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

v0.9, 2016. 03. 22.

Internet Protocol

Az osztálymentes címzés

Miért van rá szükség?

Problémák:

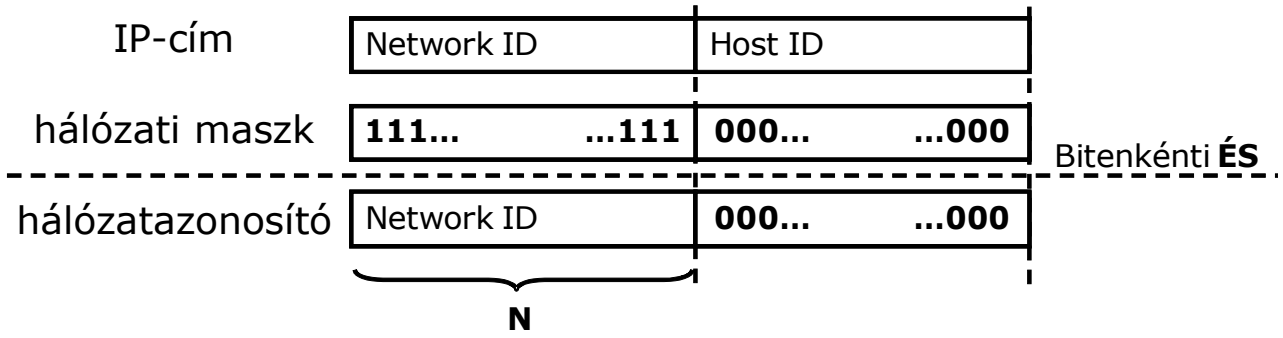
1. Az *osztály alapú címzés* (classful addressing) nem igazodik a fizikai hálózatstruktúrához
   - Egy nagy hálózat kisebb fizikai hálózatokból épül fel.
     → Az A, B osztályú hálózatokat kisebb hálózatokra KELL bontani.
2. A címzésben a két hierarchiaszint túl kevés
   - Az útválasztási (routing) 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
- **Dokumentációs célokra lefoglalt IP-címtartományok (RFC 5737)**
  - 192.0.2.0/24 (TEST-NET-1), 198.51.100.0/24 (...-2), 203.0.113.0/24 (...-3)

További speciális célra lefoglalt tartományok: RFC 6890

IPv4 link lokális címek dinamikus konfigurációja (RFC 3927)

Megjegyzés: az RFC 3927 címe: *Dynamic Configuration of IPv4 Link-Local Addresses* az IPv6 hasonló megoldását állapotmentes automatikus címkonfigurációnak (SLAAC) hívjuk.

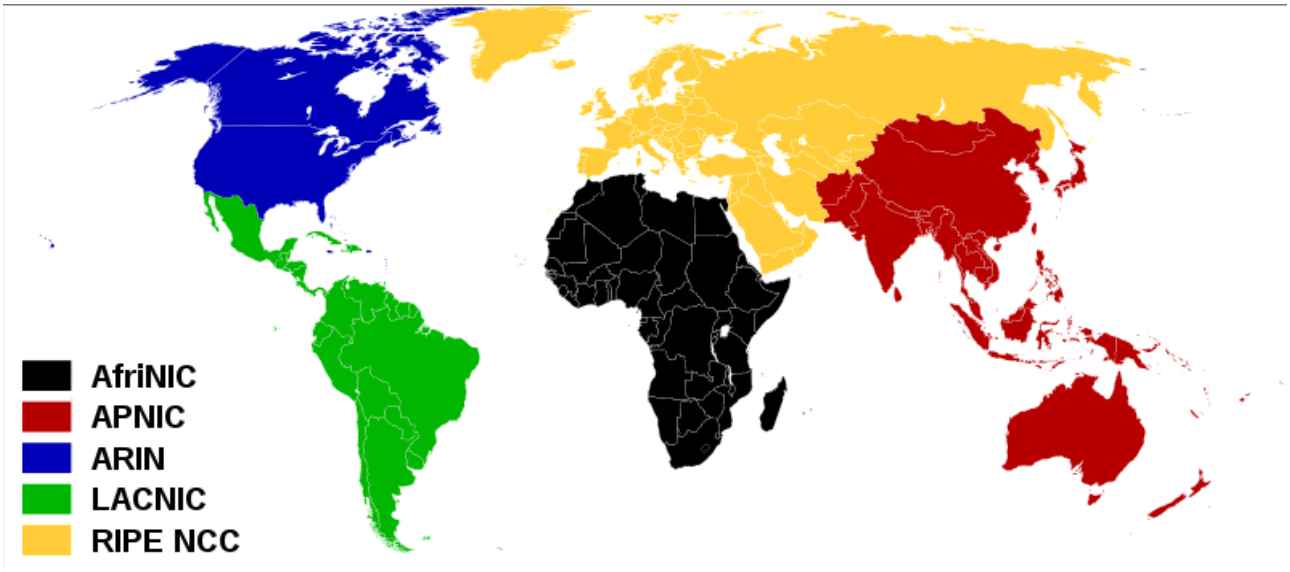
Ha nincs IP-cím statikusan beállítva, és az állomás nem kap választ egyetlen DHCP szervertől sem, akkor a 169.254.1.0 – 169.254.254. 255 tartományból véletlenszerűen választ magának egy IP-címet. Ellenőrzi (*ARP probe* segítségével), hogy már használatban van-e, és csak akkor használja, ha más nem használja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

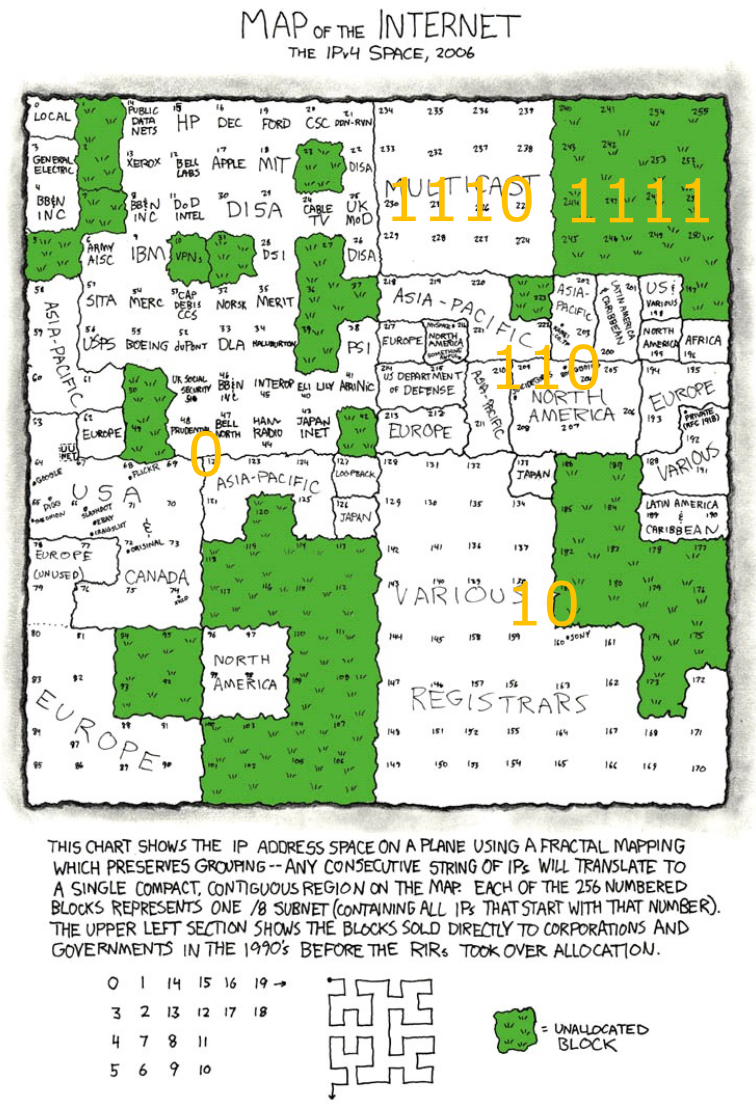


A kép forrása: http://en.wikipedia.org/wiki/File:Regional_Internet_Registries_world_map.svg

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 (eredetileg: RFC 1631, érvényes: RFC 3022) használata
  - a kezdetben kiosztott túlságosan nagy címtartományok visszaadása
- 2011. 01. 31-i igények alapján 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, a RIPE NCC-é pedig 2012. 09. 14-én kimerült

  → szigorúbb allokációs szabályok a legutolsó /8 blokkra
  → Az IPv6 bevezetése tovább Magyarországon sem halogatható!
  → Az IPv6, valamint a két rendszer együttélésének oktatása kulcsfontosságú!

IP csomagtovábbítás

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?

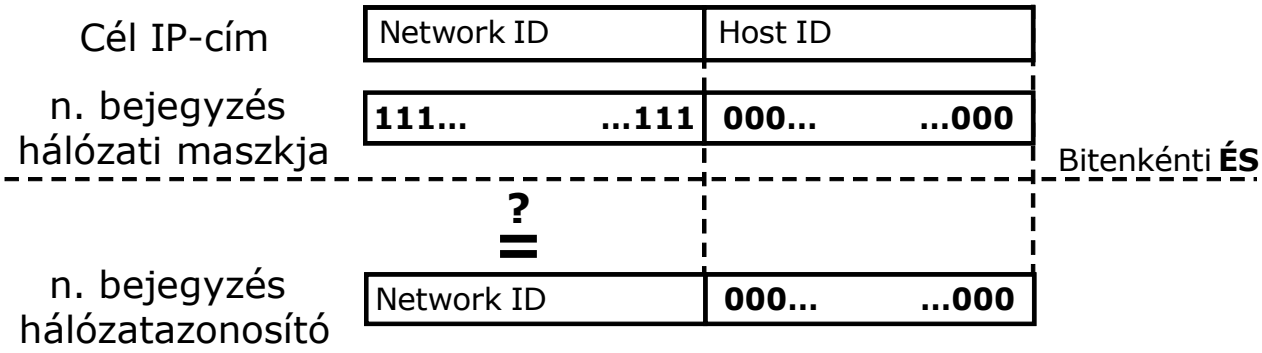
Az útválasztó tábla elvi felépítése:

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?
    - Közvetlen kézbesítés a címzettnek
    - Átadás a következő útválasztónak
    - Eldobás
- A TTL érték csökkentés után >0?
- Kell-e tördelni? Lehet-e tördelni?
- Kell-e visszajelzést küldeni?
  visszajelzés küldéséhez: ICMP (Internet Control Message Protocol)

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 terhelésnövekedés 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.

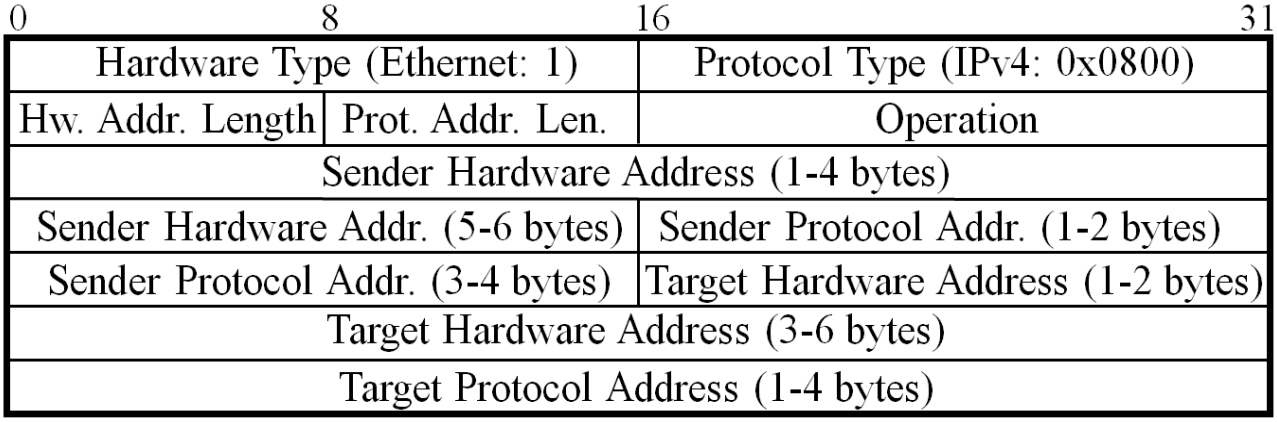
Megjegyzés: Az RFC 5681 leírja, hogy egy küszöbnél (ssthresh) mindenképpen váltani kell, illetve leír két további algoritmust is (fast retransmit, fast recovery). Ezekkel mélyebben nem foglalkozunk.

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-céméhez 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.

Szerkesztési megjegyzések:

- A TCP Congestion control témához a mérési utasításban található további anyag.
- Az IPv6 és az IPv6 áttérési technológiák témakörök kikerültek ebből a segédanyagból, mert azokkal külön anyagokban (két előadásvázlat + IPv6 könyv) foglalkozunk.