# Hálózatok (20. rész)
# ARP, RARP, BOOTP, DHCP, CDIR

Folytatjuk tovább az internet vezérlő protokolljainak ismertetését, majd rátérünk az IP protokoll egyre égetőbb problémájára: a még szabadon kioszthatóIP címek vészes fogyására.

Sorozatunk előző részében megismerkedtünk az *IP* címekkel, amelyek lehetővé teszik a gépek egyértelmű azonosítását az interneten. Felmerül azonban egy probléma: a gépek hálózati csatolói mit sem tudnak ezekről a címekről, ők ugyanis egy alacsonyabb szinten, az adatkapcsolati rétegben működnek.
Emlékezzünk vissza, hogy az adatkapcsolati réteg felelős azért, hogy egy *LAN*-on belül az állomások üzeneteket küldhessenek egymásnak keretek formájában. Mivel a kommunikáció nyílt csatornán zajlik, ezért az összes gép megkapja az összes keretet, majd ebből kell kiválogatniuk azokat, amelyek nekik szólnak. Így minden keret esetében meg kellett mondanunk a célpont hálózati csatolójának *MAC* címét. Ez például az *Ethernet* esetében egy 48 bites azonosító, amely a világ összes *Ethernet* hálózati kártyájában különböző.
Amikor az útválasztó az alhálózattól megkap egy olyan csomagot, amely egy, a saját hálózatán belüli géphez tart, akkor a kérdéses csomagot ki kell „bontania”, majd annak tartalmát az adatkapcsolati réteg segítségével keretek formájában eljuttatnia a címzetthez. Az adatkapcsolati réteg azonban egy *MAC* címet vár, a beérkezett csomag fejléce azonban *IP* címet tartalmaz. Az útválasztónak tehát ki kell derítenie az adott *IP* címhez tartozó hálózati csatoló *MAC* címét. Ehhez a legegyszerűbb módszer az, ha az útválasztó „bekiabál” a *LAN* csatornájába, hogy az adott *IP* cím gazdája jelentkezzen. Erre szolgál az ARP nevű protokoll.

## ARP (Address Resolution Protocol)

Az *ARP* protokoll tehát az *IP* címek *MAC* címekre történő leképezésére szolgál. Működése nem túl bonyolult. Ha egy gép érdeklődni szeretne például a 213.178.107.36 *IP* címhez tartozó *MAC* cím iránt, akkor csak küld egy adatszórásos üzenetet, amelyben megkérdezi: „Kié a 213.178.107.36 IP cím?”. Erre az üzenetre minden gép reagálni fog, és összehasonlítja saját IP címét a kérdésben szereplővel. Amelyik gép *IP* címe egyezik, az visszaküld egy válaszüzenetet, amelyből a kérdező megtudja a keresett gép *MAC* címét.
Ahhoz azonban, hogy az *ARP* igazán hatékony legyen, szükség van némi optimalizálásra. Beláthatjuk, hogy nem igazán járható az az út, ha minden egyes beérkező csomag esetén újra és újra érdeklődünk a *MAC* cím iránt. Mivel a gépek *IP* címei és hálózati csatolói nem változnak túl nagy gyakorisággal, ezért a birtokunkba került *MAC* címeket egy darabig eltárolhatjuk a memóriában. Persze ha mégis bekövetkezik valami változás, akkor arra gyorsan kell reagálnunk, így a gyorsítótárban tárolt információk érvényességének pár percen belül le kell járniuk.
A hatékonyságot tovább növelhetjük azzal, ha egy *MAC* cím után érdeklődő gép az *ARP* keretben a saját *IP* címét is elhelyezi. Ezt a címet pedig a *LAN* összes gépe eltárolhatja a gyorsítótárában, így az *ARP* keretet küldő gép *IP* címe már mindenki számára ismerté válik.
Amikor egy gép elindul, mindig küld egy *ARP* keretet, amelyben a saját *IP* címéhez keresi az ahhoz tartozó *MAC* címet. Ezzel egyrészt a *LAN* gépei tudomást szereznek az újonnan elindult gép *IP* címéről, másrészt így elkerülhető az *IP* cím ütközés. Ha ugyanis válasz érkezne a gép által küldött *ARP* keretre, akkor az azt jelentené, hogy a hálózaton valaki más is rendelkezik ugyanazzal az *IP* címmel. Ilyen esetben a gépnek nem szabad elindulnia.
Mi a helyzet azonban akkor, ha a keresett gép egy másik *LAN*-on található? Tegyük fel, hogy van egy *B* osztályú címtartományunk (például a 172.16.x.x), és a gépeinket két különálló *LAN*-ba szervezzük. A két hálózatot egy útválasztó köti össze, amelynek két *IP* címe van, tehát mindkét *LAN*-nak tagja.
Tegyük fel, hogy az A gép üzenetet szeretne küldeni a *C* gép számára, ám nem ismeri annak *MAC* címét, ezért egy *ARP* keretet bocsát a hálózatra. Ezt a keretet azonban a *C* gép sosem fogja megkapni, mivel az útválasztó nem továbbítja az adatkapcsolati réteg adatszórásos kereteit.
Erre a problémára két megoldás is létezik. Az első az, hogy az útválasztó válaszol azokra az *ARP* kérésekre, amelyek egy másik hálózatbeli gép iránt érdeklődnek. Ha tehát az A gép érdeklődik a *C* gép *MAC* címe iránt, akkor ennek hatására az útválasztó elküldi *A*-nak a saját *MAC* címét, így az A minden *C*-nek szóló keretét az útválasztóhoz továbbítja. Az útválasztó ezután gondoskodik arról, hogy ezek a keretek eljuthassanak *C*-hez. Ezt a technikát nevezzük ***helyettesítő ARP-nak (proxy ARP)***.

Ez a megoldás feltételezi azt, hogy az útválasztó pontosan tudja, mely alhálózatokat szolgálja ki. A másik megoldás szerint ezt a „tudást” inkább a gépekre kéne bízni, azaz fel kell ismerniük, hogy a keresett gép egy másik hálózatban van. Ebben az esetben az összes forgalmat egyből az útválasztó felé irányítja.

## RARP, BOOTP, DHCP

Láttuk, hogyan tudjuk az *IP* címeket leképezni *MAC* címekre. Időnként azonban ezt fordítva is meg szeretnénk tenni: egy ismert *MAC* címhez keressük a hozzá tartozó *IP* címet.
Erre a legjobb példa az, amikor egy olyan munkaállomást indítunk, amely nem rendelkezik háttértárolóval, így magát az operációs rendszert is a hálózatról tölti be. Amikor a gép elindul, akkor fel kell vennie a kapcsolatot a fájlszerverrel, ahonnan letölti az operációs rendszert tartalmazó képfájlt. Ehhez azonban a gépnek meg kell tudnia az *IP* címét.
Erre szolgál a ***RARP (Reverse Address Resolution Protocol)***, amely tulajdonképpen az *ARP* fordítottja, csak itt a *MAC* címet adja meg, és az ahhoz tartozó *IP* címre kíváncsi. Amikor a *RARP* szervert futtató gép meglátja ezt a kérést, akkor egy táblázatból kikeresi, mi az adott géphez rendelt *IP* cím, majd azt visszaküldi.
A *RARP*-nál is felmerül az a probléma, hogy az útválasztók nem továbbítják az adatkapcsolati réteg szintű adatszórásos forgalmat, tehát minden *LAN*-nak rendelkeznie kéne saját *RARP* szerverrel. E probléma megoldására jött létre a ***BOOTP (Bootstrap Protocol)***, amely adatkapcsolati réteg szintű keretek helyett *UDP* csomagokat használ.
A *BOOTP* működése nagyon egyszerű. Amikor a gép elindul, a 255.255.255.255-es *IP* címre (amely az adatszórásra fenntartott speciális *IP* cím) egy *UDP* csomagot, úgynevezett *BOOTREQUEST* csomagot küld, amely tartalmazza a gép *MAC* címét. Ezután vár a válaszra, ha az nem érkezik meg egy bizonyos időn belül, megismétli a *BOOTREQUEST* csomagot.
A *BOOTP* kiszolgáló a *BOOTREQUEST*-re válaszul egy úgynevezett *BOOTREPLY* csomagot küld, amely tartalmazza a gép *IP* címét, a képfájlt tároló fájlszerver címét, valamint az alapértelmezett átjárót és az alhálózati maszkot. Ezután a gép kapcsolatba lép a fájlszerverrel, és például a ***TFTP (Trivial File Transfer Protocol)*** protokoll segítségével letölti az operációs rendszert tartalmazó képfájlt.
Manapság azonban már szinte az összes számítógép rendelkezik háttértárral, így önállóan is képes betölteni az operációs rendszerét, csak a hálózati beállításokat kell „letöltenie”. Ezért a *BOOTP*-t továbbfejlesztették úgy, hogy az alkalmasabb legyen az olyan számítógépek konfigurálására, amelyeket gyakran hordoznak hálózatok között (például laptopok), ezzel megkönnyítve a felhasználók és a rendszergazdák életét. Erre szolgál a ***DHCP (Dynamic Host Configuration Protocol)***.
A *DHCP* ugyan felülről kompatibilis a *BOOTP*-vel, mégis számos dologban különbözik tőle. Ezek közül a legfontosabb az, hogy a *BOOTP* esetében a gépekhez történő *IP* címhozzárendelés általában statikusan történik, azaz egy adatbázisban előre meghatározzuk, hogy melyik gép melyik *IP* címet kaphatja meg. A *DHCP* esetében a címkiosztás általában dinamikus, tehát legtöbbször csak egy halmazt definiálunk, amelyből kiosztjuk a címeket a gépek számára. Egy állomás bekötése egy *DHCP*-t használó hálózatba igazából nem több, mint a hálózati kábelt a géphez csatlakoztatni.

## Többesküldés (multicasting)

Bizonyos alkalmazásoknak szüksége lehet arra, hogy egy csomagot egyszerre több címzetthez is eljuttathasson. Ilyen például egy videokonferenciát megvalósító program. Az adatszórás, mint megoldás, természetesen szóba sem jöhet, hiszen a csomagok olyan gépekhez is eljutnának, akik nem is érintettek a kommunikációban (nem beszélve arról, hogy az útválasztók általában nem továbbítják az adatszórásos csomagokat). Az ***unicast***, azaz amikor a csomagot egyesével elküldjük az összes címzetthez sem bizonyul mindig hatékonynak. Tegyük fel, hogy a videokonferencia két résztvevője ugyanabban a távoli hálózatban van. Az ***unicast*** használatával két ugyanolyan csomagot küldünk a hálózat felé, ami erőforrás-pazarló. Sokkal hatékonyabb lenne, ha csak egy csomagot küldenénk, amit a kommunikációban résztvevő mindkét fél megkapna.
Erre jelent megoldást a ***többesküldés (multicasting)***, amely lehetővé teszi, hogy egy csomagnak több címzettje is lehessen. A többesküldésről technikájáról már ejtettünk pár szót a hálózati réteg elméleti tárgyalásakor. Most azt tekintjük át, miként is működik ez az Internet esetében.
A többesküldés megvalósításának első kérdése az, hogy miként lehet összehozni azt az *IP* protokollal. Erre szolgálnak a speciális *D* osztályú *IP* címek ***(224.0.0.0 – 293.255.255.255)***. Minden, ebbe a csoportba tartozó *IP* cím egy többesküldési-csoportot határoz meg, így a kommunikációban résztvevő gépek külön *IP* címmel rendelkezhetnek. Könnyen kiszámolható, hogy egyszerre 250 millió csoport létezhet, amely az internet mai méretét tekintve (lásd később) nem tűnik túl soknak, minden estre ma még elégnek bizonyul.
A többesküldési-csoportoknak két fajtája van: az állandó és az ideiglenes. Az állandó csoportok olyanok, amelyek folyamatosan léteznek. Ilyen például a 224.0.0.1, amely a *LAN*-on belüli összes gépet tartalmazza. Szintén állandó csoport a 224.0.0.2, amelynek a *LAN*-on lévő útválasztók a tagjai.
Az ideiglenes csoportok viszont nem léteznek állandóan, ezeket használat előtt mindig létre kell hozni. A csoportokhoz a gépek szabadon csatlakozhatnak, illetve kiléphetnek onnan. Egy csoport akkor szűnik meg, amikor már egy gép sem csatlakozik hozzá.
A többesküldést speciális útválasztók valósítják meg, amelyek gyakran egybe vannak építve a „közönséges” útválasztókkal. Ezek bizonyos időközönként megkérdezik a hozzájuk kapcsolódó hálózat gépeit (általában többesküldéssel, a 224.0.0.1-es címen keresztül), hogy mely többesküldési csoportoknak a tagjai. Ezt az úgynevezett ***IGMP (Internet Group Management Protocol – internet csoportfelügyeleti protokoll)*** segítségével végzik, amely az *ICMP*-hez nagyon hasonló vezérlőprotokoll. A többesküldéses útválasztók feszítőfák felállításával jelölik ki a többesküldéses csomagok útvonalát (lásd sorozatunk 17. részét).

## CIDR (Classless InterDomain Routing)

Az *IP* jelenlegi verziója (az *IPv4*) jól bevált, kiforrott protokoll, bizonyítékul szolgál erre az Internet hihetetlen méretű terjedése. Ironikus módon a hálózat nagyléptékű

növekedése fogja az *IPv4* vesztét is okozni: a szabad (még ki nem osztott) *IP* címek száma vészesen csökken. Pontosabban nem is a szabad *IP* címek fogynak, hiszen abból több mint 4 milliárd létezik. A gond a címek tartományokra való tagolásából adódik. Mint utóbb kiderült, elég szerencsétlen módon sikerült az osztályokat megválasztani: az A osztályú tartományok a legtöbb hálózat számára túl nagy, a *C* osztályúak pedig túl kicsik. Sok intézmény tehát érthető módon *B* osztályú tartományokért folyamodott. Egy felmérésből azonban kiderült, hogy a *B* osztályt birtokló hálózatok jelentős része nem is használja ki igazán címtartományát, akár egy *C* osztályú címcsoporttal is vígan működhetne. Az üzemeltetők mégis egy *B* csoportot kérvényeztek, hiszen így nem okoz majd kellemetlenséget, ha a gépek száma egyszer majd 254 fölé emelkedik. Így azonban *IP* címek milliói mennek veszendőbe.

Ez a probléma valószínűleg elkerülhető lett volna azzal, ha a *C* osztályú címeknél 8 helyett 10 bitet szánunk a gépek azonosítására. Ez már elegendő lett volna a hálózatok többségének, így tényleg csak az folyamodott volna *B* osztályú tartományért, akinek tényleg szüksége is lett volna rá.

A most használt címtartományok másik problémája az útválasztók táblázatainak növekedésében jelentkezik. Az útválasztók számára az *IP* címek két részből állnak: a számukra teljesen érdektelen gépcímből, és a hálózatcímből. Az útválasztóknak minden hálózatcímhez fenn kell tartaniuk egy bejegyzést a forgalomirányító táblázatukban, amely megadja, hogy melyik kimeneten kell az arrafelé tartó csomagot továbbítani.

Ahogy egyre gyarapszik az Internethez kapcsolt hálózatok száma, úgy nőnek az útválasztók táblázatai. A nagyméretű táblázatokkal azonban máshogy kell bánni. Egyrészt több memória is kell hozzá, másrészt komplexebb kereső algoritmusok, amelyek segítségével az útválasztók gyorsabban megtalálhatják a kívánt bejegyzést. Sok, régebbi útválasztó által használt eljárás elavul, így azok az útválasztók nem tudják hatékonyan ellátni feladatukat.

A nagy méretű táblázatokkal azonban más baj is van. Az útválasztóknak folyamatosan cserélgetniük kell egymás között a táblázatuk tartalmait. Kis méretű táblázatok esetében ez könnyen ment, nagy méret esetében azonban a folyamat egyrészt lassú, másrészt sérülékeny. Egy nagyobb adatmennyiség nagyobb valószínűséggel sérül meg egy átvitel alatt. A sérült táblázatok pedig sok galibát okozhatnak a forgalomirányításban.

A forgalomirányító táblázatok azért nőnek ilyen mértékben, mert az *IP* címek nem tartalmaznak semmiféle információt arra nézve, hogy az adott hálózat merre található (nem tükrözik az Internet topológiáját). Ha például kiköthetnénk azt, hogy az *IP* címek első három bitje a földrészt, a következő öt pedig az országot jelölje, ahol az adott hálózat található, akkor a forgalomirányító táblázatban jelentős számú bejegyzést megspórolhatnánk. Sajnos erre a rendelkezésre álló 32 bit már kevés, nem is beszélve arról, hogy ez a megoldás nem használná ki hatékonyan a címeket, hiszen a kisebb országok is ugyanannyi címet kapnának, mint a nagyobbak.

Sajnos erre a két problémára nem létezik igazi megoldás (illetve létezik: az egész *IPv4* lecserélése egy újabb verzióra). Csupán annyit lehet tenni, hogy bizonyos intézkedésekkel időt nyerünk, de előbb vagy utóbb ismét szembekerülünk ezekkel a gondokkal.

Az egyik ilyen intézkedés, amely segítségével elódázható az Internet problémáinak megoldását a ***CIDR (Classless InterDomain Routing – osztály nélküli tartományok közötti forgalomirányítás)***, amely „megreformálja” a *C* osztályú *IP* címtartományok kiosztását. Először is, akinek nincs szüksége egy teljes *B* osztályú tartományra, mert például csak 400 gépet szeretne az Internethez kapcsolni, akkor az kap egy 512 darab címből álló egybefüggő címblokkot (két összefüggő *C* osztályú címtartományt). Akinek több gépe van, például 2000, az nyolc darab *C* osztályú tartományt kap.

A kiosztásnál azonban azt is figyelembe veszik, hogy az adott hálózat a világ melyik pontján található. Ha például ***Európában***, akkor a címeket a 194.0.0.0 – 195.255.255.255-ig terjedő tartományból osztják ki. Ezzel csökkenthetjük a forgalomirányító táblázatok méreteit, hiszen ha az útválasztó egy ilyen címmel találkozik, akkor egyből az alapértelmezett európai átjáróhoz küldheti.

Persze az európai útválasztóknak részletes ismeretekkel kell rendelkezniük az európai hálózatokról. Felmerül a kérdés, hogy mivel felrúgtuk az eddig használt *IP* címosztályokat, az útválasztók miként döntik el egy csomagról, hogy azt mely hálózat útválasztójához kell továbbítania? Továbbá nem áll-e fent megint a forgalomirányító táblázatok méretének megugrásának veszélye?

Az utóbbi kérdésre a válasz az, hogy nem tartunk fent minden *C* osztályú *IP* címnek külön bejegyzést, hiszen a *CIDR* használatával örökre búcsút mondunk az *IP* címek ily módon történő felosztásának. Most már mi határozzuk meg, hogy az *IP* címből hány bit szolgáljon a hálózat azonosítására. A forgalomirányító táblázatban csak a kiosztott címcsoport alapcímét és annak 32 bites maszkját tároljuk el. (Például egy intézmény igényt tart 2048 darab *IP* címre, így megkapja a 192.24.0.0 – 194.24.7.255-ig tartó címeket. A maszk azt mondja meg, hogy hány bit szolgál a hálózat azonosítására. Jelen esetben 21 darab bit, így a maszk 21 darab egyesből, és 11 darab 0-ból fog állni. Egy hálózat alapcímén a kiosztott címcsoport első címét értjük, jelen esetben ez a 192.24.0.0).

Tegyük fel, hogy beérkezik egy olyan csomag, amely a 192.24.2.5-ös *IP* címre tart. Ilyenkor az útválasztónak végig kell néznie a forgalomirányító táblája összes bejegyzését, és mindegyiken el kell végeznie a maszkolást, azaz egy logikai ÉS műveletet a csomag célpont címe és az adott bejegyzés maszkja között. Ha az így kapott érték megegyezik az adott hálózat alapcímével, akkor megtaláltuk azt a hálózatot, ahova a csomag tart, így továbbíthatjuk a megfelelő útválasztó felé.

A következő részben megismerkedünk az *IP* protokoll új generációjával, az *IPv6*-al, amely 32 bit helyett 16 bájton tárolja a címeket, kiküszöböli az *IPv4* hibáinak nagy részét, és mindezek mellett sokkal hatékonyabb és rugalmasabb is nála.

***Garzó András***
***garzo@interware.hu***