

## 20. fejezet – Hálózatok összekapcsolása

### Hálózatok összekapcsolása

Az eddigiekben is már számos alkalommal tárgyaltunk olyan modellekkről, ahol több hálózat kapcsolódott össze. Ezeket a hálózatokat (hallgatólagosan) mindig Ethernet hálózatokként képzeltük el, és így az átviteli hatóság megoldása természetes volt. A minket körül vevő világ és a minket körülvevő hálózatok azonban ennél összetettebb képet mutatnak. Nem véletlenül hivatkoztunk még az első előadások valamelyikén az Internetwork fogalomra, mely a különböző műszaki paraméterű, akár egymással közvetlenül nem összekapcsolható hálózatok közötti adatforgalom megvalósítását jelentette. Most eljött az idő, hogy ezzel a fogalomkörrel fogalkozzunk részletesebben.

A hálózatok számos paraméterükben térhetnek el egymástól. A fizikai rétegen és az adatkapcsolati rétegen elég csak az eltérő átviteli közegekre, jelszintekre, modulációs és kódolási eljárásokra, illetve az eltérő keretformátumokra gondolni. A hálózati réteg különbségei is sokfélék lehetnek:

- A hálózat jellege (összeköttetés alapú vagy összeköttetés nélküli)
- A címzés módja (tartalmaz-e hierarchiát vagy nem)
- Csomagméret (alsó illetve felső értékek, vagy csak fix méretű)
- Időzítések (a csomagok átviteli időkorlátainak eltérése)
- Sorrendiség (a csomagok sorrendje lehet szempont is, meg nem is)
- A szolgáltatás minősége (lehet fontos szempont is, meg nem is)
- Megbízhatóság (a csomagvesztés, csomagismétlés és hibajavítás, mint faktor)
- Biztonság (alkalmazható-e titkosítás, vagy nem)
- Költségek (a felhasználó fizethet például idő vagy adatmennyiség alapon)

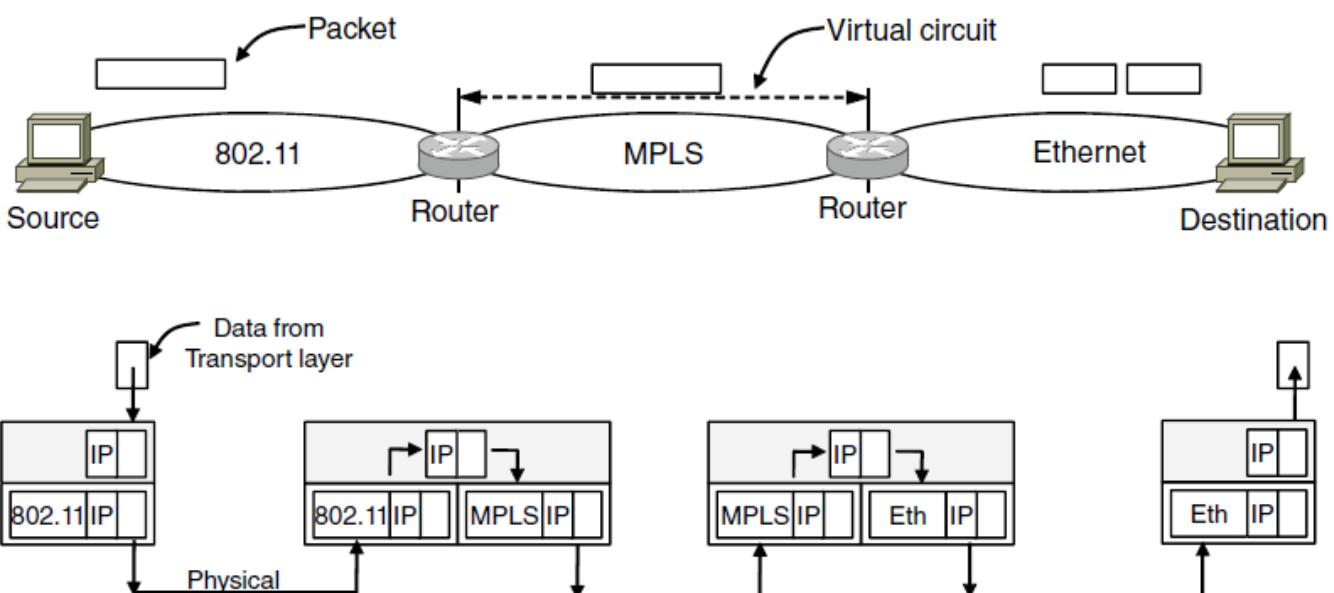
A forrás hoszt és a cél hoszt között a valóságban akár több, különböző paraméterű hálózat is lehet, melyen minden keresztül kell juttatni az adatcsomagokat. A forrás hoszt és a cél hoszt címének és elérhetőségének minden érintett hálózatban ugyanazt kell jelentenie, különben biztosan meghiúsul az információcsere.

A különböző hálózatok összekapcsolásának alapvetően két módja ismert. Az egyik megoldás a hardveres út, amikor az egyes hálózatokat összekapcsoló aktív eszközöknek ismerni kell az összekapcsolt hálózatok fent említett paramétereit, és az adatcsere folyamán az egyes hálózatok egyedi igényeit ki kell tudni szolgálni (pl. csomagdarabolás, illetve csomagegyesítés). A másik megoldás a szoftveres út, amikor a modellünkhez egy átvitási biztosító réteget – egy hálózatok feletti közös réteget – adunk hozzá.

A mai általánosan használt szoftveres megoldás (mely egyébként a hardveres megoldásnak is része) 1974-ben született meg Vinton Gray Cerf és Bob Kahn munkája nyomán. A szóban forgó protokollkészlet, az átviteli vezérlő protokoll és az internet protokoll azaz a TCP/IP (Transmission Control Protocol / Internet Protocol) eredetileg az ARPANET-hez készült, de sikeresen túlélte az ARPANET-et.

Különböző hálózatok összekapcsolásakor tehát először is egy olyan címzési módra van szükség, ami hálózati rétegen belül teszi azonosíthatóvá a hosztokat.

Például, amennyiben egy WLAN (IEEE802.11) és egy klasszikus Ethernet (IEEE802.3) hálózat között egy más típusú, például MPLS (MultiProtocol Label Switching / Többprotokolloos Címkekapsolás) hálózat található, akkor már látszik néhány nyilvánvaló különbség. Az IEEE802.11 összeköttetés nélküli szolgáltatást biztosít, az MPLS pedig virtuálisáramkör alapú összeköttetést. Ezen kívül az IEEE802.11 és az MPLS nagyobb méretű kereteket kezel, mint az Ethernet, így az MPLS-ből érkező csomagokat az Ethernet hálózatba jutva fel kell darabolni.



Az ábra alsó részén alulról felfelé a fizikai, az adatkapcsolati, és a hálózati réteg van jelölve. A forrás hoszt a szállítási rétegtől kap adatokat, olyan csomagot állít elő, amely az összes érintett hálózatban értelmezhető címet, az IP címet is tartalmazza a MAC cím mellett. Az első útválasztóig pusztán a MAC cím is célba juttatja a csomagot, itt azonban már csak az adat és az IP cím fog beágyazódni az MPLS fejléc után. Az MPLS fejlécében lévő adatok átjuttatják a csomagot a virtuális áramkörön a következő útválasztóig. (A csomagdarabolás és egyesítés az alsó ábrán nem látszik, csak a címzés módja.) A második útválasztó eltávolítja az MPLS fejlécet, és az IP cím alapján saját MAC táblából kikeresi a cél hoszthoz kapcsolódó portot.

A fenti példa is rávilágít arra, hogy a Switch (Layer 2 Switch) és az útválasztó (Router) működése közti különbség (többek közt) abból áll, hogy az útválasztó kiveszi a csomagot a keretből, és a csomagban lévő IP címet használja a cél hoszt meghatározásához. A Switch viszont a teljes keretet továbbítja a MAC cím alapján. A Switchnek nem kell ismernie a hálózati réteg protokollját, az útválasztónak viszont ismernie kell azt.

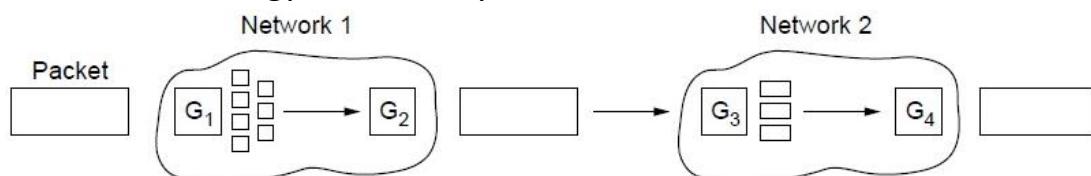
Az előző példa során is találkoztunk azzal a megoldandó problémával az átvitel során, hogy az egyes hálózatok csomagméreteinek maximuma eltérő. A megoldás a csomagok darabolása illetve egyesítése. Az eltérések az általunk tárgyalt hálózatok esetében is jelentős. Az adatmező az Ethernet (IEEE802.3) esetében 1500 bájt, WLAN (IEEE802.11) esetében 2272 bájt, az IP csomagok viszont akár 65515 bájt méretűek is lehetnek.

A csomagok átvitele darabolás nélkül is megoldható, ha ismerjük azt a legnagyobb átvihető adategységet (MTU – Maximum Transfer Unit) ami az átvitel során biztosan mindenkorrigálható. A megoldásnak előnyei és hátrányai is vannak. Ez esetben nem vesszük igénybe azokat a többlet erőforrásokat, amelyek a darabolás illetve az egyesítés elvégzéséhez elengedhetetlenek. A kisebb csomagok átvitele jellemzően több redundáns- illetve többlet információ átvitelével jár, mint a nagyobb csomagok átvitele.

Amennyiben (bármely okból is mégis) a darabolás, mint eljárás tűnik a célravezető megoldásnak, a csomagokat nyilván kisebb darabokra (Fragment) kell tördelni, majd pedig összerakni. A kérdés csak az, hogy hol történjen a darabok összerakása. Kétféle megközelítés, azaz megoldás is kínálkozik.

- Az átlátszó darabolás és összerakás

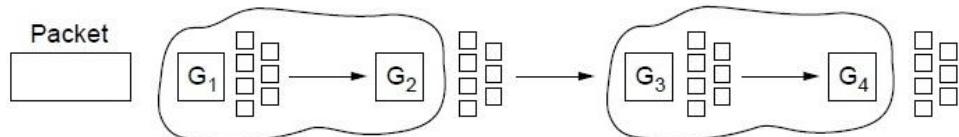
Az elnevezés arra utal, hogy a darabolás az egyes hálózatokban függetlenül történik. Az egyes hálózatok útválasztói a beérkezett csomagokat amennyiben szükséges, saját hálózatuk paramétereinek megfelelően (azaz egyedi méretre) darabolják. Azok az útválasztók, melyek megkapják az így keletkezett darabokat, azokat már egyesítve adják át egy másik hálózatnak. Így tulajdonképpen minden közbülső hálózat az eredeti csomagot kapja meg, és arról, hogy a csomagot a többi érintett hálózatban hogyan és mennyire darabolták szét nem is értesül.



A megoldásból következik, hogy minden darabnak ugyanahhoz az útválasztóhoz kell megérkeznie az egyesítéshez. Az egyesítéshez természetesen az összes darabnak egy nyilvántartás szerint meg kell érkeznie, és az utolsó darab megérkezéséig az összes többi darabot pufferelni kell az érintett útválasztóban. Több hálózaton áthaladva ez a megoldás jelentős erőforrásokat képes lekötni.

- A nem átlátszó darabolás és összerakás

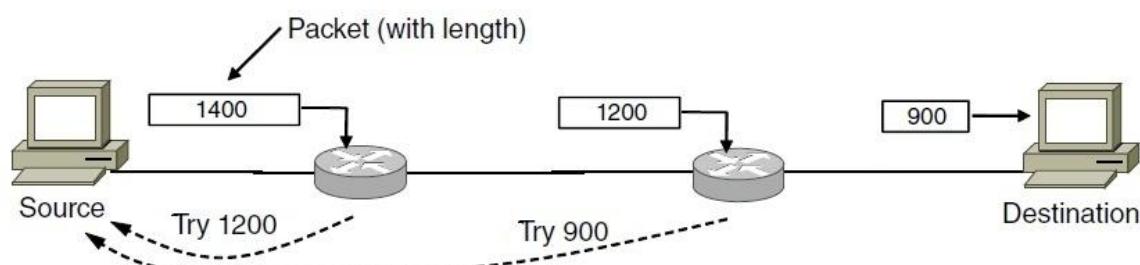
Ez az eljárás abban tér el az előző eljárástól, hogy itt a darabok összeállítása az egyes hálózatok elhagyásakor nem történik meg. A darabok összeállítása (és az összeállítást megelőző pufferelése is) csak a célállomáson, azaz a fogadó hoszon történik meg.



Az egyes darabok nyilvántartása, számozása ez esetben is elengedhetetlen feladat. Az IP protokoll is ezt az eljárást használja. A darabolás 8 bájtos határon következhet be. minden egyes darabhoz a következő adatok tartoznak

- csomagszám  
Ez azt azonosítja, hogy a darab melyik csomagnak a része.
- egy jelzőbájt  
Ami azt mutatja meg, hogy az adott darab az utolsó-e a sorban.
- Abszolút bájteltolás a csomagban  
A bájteltolás (Fragment Offset) értéke a darab első bájtjának az eredeti (nem darabolt) csomagbeli helyét jelezzi. A darabolási határ miatt az érték 8 bájtos egységen van számolva.

A csomagok átvitele darabolás nélküli és darabolásos eljárásokkal is történik a gyakorlatban, az igények és a lehetőségek szerint. Sok esetben azonban a darabolás adatvédelmi megfontolásokból, vagy műszaki korlátok miatt (például boot program esetében) nincs engedélyezve. Az MTU kézi beállításánál létezik dinamikusabb megoldás, az útvonal MTU felderítése (Path MTU Discovery). Ez esetben, az átvitelben részt vevő aktív eszközök képesek belső kommunikációval meghatározni az MTU aktuális értékét. A kommunikáció hibacsomagok visszaküldésével történik, mely jelzi az adni kívánt csomagot összeállító eszköz, a forrás számára, hogy az összeállított csomag túl nagy (nagyobb, mint az MTU). Ez után a forrás az aktuális MTU-nak megfelelő méretű csomagot állít elő. A felderítés jellemzően – az ábrán látható módon – több lépésből áll.

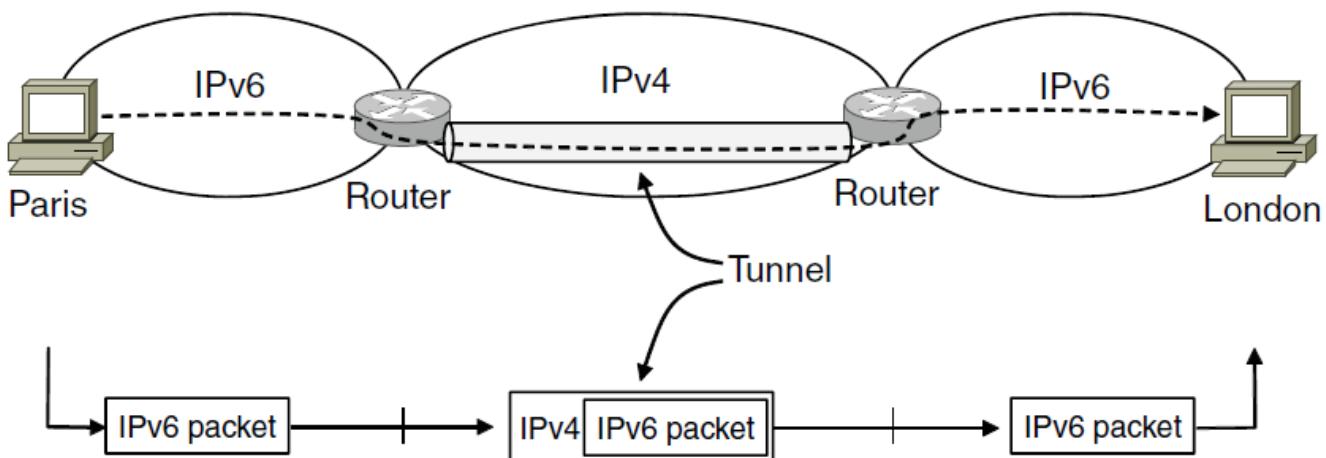


A hálózatok összekapcsolhatóságának alapfeltétele a közös hálózati réteg. A fenti példában is szereplő útválasztókat többprotokolos útválasztónak (Multiprotocol Router) nevezzük, mivel többféle protokollt képesek kezelní. A legáltalánosabban használt IP mellett számos más viszonylag gyakran használt protokollt érdemes megemlíteni.

- IPX (Internetwork Packet eXchange). A Novell által bevezetett NetWare protokollja
- SNA (Systems Network Architecture). Az IBM által használt hálózati protokoll
- AppleTalk. A Macintosh hálózatok által használt protokoll
- ATM (Asynchronous Transfer Mode). Telefonrendszerök használják pl. ISDN

### Alagút típusú átvitel

A hálózatok összekapcsolásának van egy speciális esete, amikor a forrás és cél hálózat egyforma, de az összekapcsolás egy tőlük különböző hálózaton keresztül valósul meg. Az ábra szerinti összekapcsolás egy párizsi illetve egy londoni IPv6-os hálózat összekapcsolását szemlélteti. A két hálózat között azonban IPv4-es hálózat található. A technológia lényege ez esetben az, hogy az IPv6-os csomagot a hálózat párizsi oldalán, a hálózatokat összekötő útválasztó az IPv4-es hálózatba olyan módon juttatja el, hogy beágyazza az ezt IPv4-es csomagba. A londoni oldalon, a hálózatokat összekötő útválasztó pedig kicsomagolja az IPv4-es csomagból az IPv6-os csomagot, és már csak azt továbbítja saját hálózatában.



A megoldás neve az alagút típusú átvitel (Tunneling). Az elnevezés megjegyzését segíti a példában szereplő városok kiválasztása is, hiszem a valóságban is olyan módon juthatunk el autóval (egyik hálózat) Párizsból a Csatornaalagúton (közbülső hálózat) át Londonba (másik hálózat), hogy magán az alagúton az autónkat egy vonat viszi át („beágyazva” a vonatban).

Az alagút típusú átvitel tehát olyan módon áll össze két hálózat között, hogy azok a köztük lévő hálózat(ok) hosztjait nem is érhetik el. Ez azonban sokkal inkább előny, mintsem hogy hátrány legyen. Virtuális magánhálózatok esetében (VPN) ez a tulajdonság az egyik alapvető szempont, illetve az ebből fakadó titkosítási lehetőségek. A megoldás gyakorlatilag elfedi a közbülső hálózato(ka)t, ezért hívják elfedő hálózatnak (Overlay Network).

## 21. fejezet – Az IPv4 protokoll 1

### Hálózati réteg az Interneten

Az Internet, ami mára már az életünk részévé vált, többek közt annak köszönheti sikerét, hogy tervezőinek sikerült megfelelő elvek mentén építkezniük. Ezek az elvek az IETF (Internet Engineering Task Force / Internet Működtetését Koordináló Testület) RFC1958 (Request For Comments) dokumentuma tartalmazza. Az anyag 1996 júniusában keletkezett, és természetesen egy korábbi munkákra és ötletekre épülő összefoglalás. (az 1958 a dokumentum sorszáma). <http://www.ietf.org/rfc/rfc1958.txt>

A számunkra legfontosabb 10 irányelv a következő:

1. A lényeg, hogy a gyakorlatban működjön.

A megoldásnak nem csak elméletileg, hanem gyakorlatilag is működnie kell, csak kipróbált és tesztelt prototípusok után szabad a szabványosítást elkezdeni.

2. Törekedjünk az egyszerűségre.

Itt célszerű visszanyúlni a XIV. századig, amikor William Ockhan angol filozófus (és ferencesrendi szerzetes) megfogalmazta a „lex parsimoniae” elvet (a tömörség vagy takarékosság elve). A téTEL kimondja, hogy egy jelenség magyarázatának minél kevesebb feltételezést kell magában foglalnia, kizárvva azokat, melyek nem változtatnak a magyarázó elmélet valósínűsíthető végkimenetelén.

3. Legyen egyértelmű választás.

Több lehetségesnek tűnő megoldás esetén egyértelmű priorizálást kell alkalmazni, ki kell választani az optimális, a jó megoldást. A klasszikus „Bigger is better” (A több jobb) ellenpéldája. Egy jó megoldás például a  $2+2=4$  esetében jobb, mintha az eredmény akár 3 vagy 5 is lehetne...

4. Használd ki a modularitást.

A moduláris rendszer, azaz a független rétegek egymásra épülése lehetővé teszi egy réteg megváltoztatását oly módon, hogy az a többi réteg működését ne befolyásolja.

5. Számíts heterogén környezetre.

A rugalmasság elve. Egy laboratóriumi környezethez képest a minket körülvevő valóság sokkal heterogénebb. Az Internetnek lehetőség szerint a változatosabb hardver környezetben is tudnia kell működni.

6. Kerüld a statikus opciókat és paramétereket.

Ha elkerülhetetlen néhány paraméter maximumának vagy minimumának meghatározása (pl. csomagméret), akkor is csak abban az esetben szabad statikus paramétereket rögzíteni, ha a kommunikációban részt vevő aktív elemek képtelenek maguk a paraméterek meghatározására.

7. Amit tervezel, az jó legyen, nem az abszolút tökéletes a cél.

Ez a „Good enough, is not good enough” (Az elég jó nem elég jó) elv ellenpéldája. Ez esetben arról van szó egy nem minden szempontból tökéletes, de működő megoldás többet ér egy nem működő megoldásnál. (Egy érthető példa: Egy égő ház első emeletéről biztos kényelmesebb lépcsőn lejönni, mint leugrani. De ha nincs lépcső, biztos, hogy senki se áll neki építeni egyet...)

8. Légy szigorú a küldésnél, és elnéző a vételnél.

Még ha a legszigorúbb szabályok szerint is küldi el egy hoszt a csomagjait, a fogadó félnek (bármely műszaki okból is) számítania kell arra, hogy ha a csomagok esetleg nem szabványos módon érkeznek meg, akkor is lehetőleg feldolgozhatóvá kell tenni azokat.

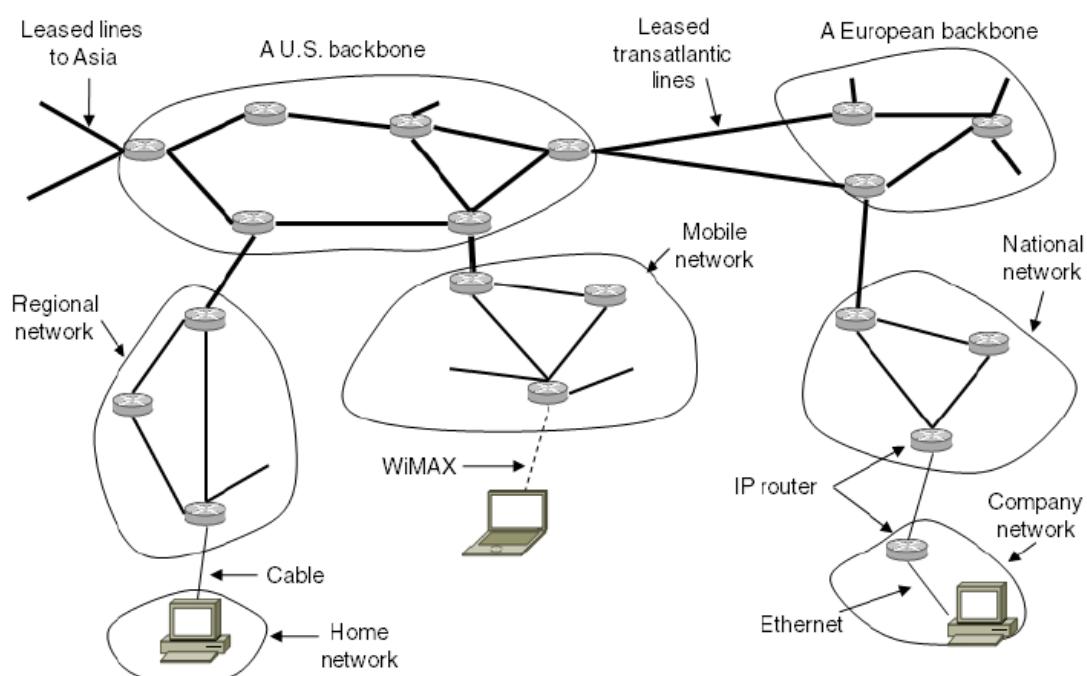
9. Gondolj a skálázhatóságra.

Egy nagy és decentralizált rendszer alapja nem lehet egy központi adatbázis. A terhelések és erőforrások hatékony elosztása, és a bővíthetőség kell, hogy a rendszer folyamatos és megbízható működésének az alapja legyen.

10. Tartsd egyensúlyban a költségeket és a teljesítőképességet.

A költséghatékonyúság és a megfelelő ár/teljesítmény viszony fontos szempont minden felhasználó számára. Ezek hiánya az ügyfelek elvesztéséhez vezető egyenes út...

Az Internet, mint világot átfogó hálózat, különböző hálózatok, autonóm rendszerek (AS - Autonomous Systems) összekapcsolásának az eredménye. Az Internet tehát kontinentális gerinchálózatokból áll, melyekhez az ISP-k csatlakoznak lefedve saját szolgáltatási területüket, több kisebb szolgáltató közreműködésével.



A hálózatok közti átjárhatóságot (tehát egy tetszőleges kiszolgáló elérését) az IP (Internet Protokoll), az internet protokoll teszi lehetővé. Az egyes kiszolgálók illetve hosztok elérése a címzési rendszernek és az erre épülő útválasztásnak köszönhető. Ennek segítségével tudunk bármely országban (egyéb okból le nem tiltott...) tetszőleges web oldalához csatlakozni, elektronikus levelet (Email) küldeni a világon bármely címre, illetve a hálózat többi szolgáltatását igénybe venni.

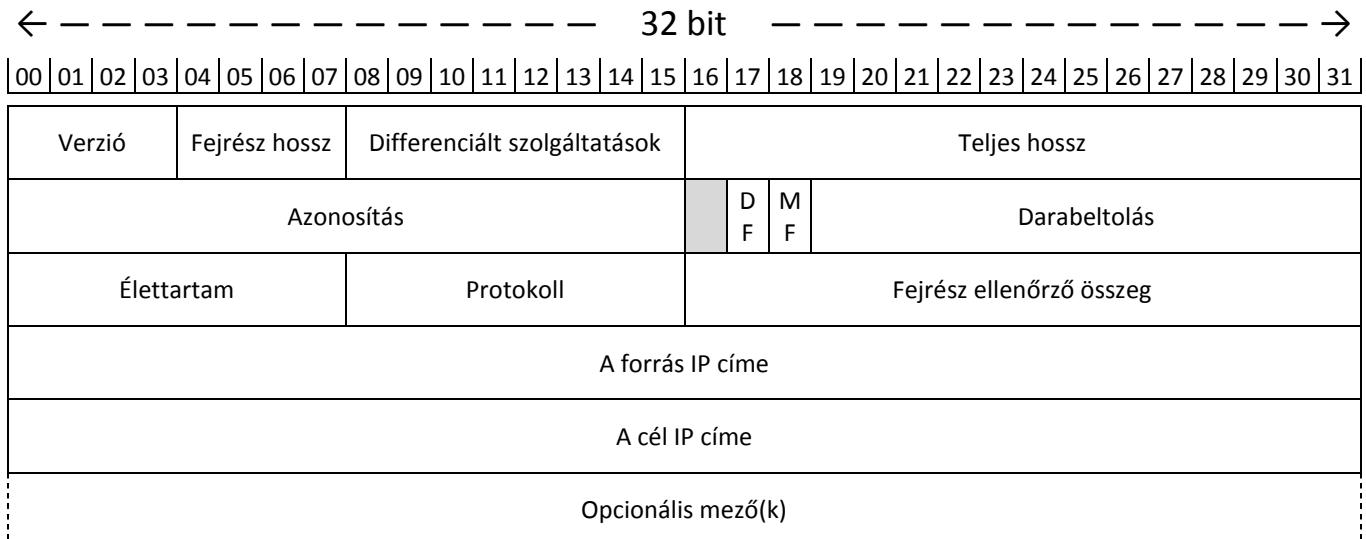
## Az IPv4 protokoll

IP (Internet Protocol, RFC791) a TCP/IP referenciamodell általános adatszállításra szolgáló hálózati réteg protokollja, és mint ilyen csomagkapcsolt, datagram jellegű, megbízhatatlan hálózati szolgáltatást nyújt a szállítási rétegnek. Bármilyen hihetetlen, de a protokoll nem garantálja sem a csomag továbbítását, sem azt, hogy jó helyre érkezik, sem azt, hogy hibátlanul – de mindenből függetlenül, a rendszer a gyakorlatban jól teljesít. Az információ csomagokban továbbítódik, a csomagok haladási útvonala azok feladásakor nem ismert. minden csomag tartalmazza a küldő és a vevő hoszt címét. A hibakezelés és hibajavítás ennek a protokollnak nem feladata.

A datagram tulajdonképpen a szállított csomagot jelenti, mely forrás és a cél hoszt közötti útja során akár a legváltozatosabb típusú és paraméterű hálózatokon haladhat keresztül. A küldő oldal szempontjából a folyamat a következő. A szállítási rétegből származó adatokat a hálózati réteg maximum 64 kB-os méretű csomagokban továbbítja, melyek az útvonal paramétereinek megfelelően esetleg csak további darabolások után érkeznek meg a vevő oldalra. A vevő oldalon a darabokból a csomagok összeállítását követően, a hálózati réteg, az összes üzenetben érintett csomag beérkezése után továbbítja az adatokat a szállítási réteg számára.

Az IPv4 protokoll megismerését célszerű az IP csomag vizsgálatával elkezdeni. Az IPv4-es datagram két fő részből áll, a fejrészről és a törzs (vagy hasznos) részből. Azonnal látszik, hogy az eddig megismert (például keret) formátumokhoz képest egy sokkal összetettebb formátumról van szó.

Az IPv4 fejrész (IPv4 Header) minimum 5 db, maximum 15 db 32 bites (azaz 4 bájtos) szóból áll, vagyis minimum 20 bájt a fix hossza, plusz az ezt követő opcionális mezők. Amennyiben IPv4 csomag található egy Ethernet keret adatmezőjében, akkor az Ethernet keret típusmezőjének értéke 0x0800.



Az átvitel során a bitek balról jobbra, és az ábra szerint soronként kerülnek továbbításra. Ebből az következik, hogy csak sikeres 32 bit átküldése után lehet az egyes sorokat értelmezni. Ez az úgynevezett felsővégi sorrend. Tekintettel arra, hogy az x86 rendszerű számítógépek alapvetően az alsóvégi sorrendet használják, az alsóvégi sorrend használata szerencsésebb lenne (illetve lett volna a protokoll megalkotásakor), de ez a változtatás már nem lehetséges. Így értelem szerűen adáskor és vételkor is (minimum sorrendi) szoftveres átalakításra van szükség, aminek természetesen van erőforrás igénye.

- Az első 32 bites szó (az ábrán az első sor) általános információkat tartalmaz.
  - Verzió: IPv4 esetében értéke 4.
  - A fejrész hosszát (IHL – Internet Header Length) jelöli, 32 bites egységenként, tehát a fejrész értéke minimum 5, azaz  $5 \cdot 32 = 160$  bit = 20 bájt. Tekintettel a 4 bites hosszra, a fejrész maximális értéke 15, azaz  $15 \cdot 32 = 480$  bit = 60 bájt.
  - Ez a differenciált szolgáltatások mező (eredetileg szolgáltatás típusa volt a neve, az új elnevezés alapja az RFC2474 és RFC3168 dokumentum). Gyakorlatilag prioritási illetve megbízhatósági információkat tartalmaz. A 8 bitből 6 bitet használunk szolgáltatási osztályok kódjára, ez a DSCP (Differentiated Services Code Point), és 2 bitet a torlódásértesítési információra, ez az ECN (Explicit Congestion Notification).
  - Teljes hossz: Ez a 16 bites mező – ahogyan az a nevéből is kiderül – a datagram teljes hosszát (azaz a fejrész és a törzsréz együttes hosszát) tartalmazza, bájtonként. Értéke tehát maximum 65535 bájt lehet.

- A második 32 bites szó csomagdarabolással kapcsolatos információkat tartalmaz.
  - Azonosítás: Ez a mező tartalmazza a darabolási információt, azaz azt, hogy egy adott darab melyik datagramhoz tartozik. Darabolás esetén tehát egy adott datagram minden darabja ugyanazt az értéket kell hogy tartalmazza ebben a mezőben. Darabolást végezhet például egy router a célhálózat paramétereinek megfelelően.
  - Itt következik 1 db kihasználatlan bit. Értéke mindig 0.
  - DF (Do Not Fragment): Azt jelzi, ha a csomag nem darabolható.
  - MF (More Fragments): Azt jelzi, hogy a szóban forgó darab az utolsó-e. Bármely darab esetében az értéke 1, az utolsó darab esetében az értéke 0.
  - Darabeltolás (Fragment Offset): Ez a mező a darab első bájtjának az eredeti (nem darabolt) csomagbeli helyét jelez. A mező 13 bit hosszú, azaz maximális értéke  $2^{13} = 8192$ , tehát maximum 8192 darab szerepelhet datagramonként. Az elemi darabméret 8 bájt, azaz a darabok mérete a 8 bájt egész számú többszöröse. Ez alól csak az utolsó darab kivételével, hiszen ez lehet rövidebb is. A maximális hossz így elvileg  $8192 * 8 = 65536$  bájt lehetne, ami azonban 1 bájttal még hosszabb is, mint a Teljes Hossz mező 65535 bájtos maximális értéke.
- A harmadik 32 bites szó csomag élettartamával és a hordozott adatok jellegével kapcsolatos információkat tartalmaz, valamint egy ellenőrző összeget.
  - Az élettartam mező (TTL – Time To Live) egy számláló, amely a csomag érvényességi idejét tartalmazza. Maximális értéke 255, mely egyrészt másodpercenként is, és ugrásonként is (routeren történő áthaladásonként) csökken 1-el. Amikor az érték eléri a 0-át, akkor a csomag kötelezően eldobandó, és a csomag eldobásáról a forrás hosztot értesíteni kell.
  - Protokoll (Transport Layer Protocol) mező arról tájékoztat, hogy a vételi oldalon összeállított datagram, melyik szállítási folyamathoz kell, hogy kerüljön. A protokollok egységes számozását az RFC1700 dokumentum tartalmazza. Legjellemzőbb a TCP (Transmission Control Protocol) és az UDP (User Datagram Protocol) használata. Előbbinél fő szempont a csomagvesztés elkerülése, utóbbinál a kis késleltetés.
  - Fejrész ellenőrző összeg (Header Checksum): Első hallásra egyszerű és magától értetődő dolognak tűnik, hogy egy olyan komplex rendszerhez, mint a fejrész ellenőrző összeget kell gyártani. Viszont vegyük észre, hogy a fejrész (például a TTL miatt) nem egy konstans valami, értéke folyamatosan változik. A kalkuláció módját és menetét az RFC1071 dokumentum tartalmazza.

- A negyedik és az ötödik 32 bites szó címeket tartalmaz.
  - A forrás címe (Source Address): A forrás 4 bájtos IPv4 címe. A gyakorlatban 4db pontokkal elválasztott 10-es számrendszerbeli értéként (0 – 255) hivatkozunk rá, de alapja a kettes számrendszer. A későbbiekben részletesen tárgyaljuk.
  - A cél címe (Destination Address): A cél 4 bájtos IPv4 címe. A gyakorlatban 4db pontokkal elválasztott 10-es számrendszerbeli értéként (0 – 255) hivatkozunk rá, de alapja a kettes számrendszer. A későbbiekben részletesen tárgyaljuk.
- A hatodik 32 bites szótól kezdve, egészen a maximálisan megengedett tizenötödik 32 bites szóig, a fejrész az opciókat tartalmazza.
  - Opciók (Options): Az opciók szolgálnak olyan, viszonylag ritkán használt, IP szintű funkciók megvalósítására, melyeknek nem volt érdemes – a minden csomagban jelen lévő – fejlécben helyen fenntartani. Az opciókat minden állomásnak értelmeznie kell tudni, és fel kell tudnia dolgozni. Fix opciókról van szó, melyeket vagy használunk, vagy nem, de meghatározott implementációjukon nem változtathatunk. Az opciókat 1 bájtos bitsorozat azonosítja. Az RFC791 dokumentum a következő opciókat definiálja:
    - Opciók vége (End Of Options): Ez egy ?0000000 tartalmú bitsorozat, amely jelzi, hogy további opciók nincsenek a csomagban.
    - Nincs művelet (No Operation): Ez egy ?0000001 tartalmú bitsorozat, amely arra használatos, hogy kitöltsse a fel nem használt egy bájtnyi helyeket, hogy a következő opció a 32 bites határon következhessen.
    - Biztonság (Security): Ez egy 10000010 00001011 tartalmú bitsorozat, melyet további 9 bájtnyi paraméter követ(het). A csomag hitelesítéséhez szükséges információkat tartalmazza, illetve azt, hogy mennyire titkos a datagram. Az egykorai katonai alkalmazások örökisége, például a Linux nem is veszi figyelembe.
    - Forrás általi útválasztás (Source Routing): Jelzi, hogy a forrás által IP címenként megadott útvonalon, tehát az állomások listában megadott sorrendjében haladhat végig a csomag. Két válfaja van, a szigorú (Strict) és a laza (Loose).
 

Az első esetben ez egy 10000011 tartalmú bitsorozat, melyet további 3 paraméter (hossz, mutató, adatok) követ. Jelzi, hogy csak a listán felsorolt állomásokon haladhat végig a csomag, és ha két szomszédosnak

felsorolt állomás mégsem szomszédos, akkor a csomag elveszik, és egy ICMP csomag (Source Routing Failed) kerül elküldésre a forráshoz.

Az ICMP (Internet Control Message Protocol) egy olyan protokoll, mely a hibákról és azok típusáról ad tájékoztatást, illetve a hálózati diagnosztika esetén is használható. A későbbiekben részletesen tárgyaljuk.

A második esetben ez egy 10001001 tartalmú bitsorozat, melyet további 3 paraméter (hossz, mutató, adatok) követ. Ha a listán két szomszédosnak feltüntetett állomás a valóságban nem szomszédos, akkor is továbbítódik a csomag a lista következő eleméhez, de már az útválasztók által kijelölt útvonalon.

- Útvonalrögzítés (Record Route): Ez egy 00000111 tartalmú bitsorozat, melyet további 3 paraméter (hossz, mutató, adatok) és a rögzített adat követ. Ez egy utasítás az útválasztók részére, hogy a csomag által érintett állomások IP címe rögzítésre kerüljön a csomagban. Tekintettel a 40 bájtos maximális hosszra, ma már nem minden esetben lehet a teljes útvonalat ezen a módon rögzíteni.
- Adatfolyam azonosító (Stream ID): Ez egy 10001000 00000010 tartalmú bitsorozat, melyet további 1 paraméter (azonosító) követ, hossza fixen 4 bájt. Jelentősége más folyam (azaz kapcsolat) orientált hálózatokkal való együttműködés során, például hálózatos játékok esetében van. Ennek az opciónak egy szétdarabolt csomag esetében annak minden darabjában szerepelnie kell.
- Időbényeg (Internet Time Stamp): Ez egy 01000100 tartalmú bitsorozat, melyet további 4 paraméter (hossz, mutató, számláló, jelzőbit) és a rögzítendő adat követ. Az útvonal azonosítóval együtt használatos, ugyanis az IP címek mellé egy 32 bites időbényeget is rögzít a csomagban.

## 22. fejezet – Az IPv4 protokoll 2, CIDR és Vezérlő és útválasztó protokollok

Az IP címek – ellentétben a MAC címekkel – hierarchikusak, azaz magunk határozhatjuk meg (természetesen bizonyos korlátok között), így lehetőségünk van lokalizációs illetve prioritálási szempontok figyelembe vételére is. Az IP cím egyszerre utal egy hálózatra és azon belül egy hosztra. Pontosabban az IP cím nem egy hosztot azonosít, hanem egy hosztnak egy adott hálózati kártyáját (NIC – Network Interface Card), más megfogalmazásban egy interfészét. Különösen szerver gépek esetében jellemző a több hálózati kártya használata. Tehát célszerű a címet két részre bontani, hálózati azonosítóra és a hosztot (Pontosabban a hoszt hálózati kártyáját) azonosító címre. A rendelkezésünkre álló 32 bites címet, azaz 4\*8 bites címet több különböző módon használhatjuk fel.

Osztály	8 bit		8 bit		8 bit		8 bit		
"A"	0	Hálózat (7)		Hoszt (24)					
"B"	1	0	Hálózat (14)		Hoszt (16)				
"C"	1	1	0	Hálózat (21)		Hoszt (8)			
"D"	1	1	1	0	Többszörös cím (Többesküldés) / Multicast (28)				
"E"	1	1	1	1	Jövőbeli használatra fenntartott / Reserved (28)				

Osztály	Hálózat		Hoszt (NIC) hálózatonként	
"A"	1-127	$2^7 - 2 = 126$ db	0.0.1 - 255.255.255	$2^{24} - 2 = 16777214$ db
"B"	128.0 - 191.255	$2^{14} = 16384$ db	0.1 - 255.255	$2^{16} - 2 = 65534$ db
"C"	192.0.0 - 223.255.255	$2^{21} = 2097152$ db	1 - 255	$2^8 - 2 = 254$ db
"D"		224.0.0.0 - 239.255.255.255		
"E"		240.0.0.0 - 255.255.255.255		

Osztály	Teljes címformátum	Adatszórás (Broadcast)	Decimális / Bináris
"A"	001.000.000.000 - 127.255.255.255	xxx.255.255.255	$127_{10} = 0111\ 1111_2$
"B"	128.000.000.000 - 191.255.255.255	xxx.xxx.255.255	$128_{10} = 1000\ 0000_2$
"C"	192.000.000.000 - 223.255.255.255	xxx.xxx.xxx.255	$192_{10} = 1100\ 0000_2$
"D"	224.000.000.000 - 239.255.255.255	-	$224_{10} = 1110\ 0000_2$
"E"	240.000.000.000 - 255.255.255.255	-	$240_{10} = 1111\ 0000_2$

### Speciális esetek:

- A 0.0.0.0 cím nincs használatban, ugyanis az összes hoszt a bekapcsolás (indulás) pillanatában ezt használja.
- A 255.255.255.255 az adatszóró (Broadcast) címek speciális esete, mely lehetővé teszi az adatszórást a helyi hálózaton, jellemzően egy LAN hálózaton belül.
- Az adott hálózaton kiosztható hosztok darabszáma esetén minden figyelembe kell venni, a fentiek szerint 2db cím (a 000 és a 255) kiesését.
- A 127.xxx.xxx.xxx címek visszacsatolásos (Loopback) tesztelésre vannak fenntartva.

A fentiekből látszik, hogy a leggondosabb és leglogikusabb tervezéssel és előrelátással sem sikerült a mai értelemben életszerű osztályozást kialakítani, anno 1981-ben. A probléma az, hogy a legtöbb céges felhasználó szempontjából az „A” osztály lehetséges hosztjainak száma túl nagy (sok esetben a „B” osztályú hálózatok is csak részben kihasználtak), a „C” osztály lehetséges hosztjainak száma pedig túl kevés. Ma már az látszik, hogy szerencsésebb lett volna a „C” osztályú hosztok esetében 8 helyett 10 bitet alkalmazni, így 254 helyett 1022 lenne csatlakoztatható hosztok száma.

Az „A”, a „B” és a „C” osztály esetében is kijelölésre került privát (Private) tartomány, melyben szereplő privát címek, közvetlenül az Internetről nem érhetők el. Az Internet elérése a hosztokról (azaz a fordított irány) azonban megfelelő útválasztással és átjáró használattal megoldott. A privát tartományok tehát megtöbbszörözök az Internetre kapcsolható hosztok számát, és jellemzően abban az esetben használatosak, amikor egy adott felhasználó hosztjait nem kell az Internet felől elérni.

Osztály	Privát IP tartomány	Hálózatok és Hosztok (NIC) száma
„A”	10.0.0.0 - 10.255.255.255	1db hálózat, 16777214 db hoszt (NIC)
„B”	172.16.0.0 - 172.31.255.255	16 db hálózat, hálózatonként 65534 db hoszt (NIC)
„C”	192.168.0.0 - 192.168.255.255	256 db hálózat, hálózatonként 254 db hoszt (NIC)

## Hálózati maszk (Netmask), alhálózati maszk (Subnet Mask) és a CIDR

Az IP címekhez hasonlóan a hálózati maszk is 32 bites. A hálózati maszk segítségével korlátozzuk a hálózatunkon használható IP címek számát. Maga a hálózati maszk bináris formában minden 1-eskből álló sorozattal kezdődik, és 0-ákból álló sorozattal végződik, tehát a hálózati maszkban 0 értéket semmiképpen sem követhet 1-es érték. Az IP cím és a hálózati maszk közti ÉS (AND) művelet a hálózat címét adja vissza, az IP cím és a hálózati maszk NEGÁLTja (Wildcard) közti ÉS művelet pedig a hoszt címét a hálózaton belül. A hálózati maszk segítségével a teljes cím szétválasztható a hálózat címére, illetve a hoszt címére. A hálózati maszk segítségével a statikus osztályba sorolás, azaz a hálózat/hoszt határ dinamikusan módosítható.

A hálózati maszk az IP címhez hasonlóan négy, pontokkal elválasztott decimális számmal (0-255) írható le, de szokásos jelölése még az, amikor csak a benne lévő 1-esek darabszámára, vagyis az előtag (Prefix) hosszára hivatkozunk, az IP cím végén, egy perjel „/” után írva.

Osztály	Alapértelmezett hálózati maszk	Bináris ábrázolás	1-esek száma
"A"	255.0.0.0	11111111.00000000.00000000.00000000	/8
"B"	255.255.0.0	11111111.11111111.00000000.00000000	/16
"C"	255.255.255.0	11111111.11111111.11111111.00000000	/24

Az alhálózati maszk a hálózati maszk által meghatározott IP címtéren belül független alhálózatok, blokkok létrehozását teszi lehetővé. A blokk kezdőcíme a hálózat címe, utolsó címe pedig az adatszóró (Broadcast) cím – ezt a két címet a hosztok nem használhatják. Ez a változó blokkméretű elosztás, az osztály nélküli (más megközelítésben: tartományon belüli) forgalomirányítás (CIDR – Classless Inter-Domain Routing), mely 1993. óta van használatban.

Alhálózati maszk	Bináris ábrázolás	Prefix	Hosztok max. száma
255.255.255.255	11111111.11111111.11111111.11111111	/32	0 db
255.255.255.254	11111111.11111111.11111111.11111110	/31	1 db
255.255.255.252	11111111.11111111.11111111.11111100	/30	2 db
255.255.255.248	11111111.11111111.11111111.11111000	/29	6 db
255.255.255.240	11111111.11111111.11111111.11110000	/28	14 db
255.255.255.224	11111111.11111111.11111111.11100000	/27	30 db
255.255.255.192	11111111.11111111.11111111.11000000	/26	62 db
255.255.255.128	11111111.11111111.11111111.10000000	/25	126 db
255.255.255.0	11111111.11111111.11111111.00000000	/24	254 db
255.255.254.0	11111111.11111111.11111110.00000000	/23	510 db
255.255.252.0	11111111.11111111.11111100.00000000	/22	1 022 db
255.255.248.0	11111111.11111111.11111000.00000000	/21	2 046 db
255.255.240.0	11111111.11111111.11110000.00000000	/20	4 094 db
255.255.224.0	11111111.11111111.11100000.00000000	/19	8 190 db
255.255.192.0	11111111.11111111.11000000.00000000	/18	16 382 db
255.255.128.0	11111111.11111111.10000000.00000000	/17	32 766 db
255.255.0.0	11111111.11111111.00000000.00000000	/16	65 534 db
255.254.0.0	11111111.11111110.00000000.00000000	/15	131 070 db
255.252.0.0	11111111.11111100.00000000.00000000	/14	262 142 db
255.248.0.0	11111111.11111000.00000000.00000000	/13	524 286 db
255.240.0.0	11111111.11110000.00000000.00000000	/12	1 048 574 db
255.224.0.0	11111111.11100000.00000000.00000000	/11	2 097 150 db
255.192.0.0	11111111.11000000.00000000.00000000	/10	4 194 302 db
255.128.0.0	11111111.10000000.00000000.00000000	/9	8 388 606 db
255.0.0.0	11111111.00000000.00000000.00000000	/8	16 777 214 db
254.0.0.0	11111110.00000000.00000000.00000000	/7	33 554 430 db
252.0.0.0	11111100.00000000.00000000.00000000	/6	67 108 862 db
248.0.0.0	11111000.00000000.00000000.00000000	/5	134 217 726 db
240.0.0.0	11110000.00000000.00000000.00000000	/4	268 435 454 db
224.0.0.0	11100000.00000000.00000000.00000000	/3	536 870 910 db
192.0.0.0	11000000.00000000.00000000.00000000	/2	1 073 741 822 db
128.0.0.0	10000000.00000000.00000000.00000000	/1	2 147 483 646 db

A CIDR leírását az RFC1700 dokumentum tartalmazza. A már korábban az osztályba sorolásra használt /8, /16 és /24 mellett bevezetett többi előtag hossz megjelenése nagyban hozzájárult az IPv4 címek elfogyásának időbeli késleltetéséhez, hiszen hatékonyabb címkiosztást tett lehetővé, akár különböző méretű hálózatok esetében is folyamatosan tölthető fel, osztható ki a címtér. Az egyetlen ökölszabály az, hogy a hálózatok mérete kettő valamelyik hatványa ( $2^x$ ) legyen, illetve az, hogy a hálózatok, a méretet meghatározó kettő hatványának a többszöröseire, mint határokra legyenek illesztve. Ettől kezdve az IP cím felépítése már „előtag + hoszt cím” modellként értelmezendő.

Semmiéppen sem szabad arról megfeledkezni, hogy az IP-protokoll számára az IP-cím és az alhálózati maszk csak együtt értelmes, mert az IP-cím minden két részből áll. Az alhálózati maszk hiányában a hoszt nem tudja meghatározni az őt tartalmazó hálózat címét, ami pedig az útválasztáshoz elengedhetetlen.

A hálózat, és az alhálózat közötti különbség fizikai és logikai megközelítésben érthető meg. Azaz hálózatról fizikailag összekötött hosztok esetében beszélünk, az alhálózat pedig a hálózatban összekötött hosztok logikai (azaz pl. alhálózati maszkkal történő) blokkokra történő szétválasztása. A blokkok közötti átjárhatóság közvetlenül nem biztosított.

A hierarchikus címzésnek – a hálózati és a hoszt cím szétválasztásának – előnye, hogy az útválasztók a számukra érdektelen hoszt címeket nem kell, hogy táblázataikban tárolják. Egy hálózaton belül az összes hoszt hálózati címe azonos, az útválasztáshoz, azaz a célhálózat megtalálásához, elég csak ezt tárolni. A nem kellő körültekintéssel történő hálózatkiosztás viszont pazarló lehet, azaz ha túl nagy alhálózatot jelölünk ki, és így sok kihasználatlan IP cím kerülhet lefoglalásra.

#### Gyakorlati példák:

1. Legyen egy belső használatú IP cím: 192.168.100.1, és legyen a hozzá tartozó hálózati maszk: 255.255.255.0. Az IP címet jelölhetjük így is: 192.168.100.1/24.

Bináris formában az IP cím: 11000000 10101000 01100100 00000001

A hálózati maszk: 11111111 11111111 11111111 00000000

Ezekből ÉS kapcsolattal a hálózat címe: 11000000 10101000 01100100 00000000

A hosztok számára marad az utolsó 8 bit, így elvileg  $2^8 = 256$  db IP cím osztható ki.

Magában a 192.168.100.0/24 jelölésű hálózatban elvileg 256 hosztot jelölhetnénk ki, de mivel a 0 a hálózatot, a 255 (ami binárisan csupa 1-esből áll) pedig a Broadcast-ot jelöli,  $256 - 2 = 254$  kiosztható hoszt (NIC) IP címünk áll gyakorlatilag rendelkezésre.

A hálózati maszk megváltoztatásával szűkíthetjük a hálózaton belül kiosztható IP címek számát, a létrehozható alhálózatok számát pedig bővíthetjük. Ha a hálózati maszkot a következő – a definíció szerint alhálózati maszkra – 255.255.255.248-ra cseréljük, a következő történik.

A hosztok számára marad az utolsó 3 bit, így elvileg  $2^3 = 8$  db IP cím osztható ki. Magában a 192.168.100.0/29 jelölésű alhálózatban elvileg 8 hosztot jelölhetnénk ki, de mivel a 0 a hálózatot, a 7 (ami binárisan csupa 1-esből áll) pedig a Broadcast-ot jelöli,  $8-2 = 6$  kiosztható hoszt (NIC) címünk áll gyakorlatilag rendelkezésre.

Az alhálózat IP címe:	192.168.100.0/29	11000000 10101000 01100100 00000000
Az alhálózati maszk:	255.255.255.248	11111111 11111111 11111111 11111000
A legkisebb IP cím:	192.168.100.1	11000000 10101000 01100100 00000 <b>001</b>
A legnagyobb IP cím:	192.168.100.6	11000000 10101000 01100100 00000 <b>110</b>
A Broadcast IP cím:	192.168.100.7	11000000 10101000 01100100 00000111

Viszont lehetőségünk van a következő szabad IP címtől újabb alhálózatot definiálni, azaz például a 192.168.100.8/29 jelölésű alhálózatot. Ez esetben az alhálózat címe a 192.168.100.8, a 6 db kiosztható cím pedig a 9-10-11-12-13-14. A 192.168.100.15 (amiben a 15 binárisan csupa 1-esből áll) pedig a Broadcast-ot jelöli.

Az alhálózat IP címe:	192.168.100.8/29	11000000 10101000 01100100 00001000
Az alhálózati maszk:	255.255.255.248	11111111 11111111 11111111 11111000
A legkisebb IP cím:	192.168.100.9	11000000 10101000 01100100 00001 <b>001</b>
A legnagyobb IP cím:	192.168.100.14	11000000 10101000 01100100 00001 <b>110</b>
A Broadcast IP cím:	192.168.100.15	11000000 10101000 01100100 00001111

Mivel egy „C” osztályú hálózat blokkokra osztásáról van szó, ezzel a módszerrel, ez esetben összesen  $2^5 = 32$  db alhálózatot definiálhatunk, egyenként  $2^3 - 2 = 6$  db kiosztható IP címmel. Az IP cím utolsó 8 bitjéből az első 5 bit az alhálózat sorszámát, az utolsó 3 bit pedig a alhálózaton belüli IP címet jelöli. A 32. alhálózat adatai:

Az alhálózat IP címe:	192.168.100.248/29	11000000 10101000 01100100 11111000
Az alhálózati maszk:	255.255.255.248	11111111 11111111 11111111 11111000
A legkisebb IP cím:	192.168.100.249	11000000 10101000 01100100 11111 <b>001</b>
A legnagyobb IP cím:	192.168.100.254	11000000 10101000 01100100 11111 <b>110</b>
A Broadcast IP cím:	192.168.100.255	11000000 10101000 01100100 11111111

Hasznos segítség a hálózatok tervezéséhez egy IP kalkulátor, ami hasonló hálózatok definiálásában segít: <http://jodies.de/ipcalc>

2. Egy nemzetközi vállalat rendelkezik 8192 db IP címmel, amit 4 különböző telephelyén (A, B, C, D) szeretne használatba venni. Az IP címek a 212.32.0.0 címtől kezdődően állnak rendelkezésre.

Telephely	Első elvi cím	Utolsó elvi cím	Igényelt hoszt	Optimális hoszt	Hálózat címe
A	212.32.0.0	212.32.7.255	2 000 db	2 048 db	212.32.0.0/21
B	212.32.8.0	212.32.11.255	1 000 db	1 024 db	212.32.8.0/22
C	212.32.12.0	212.32.15.255	1 000 db	1 024 db	212.32.12.0/22
D	212.32.16.0	212.32.31.255	4 000 db	4 096 db	212.32.16.0/20

„A” hálózat IP címe:	212.32.0.0/21	11010100 00100000 00000000 00000000
„A” alhálózati maszk:	255.255.248.0	11111111 11111111 11111000 00000000
„A” legkisebb IP címe:	212.32.0.1	11010100 00100000 00000000 00000001
„A” legnagyobb IP címe:	212.32.7.254	11010100 00100000 00000111 11111110
„A” Broadcast IP címe:	212.32.7.255	11010100 00100000 00000111 11111111
„B” hálózat IP címe:	212.32.8.0/22	11010100 00100000 00001000 00000000
„B” alhálózati maszk:	255.255.252.0	11111111 11111111 11111100 00000000
„B” legkisebb IP címe:	212.32.8.1	11010100 00100000 00001000 00000001
„B” legnagyobb IP címe:	212.32.11.254	11010100 00100000 00001011 11111110
„B” Broadcast IP címe:	212.32.11.255	11010100 00100000 00001011 11111111
„C” hálózat IP címe:	212.32.12.0/22	11010100 00100000 00001100 00000000
„C” alhálózati maszk:	255.255.252.0	11111111 11111111 11111100 00000000
„C” legkisebb IP címe:	212.32.12.1	11010100 00100000 00001100 00000001
„C” legnagyobb IP címe:	212.32.15.254	11010100 00100000 00001111 11111110
„C” Broadcast IP címe:	212.32.15.255	11010100 00100000 00001111 11111111
„D” hálózat IP címe:	212.32.16.0/20	11010100 00100000 00010000 00000000
„D” alhálózati maszk:	255.255.240.0	11111111 11111111 11110000 00000000
„D” legkisebb IP címe:	212.32.16.1	11010100 00100000 00010000 00000001
„D” legnagyobb IP címe:	212.32.31.254	11010100 00100000 00011111 11111110
„D” Broadcast IP címe:	212.32.31.255	11010100 00100000 00011111 11111111

Jusson eszünkbe a CIDR korábban említett ökol szabálya! Például a „D” hálózat esetében, ha bármely okból is 212.32.17.1 IP címnél jelöltük volna ki a hálózat IP címét, akkor csak a /24-es maszkot használhattuk volna, azaz csak 256-2 db hosztot tudtunk volna kiosztani!

„Dx” hálózat IP címe:	212.32.17.0/24	11010100 00100000 00010001 00000000
„Dx” alhálózati maszk:	255.255.255.0	11111111 11111111 11111111 00000000
„Dx” legkisebb IP címe:	212.32.17.1	11010100 00100000 00010001 00000001
„Dx” legnagyobb IP címe:	212.32.17.254	11010100 00100000 00010001 11111110
„Dx” Broadcast IP címe:	212.32.17.255	11010100 00100000 00010001 11111111

A CIDR bevezetésének az útválasztásra gyakorolt hatása az előző példa segítségével érhető meg legegyszerűbben. Az egyes telephelyek közelében lévő útválasztók mindegyikének ismernie kell az egyes telephelyeken lévő hálózatok IP címének előtagjait. Ezek az előtagok (a telephelyek földrajzi elhelyezkedés függvényében) más-más vonalakon történő kapcsolódást jelenthetnek az útválasztókban, azaz telephelyenként ez egy-egy bejegyzést jelent az útválasztók táblázataiban.

Távoli útválasztók esetében azonban az útválasztás logikája kicsit egyszerűsödik. Indulunk ki abból, hogy a közelben (az országon belüli) útválasztók átadják táblázataik tartalmát a távoli (például egy másik ország) útválasztói felé. Ez esetben azonban egy olyan helyi útválasztó, mely minden négy telephely felé rendelkezik bejegyzéssel – azaz ismeri a továbbmenő vonalat – a távoli útválasztó számára egy összefoglaló címet juttat el. Ezt azért lehet meg, mert a távoli útválasztó számára elég ezen közeli útválasztó megtalálása és az összefoglaló cím tárolása. A négy cég felé vezető irányokat elég, ha a közeli útválasztó ismeri. Ez az összefoglaló cím esetünkben a 212.32.0.0/19.

A négy hálózat összefoglaló IP címe:	212.32.0.0/19	11010100 00100000 00000000 00000000
A négy hálózat hálózati maszkja:	255.255.224.0	11111111 11111111 11100000 00000000
A négy hálózat legkisebb IP címe:	212.32.0.1	11010100 00100000 00000000 00000001
A négy hálózat legnagyobb IP címe:	212.32.31.254	11010100 00100000 00011111 11111110
A négy hálózat Broadcast IP címe:	212.32.31.255	11010100 00100000 00011111 11111111

Erre a cím összefogásra, csoportosításra (Aggregation) azért volt lehetőségünk, mert azonos régióban kerültek kiosztásra az egymást követő IP címek. Maga a címcsoportosítás folyamata automatikus, az útválasztók kezelői beavatkozás nélkül képesek azt elvégezni. Amennyiben az egyes telephelyek az Internet topolójában távol vannak egymástól, például különböző kontinenseken helyezkednek el, akkor ez az előny elveszik.

A csomagokat tehát vagy a klasszikus útválasztás módszereivel meghatározott legjobb útvonal irányába, vagy a leghosszabb egyező előtag irányába (Longest Matching Prefix) kell küldeni.

## Hálózati címfordítás (NAT)

A címfordítás, mint igény szintén az IPv4 címek beláthatón időn belüli elfogyásának következtében merült fel. Az IPv4 címek 32 bites hosszából fakadó korlátja, az, hogy az elvileg kiosztható egyedi IP címek száma  $2^{32} = 256^4$ , azaz összesen „4 294 967 296 db”, ráadásul ez a szám már magába foglal jó pár ki nem osztható címet is (Broadcast, Loopback, Private). Tekintettel az Internet folyamatos térhódítására a négy milliárd nem is tűnik olyan nagy számnak. Az IPv4 címek elfogyásának időbeli késleltetéséhez a CIDR 1993-as megjelenése nagyban hozzájárult, amit a 2001-ben a NAT megjelenése követett. A címfordítás technológiája miatt nem került gyorsabban bevezetésre az IPv6 szabvány, amely kifejlesztésének egyik oka az IPv4 fogyatkozó címtartományának kiváltása volt. Hosszabb távon természetesen az IPv6 bevezetése a megoldás.

A hálózati címfordítás (NAT – Network Address Translation) a címfordításra képes hálózati eszközök (útválasztók, tűzfalak) kiegészítő szolgáltatása, mely lehetővé teszi a belső hálózati hosztok közvetlen kommunikációját tetszőleges protokollokon, keresztül külső hálózati (jellemzően Interneten található) hosztokkal. A kommunikációhoz tehát a belső hálózat hosztjainak így nem kell nyilvános IP címmel rendelkezniük. A NAT leírását az RFC3022 dokumentum tartalmazza.

A hálózati címfordítást végző eszköz egy belső hálózatban lévő hosztokról érkező csomagokat az Internetre továbbítás előtt úgy módosítja, hogy azok feladójaként saját magát tünteti fel. Ezért az azokra érkező válaszcsomagok is hozzá kerülnek majd továbbításra, amiket – a célállomás címének visszamódosítása után – a belső hálózaton elhelyezkedő eredeti feladó hoszt részére ad át. A címfordítás tehát egy aktív hálózati eszközt igényel, amely folyamatosan figyeli az érkező csomagokat és azok feladói és címzettjei alapján elvégzi a szükséges módosításokat. A címfordítást általában egy tűzfal végzi el, amely megfelelően szétválasztja a külső Internetet a belső hálózattól. Innen származik a külső, illetve belső hálózat elnevezés is. A belső hálózatnak olyan címtartományt kell adni, amelyet minden hálózati eszköz a nemzetközi szabványoknak megfelelően belsőnek ismer el, és így azokat nem irányítja közvetlenül a külső hálózat felé. A privát, vagy belső tartományokról már volt szó, ezek azok:

Osztály	Privát IP tartomány	Hálózatok és Hosztok (NIC) száma
"A"	10.0.0.0 - 10.255.255.255	1db hálózat, 16777214 db hoszt (NIC)
"B"	172.16.0.0 - 172.31.255.255	16 db hálózat, hálózatonként 65534 db hoszt (NIC)
"C"	192.168.0.0 - 192.168.255.255	256 db hálózat, hálózatonként 254 db hoszt (NIC)

A címfordítás segítségével megoldható, hogy akár egy egész cég teljes belső hálózati forgalma egyetlen külső IP cím mögött legyen, azaz gyakorlatilag egyetlen külső címet használ el egy több száz hosztból álló hálózat. A belső forgalomban természetesen szükség van az egyedi belső címekre, de erről csak a címfordítást végző hálózati eszközöknek kell tudnia, kifelé ennek részleteiről már nem láthatók információk. Így létrejöhét egy olyan konfiguráció is, hogy egy viszonylag nagy cég teljes külső címfoglalása csak kb. 10-20 db IP cím, míg a belső forgalmukban akár több ezer belső IP cím is lehet. Nagy előnye ennek a technikának, hogy ugyanazt a belső tartományt nyugodtan használhatja bárki más is, amíg minden egyedi külső cím mögé van fordítva. Akár az összes NAT-ot használó cég belső hálózatában lehet minden gép a 10.0.0.0 vagy a 192.168.0.0 tartományban, ha kifelé valóban egyedi címmel látszanak.

## Az Internet vezérlő protokolljai

A hálózati rétegen nem kizárolag az adatok továbbítására szolgáló IP protokoll használatos, hanem számos egyéb protokoll is. Ezek vezérlési, címszervezési illetve kényelmi feladatokat látnak el. A leggyakrabban a következő protokollokkal találkozhatunk.

- ICMP (Internet Control Message Protocol / Internetes vezérlőüzenet protokoll)  
Az ICMP egy olyan protokoll, mely a hibákról és azok típusáról ad tájékoztatást, illetve a hálózati diagnosztika esetén is használható. Leírását az RFC792 dokumentum definiálja. A szabvány 8 biten ábrázolja az ICMP üzenetek típusait, azaz elvileg 256 féle üzenet létezhet. A 256 lehetséges üzenetből azonban csak 40 féle üzenet van definiálva. Ezek közül is csak a leggyakoribb 11 db üzenet kerül megemlítésre.
  - „0 – Echo Replay” (Ping)  
A „8”-as típusú (visszhang csomag) csomagra érkező válasz.
  - „3 – Destination Unreachable”  
A célállomás nem érhető el, a csomagot nem lehet kézbesíteni. Ennek az üzenetnek 16 altípusa van.
  - „4 – Source Quench”  
Ez egy útválasztók által küldhető üzenet, a forrás elnyomás, ami azt jelzi, hogy az útválasztónak nincs elég memóriája a kérés feldolgozására, ezért kéri a bejövő forgalom csökkentését.

- „5 – Redirect Message”
 

Egy valószínűleg rosszul irányított csomaggal kapcsolatos üzenet másik hálózat vagy hoszt felé. Célja, hogy az adott hosztnak küldött üzenetek a megfelelő irányba legyenek elküldve. Ennek az üzenetnek 4 altípusa van.
- „8 – Echo Request” (Ping)
 

Visszhang kérése, azaz annak ellenőrzése, hogy a keresett hoszt elérhető-e.
- „9 – Router Advertisement”
 

Az útválasztó saját adatainak hirdetése a többi (közeli) útválasztó felé.
- „10 – Router Solicitation”
 

A (közeli) útválasztók adatainak kérelmezése.
- „11 – Time Exceeded”
 

Egy csomag vagy darab (Fragment) érvényességi idejének (TTL) lejártára figyelmeztető üzenet. Ennek az üzenetnek 2 altípusa van.
- „12 – Bad IP Headers”
 

Érvénytelen IP fejrészre, vagy hibás paraméterre figyelmeztető üzenet. Ennek az üzenetnek 3 altípusa van.
- „13 – Timestamp”
 

Ugyanaz, mint a „8”-as (Ping) de időbényeggel együttes kérésről van szó. Célja az időszinkronizáció.
- „14 – Timestamp Reply”
 

Ugyanaz, mint a „0”-ás (Ping) de időbényeggel együttes válaszról van szó. Célja az időszinkronizáció.

- ARP (Address Resolution Protocol / Címfeloldási protokoll)

Az IP címeket a hálózati kommunikáció során valamely módszerrel mindenkorban az adatkapcsolati rétegben használatos fizikai MAC címekké kell alakítani, hiszen a keretek a fizikai MAC címek segítségével érik el a célállomásokat. Az ARP az IP címek és fizikai címek egymáshoz rendelésének módszere. Segítségével az IP cím ismeretében hozzájuthatunk a 48 bites (az eszköz gyártója által meghatározott) fizikai MAC címhez. Az ARP leírását az RFC826 dokumentum tartalmazza.

Az ARP protokoll az összerendelt adatokat a memóriájában (ARP Cache) tárolja. Amennyiben egy keresett összerendelés itt nem található meg, akkor azt fel kell kutatni a hálózaton. Ez egy speciális Ethernet Broadcast üzenet küldésével történik az ff:ff:ff:ff:ff:ff MAC címre, (ez egy adatkapcsolati régen lezajló folyamat, nem összekeverendő az IP Broadcast-tal) melyet az összes, a szegmensbe bekapcsolt hoszt megkap. A keresett IP címet mindegyik hoszt összehasonlítja a saját IP címével, és egyezés esetén az érintett hoszt egy ARP választ küld a kérdezőnek.

A kérdező ebből a válaszból olvassa ki a szükséges IP cím és Ethernet cím összerendelést. Az összerendelés ezután bekerül az ARP memóriájába, és ott egy megadott ideig (ARP Refresh time) megőrzésre kerül. Egy adott hoszt ARP memóriájában tehát csak az adott Ethernet szegmensen levő hosztok IP cím és Ethernet cím összerendelései találhatók, mert egy másik szegmensen levő hosztal a kommunikáció már (általában) útválasztón keresztül történik.

Két hoszt a következő négy alapesetben veszi igénybe az ARP protokolلت:

- Ha a két hoszt ugyanazon a hálózaton található, és az egyik szeretne csomagot küldeni a másik számára.
- Ha a két hoszt különböző hálózaton található, és így útválasztón vagy az alapértelmezett átjárón (Default Gateway) keresztül érik el egymást.
- Ha egy útválasztónak tovább kell küldenie egy hoszt csomagját egy másik útválasztón keresztül.
- Ha egy útválasztónak tovább kell küldenie egy hoszt csomagját a címzettnek, ami ugyanazon a hálózaton található.

Az első esetben a két hoszt ugyanazon a fizikai hálózaton található, vagyis képesek közvetlenül kommunikálni egymással útválasztó igénybevétele nélkül is. A másik három eset – ami az Interneten leggyakoribb – az, amikor bármely két hoszt (jellemzően számítógép) több mint 3 ugrás (Hop) távolságra van egymástól.

- **DHCP**

(Dynamic Host Configuration Protocol / Dinamikus hosztkonfigurációs protokoll)

A DHCP protokoll feladata az, hogy a TCP/IP hálózatra csatlakozó hosztok automatikusan megkapják a hálózat használatához szükséges beállításokat, IP címet, a hálózati maszkot és az alapértelmezett átjáró IP címét. A protokoll leírását az RFC1541 és RFC2131 dokumentumok definiálják. Hárrom féle IP cím kiosztási módszer használatos a DHCP segítségével.

- statikus
  - A kiosztás alapja egy, a MAC címekre épülő algoritmus, illetve lehetőség van manuális IP cím kiosztásra is.
- automatikus
  - Címkiosztás az IP tartomány megadásával.
- dinamikus
  - Címkiosztás az IP tartomány megadásával, és az IP címek „újrahasznosításával”. Az újrahasznosítás paramétere a bérleti idő (Lease Time), amely a már egyszer kiosztott IP cím újra kioszthatóságát jelenti. Csak a bérleti idő lejárta után osztható ki az adott IP cím másik hoszt részére.

DHCP szerverek használata esetén bárki, akinek fizikai kapcsolata van a hálózattal, könnyen juthat IP címhez, így a DHCP használata megkönnyíti a hálózati betörésekkel való védelem nélküli (vagy nem kellően védett) vezeték nélküli hálózatok esetében a DHCP egyszerű hozzáférést biztosít a sugárzás hatókörén belül a hálózathoz – hiszen ez esetben fizikai kapcsolata sincs szükség. A behatoló így elérheti az Internet használatot és a (nem kellőképpen védett) megosztott erőforrásokat is.

## Az útválasztás protokolljai (Routing Protocol)

Az útválasztás az a folyamat, ami során egy hálózati protokoll egy csomagja az útválasztók sorozatán keresztül a feladó hosztól eljut a címzett hosztig. mindenéppen szükség van arra, hogy az útválasztók kommunikáljanak egymással, hogy korábban tárgyalt útválasztó algoritmusok segítségével eldönthessék, hogy egy adott végcél (hoszt) felé melyik irányba kell továbbítani a csomagot. Tekintettel arra, hogy az Internet autonóm rendszerekből (AS) áll, az egyes rendszerekben más-más protokollok használatosak. Maguk a rendszerek is szétválaszthatóak az útválasztás szempontjából körzetén belüli (Intradomain Routing) illetve körzetek közötti (Interdomain Routing) útválasztást használó rendszerekre. Előbbieket belső átjáró protokollnak (IGP – Interior Gateway Protocol), utóbbiakat külső átjáró protokollnak (EGP – Exterior Gateway Protocol) nevezünk. Az útválasztás protokolljai a kommunikáció módját és az útvonal kiválasztásának mikéntjét is meghatározzák. Más megfogalmazásban az útválasztás protokolljai a korábban tárgyalt útválasztó algoritmusok gyakorlati megvalósításának módjai.

- RIP (Routing Information Protokoll)

Az első IGP protokoll, az útválasztási információs protokoll (RIP) egy távolságvektor alapú protokoll, amely első verziójában maximum 15 ugrással (Hop) valamint egy időzítő segítségével és a hálózati útvonalak költségei alapján igyekezett az útválasztás feladatát megoldani. 1988-ban vezettek be, és leírását az RFC1058 dokumentum tartalmazza. A maximum 15 ugrás a kompatibilitás miatt a későbbi verziókban is megmaradt, de a protokoll eszköztára már a CIDR lehetőségeihez is alkalmazkodott. Legnagyobb problémája, hogy nem mentes a végtelenig számolás problémájától, a hurokmentességtől és hogy az egyre nagyobb hálózatok kiszolgálására csak korlátozottan alkalmas.

- **OSPF (Open Shortest Path First)**

Az OSPF szintén IGP protokoll – a nyílt hozzáférésű, a legrövidebb utat preferáló protokoll – melyet 1990-ben vezettek be, és leírását az RFC2328 (v1) és RFC1247 (v2 1991) dokumentumok tartalmazzák. Az OSPF főleg a RIP hibáinak a kijavítását célozta meg sikkerrel, hiszen az OSPF egy kifinomultabb, kevesebb sávszélességet igénylő hurokmentes megoldást kínál. A távolság vektoros útválasztás (Distance Vector Routing) módszere helyett az OSPF a kapcsolatállapot alapú útválasztás (Link State Routing) módszerét használja.

- **BGP (Border Gateway Protocol)**

A határátjáró protokoll (BGP) tulajdonképpen egy EGP protokoll, hiszen a különböző AS-ek közötti útválasztás a feladata. A BGP-nek is komoly evolúciója van, első verziója 1989-ben jelent meg, leírását az RFC1105, RFC1163, RFC1164 dokumentum tartalmazta. Ezeket követte az RFC1267 (v2 1991), RFC1771 (v3 1995, ez már a CIDR lehetőségeihez is alkalmazkodott) és az RFC4271 (v4 2006) dokumentum.

A BGP segítségével, felhasználás jellege miatt tetszőleges topológiákat is tudunk kell támogatni, de közben gondoskodni kell a hurokmentességről is. A távolság vektoros útválasztás itt optimálisan azért nem használható, mert nem minden legolcsóbb útvonal a kívánatos. A kapcsolatállapot alapú útválasztás pedig azért használhatatlan, mert ehhez az egész Internet topológiáját tárolni kellene, ami még az AS szinten is megoldhatatlan. Az IETF ezért alkotott meg egy közbülső, új megoldást, az út-vektorokat (Path Vectors). A módszer lényege az, hogy minden terjesztett útvonalban a célpontig vezető teljes útvonalat leírjuk. Így a hurokmentességet minden útválasztó könnyen ellenőrizhet. Ha egy megkapott útvonalban már szerepel, akkor azzal az útvonallal már nem foglalkozik a továbbiakban. Emellett nincs szükség valamilyen, az egész Internetben egységes költség definiálására, hiszen mindenki a teljes útvonalat saját szempontjai szerint pontozza. Az eljárás legfőbb előnye a hatékonyság, egyetlen hátránya a nagy memóriaigény, az útválasztók tábla bejegyzéseinek jelentős növekedése.

## Többesküldés (Multicast) az Interneten

Az eddigiekből már kiderült, hogy a „D” osztályú címek a Multicast céljára vannak fenntartva. minden „D” osztályú cím tehát egy hoszt csoportot jelöl, így – elvileg – akár egyszerre sok millió ügyfélnek küldhetünk IP csomagokat. A gyakorlatban ez azonban szinte kivitelezhetetlen, hiszen minél több hosztról beszélünk, annál változatosabb földrajzi eloszlásban helyezkedhetnek el az egyes hosztok. Azaz elküldhetjük a

csomagokat, de arra semmi garancia sincs, hogy azok minden érintett hoszthoz meg is érkeznek. Kijelenthetjük tehát, hogy a Multicast mint feladat, nem tökéletesen megoldott, hiszen a Multicast alapesetben úgy valósítható meg, hogy minden a csoporthoz tartozó útválasztó számára egyesével elküldjük a Multicast csomagokat, miközben az is fontos lenne, hogy egy csomag csak egyszer haladjon végig egy útvonalon. Az útválasztóknak azt a feladatot kell megoldaniuk, hogy megvizsgálják az összes „D” osztályú címmel feladott csomagot és kiszűrjék azokat, melyek nekik nem szólnak. Valamilyen módon azonban az útválasztók tudomására kell hozni, hogy az adott útvonalon van-e olyan hoszt, amely tagja a megcímzett Multicast csoportnak.

Ennek az összetett feladatnak a megoldására hozták létre az Internetes csoportkezelő protokollt (IGMP – Internet Group Membership Protocol) 1989-ben, melynek a leírását az RFC1112 (v1), RFC2236 (v2, 1997), RFC3376 (v3, 2002) és RFC4604 (2006) dokumentumok tartalmazzák. A feladat nem egyszerű, ahogyan azt a protokoll evolúciója is jelzi. Az IGMP olyan pont-multipont alkalmazások esetében használatos, mint a Video Streaming vagy a csoportos játék, és célja az igénybevett erőforrások hatékonyabb kihasználása. Az IGMP protokoll működésének lényege az, hogy az érintett útválasztók lekérdező- és válaszcsomagok segítségével egy speciális többesküldés-feszítőfát (Multicast Spanning Tree) hoznak létre. Tehát kétféle IGMP üzenet létezik, az egyik segítségével az útválasztók kérdezik le a csoporttagságot, a másikkal pedig az hosztok válaszolnak, egy-egy választ küldve minden csoporthoz, melynek tagjai. A válaszokból áll össze a csoporttagságokat tartalmazó táblázat.

## **23. fejezet – Az IPv6 protokoll**

### **Az IPv6 protokoll**

Az IPv6 protokoll tervezésének és megjelenésének fő szempontja az IPv4 protokoll lecserélése volt, amire az IPv4 ismert korlátai miatt volt szükség. Az első IPv6-tal kapcsolatos szabványok 1992. év végére készültek el, és egy evolúció során további hét változatból 1994-re született meg a ma IPv6-nak nevezett protokoll, amelyet 1994. november 17-én az Internet Engineering Steering Group (IESG) is elfogadott és felhasználásra javasolt. Az IPv6 protokoll leírását az RFC2460 dokumentum tartalmazza. 2011. június 8-ára ismert tartalomszolgáltatók, mint a Google, a Facebook és a Yahoo világméretű tesztnapot kezdeményeztek, "World IPv6 Day", azaz az IPv6-világnap néven. A protokoll tervezésekor nemcsak az IPv4 hibáit igyekeztek megszüntetni, hanem új szolgáltatásokat is bele kívántak építeni, amelyek gyorsabbá és az új felhasználói igényeknek jobban megfelelővé teszik.

Az IPv6 az IPv4 szerves folytatása. Sem a TCP, az UDP, a DNS, (ezek később kerülnek részletesen ismertetésre) sem az egyéb alkalmazói protokollok nem változnak, csupán maga az IP, amely viszont továbbra is megmarad megbízhatatlan szolgáltatást nyújtó datagram hálózatnak. Az egyetlen lényeges változás az architektúrában az, hogy az ARP funkciót többé nem külön definiáljuk minden kapcsolat típushoz, hanem maga az IP tartalmazza szomszéd felismerő protokoll (NDP – Neighbour Discovery Protocol) néven. Az egész protokollcsaládban általános lett a változó hosszúságú opciók beillesztésének lehetősége, melyek mindig egy hossz, egy típus és egy adatmezőből állnak, valamint minden esetben meghatározzák, hogy mit kell tenni a fel nem ismert opciókkal. Így egy későbbi bővítés esetén is biztosan tudhatjuk azt, hogy a régi berendezéseink hogyan viselkednek az új környezetben.

Az IPv6 protokoll megalkotásakor a következő célokat tüzték ki, és próbáltak elérni.

1. Támogatni kell az eddiginél (IPv4) sokkal több hosztot, lehetőleg milliárdos nagyságrendben. A kiosztható hosztok száma fontosabb, mint a hatékony címtartomány hozzárendelés.
2. Csökkenteni kell az útválasztók táblázatainak méretét.
3. A több hoszt miatt mindenképpen gyorsítani kell az útválasztókban a csomagok feldolgozását. Ehhez a protokoll egyszerűsítésére van szükség.
4. Támogatni kell az eddiginél (IPv4) sokkal jobb biztonságot, hitelesítést és titkosítást.
5. Kiemelt figyelmet kell fordítani a valós idejű szolgáltatásokra.
6. A többesküldés (Multicast) hatékonnyá tétele hatósugár megadásával.

7. Lehetővé kell tenni a hosztok barangolását anélkül, hogy IPv6 címük eközben megváltozna.
8. A protokollnak képesnek kell lenni a további fejlődésre.
9. Meg kell engedni, hogy a régi és az új protokoll akár évekig képes legyen egymás mellett létezni, egymás zavarása nélkül.

Az IPv6 legfontosabb jellemzői végül a következők lettek.

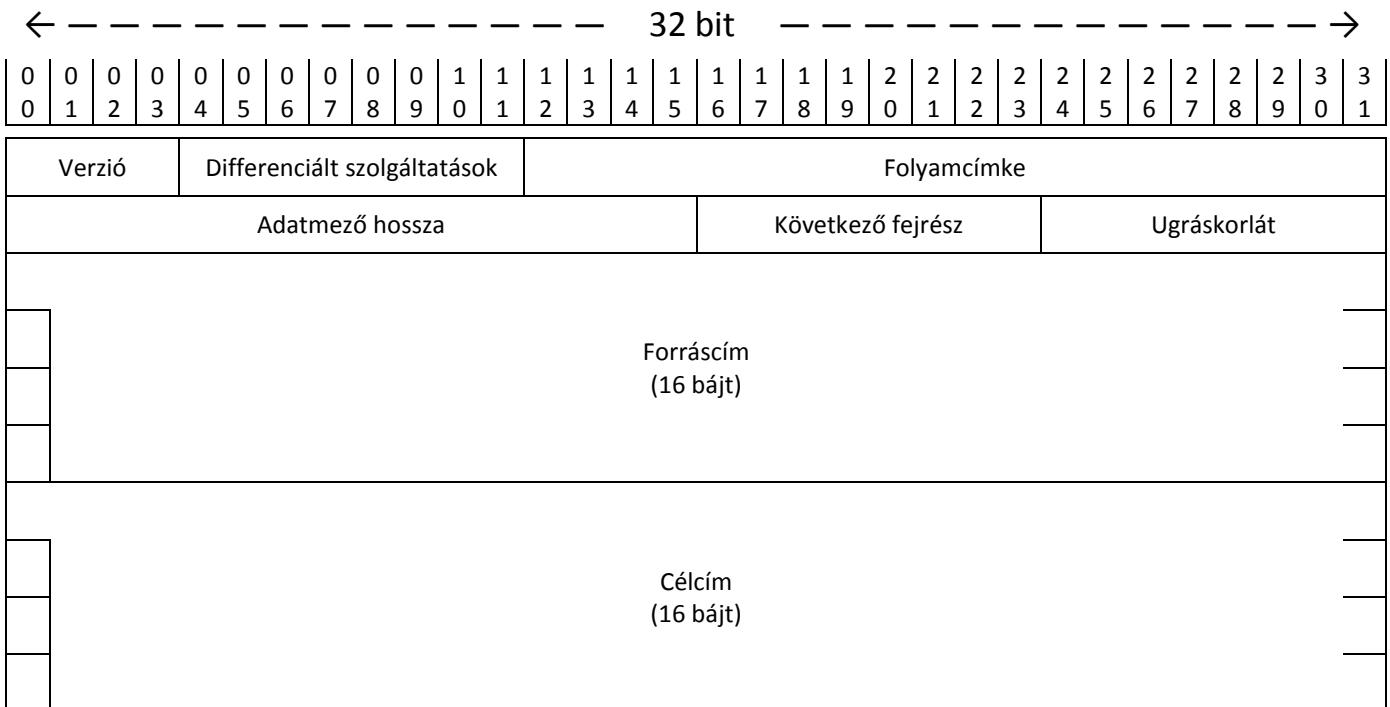
- Megnövelt, nagyobb címtartomány.
- Közvetlen végponti címezhetőség.
- Automatikus konfiguráció, vagyis a hosztok automatikus hálózati konfigurálását egy támogató rendszer végzi.
- Hálózati mobilitás, vagyis egy hálózati csatolóhoz egy időben több címet rendelhetünk. Ez hasonló a mobilszolgáltatók roaming szolgáltatásához.
- Titkosítás, azonosítás. Az IPv6 címzés szerves része az IPsec biztonsági protokoll, ez hálózati szinten nyújt lehetőséget arra, hogy a kommunikációban résztvevő felhasználók hitelesen azonosítsák egymást, és az egymás között zajló adatforgalmat titkosítsák egy biztonságos úgynevezett alagúton (Tunnel) keresztül anélkül, hogy az Internetről bárki le tudná hallgatni őket – ez persze az elmélet, a lehallgatás a valóságban nemzetbiztonsági érdek.
- Többszörös címezhetőség, szabványosított Multicast.

Az átállás az IPv4-ről IPv6-ra nem tud az egész Interneten egy időben lezajlani, ezért szükséges, hogy a két rendszer egymás mellett működhessen, akár az Interneten vagy akár egy hoszon belül is. Ezt az átmenetet a kompatibilis címek – az IPv4 címek egyszerűen átalakíthatók IPv6-címekké – és a különféle alagutak alkalmazása biztosítja. Használható egy kettős protokollcsomag (Dual Stack IP) nevű technika is, amely minden két protokollt egy időben támogatja. A két teljesen különálló hálózati alrendszer és a két különböző protokollverzió ebben az esetben nincs zavaró hatással egymásra.

A végfelhasználók szempontjából a legfontosabb változás az, hogy minden végfelhasználó fix IPv6 címhez juthat, azaz az IPv4-ben megismert NAT ezzel elvileg okafogyottá válik. (Nem kell üldözési mánia ahhoz, hogy belássuk, ez bizonyos szempontból hátrány is lehet...) Mivel a címtartományt az IPv6 esetében 128 bitre növelték, így több mint hárromezer milliárd ( $2^{128} = \text{kb. } 3,4 \cdot 10^{38}$ ) darab IPv6 cím osztható ki.

## Az IPv6 fejrésze (IPv6 Header)

Az IPv4 ismeretében lehetőségünk van a fejrészek összehasonlításra. Az IPv6 céljai között szerepelt a protokoll egyszerűsítése, ami már a fejrészen is észrevehető. Az IPv6 fejrész csak 8 mezőt tartalmaz (szemben az IPv4 13 mezőjével), és bár a címek 16 bájtosak, mégis csupán dupla olyan hosszú, mint a régi IPv4 fejrész. Maga az IPv6 cím továbbra is 32 bites szavakból áll, pontosan 10 ilyen szóból, tehát a hossza 320 bit, azaz 40 bájt.



Az IPv6 fejrész mezői a következők:

- Verzió: IPv6 esetében értéke 6.
- A differenciált szolgáltatások a második mező – eredetileg forgalmi osztály (Traffic Class) volt a mező neve, érdekes hogy már az IPv4 esetében ennél a mezőnél volt névcsere... Funkciója is az IPv4-ben megismert, azaz az első 2 bit az explicit torlódás jelzésére szolgál, az utolsó 6 bit pedig prioritási szinteket jelez.
- A folyamcímke (Flow Label) vagy folyam azonosító alkalmas az ugyanattól a feladó hosztól ugyanaddig a vevő hosztig futó logikailag egybetartozó csomagok megjelölésére. Így az egybe tartozó csomagoknak – folyamoknak – egyedi sávszélességbeli vagy késleltetési igénye lehet. Például egy TCP kapcsolat lehet egy folyam. A folyamcímke 0 értéke pedig azt jelzi, hogy a csomag nem tartozik egyetlen folyamhoz sem. A módszer lehetőséget teremt a datagram alapú hálózat rugalmasságának és a virtuálisáramkör alapú hálózat garanciáinak ötvözésére. A folyamcímke leírását az RFC1809 dokumentum tartalmazza.

- Az adatmező hossza mező mondja meg, hogy a fejrész 40 bájtja után mennyi adatbájt következik. Fontos változás, hogy az adatmező hosszában az IPv6 esetében a fejrész már nem számít bele a hosszba. Így a rendelkezésre álló 16 bit segítségével az adatmező hossza maximum 65535 bájt lehet.
- A következő fejrész mező arra utal, hogy az IPv6 fejrész után további fejrészek is következhetnek, melyek az IPv4-ben még opciók voltak. A kiegészítő fejrészek formátuma eltér az IPv6 fejrész formátumától, azaz specifikus mezőket tartalmaznak. Jelenleg 8 fajta ilyen további fejrész következhet. Ezen fejrészek sorrendje kötött. minden fejrész tartalmazza a következő fejrész típusát, kivéve az utolsót (ami a felsőbb rétegek fejrésze, mivel a sorrend is kötött). minden fejrészenben benne van, hogy mit kell tennie annak az útválasztónak, aki nem ismeri fel az egyes fejrészeket. A lehetőségek: dobja el, továbbítsa, vagy küldjön a feladónak ICMP üzenetet. A sorban elől a minden ugrás által feldolgozandó fejrész van, majd a közbülső- illetve a végpontok által feldolgozandó fejrészek következnek, végül pedig a felsőbb rétegekben feldolgozandó fejrész a zárja a sort.
  1. Hop-by-Hop Options header  
Egyetlen opciója a Jumbogram, azaz a 64kB-nál nagyobb datagramok támogatása, amit minden érintett útválasztónak fel kell tudnia dolgozni.
  2. Destination Options header (első előfordulás)  
Az egyetlen opció, mely kétszer is előfordulhat. Ebben a pozícióban a közbülső állomások számára tartalmaz adatokat.
  3. Routing header  
Az IPv4 szerinti forrás általi útválasztás (Source Routing Option) megfelelője, azaz a laza vagy szigorú útválasztást, útvonaljelölést tartalmazza.
  4. Fragment header  
A csomag tördelésével kapcsolatos információkat tartalmazza. Közbülső útválasztók részére tilthatja vagy engedélyezheti a csomagdarabolást.
  5. Authentication header  
Hitelesítési azaz autentikációs információkat tartalmaz. Segítségével ellenőrizhető, hogy valóban a küldő küldte-e, illetve, hogy történt-e változás az adatokban az átvitel során.
  6. Encrypted Security Payload header  
Célja a titkosítás, azaz hogy csak a valódi címzett tudja az adatokat elolvasni.
  7. Destination Options header (második előfordulás)  
Ebben a pozícióban csak a célállomás számára tartalmaz információkat.
  8. Upper Layer Header  
A célállomás felsőbb rétegei számára tartalmaz információkat.

- Az ugráskorlát mező funkciója az IPv4 TTL mező funkcióhoz hasonlít. Feladata, hogy a csomagok élettartamát ugrásonként csökkentse. A csökkentésnek az IPv6 esetében időhöz nem, csak az ugrások számához van köze – erre utal az elnevezés megváltoztatása is.
- A forráscím mező hossza 16 bájt, és a küldő hoszt címét tartalmazza.
- A célcím mező hossza 16 bájt, és a cél hoszt címét tartalmazza.

Amit mindenkor illik észrevenni – mint változást az IPv4-hez képest – az a darabolással kapcsolatos mezők és az ellenőrző összeg mező eltűnése.

A csomagdarabolás filozófiájában történt a változtatás, ugyanis az IPv6 esetében sokkal kevésbé preferált az csomagok út közbeni, azaz útválasztók általi darabolása. A preferált módszer az útvonal MTU-jának gyors felderítése után az eleve megfelelő méretű csomagok előállítása.

Az ellenőrző összeget – ami az IPv4-ben főleg a TTL-nek köszönhetően amúgy is bonyolult, ezáltal az egész átvitelre nézve lassító (erőforrás elvonó) hatású volt – azért lehetett elhagyni, mert egyrészt a hálózatok megbízhatósága a hálózatok evolúciója során sokat javult, illetve indokolatlanná is vált, hiszen a szállítási rétege amúgy is rendelkezik ellenőrző összeggel.

## Az IPv6 címek

Az IPv6 címzési rendszere nem tartalmaz az IPv4-ben megismert osztályokat. Egy 16 bájtos címnek pusztán a leírása is komoly feladat, ezért többféle ábrázolásmód van használatban.

- Leírhatjuk az IPv6 címet 8 db, egyenként 4 db hexadecimális szám kettősponttal elválasztott sorozataként:  
8000:0000:0000:0000:0123:4567:89AB:CDEF
- A hatalmas címtartomány sok esetben tartalmaz sok egymás utáni „0” értéket. A „0”-ák elhagyására két egyszerűsítés létezik. Egyszerűbben a felvezető „0”-kat hagyhatjuk el (egy címben ezt csak egy alkalommal engedi a szabvány), a másrészről a csak „0”-ákból álló csoportokat hagyhatjuk el:  
8000:0:0:123:4567:89AB:CDEF  
8000::123:4567:89AB:CDEF
- A régi (IPv4) címe írásmódja viszont maradt decimális, azaz a következő:  
::192.168.50.25
- A hálózati maszkot kizárolag a prefix értéke jelöli:  
8000::123:4567:89AB:CDEF/64

## A fontosabb nevezetes címtartományok, címcsoportok.

- Reserved by IETF      Az IETF által szabványba rögzítetten lefoglalt tartományok.  
<http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml>
- ::/128                  Unspecified Address  
A hosztok inicializálásakor használatos, végig nullákból áll.
- ::/0                    Alapértelmezett Unicast átjárócím.
- ::1/128                Loopback Address  
A visszacsatolásos tesztelés címtartománya.
- ::/96                   IPv4 Compatible IPv6 Address  
Ma már nincs érvényben, mert az IPv4 – IPv6 átmenetre született jobb (éppen a soron következő) megoldás.
- ::FFFF:0:0/96           Az IPv4 – IPv6 átmenetre fenntartott tartomány.
- 2000::/3                Global Unicast
- FC00::/7               Unique Local Unicast  
A privát címek tartománya, melyek olyan helyeken is használhatóak, ahol nincs hivatalosan kiosztott IPv6 prefix. (RFC4139)
- FE80::/10              Link-Local Unicast  
Olyan címek tartománya, melyek csak egy adott fizikai link-en, például szegmensen érvényesek. (RFC4862)
- FEC0::/10              Site-Local Unicast  
Olyan címek tartománya, melyek csak egy Site-on belül érvényesek. Nincs használatban, mert a Site fogalma sem pontosan meghatározott. (RFC3978)
- FF00::/8               Multicast  
Broadcast és Multicast céljára felhasználható címek tartománya.

## Az NDP protokoll

A korábban megismert ARP által megvalósított funkció (az IPv4 és a MAC címek összerendelése) az IPv6 esetében számos további funkcióval kiegészítve, szomszéd felismerő protokoll (NDP – Neighbour Discovery Protocol) néven működik.

A környezet felismerő protokoll a hálózatban elhelyezkedő aktív hálózati eszközök, entitások (hosztok, útválasztók) feltérképezésére szolgál. Információt szolgáltat az útválasztóknak, hosztnak a környezetükben található hálózati eszközök elérhetőségéről, címéről. Az NDP leírását az RFC4861 dokumentum tartalmazza.

A protokoll feladatai a következők:

- **Útválasztó felderítése (Router Discovery)**  
Ez egy eljárás a hosztok számára, arról, hogy hogyan deríthetik fel a hálózathoz kapcsolódó útválasztókat.
- **Cím előtagjának felderítése (Prefixr Discovery)**  
Ez egy eljárás a hosztok számára, arról, hogy képesek legyenek a cím előtag beállítására.
- **Paraméter felderítés (Parameter Discovery)**  
Ez egy eljárás a hosztok számára, arról, hogy honnan szerezhetnek tudomást az egyes paramétereiről, mint például a kapcsolatparamétereiről – például MTU, vagy az Internet paramétereiről – például, hogy hány pontot érinthet maximálisan az útja során a csomag (Hop Limit)
- **Automatikus címkonfigurálás (Address Autoconfiguration)**  
Ez egy eljárás a hosztok számára, arról, hogy hogyan végezhetik el az egyes kapcsolói pontok címének automatikus konfigurálását.
- **Cím feloldás (Address resolution)**  
Ez egy eljárás a hosztok számára, arról, hogy hogyan határozhatják meg az IP cím ismeretében a fizikai címet.
- **Következő ugrás meghatározása (Next-hop determination)**  
Ez egy algoritmus, amely leképezi a cél IP címét annak a csomóponti hosztnak a címére, ahova a csomagot továbbítani kell. Ez lehet maga a célpont vagy egy közbuli útválasztó.
- **Elérhetetlen szomszéd érzékelés (Neighbor Unreachability Detection)**  
Ez egy módszer arra, hogy a hosztok hogyan érzékelhetik, ha valamelyik, velük egy hálózaton levő hálózati eszköz elérhetetlenné válik.

- Cím ismétlődés érzékelése (Duplicate Address Detection)  
Ez egy eljárás a hosztok számára, arról, hogy hogyan érzékelhetik azt, hogy az általuk használni kívánt címet használja-e valamely más hálózati eszköz.
- Átirányítás (Redirect)  
Ez egy eljárás az útválasztók számára arról, hogy hogyan tudja értesíteni a hosztokat arról, hogy létezik az általa eddig használtan jobb (gazdaságosabb) csomagküldési útvonal is.

Az NDP protokoll, mivel az IPv6 protokoll része, ICMPv6 csomagokat felhasználva oldja meg a feladatait. Az IPv6 természetesen tartalmazza az összes IPv4 által használt ICMP üzenetet is. A környezet felmérés céljaira az ICMPv6 ötféle üzenetet használ.

- Útválasztó kérelmezés (Router Solicitation)
- Útválasztó hirdetés (Router Advertisements)
- Szomszéd kérelmezés (Neighbor Solicitation)
- Szomszéd hirdetés (Neighbor Advertisements)
- Átirányítás (Redirect)

## **24. fejezet – A szállítási réteg**

### **A szállítási réteg**

A rétegek közül a szállítási réteg az alsó három réteg logikai folytatásának tekinthető, hiszen ha egy hoszt üzenetet küld a másiknak, akkor az üzenet továbbítása előtt ezt általában csomagokra kell darabolni, ezeket a hálózati rétegnek átadva át kell vinni a hálózaton és a célhosznak átadni, ahol az üzenet összerakásra kerül. Az üzenetben leírt különféle típusú tevékenységet pedig végre kell hajtani.

A szállítási réteg feladata megbízható adatszállítás biztosítása a forráshoszt és a célhoszt között, függetlenül az alatta lévő rétegek kialakításától. A szállítási réteg feladata ellátásához a hálózati réteg által nyújtott szolgálatokra támaszkodik. A feladat itt már a tényleges hoszt-hoszt kapcsolat hibamentes megvalósítása. A szállítási réteg tehát már valódi két végpont között réteg, ami azt jelenti, hogy itt a forráshoszt és a célhoszt egymással kommunikál, míg az alsóbb rétegeknél ez nem igaz, ott a hosztok a szomszédjukkal (azaz azzal az aktív hálózati eszközzel, amihez közvetlenül kapcsolódnak) folytatnak párbeszédet. A használt protokollok sok esetben hasonlítanak az adatkapcsolati réteg protokolljaira, de itt az IMP-ket összekötő fizikai csatornát, a két hoszt között található teljes hálózat jelenti.

A szállítási réteg adategysége a szegmens (Segment), illetve használatos még a TPDU (Transport Protocol Data Unit / Szállítási Protokoll Adategység) elnevezés is. Az átvitel során a szegmensek, melyeket szállítási réteg használ csomagokba ágyazódnak, melyeket a hálózati réteg használ; a csomagok tartalma pedig keretekben folytatja útját az adatkapcsolati rétegen.

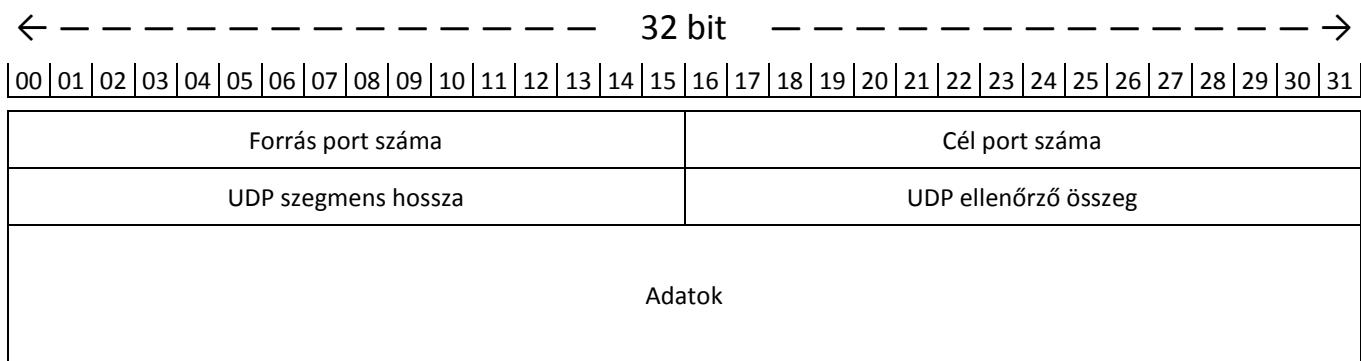
Az Internet két fő protokollt használ a szállítási rétegen, ezek az UDP és a TCP. Fontos megemlíteni továbbá a RTP protokollt, mely a valósidejű átvitelt biztosítja.

### **Az UDP protokoll**

A felhasználói datagram protokoll (UDP – User Datagram Protocol) egy nem megbízható, összeköttetés nélküli protokoll. Az UDP protokoll leírását az RFC768 dokumentum tartalmazza. Elsősorban olyan egylövetű, kliens-szerver típusú kérdés-válasz alkalmazásokban terjedt el, ahol a gyors válasz sokkal fontosabb, mint a pontos, megbízható válasz, ugyanis az UDP nem garantálja a csomagok megérkezését. UDP-t használnak például az audio-video streaming alkalmazások, a DNS (Domain Name Server), az SNMP (Simple Network Management Protocol) és a DHCP (Dynamic Host Configuration Protocol) is.

Az UDP működési mechanizmusa a következő. Az UDP az információját egy IP csomagba helyezi el, ellenőrző összeget számol hozzá és feladja a csomagot. Így a kézbesítést nem garantálja, de a hibás kézbesítést észlelhetővé teszi. Ha a kérdés, vagy a válasz az átvitel során elveszik, a hiba egyszerű újrakérdezéssel megoldható, amennyiben az adott információ időben még releváns. Az UDP – mivel egy szállítási protokoll az jellemzően az operációs rendszer része – alkalmazásnak tekinthető. Az UDP egyszerűsége miatt kíméletesen bánik a hálózati erőforrásokkal.

Az UDP adatszerkezete is fejrészből valamint az adatokból áll. A fejrész 32 bites és minden össze 8 bájt hosszú, az adatok maximális méretét pedig az IPv4 csomag törzs részének maximális hossza, mínusz az UDP fejrész hossza határozza meg, tehát  $65515 - 8 = 65507$  bájt.



Az UDP adatszerkezetében egy új fogalom jelent meg, a port (Port), amit nagyrétkén kapu néven is említi néhány forrás. A portok a hosztokon belül futó alkalmazások megkülönböztetésére szolgálnak. Ez azt jelenti, hogy bizonyos alkalmazások a számukra meghatározott portokat használják a kommunikációra.

Az UDP szegmens adatszerkezetének mezői a következők:

- Forrás port száma (Source Port)

A forrás címe a forrás hosztot azonosítja, a forrás port száma pedig a forráson futó alkalmazást. Válaszküldés esetén ebből a forrás portból lesz a címpor, azaz a forrás port ismerte nélkül nem lenne lehetőség válasz eljuttatására a forrás megfelelő alkalmazáshoz.

- Cél port száma (Destination Port)

A cél hoszon futó alkalmazást azonosítja.

- UDP szegmens hossza (UDP Total Length)

A szegmens hossza egyenlő a 8 bájtos fejrész és az adatmező együttes hosszával. A hossz értéke bájtonként értelmezendő, azaz 8 bájt (ebben az esetben csak a fejlécet tartalmazza, adattartalom nélkül) és 65515 bájt között lehet.

- UDP ellenőrző összeg (UDP Checksum)  
Az ellenőrző összeg használata nem kötelező, de növeli az adatbiztonságot.
- Adatok (Application Data / Message)

A portszámokat tartalmazó mezők, azaz a portok ismerete nélkül a szállítási réteg nem képes a szegmenseket a megfelelő alkalmazáshoz eljuttatni. Az, hogy az UDP a portokat már a fejrészben használja, jelentős idő- és teljesítmény előnyt jelent egy hagyományos IP csomag feldolgozhatóságához képest. A szegmensek IP csomagba történő beágyazása tehát megoldja az adatok megfelelő alkalmazáshoz történő gyors eljuttatását.

A port tartományokat eredetileg az RFC1700 dokumentum tartalmazta, de tekintettel a hatalmas tartomány szinte folyamatos változásaira (főleg a magasabb portszámok esetében), az aktuálisan hivatalos állapotot célszerű az Interneten megtekinteni itt:

<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>

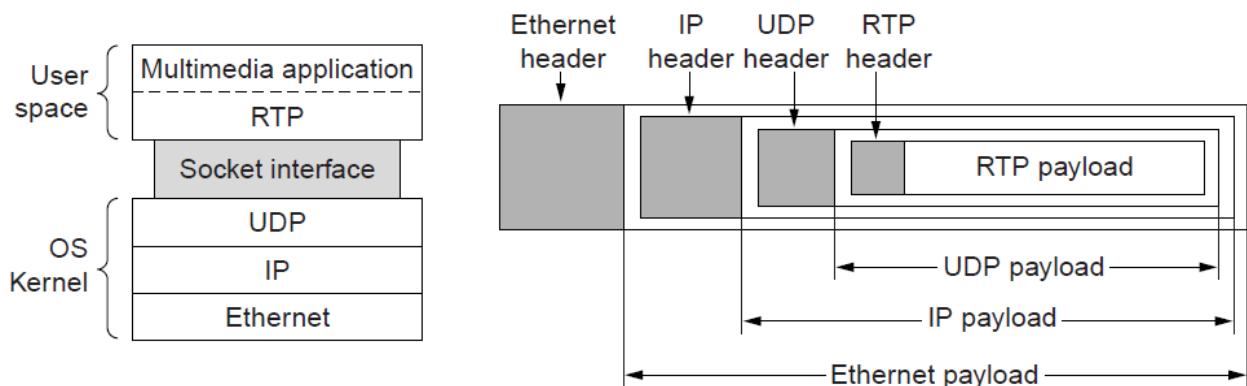
A port tartományok rövid áttekintése:

- 1-1023 Közismert portok (Well-known) [csak a leggyakoribbak vannak felsorolva]
  - 20, 21    FTP                állományátvitel
  - 22        SSH                távoli bejelentkezés, telnet
  - 25        SMTP              e-mail küldés
  - 53        DNS                névszolgáltatás
  - 80        HTTP              web elérés
  - 110      POP3              e-mail hozzáférés
  - 143      IMAP              e-mail hozzáférés
  - 443      HTTPS             biztonságos web elérés
  - 543      RTSP              médialejátszó vezérlés
  - 631      IPP               nyomtatómegosztás
- 1024-49151 Regisztrált portok (Registered)
  - alkalmazás specifikus portok (NFS, BitTorrent, Antivirus, hálózatos játék, stb.)
- 49152-65535 Ideiglenes portok (Temporary / Dynamic)
  - nem fixen kiosztott tartomány, kliens-szerver és ideiglenes egyedi alkalmazásokhoz

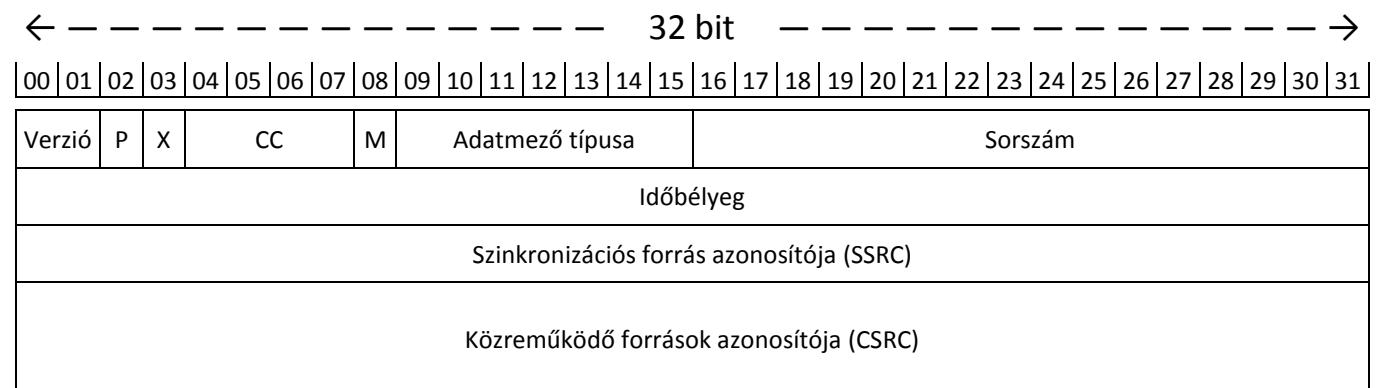
## Az RTP protokoll

A valósidejű átviteli protokoll, az RTP (Real-time Transport Protocol) az UDP protokollra épülve, és így kihasználva annak multiplexelő és hibadetektáló funkcióit – a szegmensek valósidejű továbbítását biztosító megoldás. Az RTP protokoll leírását az RFC1889 dokumentum tartalmazza. Az RTP protokoll nem közvetlenül foglalkozik a szegmensek gyors továbbításával, vagy QoS biztosításával, ezeket az alatta levő rétegekre bízza. Az RTP feladata csupán a valósidejű kommunikáció menedzsmentje.

Az RTP adatfolyamokat nem az operációs rendszer, hanem maga a valósidejű alkalmazás darabolja szegmensekre, tehát az RTP nem az operációs rendszer része, hanem az alkalmazásoké. A szegmensek belső formátuma a fejléctől eltekintve az alkalmazásra van bízva, így maga az RTP csak egy keretprotokoll, konkrét alkalmazásához ki kell egészíteni a használt szegmensek és csomagok típusával, a típuskódok és az egyes típusú adatok kódolásának leírásával. Mindezekkel az RTP nem foglalkozik, csupán a szegmensek átvitelével, a szinkronizációs információ előállításával és kezelésével, valamint a kapcsolat minőségének felügyelésével. Az RTP helyzetét az egymásra épülő protokollokban az ábra szemlélteti.



Az RTP fejrésze 32 bites. Fixen 3 db 32 bites szóból, azaz 12 bájtból, és néhány kiterjesztésből áll.



Az RTP fejrészének mezői a következők:

- **Verzió (Version)**  
Éz a mező csak 2 bites, értéke jelenleg 2, azaz további két verziót képes jelezni.
- **P (Padding)**  
Értéke 1, ha maga a TPDU nincs teljesen kitöltve adattartalommal, azaz a formai követelmények miatt ki kellett egészíteni 4 bajtra, vagy annak valamelyik többszörösére. Az utolsó, a kiegészítő bajt tartalmazza, hogy hány bajtot kell figyelmen kívül hagyni, azaz hogy mennyi volt a töltelék, a kitöltés.
- **X (Extension)**  
Értéke 1, ha a fejrész kiegészítő fejrésszel rendelkezik.
- **CC (CSRC Count)**  
Az CSRC azonosítók számát tartalmazza.
- **M (Marker)**  
Alkalmazásfüggő jelölőbit, egy TPDU folyam szignifikáns eseményeit jelölheti, például a kép- vagy hangkeret elejét vagy végét.
- **Adatmező típusa (Payload Type)**  
A használt kódolási algoritmus azonosítója. Például: mp3, mkv, stb.
- **Sorszám (Sequence Number)**  
A TPDU-k számolására szolgál, segítségével detektálható az adatvesztés.
- **Időbélyeg (Timestamp)**  
A valósidejű átvitelt segítő mező. Értéke a vevő oldali időzítési szórás (Jitter) csökkentésével segíti elő a multimédiás tartalom – csomagok érkezési idejétől független – folyamatos lejátszását.
- **SSRC (Synchronization Source Identifier)**  
A szinkronizációs forrás azonosítója az RTP folyam forrását azonosítja.
- **CSRC (Contributing Source Identifier)**  
Az RTP által létrehozott kombinált (mixelt) folyam közreműködő komponenseit azonosítja.

Az RTP protokoll szerepe tehát leginkább a valósidejű multimédia információk továbbításának végrehajtásában van, UDP protokoll igénybe vételével, az IP protokollon, azaz jellemzően az Interneten.

## A TCP protokoll

Az átvitelvezérlő protokoll (TCP – Transmission Control Protocol), egy megbízható összeköttetés alapú protokoll. A TCP protokoll első verziójának a leírását 1981-ből az RFC793 dokumentum tartalmazza. Evolúciója során az RFC1122 (hibajavítások), az RFC1323 (teljesítőképesség növelése), az RFC2018 (szelektív nyugtázás), az RFC2581 (torlódáskezelés), az RFC2873 (fejrész továbbfejlesztése), az RFC2988 (időzítők), az RFC3168 (explicit torlódáskezelés) és az RFC4614 (a jelenlegi verzió) dokumentumok is a TCP-vel foglalkoztak. Felépítése jóval bonyolultabb, mint az UDP felépítése. Feladata az, hogy hibamentes bájtos átvitelt biztosítson bármely két hoszt között az Interneten, az egyébként megbízhatatlan összekapcsolt hálózaton – ezért ellenállónak kell lennie a lehetséges meghibásodásokkal szemben. A TCP átvitele tehát egy bájtfolyam, a rendszer az üzenethatárokkal nem foglalkozik. A TCP forgalomszabályozást is végez annak érdekében, hogy egy gyors forráshoszt csak annyi üzenetet küldjön egy lassabb célhosztnak, amennyit az fogadni képes. A TCP mivel egy szállítási protokoll az jellemzően az operációs rendszer, azon belül is leggyakrabban a kernel része.

A TCP-t használó hosztok az adatokat szegmensekben cserélik egymás között. A szegmensméret korlátja csak az, hogy a TCP fejrésznek és a szegmensnek egyaránt be kell férnie az IPv4 csomag törzs részébe. A szegmens maximális hossza az IPv4 törzs részénél a TCP fejrész hossza, azaz  $65515 - 20 = 65495$  bájt.

A TCP fejrésze 32 bites. Fixen 5 db 32 bites szóból, azaz 20 bájtból, és néhány opcionális mezőből áll.

32 bit																					
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31																					
Forrás port száma										Cél port száma											
Sorszám																					
Nyugtaszám																					
TPC fejrész hossza		C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Ablakméret											
TCP ellenőrző összeg										Sürgősségi mutató											
Opciók (0 vagy több 32 bites szó)																					
Adatok																					

A TCP szegmens adatszerkezetének mezői a következők:

- Forrás port száma (Source Port)

A forrás címe a forrás hosztot azonosítja, a forrás port száma pedig a forráson futó alkalmazást. Válaszküldés esetén ebből a forrás portból lesz a címport, azaz a forrás port ismerte nélkül nem lenne lehetőség válasz eljuttatására a forrás megfelelő alkalmazáshoz.

- Cél port száma (Destination Port)

A cél hoszton futó alkalmazást azonosítja.

- Sorszám (Sequence Number)

A mező tartalma egy 32 bites, átforduló szám, amelynek a kezdeti, véletlenszerűen választott sorszámrétkhez képesti eltolása megegyezik az adott szegmens első adatbájtjának az bájtfolyamban elfoglalt pozíójával. A vett szegmens sorszáma alapján dönt a vevő arról, hogy ez beleillik-e a vételi ablakába vagy el kell-e dobni. A 32 bittel nagyából 4 GB adat címezhető meg.

- Nyugtaszám (Acknowledgement Number)

A mező azt a következő adatbájt sorszámát tartalmazza, amire a hoszt éppen várakozik. Az ACK flag azt jelzi, hogy ez a mező érvényes adatot tartalmaz-e.

- TCP fejrész hossza (TCP Header Length)

Azt mutatja meg, hogy a TCP fejrész hány darab 32 bites szót tartalmaz. Az opció mező változó hossza miatt, erre az információra feltétlenül szükség van.

- Használaton kívüli 4 bit (Not In Use)

- 8 db egy bites jelző mező (Flag)

A jelzőbőlök a TCP kapcsolat adataira, állapotaira, állapotváltozásaira vonatkozó kérésekben és jelzésében, illetve a torlódáskezelésben játszanak szerepet.

- CWR (Congestion Windows Reduced)

- ECE (ECN Echo)

- URG (Urgent Pointer)

- ACK (Acknowledgment)

- PSH (Push Function)

- RST (Reset the Connection)

- SYN (Syncronize Sequence Number)

- FIN (Final, No more Data from the Sender)

- Ablakméret (Window)

A forgalomszabályzásnál használt változó méretű csúszóablak méretét jelzi.

- TCP ellenőrző összeg (Checksum)

Az UDP-vel ellentétben az ellenőrző összeg használata TCP esetén kötelező.

- **Sürgősségi mutató (Urgent Pointer)**  
Ha az URG flag értéke 1, akkor ez a 16 bit az aktuális sorszámtól számolva kijelöli a sürgős adatok kezdetét.
- **Opciók (Options)**  
Amennyiben a hosztokon futó alkalmazások igénylik, a TCP esetében lehetőség van opciók egyedi használatára. Leggyakrabban a puffer méret egyeztetésre használják a hosztok vonatkozó alkalmazásai.
- **Adatok (Data)**

A portok szerepről az UDP protokoll tárgyalásakor már volt szó. Annak érdekében, hogy a kapcsolatban részt vevő alkalmazást azonosítani lehessen, illetve, hogy egy hoszt egyszerre több élő TCP kapcsolattal rendelkezhessen, a TCP adatot hordozó IP csomagokban nemcsak a céloszt címet kell megadni, hanem a TCP port számát is. A TCP összeköttetés duplex pont-pont összeköttetés, azaz a forgalom egyszerre két irányba halad, és az adatszórás (Broadcast) valamint a többesküldés (Multicast) nem támogatott. A tehát TCP egy kapcsolat-orientált protokoll, fő feladata egy megbízható, és biztonságos kapcsolat kiépítése és fenntartása két folyamat között. A megvalósítás menetét alapvetően három részre bonthatjuk:

1. Létrejön a megbízható kapcsolat két állomás között.
2. Megkezdődik a tényleges adatátvitel.
3. A kapcsolat lezárása, és a számára elkülönített erőforrások felszabadítása.

A protokoll a hibamentes átvitelhez az úgynevezett pozitív nyugtázás újraküldéssel (Positive Acknowledgement with Retransmission) eljárást használja. A TCP kapcsolatok egyes lépéseiit állapotoknak nevezzük. A kapcsolat az élettartama alatt különböző állapotváltozásokon megy keresztül:

- **CLOSED**  
Ez az alapértelmezett állapot, amelyből a kapcsolat kiépítésének folyamata indul. Elméleti állapot, a hosztok között nincs élő, létező kapcsolat (vagyis még nem jött létre, vagy már lezárult).
- **LISTEN**  
A hoszt (általában a szerver) szinkronizálási kérésre várakozik (SYN), saját SYN üzenetét még nem küldte el.
- **SYN-SENT**  
A hoszt (általában a kliens) elküldte a SYN üzenetet, és várakozik a másik hosztól (általában a szervertől).

- **SYN-RECEIVED**  
Kapcsolódási kérés (SYN) elküldve és fogadva is, várakozás a másik hoszt általi nyugtázás beérkezésére (ACK).
- **ESTABLISHED**  
A létrejött TCP kapcsolat stabil állapota. Miután minden hoszt ebbe az állapotba kerül, megkezdődhet az adatok átvitele, ami addig folytatódhat, amíg valamelyik hoszt a kapcsolat lezárását nem kezdeményezi.
- **CLOSE-WAIT**  
Az egyik hoszt kapcsolatbontási kérést (FIN) kapott a másik hosztól. Várakozik a helyi alkalmazás nyugtázására, mielőtt elküldné a megfelelő válaszüzenetet.
- **LAST-ACK**  
A hoszt már fogadott és nyugtázott egy kapcsolatbontási kérést, el is küldte a saját FIN üzenetét, és várakozik a másik hoszt ezen kérésre érkező nyugtájára (ACK).
- **FIN-WAIT-1**  
Várakozás az elküldött FIN üzenet nyugtázására, vagy a kapcsolatbontási kérés érkezésére másik hosztól.
- **FIN-WAIT-2**  
Megérkezett a nyugta az elküldött kapcsolatbontási üzenetre, várakozás a másik hoszt FIN üzenetére.
- **CLOSING**  
A hoszt megkapta a másik hoszt FIN üzenetét, és nyugtázta azt, de a saját FIN üzenetére nyugtát még nem kapott.
- **TIME-WAIT**  
A kapcsolatbontási kérést és a nyugtát (FIN, ACK) a hoszt megkapta és kiküldte, a kapcsolat lezárult. Egy rövid ideig várakozik még, hogy biztosítsa azt, hogy a másik hoszt is megkapja a nyugtát, és hogy ne legyen átfedés az újonnan létrejövő kapcsolatokkal.

## 25. fejezet – Az alkalmazási réteg

### Az alkalmazási réteg

Az alkalmazási réteg alatt elhelyezkedő rétegek a felhasználó számára láthatatlan, érzékelhetetlen módon végzik a dolgukat, így a felhasználó már csak az általa az alkalmazási rétegben futtatott programok közvetlen eredményét, hatását tapasztalja. Az alkalmazási réteg széles körben igényelt és használt szolgáltatásokat tartalmaz, például fájlok hosztok közötti másolása, elektronikus levelezés, web böngészés, távoli terminálok elérése. Az alkalmazási réteg tartalmazza az összes magasabb szintű protokollt.

### A DNS

A körztnév kezelő rendszer (DNS – Domain Name Service) teszi lehetővé azt, hogy az Interneten, a világhálón képesek legyünk egy általunk keresett oldalt csupán a neve alapján, azaz az IP címe ismerete nélkül is megtalálni. A gyakorlatban ez azt jelenti, hogy például a [www.index.hu](http://www.index.hu) oldalt egyszerűen a névnek a web böngészőbe történő begépelésével is elérhetjük, anélkül, hogy tisztába lennénk azzal, hogy a név mögött a 217.20.130.99 IPv4 cím áll. Természetesen a web böngészőbe az IP címet is beírhatjuk, de lássuk be, hogy ez a kevésbé életszerű megoldás.

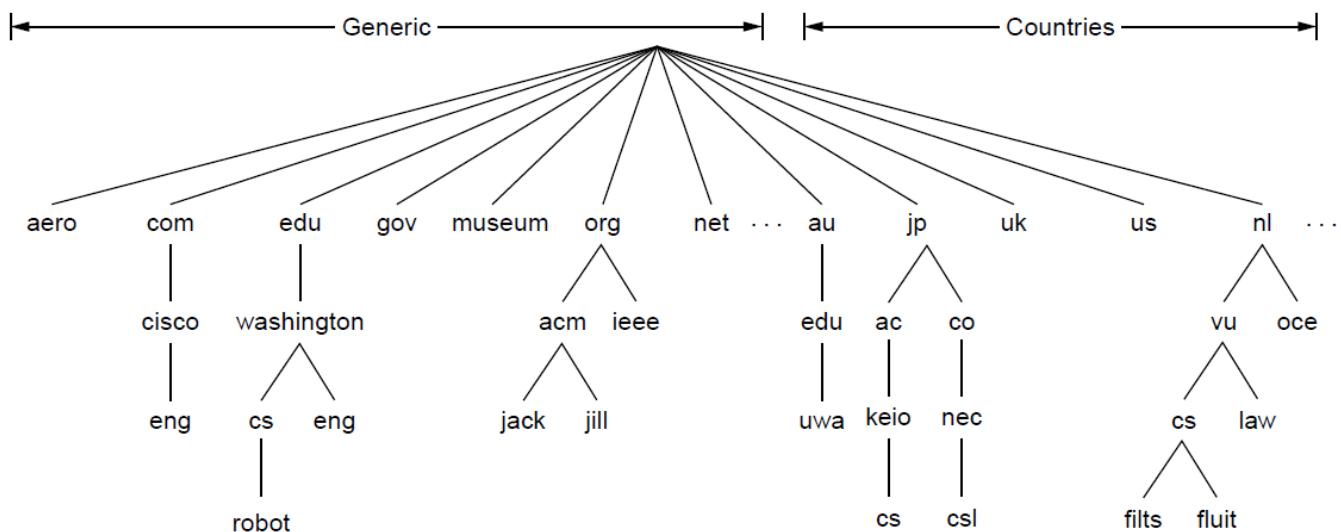
Az általános megoldás a webcím, mint egységes erőforrás azonosító, azaz az URL (Uniform Research Locator) szerinti hivatkozás. Az URL megalkotója Tim Berners-Lee 1991-ben, leírását az RFC1738 dokumentum tartalmazza. Az első weblap a CERN oldala a <http://info.cern.ch:80> volt.

Az URL részeit képezi a használt protokoll neve, ezt követi „//” után a tartomány neve valamint opcionálisan „:” után a használt port száma. Lehetőség van továbbá egy „/” jel után további elérési út megadására is. Az URL által bevezetett webcím alapvető újítás volt, hiszen egy címben össze tudta foglalni a keresett dokumentum megtalálásához szükséges négy információt.

- a protokollt, amit a célhosztval való kommunikációhoz használunk
- a szóban forgó hoszt vagy tartomány nevét
- a hálózati port számát, amin az igényelt szolgáltatás elérhető a célhoszon
- a fájlhoz vezető elérési utat a célhoszon belül

A DNS 1983-ban, érdekes módon időben sokkal az URL előtt jelent meg, leírását az RFC1034, RFC1035 és RFC2181 dokumentumok tartalmazzák. Az URL tulajdonképpen a DNS szolgáltatásait teszi a felhasználó számára még produktívabbá.

A DNS-ben a domén (Domain) nevek kiosztásának és az IP-címek hozzárendelésének a felelősséget oly módon delegálják, hogy minden tartományhoz mérvadó névkiszolgáló szerver (Authoritative Name Server) tartozik. A mérvadó névkiszolgáló szerverek felelősek a saját doménjeikért. Ezt a felelősséget tovább delegálhatják, így az al-doménekért más névkiszolgáló (azaz DNS szerver) felelhet. Ez a mechanizmus áll a DNS elosztott és hibatűrő működése mögött, és ezért nem szükséges egyetlen központi címtárat fenntartani és állandóan frissíteni. A domén nevek és az IP címek közötti kapcsolat biztosítása, a névfeloldás dinamikus kell, hogy legyen, hiszen a rendszer nem statikus.



A tartomány alapú névrendszerben egyéb információk is tárolódnak, például egy adott internetes tartomány számára e-mailt fogadó levelező kisszolgálók listája. Az egész világot behálózó, elosztott, kulcsszó-alapú átirányítási szolgáltatásként a DNS az Internet funkcionalitásának alapvető fontosságú eleme.

A DNS névtérben, a legáltalánosabban használt „.com” (Commercial) mellett számos elsődleges illetve ország specifikus körzet létezik. Ezeket az ISO3166 szabvány foglalja össze. Leggyakrabban ezekkel a végződésekkel találkozhatunk:

• com	commercial	kereskedelemi
• edu	education	oktatási
• gov	government	kormányzati
• int	international	nemzetközi
• mil	military	katonai
• net	network	hálózati
• org	organization	nonprofit szervezet
• biz	business	üzleti
• info	information	információ

A domének egy része üzleti lehetőségeket is rejt, hiszen például Tuvalu (egy kis óceániai állam) „.tv” doménje több TV társaság érdeklődését is felkeltette. Hozzáállás kérdése, hogy ötletesnek vagy a „üzleti megfontolásból előrelátó” szándékúnak nevezzük az úgynévezett domén hack (Domain Hack) oldalak bejegyzését, például: [instagr.am](http://instagr.am) (am = Örményország), [goo.gl](http://goo.gl) (gl = Grönland), [gondol.at](http://gondol.at) (at = Ausztria), [favor.it](http://favor.it) (it = Olaszország), stb.

Elvi szinten akár egy darab szerver is elláthatná a névfeloldás feladatát és a teljes DNS adatbázis tárolását, de gyakorlatilag ez így megoldhatatlan, hiszen akkora terhelésnek lenne ez a szerver kitéve, amit lehetetlen lenne kiszolgálni, illetve a rendszer hibatűrése sem lenne így biztosítva. Az ábra a névhierarchia csúcsán álló gyökérnévszervereket (Root Name Server) és a hozzájuk kapcsolódó kiszolgáló rendszereket mutatja. A rendszer áttekintéséhez célszerű felkeresni a <http://root-servers.org> oldalt, ahol a térkép tovább nagyítható. A 13 darab gyökérnévszervert „A”-tól „M”-ig jelöljük, elnevezésük pedig a <http://a.root-servers.net>-től kezdődik és a <http://m.root-servers.net>-ig tart. A gyökérnévszerverek IPv4 és IPv6 címére minden névszervernek szüksége van.



A feladatot tehát több szerver között kell szétosztani, ezért maga a DNS névtér is egymást nem átlapoló zónákra (Zone) van elosztva. A DNS névtér – mint fa struktúra – egyik egyben delegált ágát, melyért egy kitüntetett szerver felelős, zónának nevezzük. Az, hogy egy zónáért egy szerver – egy láttató- vagy autoritatív szerver – felelős, egyben arra is utal, további „nem felelős” szerverek is találhatóak a zónában, melyek a szerepe a szolgáltatás megfelelő színvonalú elérésének biztosításában van.

A DNS kliens oldali szoftvermodulja a DNS-névfeloldó (DNS Resolver). A névfeloldó felelős a kliensen – azaz egy hoszton – egy erőforrás teljes névfeloldásáért, egy tartománynév IP-címre fordításáért, illetve a nem teljes névnek az alapértelmezett utótaggal (Fully Qualified Domain Name) történő kiegészítéséért.

A DNS-lekérés lehet rekurzív vagy nem rekurzív kérés.

- Nem rekurzív kérés esetén a DNS szerver vagy olyan tartományról szolgáltat információt, amelyre nézve saját maga láttató- vagy autoritatív szerver. További lehetőség, hogy a DNS szerver más kiszolgálók lekérdezése nélküli, részleges eredményt ad csupán.
- Rekurzív kérés esetén a DNS szerver teljes mértékben megválaszolja a kérést vagy hibajelzést ad – szükség esetén akár további DNS szerverek lekérdezésével. A DNS szerverekkel szemben nem általános követelmény vagy elvárás a rekurzív lekérdezések támogatása.

A DNS-névfeloldó vagy a DNS-névfeloldó nevében eljáró DNS szerver a lekérdezés fejléceiben a megfelelő bitek beállításával jelzi, hogy a lekérdezés rekurzív lesz-e.

Maga a DNS-névfeloldó is dolgozhat rekurzív vagy iteratív üzemmódban.

- Rekurzív üzemmódban a teljes munkát a beállított rekurzív DNS szerver végzi.
- Iteratív módban, a nem rekurzív kérésre a DNS szervertől visszakapott válasz kétféle lehet. Egyik lehetőség, hogy a keresett információ érkezik meg a kiszolgálótól, másik lehetőség az, hogy egy másik DNS szerverre való hivatkozás érkezik meg válaszként. Utóbbi esetben ezt a DNS szervert kell a kéréssel megkeresni. Így a DNS-névfeloldó lépésről lépésre haladva annyi kérdést tesz fel az adott névszervereknek, amennyi az információ beszerzéséhez szükséges.

A DNS-névfeloldó az így kapott választ átadja a programnak, amely az információt kérte, jellemzően például a web böngészőnek. Az egyszerű felhasználói DNS-névfeloldók általában mindenki szükséges egy elérhető rekurzív DNS szerver. A DSN szervereknek általában saját, iteratívan működő DNS-névfeloldójuk van.

Az egyes szolgáltatók különböző DNS szerverei dinamikusan adatot cserélnek egymással. A DNS szerverek funkciójukat tekintve a következő feladatokat láthatják el, illetve a következő (Root alatti) hierarchia szinteken helyezkedhetnek el.

- Láttató- vagy autoritatív (Authoritative) szerver

Ezek azok a DNS szerverek, melyeknek az is feladata, hogy bizonyos neveket ők mutassanak meg mások számára. Az egy zónáért felelős DNS szerverek között van egy kitüntetett, amelyet az adminisztrátor konfigurál. A többi szerver ezt a zónát tükrözi. A kitüntetett szerverre használt kifejezés az elsődleges (Primary), a tükröző szerverekre pedig a másodlagos (Secondary) elnevezés. A gyakorlatban a kód úgy működik, hogy a többi szerver egy-egy zóna autoritatív szerverei közül igyekszik azzal kommunikálni, amelyik a leggyorsabban válaszol. Ennek érdekében a szerverek egy olyan algoritmust használnak, hogy kezdetben minden egyik névszervert megkérdezik, majd megmérik a válaszidőt, és azután azt a szervert preferálják, amelyik gyorsabban válaszolt.

- Caching only szerver

A DNS szerverek egy része nem autoritás semmilyen névre – azaz nincs úgynevezett láttató feladata – hanem csak arra szolgál, hogy feloldja a neveket a kliensek számára. Ezek a Caching Only, azaz csak cache-elő DNS szerverek. Célszerű minden lokális hálózaton legalább egy ilyen DNS szervert működtetni, mivel ezzel a módszerrel a hálózat terhelése csökkenhető, a névfeloldás pedig gyorsítható.

- Forwarder szerver

A továbbítás (Forwarding) lényege az, hogy a helyi forwarder szerver csak továbbítja a kéréseket a szolgáltató (ISP) DNS szervere felé, illetve a válaszokat a kliensek felé.

- Slave szerver

Az olyan DNS szervert, ami csak forwarder szervert (esetleg többet is) használ a nevek feloldására, slave szervernek nevezük. Slave szerverre van szükség például tűzfal mögött, ahol a szervernek arra nincs módja, hogy közvetlenül jussanak ki az Internetre.

Ne feledkezzünk meg arról sem, hogy a DNS szervereket (helyezkedjenek el bárhol is a fenti hierarchiában) a gyakorlatban nagyszámú hoszt folyamatos névfeloldási kérésekkel tereli. A kéréseket és a válaszokat UDP üzenetek hordozzák, az 53-as szolgáltatási porton keresztül. Mindkét esetben tehát a legkívánatosabb cél a kis késleltetés, vagyis a gyorsaság. Ennek érdekében a szerverek a fenti módokon gyorstárazást (Cache) használnak. A gyorstárba mindenkor a legutóbb használt elemek kerülnek egy élettartamjelzővel együtt, hiszen az adatok érvényessége a rendszer lehetséges dinamikus változásai miatt véges.

Az eredeti koncepcióban a biztonság kérdése még nem szerepelt, de a gyakorlat megkövetelte azt. A felhasználói bizalom ez esetben például azt jelenti, hogy a felhasználó biztosan a saját bankjának a honlapját éri el a bank URL címének böngészőbe történő begépelése után, nem pedig egy olyan rosszindulatú csalás áldozata lesz ahol egy, a saját bankjának a honlapjával szinte 100%-ban megegyező, de csak adatlopásra szakosodott tartalmat talál, ahol első lépésében hozzájutnak a belépési adataihoz, második lépésben pedig akár a pénzéhez is. A DNS információk meghamisításával, DNS megtévesztéssel (DNS Spoofing) ehhez hasonló károk okozhatók, illetve a támadó lehallgathatja vagy meghamisíthatja az üzeneteket.

A DNS-biztonság (DNSSEC – Domain Name System Security Extensions) az Internet egyik alapvető, névfeloldást biztosító szolgáltatásának, a DNS-nek, az IETF által specifikált, biztonsági célú kiterjesztése. Első verziójának leírását az RFC2065 (1997), 1998-as módosítását az RFC2535, 2005-ben történt módosítását pedig az RFC4033, RFC4034 és RFC4035 dokumentumok tartalmazzák. A megoldás bonyolultsága többek közt abban áll, hogy visszamenőlegesen, az a régebbi eszközökön is alkalmazhatónak kell lennie. Ez utóbbi szempont teljes körűen sajnos a mai napig sem megoldott. A DNSSEC célja a kliensek számára a DNS-ben található adatok integritásának és autentikusságának biztosítása, illetve egy rekord nem létezésének autentikus bizonyítása – nem célja viszont sem a tényleges elérhetőség igazolása, sem az adatok titkos kezelése. A DNSSEC-ben minden válaszüzenet digitális aláírással van ellátva. Az aláírás ellenőrzésével igazolható, hogy a kapott információ megegyezik az autoritatív DNS szerver által nyújtott információval.