Hálózatok és protokollok
(jegyzet)

Uhlár László, Bérci Norbert

2015. október 8-i óra anyaga

Tartalomjegyzék

1. Egy kis történelem

1.1. A kezdetek

Az igény, hogy a számítógépek egymással valamiféle összeköttetésben legyenek, szinte egy idős az első elektronikus számítógépekkel. Kezdetben önálló, szinte egész termeket kitöltő számítógépeken dolgoztak az emberek. Korán megjelent az igény, hogy az egyik gépen megtalálható adat, program minél könnyebben átvihető legyen egy másik gépre anélkül, hogy ehhez külső adathordozót kelljen igénybe venni. A számítógépek méretének és árának csökkenésével egyre inkább elterjedt az a modell, hogy nem egy hatalmas gépen dolgoztak a felhasználók, hanem több kisebb számítógép volt például egy cég irodaházában. Mivel fizikailag egymáshoz közel voltak, jogos igény volt, hogy a viszonylag ritkán használt de drága perifériákból ne kelljen minden géphez külön-külön beszerezni egy példányt (pl.: nyomtató), hanem közösen használhassanak egy ilyen eszközt. Tehát a számítógépes hálózatok létrehozásának célja:

- Lehetővé teszi az erőforrások megosztását. A rendszerben levő erőforrások (háttértárak, nyomtatók, scannerek, egyéb perifériák) a jogosultságtól függően elérhetők bárki számára.
- Nagyobb megbízhatóságú működést eredményez, hogy az adatok egyszerre több helyen is tárolhatók, az egyik példány megsemmisülése nem okoz adatvesztést. Az azonos funkciójú elemek helyettesíthetik egymást. (Több nyomtató közül választhatunk.)
- Gazdaságosan növelhető a teljesítmény. A feladatok egy nagyszámítógép helyett megoszthatók több kisebb teljesítményű eszköz között. Sőt, egyes esetekben magát a nagy teljesítményű szervert is helyettesíthetik (cluster computing).

[0] $Revision : 48$ ($Date : 2013 - 10 - 0707 : 32 : 38 + 0200(Mon, 07Oct2013)$)

- Elérhetővé válnak a központi adatbázisok. Ezek az adatbázisok sok helyről lekérdezhetők, és sok helyről tölthetők. Csak így képzelhető el pl. egy valóban aktuális raktár vagy megrendelés állomány kezelés egy nagyvállalatnál.
- A hálózati rendszer kommunikációs közegként is használható (IP telefon, üzenetküldő szolgáltatások, email).

1.2. Az ARPA project

Az 1960-as évek közepén (dúl a hidegháború) az Amerikai Védelmi Minisztérium (U. S. Department of Defense) olyan parancsközlő hálózat kialakítását tűzte ki célul, mely átvészel egy esetleges atomcsapást. A fejlesztéseket a minisztérium ösztöndíjakkal támogatta. Az elméleti kutatások után olyan hálózat kialakítására írtak ki pályázatot, amely csomóponti gépekből áll, adathálózat köti ezeket össze és néhány csomópont megsemmisülése esetén is működőképes marad a hálózat többi része. A tenderre több cég is nevezett, a győztes 1969-ben állította üzembe az első csomópontot, 1972-re 37-re nőtt a csomópontok száma. Ekkoriban kapta az ARPAnet nevet ez a hálózat (Advanced Research Project Agency). A 70-es évek végére összeköttetések épültek ki más helyi hálózatok és az ARPAnet között, mára ez a hálózat behálózza az egész Földet. A 80-as évektől nevezik a hálózatok ezen hálózatát internetnek.

2. Rétegezett felépítés

2.1. Okok és célok

Bizonyára el tudjuk képzelni, hogy a fentebb vázolt hálózatokon a kommunikáció meglehetősen összetett és bonyolult dolog. Nem lenne szerencsés, ha a programozónak olyan hálózati kommunikációra képes programokat kellene írnia, amely a teljes kommunikáció minden aspektusát megoldja. Ugyanannak a programnak kellene gondoskodnia a megfelelő feszültségszintek előállításától kezdve a megcímzett gép azonosításáig mindenről. Ha a hálózati működés valamely részén változtatnak, akkor az egész program módosítására szükség lehet. Ennek elkerülése érdekében a hálózati kommunikáció folyamatát logikailag több részre bontjuk: az egyes részek a folyamat egy jól meghatározott részéért felelnek, azt kell megvalósítaniuk. Csak arról kell gondoskodni, hogy az egyes részek (rétegek) egymást megértsék: egy jól definiált interfészt kell egymás felé mutatniuk.

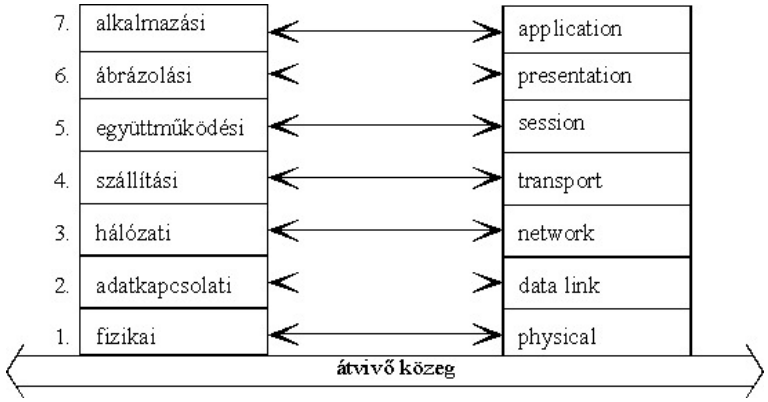
Például a postai levelezést mint kommunikációs hálózatot tekintve, a postaládát kiürítő dolgozónak nem kell tudnia repülőt vezetni vagy hajót építeni és átszelni az óceánt, ha oda szól a levél, neki csak a ládát kell tudnia kiüríteni (de azt hiba nélkül) és el kell juttatnia a borítékokat a megfelelő helyre. Az ottani dolgozónak pedig nem kell tudnia, hogy a város mely részén vannak ürítendő postaládák és azokat hogyan kell kinyitni, neki csak a küldeményeket kell bizonyos szempontok szerint szétválogatnia, stb. Hasonló módon az egyes rétegek szolgáltatásait implementáló programozóknak sem kell az egész kommunikációs problémát egyben vizsgálniuk, nekik elég csak az adott rétegre koncentrálniuk. Feltéve persze, hogy azért van valaki, aki átlátja a teljes hálózati működést, és úgy tervezi meg az egyes rétegeket illetve a közöttük lévő interfészeket, hogy összességében a hálózat a kívánalmaknak (specifikációnak) megfelelően működjön.

2.2. ISO OSI

A Nemzetközi Szabványügyi Szervezet (International Organization for Standardization, ISO[1]) létrehozta az OSI (Open Systems Interconnection - nyílt rendszerek összekapcsolása) modellt (ISO/IEC 7498), ami hét rétegre bontja logikailag a számítógép hálózatok működését (lásd az 1. ábrán).

Minden egyes réteg az alatta lévő réteg szolgáltatásait veszi igénybe, és annak segítségével tud kommunikálni. A kommunikáció célja ugyanakkor, hogy egy másik hálózatba kötött eszköz

[1] a rövidítés nem a szervezet elnevezésének rövidítése, mivel az sok nyelven másképp hangzana, hanem a görög *isos* szóból származtatták (jelentése: egyenlő) lásd: http://www.iso.org/iso/about/about



1. ábra. Az OSI rétegek

ugyanilyen rétegével adatot cseréljen, aminek a specifikációját a kettejük közötti *protokoll*nak nevezzük. Az alsóbb réteg feladata, hogy ezt lehetővé tegye (ami ennek megvalósításához igénybe veheti az azalatti réteg szolgáltatásait, és így tovább). A két, egymás alatti réteg közötti kommunikáció az *interfész*en keresztül történik. A egyes hálózati eszközök közötti tényleges fizikai jeltovábbítás a fizikai réteg segítségével, az átviteli közegen zajlik.

Segítheti a megértést, ha két magas rangú államférfira gondolunk, akik tolmácsok segítségével kommunikálnak egymással: az egyik vezető a saját tolmácsának mondja, az elmondja a másik tolmácsnak, az lefordítja a saját főnökének. Kik kommunikálnak egymással? Bár a vezetők beszélni csak a saját tolmácsaikkal beszélnek (illetve a tolmácsok egymással), de mégis a két államférfi cserél eszmét a beszélgetés során.

3. Az egyes rétegek feladata

Ebben a részben a hétköznapokban manapság leggyakrabban használt technológiákat tekintjük át: az Ethernetet és a TCP/IP protokollcsaládot. Természetesen ettől eltérő protokollok használatára is lehetőség van, ezekről bővebben a Számítógép hálózatok felsőbbéves tárgyban lesz szó.

3.1. A fizikai réteg

Az adatokat (biteket) valamilyen fizikai jellé alakítva az adott átviteli közegen tudni kell továbbítani illetve fogadni. A fizikai réteg tehát meghatározza az átviteli közeget, annak elektromos-, egyéb jellemzőit, az esetleges csatlakozók méretét, formáját, a bekötés módját, a használható frekvenciákat, az alkalmazott kódolást, az esetleges ütközések érzékelésének módját, stb, azaz minden olyan paramétert, aminek specifikálása ahhoz szükséges, hogy a fizikai közegen biteket lehessen továbbítani két eszköz között. Két nagy csoportját különböztetjük meg a jeleknek:

analóg : az adott jellemző (megadott határok között) bármilyen értéket felvehet

digitális : az adott jellemző csak fix értékeket vehet fel

bináris : két lehetséges értéket vehet fel

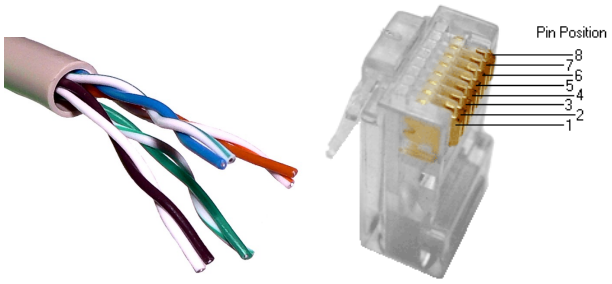
Gyakori, hogy a közeg analóg jelek továbbítására képes, emiatt meg kell oldani a digitális jelek analóg jellé konvertálását (DAC: digitális→analóg konverter), majd a vonal túlvégén az érkező analóg jeleket digitalizálni kell (ADC: analóg→digitális konverter). Ilyen konvertereket használunk például az audio CD-n tárolt digitális jelek analóggá konvertálásakor (DAC, például egy digitális bemenettel rendelkező rádióerősítő is tartalmaz ilyet), illetve a mikrofonnal érzékelt analóg jel digitálissá konvertálásakor (ADC).

Fontos fogalom az átviteli sebesség (sávszélesség), ami megadja az egy másodperc alatt átvitt bitek számát, mértékegysége a bps (bit per secundum): például 10 bps a sebesség, ha 10 bit

adat továbbítódik egy másodperc alatt. Elterjedt prefixált mértékegységek még a kbps (ezer bit per másodperc), az Mbps illetve a Gbps hasonlóan megabit illetve gigabit egységekben. Nem összekeverendő a Bps-el, ami bájtban adja meg az átvitt adatmennyiséget.

3.1.1. UTP

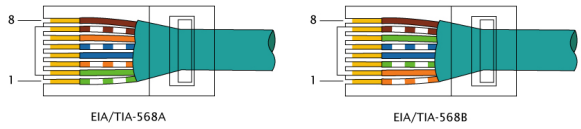
UTP (Unshielded Twisted Pair: árnyékolatlan csavart érpár). Felhasználóként ezzel a kábellel találkozunk napjainkban a legtöbbet. 8 szál vezeték párosával egymás köré tekerve alkotja a kábelt, külön árnyékolás nélkül. A vezeték párok egymás köré tekerésével csökkentik a környezet zavaró jeleit, a megcsavarás számának változtatásával pedig a párok közötti áthallás csökkenthető (lásd a 2. ábrán).



(a) UTP kábel (b) 8P8C csatlakozó

2. ábra. UTP kábel és csatlakozó

A csatlakozóban az egyes vezetékek sorrendje fontos, mivel azok kettesével vannak csavarva: nem elegendő az, hogy például a csatlakozó négyes pinje a kábel másik végén lévő csatlakozó négyes pinjéhez csatlakozik, illetve, hogy az ötös pin az ötös pinhez, hanem az is fontos, hogy a négyes és az ötös pinekhez csatlakozó vezetékek ugyanannak a vezetékpárnak a tagjai legyenek. Sőt, az sem mindegy, hogy az egyes érpárok közül melyek kerülnek egymás mellé. A leghasznosabb követni egy szabványos színkód alapú bekötési módot: ilyenből kétféle is létezik: ismertebb nevükön TIA/EIA-568A és TIA/EIA-568B, de mindkettő része a nemzetközi ISO/IEC 11801 szabványnak (lásd 3. ábrán).



(a) TIA/EIA 568A (b) TIA/EIA 568B

3. ábra. TIA/EIA 568 bekötési módok

Azért van belőlük kettő, mert a régebbi Ethernet szabványoknál ha két gépet közvetlenül akartunk összekötni, akkor a kábel egyik végére az egyik, a másik végére a másik változat szerinti kiosztás szerint kellett az érpárok bekötését elvégezni, amit *cross link* kábelnek nevezünk. Manapság is lehet ilyen kábeleket kapni hálózati termékeket forgalmazó szakboltokban, de a modern eszközök (újabb hálózati kártyák) automatikusan érzékelik, hogy milyen kábelt csatlakoztattunk, így használhatók cross link és straight-through kábellel is.

Az UTP kábeleket több kategóriába sorolják a paramétereik szerint (sintén az ISO/IEC 11801 illetve a vonatkozó TIA/EIA 568 szabványokban): CAT3: 16 MHz, CAT4: 20 MHz, CAT5: 100 MHz, CAT6: 250 MHz, CAT6A: 500 MHz, CAT7: 600 MHz, CAT7A: 1000 MHz frekvenciáig bevizsgált és minősített kábel (lásd a 4. ábrán).

4. ábra. Egy UTP kábelen feltüntetett jelölések (többek között a kategória)

3.2. Az adatkapcsolati réteg

Az adatkapcsolati réteg (Data Link Layer) alapvető feladata, hogy egy bitfolyam átvitelére képes fizikai rendszert egy olyan eszközzé alakítsa, ami adatintegritás ellenőrzött kommunikációt tesz lehetővé két, helyi hálózatba kötött eszköz között. Az adó oldal a bemenő adatokat meghatározott hosszúságú darabokra – *keret*ekké – tördeli, majd a protokollnak megfelelő kiegészítő információkkal egészíti ki (pl. a keretek előtt és mögött speciális bitmintákat helyez el a keret elejének és végének felismeréséhez; a hálózati eszközök címzéséhez szükséges adatokat fűz hozzá; adatintegritás ellenőrző kódokkal egészíti ki). A vevő oldal a fogadott adatkereteket a megfelelő részek értelmezése és levágása után továbbítja a felsőbb rétegnek.

3.2.1. Az Ethernet

Környezetünkben legelterjedtebben az Ethernettel találkozhatunk (IEEE 802.3 szabványcsalád), ami egy lokális hálózati (LAN) protokoll, azaz Ethernettel csak ugyanabban az Ethernet hálózatban létesíthető kommunikáció egy másik eszközzel! Az Ethernet a MAC címmel biztosítja a hálózati eszközök címezhetőségét: minden egyes keret tartalmazza a küldő állomás MAC címét és a címzett MAC címét. A címzett MAC cím jelölhet egyetlen eszközt, de jelölhet több eszközt is (ebben az esetben *broadcast* címről beszélünk). A MAC cím az egész világon egyedi szám, 6 bájtból áll, 12 jegyű hexadecimális szám formájában szoktuk megadni. Pl.: 38:60:77:df:94:f3. Napjaink hálózati eszközei már lehetőséget biztosítanak arra, hogy a MAC címet megváltoztassuk, így az egyediség nem feltétlenül biztosított.

Az Ethernet manapság leggyakrabban használt verzióiban (100BASE-TX, 1000BASE-T) egy központi egységhez csatlakoznak az állomások egyesével, UTP kábellel (csillag topológiában). Ennek egyik előnye, hogy kábelhiba esetén csak az érintett gép esik ki a kommunikációból, továbbá amiatt, hogy a kábelhez (az átviteli közeghez) csupán két eszköz kapcsolódik, lehetőség van *full duplex*[2] átvitelre: az átviteli közegen mindkét hálózati eszköz egyszerre tud kommunikálni úgy, hogy egymás adását nem zavarják, nem korlátozzák, mintha egy-egy párhuzamos csatornán zajlana a két irányú kommunikáció.

A különböző UTP kábelekből a következő, leggyakrabban használt Ethernet hálózatok építhetők:

- 10BASE-T: 10 Mb/s sebességű Ethernet 2 érpáron, CAT3 kábelen (IEEE Std 802.3 Clause 14.4.1)
- 100BASE-TX: 100 Mb/s sebességű Ethernet 2 érpáron, CAT5 kábelen (IEEE Std 802.3 Clause 25.4.9)
- 1000BASE-T: 1000 Mb/s (=1Gb/s) sebességű Ethernet 4 érpáron, CAT5 kábelen (IEEE Std 802.3 Clause 40.7.1)
- 10GBASE-T: 10Gb/s sebességű Ethernet 4 érpáron, CAT6, CAT6A, CAT7, CAT7A kábelen (IEEE Std 802.3 Clause 55.7.1)

[2] néhol csak simán duplexnek hívják ezt a kommunikációs módot

A központi egység régebben hub volt, ma már legtöbbször switch. A két eszköz közötti különbség a következő: A hub az egyik csatlakozóján beérkező jelet az összes csatlakozón újra kiküldi, azaz a jelszintek és időzítések helyreállítása után egyszerűen csak megismétli azokat. Ennek egyik hátránya, hogy minden keret minden egységhez eljut, aminek biztonsági kockázata lehet (harmadik félhez is eljutnak két másik fél közötti adatok), a másik hátránya pedig, hogy a hálózat által biztosított sávszélességet nem hatékonyan használjuk fel, hiszen olyan keretek is közlekednek a kábelen, amelyek nem szólnak a kábelen lévő egyik egységnek sem, és nem is tőlük érkeznek. A switch erre megoldást kínál: a keretekben található fizikai cím (MAC) alapján eldönti, hogy mely csatlakozóján (portján) küldi tovább az adatot. A switch működése egy dinamikusan karbantartott táblázatra épül, amelyben a kapcsoló minden portjához feljegyzi az adott porton beérkező keretek küldőjének MAC címét. Ezzel a kapcsoló megismeri a hozzá kapcsolódó gépek helyzetét, tehát azt, hogy az egyes gépek a kapcsoló melyik interfészéhez kapcsolódnak. Egy beérkezett keret továbbításához csak meg kell vizsgálnia a táblázatot, hogy a keretben szereplő cél MAC cím melyik interfészén keresztül érhető el. Egy interfészhez több gép MAC címe is feljegyezhető, ezért nincs akadálya hub-switch vagy switch-switch kapcsolatnak sem. A switch bekapcsolásakor kezdődő tanulási folyamat során fokozatosan alakul ki a kapcsoló táblázata, ezért normál jelenségnek tekinthető, ha egy olyan keretet kell továbbítania, amelynek a címzettje még a switch számára ismeretlen irányban van. Ekkor az ún. elárasztást alkalmazza, azaz a beérkezett keretet a fogadó port kivételével az összes többi portján kiküldi (és az erre érkező válaszból fogja megtanulni, hogy az az eszköz melyik portján érhető el).

Nagyon fontos kiemelni, hogy az Ethernet bármely más protokoll alkalmazása nélkül is lehetőséget ad a kommunikációra egy Ethernet hálózaton belül. Probléma akkor merül csak fel, ha több ilyen hálózatot kell összekapcsolni és közöttük is biztosítani kell az adatok továbbítását. Mivel az Ethernet hálózatok mérete korlátozott, erre előbb-utóbb szükség lesz, nem is beszélve arról, ha az internethez szeretnénk kapcsolni a helyi hálózatunkat.

3.3. A hálózati réteg

A hálózati réteg feladata a csomagok eljuttatása a forrástól a célig úgy, hogy azok akár több lokális hálózaton (LAN-on) is áthaladhatnak. Pontosan ez különbözteti meg az adatkapcsolati rétegbeli protokolloktól (pl. Ethernet), amik egy hálózaton belül képesek keretek célba juttatására. A célig egy csomag valószínűleg több csomópontot is érint, sőt, az is elképzelhető, hogy több párhuzamos útvonal is létezik. Az útvonal megválasztásához természetesen (részben) ismerni kell az átviteli hálózat felépítését, azaz a topológiáját, és ki kell tudni választani egy megfelelő útvonalat. A csomagoknak tartalmazniuk kell mind a forrás, mind a cél hálózati címet (ami általában különbözik az adatkapcsolati rétegben alkalmazott címtől!). A következőkben az IP-vel (Internet Protocol), mint hálózati rétegbeli protokollal foglalkozunk, a pontos protokoll specifikáció megtalálható a http://tools.ietf.org/html/rfc791 oldalon.

3.3.1. IP (v4) címek

Ahhoz, hogy a hálózati réteg megtalálja a csomagok címzettjét, minden gépnek rendelkeznie kell egy egyedi címmel, ez az IP cím. A jelenleg (még) legelterjedtebb IPv4 szerint ez a cím egy 32 jegyű bináris szám, amit a jobb olvashatóság miatt 8 bitenként decimális számmá alakítunk (pontozott négyes jelölés [dotted decimal notation]), például: 193.224.69.67.

A hálózati eszközöket fizikai vagy logikai szempontok alapján alhálózatba soroljuk. Az egy alhálózatban lévő gépek egymással közvetlenül (azaz hálózati rétegbeli útválasztók nélkül) tudnak kommunikálni. Annak eldöntése, hogy két gép egy alhálózatban található-e, az IP címükből eldönthető. Kezdetben a lehetséges címeket osztályokba sorolták, így beszélhetünk A, B, C, stb. osztályú címekről.

- A osztály: az első bit (a 32 -ből) 0, és 8 bit azonosítja a hálózatot (ebből a már említett első bit fix), a maradék 24 bit a hálózaton belül az egyes hálózati eszközöket. A legkisebb ilyen hálózat azonosító a 0 lehet, a legnagyobb a 127. Egy hálózaton belül kb. 2^{24} eszköz címezhető meg.

- B osztály: a cím első két bitje 10, és 16 bit azonosítja a hálózatot (ebből a már említett első két bit fix), a maradék 16 bit az egyes hálózati eszközöket. Így a legkisebb hálózatszám a 128.0 lehet, a legnagyobb a 191.255. Egy hálózaton belül kb. 2^{16} eszköz címezhető meg.
- C osztály: a cím első három bitje 110, és 24 bit azonosítja a hálózatot (ebből a már említett első három bit fix), a maradék 8 a hálózaton belül az egyes hálózati eszközöket. A legkisebb ilyen hálózatcím a 192.0.0, a legnagyobb a 223.255.255. Egy hálózaton belül kb. 2^8 eszköz címezhető meg.
- A többi osztállyal (a fennmaradó IP címekkel) jelen jegyzetben nem foglalkozunk, speciális célokra fenntartottak.

Minden hálózaton belül található két speciális cím: a *network address*, ami csupa 0 host részből áll, és a *broadcast address*, ami a csupa 1 host részből áll. Mivel ezeket a címeket nem lehet egyik gépnek sem kiosztani, minden egyes hálózatban az elméletinél kettővel kevesebb IP cím osztható ki hálózati eszközöknek.

3.3.1. példa. Az 19 -es hálózatban (A osztályú cím) a network address: 19.0.0.0, a broadcast address: 19.255.255.255, a kioszthatók gépek száma: $2^{24} - 2$

3.3.2. példa. Az 155.13 -as hálózatban (B osztályú cím) a network address: 155.13.0.0, a broadcast address: 155.13.255.255, a kioszthatók gépek száma: $2^{16} - 2$

3.3.3. példa. Az 210.15.9 -ás hálózatban (C osztályú cím) a network address: 210.15.9.0, a broadcast address: 210.15.9.255, a kioszthatók gépek száma: $2^8 - 2$

Minden osztályban kijelöltek olyan címtartományt[3], amelyek a nyílt interneten nem használhatók (nem „publikusak"). Ezeket a privát címeket egy-egy belső alhálózatban lehet használni, de ilyen címek az internetre nem továbbíthatók, illetve nem is érkezhetnek onnét. Azaz ilyen privát IP címmel lehet, hogy egy időben több számítógép is rendelkezik a világon, de ebben a formában nem tudnak kommunikálni a külvilággal (és egymással sem). A tartományok a következők:

- A osztályban: 10
- B osztályban: 172.16–172.31
- C osztályban: 192.168.0–192.168.255

A publikus tartomány mellett két A osztályú speciális hálózatot[4] kell még megemlíteni: a 0 és a 127 hálózati azonosítójút: Az előbbi a „saját hálózatom" azonosítására szolgál, és csak abban a speciális esetekben továbbítható, ha az IP cím használata kötelező, de mégsem tudjuk azt a valós (érvényes) értékkel kitölteni. A 127-es hálózat a *loopback* hálózat, bármelyik címe pedig a „saját magam" IP címe, azaz olyan esetekben használjuk, amikor ugyanannak a gépnek akarunk küldeni valamit, amin éppen dolgozunk. Például fut egy program a saját gépünkön, és ki akarjuk próbálni, hogy működik-e: ekkor nem kell a gép valódi IP címét használni, hanem elegendő például azt, hogy 127.0.0.1. Másik példa: ha felinstalláltunk egy web szervert a gépünkre, akkor a 127.0.0.1 címet beírva a böngészőbe, kipróbálhatjuk, hogy működik-e, anélkül, hogy ehhez IP címet kellene a gépnek kiosztani.

A jelen jegyzetben tárgyalt speciális tartományok közül még egy fontos, a 169.254 számú hálózat[5]: ezt autokonfigurációs esetben használja az operációs rendszer: ha más módon a gép IP címe nem deríthető ki, de helyi hálózatban mégis használni szeretnénk a gépet mindenféle kézi beállítás nélkül.

A `ping` paranccsal megpróbálhatunk elérni egy IP címet: a megadott IP című gép általában válaszol a PING kérésre, így még a gép elérési idejéről is információt szerezhetünk (mennyi időbe kerül a PING csomagot a gépnek eljuttatni és onnét a választ megkapni [round trip time]):

[3] RFC 1918 - http://tools.ietf.org/html/rfc1918
[4] RFC 1122 - http://tools.ietf.org/html/rfc1122
[5] RFC 3927 - http://tools.ietf.org/html/rfc3927

```
bercin@users:~$ ping 173.194.44.55
PING 173.194.44.55 (173.194.44.55) 56(84) bytes of data.
64 bytes from 173.194.44.55: icmp_req=1 ttl=51 time=15.4 ms
64 bytes from 173.194.44.55: icmp_req=2 ttl=51 time=15.4 ms
64 bytes from 173.194.44.55: icmp_req=3 ttl=51 time=15.5 ms
^C
--- 173.194.44.55 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 15.487/15.501/15.522/0.102 ms
bercin@users:~$
```

A ping működik a már említett 127-es (loopback) hálózatra is:

```
bercin@users:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_req=4 ttl=64 time=0.039 ms
^C
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.028/0.033/0.039/0.008 ms
bercin@users:~$
```

Az elérési idő nagyságrendekkel kevesebb, ami abból következik, hogy a loopback IP címek nem hagyják el a gépet, azt az operációs rendszer kezeli.

Kezdetben az egyes hálózati eszközök tehát az IP címből meg tudták állapítani, hogy két gép egy alhálózaton van-e (a fenti módszerrel). Viszont már elég korán felismerték annak a veszélyét, hogy így nem túl gazdaságos a címek kiosztása, pl. ha valaki megszerzett egy A osztályú hálózatot, akkor $2^{24}-2$ darab különböző gépet helyezhetne el benne, de ennyire valószínűleg nincs szüksége. Felmerült az igény arra, hogy ezek helyett az alapértelmezett hálózatszámok helyett szabadabban lehessen gazdálkodni a még meglevő címekkel.

3.3.2. A CIDR hálózati osztályozás

A CIDR (Classless Interdomain Routing) használatával már nem dönthető el pusztán a hálózat címéből, hogy hány bit alkotja a hálózati részt és hány bit a host részt (ahogy azt az osztály alapú címzésnél tettük). Az IP cím mellé meg kell adni egy újabb 32 jegyű bináris (32 bites) számot, amely az elején csupa 1-t tartalmaz, utána csupa 0-t, amit *netmask*nak nevezünk. Ezt szintén pontozott négyes jelöléssel írjuk le, például: 255.255.255.0.

A network addresst úgy kaphatjuk meg, hogy a netmaskkal és egy IP címmel bináris ÉS műveletet végzünk, míg a broadcast address a netmask negáltja és az IP cím bináris VAGY művelet végzésével számítható ki.

3.3.4. példa. Adott egy gép a 193.224.69.67 IP címmel és 255.255.255.0 netmaskkal. A network address kiszámítása:

IP	193.224.69.67	11000001111000000100010101000011
netmask	255.255.255.0	11111111111111111111111100000000
bináris ÉS művelet		
network address	193.224.69.0	11000001111000000100010100000000

3.3.5. példa. Adott egy gép a 193.224.69.67 IP címmel és 255.255.255.0 netmaskkal. A broadcast address kiszámítása:

IP	193.224.69.67	11000001111000000100010101000011
netmask negáltja	255.255.255.0	00000000000000000000000011111111
bináris VAGY művelet		
broadcast address	193.224.69.255	11000001111000000100010111111111

Sokkal kényelmesebb leírást, használhatóságot biztosít az úgynevezett CIDR jelölés, amely esetén a hálózat pontozott négyes jelölése helyett az alhálózati maszk egyeseinek a számát adják meg. Például az előző példában szereplő 193.224.69.67 IP című és 255.255.255.192 netmaskkal rendelkező gép a rövidített CIDR jelöléssel: 193.224.69.67/26. Érdemes kipróbálni a következő oldalt: http://www.fport.hu/index.php?site=cidr

3.3.6. feladat. A CIDR kompatibilis a régi, osztály alapú címzéssel? Ha nem, miért? Ha igen, adjuk meg az egyes osztályokhoz tartozó netmaskokat!

3.3.7. feladat. Hány gép címezhető meg egy 255.255.255.128 netmaskú hálózatban?

3.3.8. feladat. Mennyi legyen a netmask, ha legalább 20, de legfeljebb 30 gép számára szeretnék IP címeket kiosztani?

3.3.3. Az ARP protokoll

Ha Ethernet hálózatról van szó, akkor az IP címen kívül az adatok tényleges elküldéséhez szükség van a címzett MAC címére is, mivel az adatkapcsolati réteg szintű kommunikáció Ethernet protokollal történik (emlékezzünk vissza a rétegezett felépítésre), így az Ethernet keretet ki kell töltenie, amiből a saját MAC címünket nyilván ismerjük, így csak a cél gép MAC címére van szükség.

Egy IP címhez tartozó MAC cím kiderítése az ARP (Address Resolution Protocol) protokoll feladata: Az A gép egy speciális Ethernet keretet küld a hálózatra, amely minden géphez eljut (broadcast), és az a gép, amelyiknek a keretben szereplő IP cím a saját IP címe (B gép), egy válasz keretet küld. Mivel a válasz is egy Ethernet csomag, és abban ki kell tölteni a forrás MAC címet (azaz B MAC címét), az A gép amikor megkapja a válasz csomagot, egyben megtudja az IP címhez tartozó MAC címet is. Mindezek után az eredetileg szándékozott adat elküldése már IP szinten is lehetségessé válik.

Az operációs rendszer az ARP válaszok fogadásakor egy belső táblázatot (ARP cache) tart karban, és ennek adatait használja az IP címhez tartozó MAC cím meghatározásakor, hogy ne kelljen minden egyes csomag küldésekor az ARP kérdés-válasz kommunikációt lejátszani. Ha a kérdéses MAC cím nem szerepel az ARP táblában, akkor az ARP kérdés-válasz lejátszódik; ha szerepel, akkor az Ethernet csomag egyéb kommunikáció nélkül kitölthető. Az ARP cache tartalmát lekérdezhetjük az arp paranccsal:

```
NorbiMBPr:trunk bnorbert$ arp -a
? (10.0.1.1) at b8:c7:5d:cf:59:2e on en0 ifscope [ethernet]
appletv (10.0.1.105) at b8:c7:5d:cf:59:2e on en0 ifscope [ethernet]
jcidev.vmnet (192.168.153.130) at 0:c:29:b0:55:d6 on vmnet8 ifscope [ethernet]
NorbiMBPr:trunk bnorbert$
```

A számunkra jelenleg érdekes részek a zárójelben lévő IP címek és az utánuk következő (hozzájuk rendelt) MAC címek.

3.3.4. Az útvonal meghatározása (routing)

Ha a forrás IP és a cél IP ugyanabban a hálózatban (network address) vannak, akkor a csomag a célgépnek közvetlenül elküldhető:

3.3.9. példa. Az A gép IP beállítása: 193.224.69.67/26, és szeretne kommunikálni a 193.224.69.121 című géppel (B gép). A saját hálózatunkban van? Először számítsuk ki az A gép network addressét:

IP A	11000001111000000100010101000011	193.224.69.67
netmask	11111111111111111111111111000000	255.255.255.192
network address	11000001111000000100010101000000	193.224.69.0

Majd számítsuk ki a B gép network addressét a saját hálózatunk netmaskját felhasználva (azért a saját hálózatunk netmaskját használjuk, mert azt akarjuk eldönteni, hogy a célgép ebben a hálózatban van-e):

IP B	11000001111000000100010101111001	193.224.69.121
netmask	11111111111111111111111111000000	255.255.255.192
network address	11000001111000000100010101000000	193.224.69.0

Mivel a két network address megegyezik, a két gép egy hálózatban (a saját hálózatunkban) van, így közvetlenül kommunikálhat vele az alsóbb rétegbeli protokoll segítségével (Ethernet).

Ha a cél IP cím nem a forrás IP címmel azonos hálózatban van, akkor a csomagot egy routernek (útvonal választónak) kell továbbítania, ennek a dolga valahogyan azokat a címzetthez eljuttatni. Ahhoz, hogy ez megtörténhessen, a router IP címét is ismerni kell.

3.3.10. példa. A kommunikáció kezdeményezője legyen megint a fenti példában szereplő 193.224.69.67/26 című gép. Kezdeményezzen kommunikációt a 193.224.69.51 IP című géppel. Egy alhálózatban vannak? Ehhez a címzett címének veszi az első 26 jegyét: 11000001111000000100010100 (pontozott decimális alakban: 192.224.69.0). Látható, hogy különböznek, azaz nem egy alhálóban vannak, így a routernek kell küldeni a csomagot.

3.3.11. feladat. Mi az oka annak, hogy ha a saját hálózatunk netmaskját használjuk fel annak eldöntésére, hogy a célgép a saját hálózatunkban van-e, akkor nem követünk el hibát?

3.3.12. feladat. Lehetséges-e, hogy a router nem a saját alhálózatunkban van?

Lehetséges, hogy egy gép nem csak egy hálózati rétegbeli hálózathoz csatlakozik, ezért minden eszköznek van routing táblája, amely arról tartalmaz információkat, hogy adott alhálózatok esetén mely hálózati csatolón (azaz adatkapcsolati rétegbeli eszközön) küldje ki az csomagokat/kereteket. A routing tábla (többek között) a következő bejegyzéseket tartalmazza: hálózati cím, netmask, router címe, hálózati kártya. Linuxon ezt a `route` paranccsal listázhatjuk ki:

```
bercin@kanape:~$ /sbin/route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.140.2   0.0.0.0         255.255.255.255 UH    0      0        0 tun0
192.168.140.0   192.168.140.2   255.255.255.0   UG    0      0        0 tun0
10.3.18.0       0.0.0.0         255.255.254.0   U     0      0        0 eth0
0.0.0.0         10.3.19.254     0.0.0.0         UG    0      0        0 eth0
bercin@kanape:~$
```

Az első sor szerint a 192.168.140.2 IP-vel és 255.255.255.255 netmaskkal megadott hálózatba router nélkül lehet csomagokat küldeni (mert a gateway IP címe a 0.0.0.0) a tun0 hálózati csatolón keresztül (sor vége). A második sor szerint a 192.168.140.0 IP-vel és 255.255.255.0 netmaskkal megadott hálózatba a 192.168.140.2 routeren keresztül lehet csomagokat küldeni, és a router a tun0 hálózati csatolón érhető el.

Az operációs rendszer a lista sorrendjében próbálja végig a lehetséges útvonalakat, és az első egyezésnél szereplő adatokat használja fel.

3.3.13. példa. A 10.3.19.7 IP címre küldendő csomag esetében az első sor nem mutat egyezést (mivel 10.3.19.7 & 255.255.255.255 != 192.168.140.2), a második sor sem mutat egyezést (10.3.19.7 & 255.255.255.0 != 192.168.140.0) viszont a harmadik sor igen (mivel 10.3.19.7 & 255.255.254.0 = 10.3.18.0), így a csomagot a lokális hálózaton kell továbbítani (mivel ebben a sorban a gateway címe 0.0.0.0), az eth0 hálózati csatolón keresztül.

3.3.14. feladat. Milyen hálózatot jelöl a 255.255.255.255 netmask?

3.3.15. feladat. Milyen hálózatot jelöl a 0.0.0.0 netmask és a 0.0.0.0 network address? Milyen IP címekre fogja ezt a bejegyzést választani az operációs rendszer?

Fontos információkhoz juthatunk a `traceroute -n` parancs segítségével, amivel az adott géphez vezető útját, az egyes közbülső routereinek elérési idejeit láthatjuk:

```
NorbiMBPr:trunk bnorbert$ traceroute -n 217.20.130.97
traceroute to 217.20.130.97 (217.20.130.97), 64 hops max, 52 byte packets
 1  10.0.1.1  1.113 ms  0.753 ms  0.739 ms
 2  80.99.195.254  94.095 ms  22.981 ms  17.520 ms
 3  89.135.214.94  9.283 ms  10.238 ms  10.511 ms
 4  193.188.137.25  9.619 ms  9.788 ms  10.651 ms
 5  217.20.130.97  10.892 ms  8.904 ms  11.312 ms
NorbiMBPr:trunk bnorbert$
```

Az első oszlop a router sorszámát tartalmazza (hányadik router az útvonalon), a második oszlop a router IP címe. A többi oszlopban az adott router elérési ideje látható (bővebben lásd: `man traceroute`). Ez a parancs windowsban is megtalálható, de ott `tracert` -nek hívják.

3.3.5. A DNS

Igen ám, de ha mi beírunk a böngészőbe egy címet, az a legritkább esetben pontozott négyes jelölés, legtöbbször szövegesen megadott címek: például `www.index.hu` Hogyan lesznek a szöveges címekből IP címek? Erre szolgál a DNS (Domain Name Service).

Egy világméretű elosztott (azaz nem egy gépen tárolt, hanem részenként szétosztott) adatbázisban találhatóak a névhez IP címet rendelő információk. Ha a gépünknek szüksége van pl. a `turdus.itk.ppke.hu` gép címére, akkor egy kérdést küld a beállított DNS szervernek, ami általában tovább küldi a kérést: a .hu legfelső szintű (top level) domain valamely DNS szerveréhez (több gépen vannak ugyanazok az adatok), hogy mely gép tárolja a ppke.hu tartomány címeit. Ekkor visszakapja a gépünk az egyik ilyen DNS szerver címét. Egy második körben már ehhez fordul egy újabb kéréssel, hogy megtudja, hogy mely gép tárolja az itk.ppke.hu tartomány címeit. Ekkor innét is visszakapja egy olyan DNS szerver címét, amelyhez fordulva kéréssel, már megkaphatja a turdus.itk.ppke.hu gép IP címét. (A kapott eredményt egy ideig megőrzi, legközelebb ne kelljen az egész utat újra bejárni.)

Linuxban a `host` paranccsal tudjuk ezt kipróbálni:

```
bercin@users:~$ host www.index.hu
www.index.hu has address 217.20.130.97
bercin@users:~$
```

Néhány esetben (a terheléselosztás miatt) több IP címet is visszakaphatunk:

```
bercin@users:~$ host www.google.hu
www.google.hu has address 173.194.44.55
www.google.hu has address 173.194.44.56
www.google.hu has address 173.194.44.63
www.google.hu has IPv6 address 2a00:1450:4016:803::101f
bercin@users:~$
```

3.3.6. A DHCP

A fentiekből látható, hogy minden eszköznek ki kell osztani egy IP címet, meg kell adni a hálózatához tartozó netmaskot, tudnia kell a router címét, és a DNS címet is (amennyiben szeretnénk szöveges címeket használni). Így elég bonyolulttá válik egy hálózatban az IP címek karbantartása: ügyelni kell, hogy mindegyik eszköz jó címeket kapjon, ne legyen ütközés (ugyanolyan IP címet ne kapjon két eszköz), stb. Ennek a problémának a megoldására dolgozták ki a DHCP-t. (Dynamic Host Configuration Protokol). Ez (többek között) az előző beállítások automatikus megoldását teszi lehetővé, ezzel segítve a rendszergazda, a felhasználó munkáját.

Ha a DHCP szolgáltatás nem elérhető, a gépek általában a fentebb említett 169.254 -es hálózatból automatikusan osztanak maguknak IP címet, amivel lehetővé válik, hogy a LAN-on belül kommunikáljanak.

3.3.7. Az IPv6

Napjainkra égető problémává vált, hogy a fentebb vázolt 32 bites IP címzés nem elegendő, elfogynak a kiosztható címek. Erre kínál megoldást az IPv6, amely esetén 128 bites címe van minden hálózati csatolónak. Bővebben: `http://en.wikipedia.org/wiki/IPv6` `http://ipv6forum.hu` `http://www.worldipv6launch.org`

Az IP címek elfogyása annak is „köszönhető”, hogy az internet őskorában az A osztályú címeket is előszeretettel kiosztották az akkoriban jelentős vállalatoknak. Érdekes olvasmány ezek listája (mivel az akkoriban kapott tartomány a legtöbb esetben még mindig az adott cég birtokában van - ha a cég még létezik). Például: az USA kormányának különböző kutatási, technológiai, katonai részlegei; IBM; AT&T; Xerox; HP; DEC; Apple; MIT; Ford; UK Védelmi Minisztérium; UK Munka- és Nyugdíjügyi Minisztérium; duPont; US Postal Service; Bővebben például a `http://en.wikipedia.org/wiki/List_of_assigned_/8_IPv4_address_blocks` oldalon...