzenetek sorrendjét és ICMPv6 típusát!
- Az ICMPv6 üzenetekben milyen IPv6 és MAC címek vannak?
- Ellenőrizzük a módosított EUI-64 generálását!
- Végül milyen prefixet kapott a kliens?

A módosított EUI-64 algoritmus

Célja a hálózati interfész 48 bites MAC címéből 64 bites azonosító létrehozása.

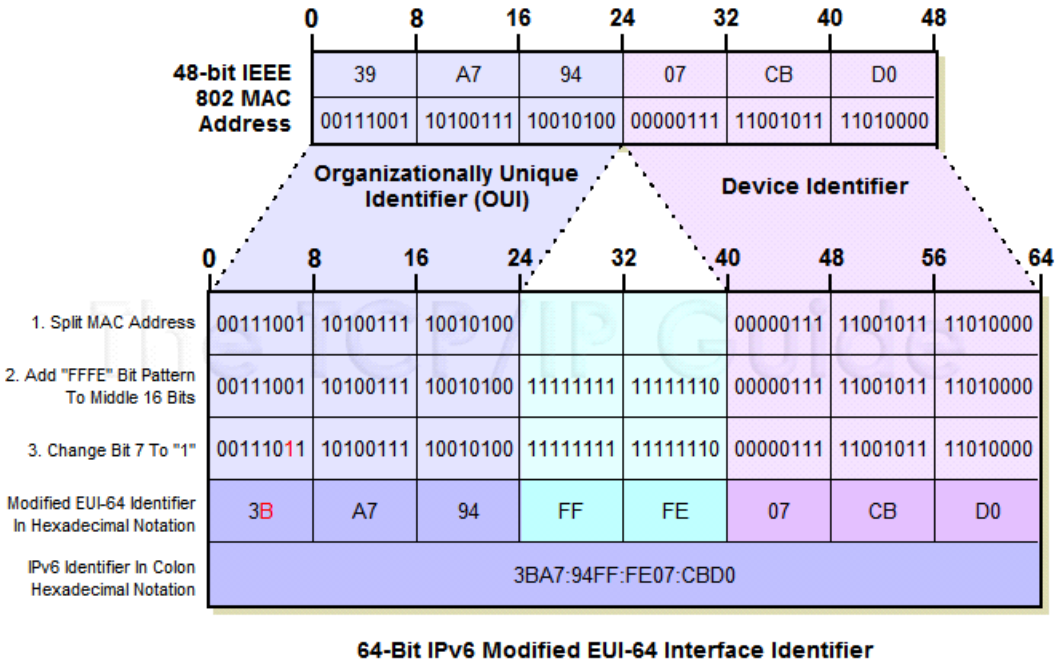
Az algoritmus lépései:

- A 48 bites címet középen kettévágva a két fél közé beszúrjuk az FFFE 16 bites értéket.
- A MAC cím első bájtjának második legkisebb helyiértékű bitjét 1-re állítjuk. (Ez a MAC cím OUI részének U/L bitje, tehát azt jelezzük vele, hogy nem univerzálisan egyedi, hanem lokálisan adminisztrált címről van szó.)

Az algoritmus működését egy példával illusztráljuk:

- Az eredeti MAC cím: 00:21:70:C2:BA:33
- Az első lépés eredménye: 00:21:70:FF:FE:C2:BA:33
- A második lépés eredménye: 02:21:70:FF:FE:C2:BA:33
- Az IPv6-nál használt alakban: 221:70FF:FEC2:BA33

Az alábbiakban bemutatunk egy ábrát, ami az algoritmus működését bit szinten illusztrálja. Forrás: http://www.tcpipguide.com/free/t_IPv6InterfaceIdentifiersandPhysicalAddressMapping-2.htm



64-Bit IPv6 Modified EUI-64 Interface Identifier

Internet Control Message Protocol version 6 (ICMPv6)

Az ICMP IPv6-os implementációja . Üzenetei az IPv6 felett utaznak (next header=0x3A), de minden IPv6 implementáció kötelező része! Aktuális dokumentáció: RFC 4443.

Üzenetei két típusba sorolhatók: hibaüzenetek és információs üzenetek.

Az ICMPv6 üzenetek formátuma egyedi. Ami közös bennük, az az ICMPv6 fejrész első 32 bitjén található 3 adatmező:

- **Type** (8 bit)

  0-127: hibaüzenetek, 128-255: információs üzenetek

- **Code** (8 bit)
- **Checksum** (16 bit)

A további rész kiosztása függ az üzenet típusától.

Az alábbiakban felsoroljuk a fontosabb ICMPv6 üzeneteket.

1 – Destination Unreachable (cél nem elérhető)

Mint az azonos nevű ICMP üzenet.

2 – Packet Too Big (a csomag mérete túl nagy)

Az IPv6 útközben nem tördel. Ha egy csomag nem fér bele az MTU-ba, akkor a router eldobja, és visszajelzést küld.

3 – Time Exceeded (időtúllépés)

Mint az azonos nevű ICMP üzenet.

4 – Parameter Problem (paraméter értelmezési hiba)

A hiba okát a Code mezőben jelzi: 0: Hibás IPv6 fejrész mező,
1: Ismeretlen Next Header típus, 2: Ismeretlen IPv6 opció.

128 – Echo Request (visszhang kérés)

Mint az azonos nevű ICMP üzenet.

129 – Echo Reply (visszhang válasz)

Mint az azonos nevű ICMP üzenet.

133 – Router Solicitation (router info kérése)

Az NDP része.

134 – Router Advertisement (router info adása)

Az NDP része, lehet kérésre válasz, de a routerek kérés nélkül is hirdetik.

A kéretlen hirdetés szándékosan nem pontosan periodikus.

135 – Neighbor Solicitation (MAC-cím kérése)

Az NDP része, az ARP megfelelője (pl. ARP Request és Probe funkciók).

136 – Neighbor Advertisement (MAC-cím hirdetése)

Lehet kérésre válasz, ekkor Solicited Neighbor Advertisement.

Kérés nélkül is küldhető (pl. IP-cím változásakor), ez az Unsolicited N.A.

IPv6 Transition

Az IPv4 → IPv6 áttérés időzítése

Az IPv4-ről IPv6-ra való átállás időbeli lefolyásának két (csak elvi) szélsőséges esete:

- Áttérés „óraütésre”:
    - Az Internet történelmében egyszer sikerült: Az ARPANET-en 1983. 01. 01-én volt az áttérés NCP-ről (Network Control Program) TCP/IP-re (RFC 801).
    - Ma ez lehetetlen feladat. Több milliárd csomópont van, lehetetlen őket egyszerre „átkapcsolni”. Sőt jelenleg rengeteg hardver és szoftver eleve alkalmatlan IPv6-ra.
- Időben elhúzódó átmenet:
    - A két protokoll tartósan egymás mellett fog élni, a fentieken túl azért is, mert:
        - Bizonyos hardver és szoftver szállítók nem fogják megoldani az IPv6 kompatibilitást.
        - A régi eszközökhöz a felhasználók ragaszkodnak.
    - Meg kell oldani az IPv4 és az IPv6 alapú rendszerek együttműködését!

Az IPv4 és IPv6 alapú rendszerek együttműködésének fontosabb esetei

A legtöbb alkalmazásunk kliens-szerver konfigurációban működik. Egyik szempont, hogy mely protokollokra képes a kliens és a szerver. A másik szempont, hogy a hálózat mely protokollokra képes a klienstől a szerverig terjedő úton.

Az egyszerű eset: dual stack használata

Ha a kliens és a szerver közül bármelyik is képes mindkét protokoll használatára (dual stack), akkor a „közös nyelv” használatával a kommunikáció megoldott. A probléma az, hogy már nincs elég IPv4 cím! (Egy kényszermegoldás a Dual-Stack Lite.)

IPv6 kliens és IPv4 szerver

Csak IPv6-ra képes a kliens (már csak IPv6 cím jutott neki) és csak IPv4-re képes a szerver (régi, az IPv6-ot nem támogatja). Egy jó megoldás: DNS64 szolgáltatás + NAT64 átjáró használata.

IPv4 kliens és IPv6 szerver

Csak IPv4-re képes a kliens (régi hardver és/vagy szoftver) és csak IPv6-ra képes a szerver (ilyenek is vannak, és számuk nőni fog). Egy jó megoldás lehet majd: NAT46, de ez még nem kiforrott.

IPv6 képes kliens IPv4-only környezetben és IPv6 szerver

A kliens képes IPv6-ra, de az internetszolgáltató (ISP) csak IPv4-címet ad a kliensnek, a szerver pedig kizárólag IPv6-ra képes. Egy jó megoldás: 6to4 mechanizmus és 6to4 relay használata.

IPv6 kliens és IPv6 szerver DE útközben egy szakaszon csak IPv4 van

Tipikus eset, az új IPv6 „szigeteket” össze kell kötni. A megoldás az IPv6 datagramok szállítása IPv4 fölötti „alagútban” (6in4 tunnel).

IPv4 kliens és IPv4 szerver DE útközben egy szakaszon csak IPv6 van

Ma még nálunk nem igazán jellemző, de majd lehet. A megoldás az IPv4 datagramok szállítása IPv6 fölötti „alagútban” (4in6 tunnel).

Kitérő: hálózati címfordítás (NAT)

A TCP/IP protokollcsaládot eredetileg végponttól végpontig való kommunikációra tervezték. Az alkalmazások arra számítanak, hogy a címek és portszámok a hálózati átvitel során változatlanok.

Az IPv4-címek szűkössége miatt terjedt el az a megoldás, hogy *egy hálózatban, ahol privát IP-címeket használnak, de szükség van külső kommunikációra is*, *a problémát címfordítással oldják meg*.

Legyen a feladat először az, hogy a *privát IP-címmel rendelkező gépek elérhessenek külső, publikus IP-címmel rendelkező gépeket*. Ehhez rendelkezésünkre áll egy router, aminek van publikus IP-címe.

A megoldás alapötlete a következő:

- A privát IP-címmel rendelkező gépről a célcím alapján küldjük el a csomagot az Internet felé! (Ekkor a csomag ugyan odaérne, de a válasz nem találna vissza.)
- A kimenő router cserélje ki a forrás privát IP-címét a saját publikus IP-címére!
- Elküldése után a csomag megérkezik a címzetthez, és így már a válasz visszaér a routerhez.
- A router továbbítsa a választ a megfelelő privát IP-című gépnek! – De hogyan?

Ahhoz, hogy a router a választ az eredeti feladónak vissza tudja küldeni, nyilván kell tartania, és egy válaszcsomag érkezésekor tudnia kell, hogy ki volt a küldője annak a csomagnak, amire a válasz érkezett. Ennek érdekében a nyilvántartáshoz az IP-címen túl mást is felhasznál. TCP és UDP esetén ezek a forrás portszámok. De a routernek nem elegendő ezeket megjegyeznie, hiszen a forrás portszámok csak *gépenként egyediek*, a forrás IP-címet viszont a sajátjára cseréli. Tehát a megoldás:

- A router a kimenő csomagokban a forrás IP-cím cseréjekor a forrás portszámokat is kicseréli a *routeren egyedi* portszámokra: az IP-címek és a célportszám mellett ezekkel már egyértelműen azonosítani tudja a kapcsolatokat. Kapcsolatonként nyilvántartja, hogy mit mire cserélt ki!
- A bejövő csomagoknál az IP-címen kívül a portszámokat is vissza kell cserélnie. (Itt most az irány változása miatt a cél IP-cím és a célport az, amit átír.)

Megjegyzés: terminológia nem egységes, de eredetileg a NAT csak az IP-címek cseréjét jelentette, ezt hívják ma *basic NAT*-nak, vagy *one-to-one NAT*-nak. A fenti megoldás precíz neve a *NAPT* (Network Address and Port Translation). Nevezik *many-to-one NAT*-nak is.

A NAT célja és működése szempontjából a fent ismertetett megoldást *Source NAT*-nak (SNAT) hívjuk akkor, ha a router publikus IP-címe fix, és *Masquerade*-nek, ha DHCP-vel kapta az interfésze az IP-címet.

A másik irányú feladat az, hogy *privát IP-címmel rendelkező gépeket elérhetővé tegyünk az Internet felől*. Erre a megoldás a *Destination NAT* (DNAT), ahol a router az adott portjára érkező csomagokat egy meghatározott privát IP-című gépnek továbbítja úgy, hogy a célcímet kicseréli a csomagban. Például a 80-as portra érkező csomagokat a 10.1.1.2 webszerver, a 25-ösre érkezőket pedig a 10.1.1.3 SMTP szerver felé továbbítja.

ICMP esetén nincs portszám, de segíthet az, hogy egy hibaüzenetben benne van az azt kiváltó TCP vagy UDP adategység első 64 bitje a portszámokkal. Amennyiben nem hibaüzenetről van szó, akkor is van megoldás, az ICMP üzenet valamilyen azonosító jellegű mezőjét használják fel.

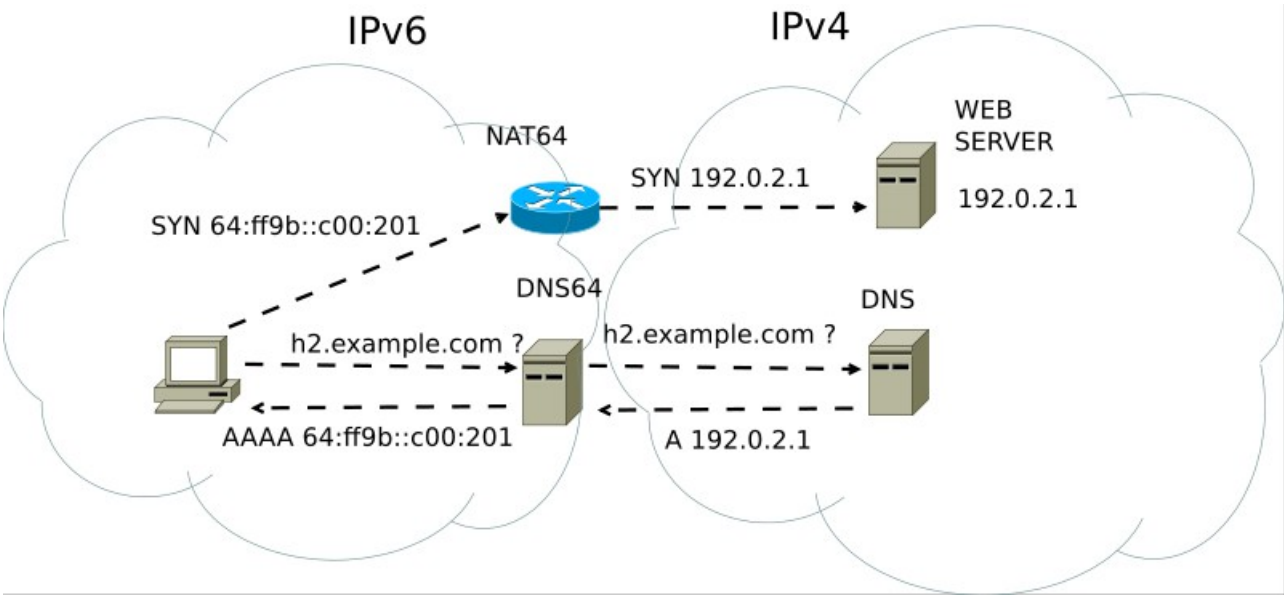
A NAT-ról bővebben az RFC 3022-ben olvashatunk, az IP, TCP, UDP, ICMP fejrészek mezőinek

módosításával a 4.1. rész, az ellenőrző összeg hatékony újraszámításával a 4.2. rész foglalkozik.

A DNS64 + NAT64 megoldás

Az IPv4 címek kifogyása miatt várhatóan az lesz az első tipikus megoldandó probléma, hogy az internetszolgáltatók csak IPv6 címet tudnak adni az új ügyfeleiknek. A csak IPv6 címmel rendelkező klienseknek el kell érniük a csak IPv4 címmel rendelkező szervereket is.

A probléma elvi megoldását az alábbi példán mutatjuk be:



Forrás: http://en.wikipedia.org/wiki/File:NAT64.svg

(a fenti ábrán mindkét helyen javítandó: 64:ff9b::c00:201 → 64:ff9b::**c000**:201)

A *DNS64 szerver* (RFC 6147) egy *caching-only* névkiszolgálóhoz hasonlóan *recursive query*kre válaszol. Ennek során, egy adott szimbolikus névhez

- ha van IPv6 cím, akkor továbbítja a válaszában a kliensnek
- ha nincs IPv6 cím, de IPv4 cím van, akkor az IPv4 cím alapján generál egy speciális IPv6 címet, és azt adja vissza a kliensnek.

A speciális IPv6 cím a példában használatos ún. *Well-Known Prefix* esetén a 64:FF9B::/96 prefix + az utolsó 32 biten a szimbolikus névhez tartozó IPv4 cím.

A megoldás működéséhez szükséges, hogy a kliensben névkiszolgálóként a DNS64 szerver legyen beállítva, és az útválasztási táblázatok szerint a Well-Known Prefix (mint hálózat) felé az út egy *NAT64 átjáró*n (RFC 6146) keresztül vezessen (anycast címzés használható).

Kövessük végig a példát!

- 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.
- TCP SYN szegmenst tartalmazó IPv6 csomagot küld a kapott címre.

- Az IPv6 csomag megérkezik a NAT64 átjáróhoz.
- Az átjáró az IPv6 csomag alapján egy IPv4 csomagot készít, benne
    - a célcím a speciális IPv6 célcím utolsó 32 bitje
    - a forráscím a NAT64 átjáró IPv4 címe.
- Az átjáró a csomagot elküldi a címzettnek.
- Az IPv4-only szerver megkapja a TCP SYN szegmenst tartalmazó IPv4 csomagot és a megszokott módon válaszol (SYN+ACK).
    - A kapott IPv4 csomagban a forráscím a NAT64 átjáró IPv4 címe volt.
    - A válasz címzettje is a NAT64 átjáró IPv4 interfésze lesz.
- A választ megkapja a NAT64 átjáró, és elkészíti a neki megfelelő IPv6 csomagot, melybe
    - forráscímként ugyanaz a speciális IPv6 cím kerül, amit a DNS64 szerver generált
    - célcímként az IPv6-only kliens IPv6 címe kerül
    - mindez a NAT által korábban is használt módon, kapcsolattábla alapján történik.
- Az IPv6 csomagot a NAT64 átjáró elküldi az IPv6-only kliensnek.
- Az IPv6-only kliens megkapja a csomagot.
- A kommunikáció a fentiek szerint tovább folytatódik…

Megjegyzés: A NAT64-től való megkülönböztetésül a NAT-ot NAT44-nek is hívják, amikor mind a lecserélt, mind az új IP-cím verziószáma 4-es.

A fentiekben bemutatott megoldás az RFC 6052 szerinti 64:FF9B::/96 előre lefoglalt, ún. *Well-Known Prefix*-et használja. A gyakorlatban számos hátránnyal járna, ha világszerte mindenki ezt a prefixet használná (RFC 6052 3.1. és 3.2.). A NAT64 átjáró megvalósításakor a gyakorlatban az egyes IPv6 hálózatokból szoktak erre a célra lefoglalni egy részt, ezt *hálózat-specifikus prefix*nek (Network-Specific Prefix) nevezik. Az ilyen IPv4 címeket tartalmazó IPv6 címeket úgy hívják, hogy *IPv4-Embedded IPv6 Addresses*.

Az IPv4-címeket beágyazó IPv6-címekről

Az *IPv4-címeket beágyazó IPv6-címek*et (IPv4-Embedded IPv6 Addresses) még két további névvel illetik a felhasználásuk céljától függően, bár szerkezetük és előállításuk azonos.

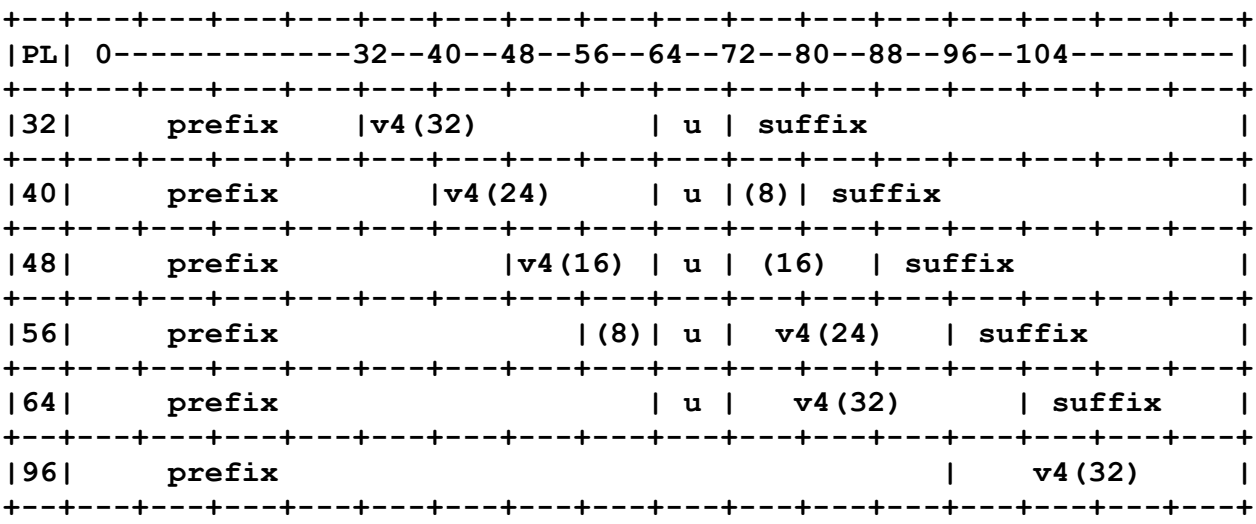
- Azokat az IPv6 címeket, amiket arra használunk, hogy IPv4 állomásokat képviseljenek IPv6 hálózatokban, úgy nevezzük, hogy *IPv4-Converted IPv6 Addresses*. (A fentiekben pontosan erről volt szó.)
- Az *IPv4-Translatable IPv6 Addresses* megnevezést pedig akkor használjuk, ha a cím egy IPv6 állomáshoz tartozik, és a fordítás célja, hogy a csak IPv4-re képes eszközök is el tudják érni az IPv6 állomást. (Ezzel az esettel most nem foglalkozunk.)

Az RFC 6052 definiálja, hogy hogyan kell az IPv4-címeket beágyazó IPv6-címeket képezni. Hálózat-specifikus prefixnél a hálózat adminisztrátora dönti el, hogy rendelkezésére álló címtartományból mekkora tartományt szeretne erre a célra szánni. Ennek megválasztása során tekintettel kell lennie az alábbiakra:

- A prefix mérete szigorúan csak 32, 40, 48, 56, 64 vagy 96 lehet.

- Az IPv6 címben a 64. – 71. sorszámú biteknek 0 értékűeknek kell lenniük.
- Az IPv6 cím 32 bitjét a választott méretű prefix után írjuk, de a fenti követelmény kielégítése érdekében a szigorúan 0 értékű bitek helyét „átugorjuk”.
- A cím végét szükség esetén ugyancsak 0 értékű bitekkel töltjük ki.

A címek lehetséges formátuma:



Ha például egy /48-as címtartományunk van, akkor a hálózat-specifikus prefix mérete /56, /64 vagy /96 lehet. A választás szempontjairól az RFC 6052 3.3. és 3.4. részében olvashatunk.

A 6in4 tunnel elve

Ezt a megoldást akkor használjuk, ha IPv6 szigetek csak IPv4 hálózaton keresztül tudnak egymással kommunikálni (IPv6-over-IPv4).

Ekkor az IPv6 csomagokat IPv4 csomagokba becsomagolva visszük át az IPv4 hálózaton. Az IPv4-ben a 41-es protokollazonosítót használjuk az IPv6 csomagok azonosítására. (Amint az IP fölött a Protocol mezőben a TCP protokollt a 6, az UDP-t a 17, az ICMP-t pedig az 1 érték azonosítja.) A be- és kicsomagolást az IPv6 szigetek határán levő átjárók végzik.

A 6to4 megoldás elve

Ezt a megoldást akkor használjuk, ha egy IPv6 képes kliens IPv4-only környezetben van, és IPv6-only szervert szeretne elérni.

A 6to4 megoldás (RFC 3056) egy „automatikus” tunnel, ami az IPv6 csomagokat IPv4 csomagokba csomagolja be. A 6in4-hez hasonlóan a 41-es protokollazonosítót használja. Megvalósítás szempontjából kétféle konfiguráció lehetséges:

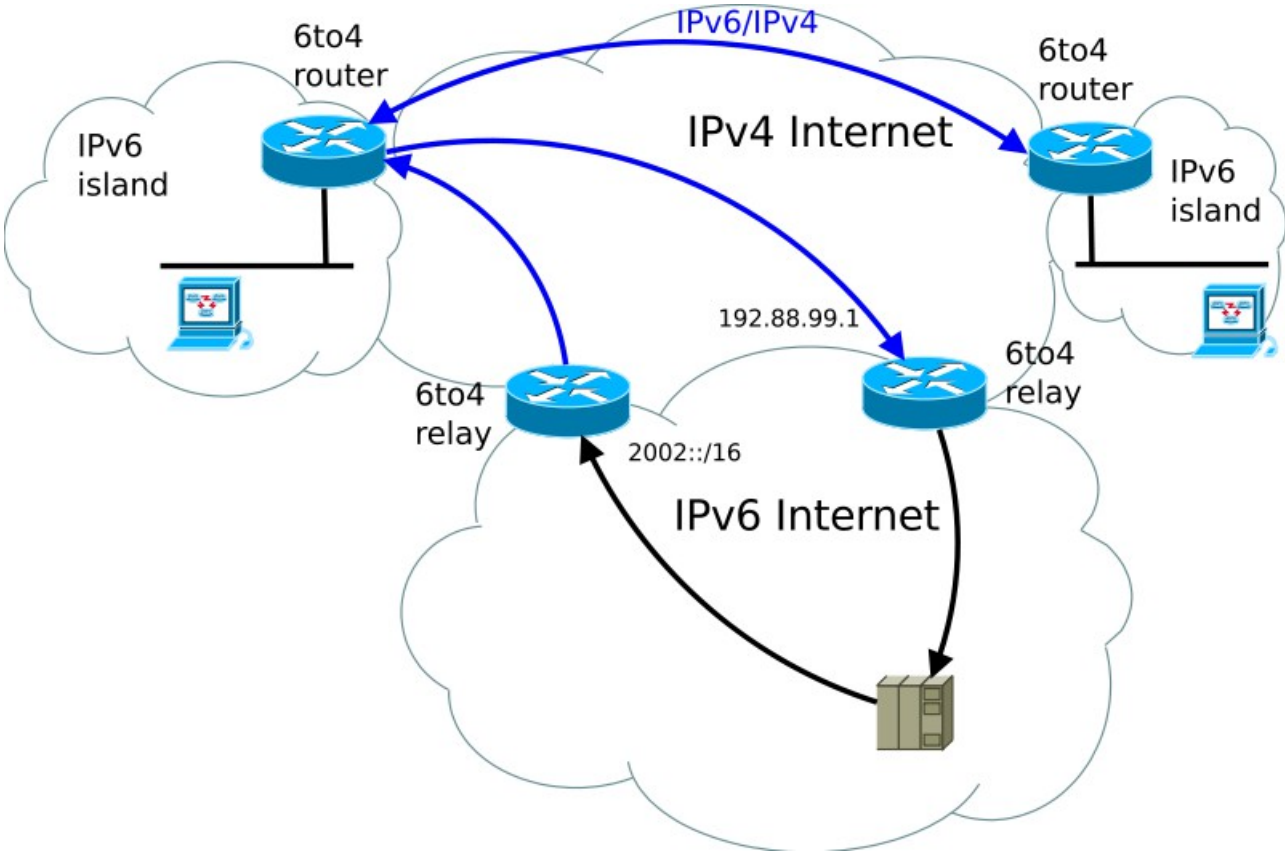
- lehet egyetlen host, amin az IPv6 kliens fut, az végzi a becsomagolást (*6to4 host*)
- lehet több IPv6-os gép egy IPv6 hálózaton, aminek a routere végzi a becsomagolást (*6to4 router*).

A megoldás működése dióhéjban a következő:

- 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 relay*nek.
- A 6to4 relay megkapja az IPv4-be ágyazott IPv6 csomagot, kicsomagolja, majd továbbítja a natív IPv6 hálózatba.
- A natív IPv6 hálózatban a csomag megérkezik a címzetthez, ami válaszol.
- A válasz visszaér a 6to4 relay-hez.
- Visszafele a 6to4 relay az IPv6 csomagokat IPv4-be csomagolva küldi a kliensnek, pontosabban a kliens IPv6 csomagját IPv4-be csomagoló eszköznek (host vagy router). Ehhez szükséges feltétel: a becsomagolónak legyen publikus IPv4-címe, különben nem találna vissza hozzá a válasz!
- A host vagy router kicsomagolja az IPv4 csomagból az IPv6 csomagot és eljuttatja a kliensnek.

Ha a becsomagolást végző hostnak vagy routernek nincs publikus IPv6 címe, akkor a 6to4 helyett *Teredo*t kell használni (RFC 4380). A 6to4 továbbfejlesztett változata a 6rd (RFC 5969).

Bár a 6to4 megoldást most arra a célra mutattuk be, hogy egy IPv4 környezetben működő IPv6 host elérhesse a natív IPv6 Internetet, annak sincsen akadálya, hogy két IPv6 szigetet kössünk össze a 6to4 segítségével az IPv4 Internet felhasználásával. A 6to4 lehetőségeit mutatja be az alábbi ábra.



Forrás: http://en.wikipedia.org/wiki/File:6to4.svg

A 6to4 megoldás címzése

A 6to4 címzéshez 2002::/16 prefixet foglalták le. A 6to4 IPv6 címek képzése az alábbi módon történik.

- Hálózatcím: 2002::/16 prefix + publikus IPv4 cím 32 bitje + 16 bit subnet ID
    - Host esetén a subnet ID egy generált véletlenszám.
    - 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 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.

Kiegészítés az általános IPv6 témához: IPv6 szabványos dokumentációs prefix

Annak érdekében, hogy a dokumentációkban szereplő példák ne okozzanak zavart (a fejekben) vagy működő rendszerekkel való ütközést, lefoglaltak egy globális unicast prefixet kifejezetten dokumentációs célra. Ez a 2001:DB8::/32 (RFC 3849). Ezt a prefixet soha senkinek sem fogják kiosztani és nem is routolják. De természetesen ettől még nem számít lokális unicast prefixnek!

(Így ha egy példában ezt a prefixet használják, és valaki a példát szó szerint begépeli, az nem fog kárt okozni, mert a prefixet a valós életben soha semmire nem használjuk. A prefixet helyi hálózatokban is kiszűrhetik; senki ne számítson rá, hogy működni fog, ha használni próbálná!)